Nexthink V6.28

Installation and Configuration

Generated: 2/11/2021 10:54 pm

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Planning your installation

Overview of the installation process

Overview

This article describes the basic steps to install Nexthink in a corporate network. The current article includes pointers to other articles in the Installation and Configuration manual that explain each step in detail. Accordingly, the Installation and Configuration manual is largely organized to follow the sequence of steps presented in this article. However, because a particular installation of Nexthink depends on multiple factors, such as the number of licensed devices, the network infrastructure, the connectivity of the Appliances, etc., the exact procedure may vary from one customer to another. For example, the order of the steps changes when installing Nexthink for a proof of concept (POC).

For guaranteed satisfaction, contact Customer Success Services to help you install and deploy Nexthink throughout your organization. The intervention of Customer Success Services may be required in the case of installations with particular needs.

From a high-level view, the installation of Nexthink follows these steps:

1. Installing the Appliances.
2. Installing and deploying the Collectors.
3. Installing the Finder.

Once explained how to install Nexthink, the article ends with a couple of pointers to the most common configuration and maintenance activities that come after the installation procedure.

Installing the Appliances

The server components of Nexthink are the following:

- The Portal.
- The Engine.
- The Web Console.

A Nexthink Appliance is a virtual or physical machine that holds an instance of

either the Portal or the Engine, as well as an instace of the Web Console.

The Portal and Engine components obey a master/slave architecture, with the Portal being the master component and one or more Engines being the slaves. A typical installation of Nexthink is thus made up of one or several Engine Appliances that connect to a single Portal Appliance. For small setups, it is possible to host the Portal and one Engine in a single appliance.

The Web Console is a helper component that is installed on every Nexthink Appliance, alongside the Portal or the Engine. The purpose of the Web Console is to help you manage your Appliances and configure the specific settings of the Portal and the Engine, as well as globally controlling some features of the Finder.

Although Nexthink Appliances are typically installed on premises, images for both Microsoft Azure and Amazon Web Services are available to install Nexthink on the cloud. Refer to the appropriate specialized articles to install your Nexthink Appliances on the computation nodes supplied by your favorite cloud provider.

To install the Nexthink Appliances:

1. Review the hardware and connectivity requirements for both the master and slave Appliances.
    - ♦ The hardware requirements of each Appliance depend mainly on the number of devices that they must support and on the activated optional features.
    - ♦ The connectivity requirements are the same for all Appliances of the same type (master or slave).
2. Select a reference architecture.
    - ♦ Base your setup on one of the reference archictectures proposed, depending on the size and characteristics of the IT infrastructure in your organization.
3. Install the Appliances.
    1. Get the network parameters (IP address and subnet) for each Appliance ready, including the proxy settings (if using a proxy to connect your Appliances to the Internet).
    2. Install one Portal and all the required Engines. Prefer an online installation whenever possible.
        - ◊ Add more Engines later, as needed, if your requirements grow.
    3. Log in to the Web Console of each Appliance for the first time and configure its accounts.
    4. Set the names of the Portal and Engine Appliances from the Web Console. These are the FQDN names that will be registered by the

DNS and in the digital certificates that secure the communication with the Appliances.

4. Federate the installed Appliances.
    - The federation establishes the master/slave dependence of Portal and Engines and creates a public key infrastructure (PKI) that later serves to secure the communication with the Collectors.
5. Connect the master Appliance to the slave Appliances.
    - Once federated, connect the Portal to the Engines to collect their data daily.
6. Request and install a license.
    - To enable the product, request a license, install it from the Portal and distribute the licensed devices among the connected Engines.

As a best practice, repeat the previous procedure to install the Nexhtink Appliances for test, QA and production environments.

Installing and deploying the Collectors

The Collector is the light-weight Nexthink component that gathers hardware, software and activity data from the devices within your organization. The Collector also enables the engagement of the end-user through feedback retrieval as well as remotely acting on the device when required. As such, deploy the Collector to all corporate devices that run a supported version of either Microsoft Windows or Apple macOS operating systems.

Instructions on enterprise deployment of the Collector are given for Microsoft SCCM, Active Directory Group Policy (GPO), and VMWare Airwatch. These tools are only required for the initial installation, as subsequent upgrades can be automatically managed by the product when Nexthink makes them available.

Applies to platforms:
Assigning Collectors to Engines

Starting from V6.19, the rule-based Collector assignment feature greatly simplifies the deployment of Windows Collectors in multi-Engine environments:

1. Generate a single Collector installer that points to the Portal instead of multiple installers that point to a different Engine each.
2. Write a set of rules to assign a different Engine to each group of Collectors and let the Portal manage the assignment process.

These rules also replace the conventional method of assigning Collectors, either Windows or Mac, to entities, which constitute the basis of your hierarchies (see

the next steps below).

Applies to platforms:

Installing the Finder

The Finder is a rich-client Windows application that lets you query the Engine in real-time and visualize the results either as lists of records or through convenient graphical views. The Finder is also the tool that enables the creation of metrics, which are displayed as widgets in the dashboards of the Portal.

The easiest method for each Nexthink user to install the Finder on a Windows device is by downloading the Finder installer from the Portal.

Applies to platforms:

Next steps

Once the installation of Nexthink is complete, these are the most common activities that usually follow:

1. Replacing the certificates.
   Replace the default self-signed certificates that the server components of Nexthink use to identify themselves and, optionally, the certificates generated by the federation to secure the communication with the Collectors. Substitute them for certificates signed by either a public or an in-house CA.
2. Define a hierarchy.
   Organize your corporate infrastructure into levels and domains to delimit the view of the different user groups over it.
3. Adding more users
   Because working with a single admin account is neither secure nor convenient, create the accounts required for other users to log in to Nexthink. Define profiles and roles to assign different responsibilities and separate groups of users.
4. Backup procedures.
   Prepare your Appliances for recovery in case of disaster.

To get the most out of your Nexthink setup, browse the rest of topics in the Installation and Configuration manual. Find other configuration settings and customization procedures to help you adapt Nexthink to your specific needs.

Related tasks

- Installing the Appliance

- Installing the Appliance on Azure
- Installing the Appliance on Amazon
- Managing Appliance accounts
- Setting the names of the Portal
- Setting the names of the Engines
- Federating your Appliances
- Connecting the Portal to the Engines
- Setting up a software license
- Installing the Collector on Windows
- Assigning Collectors to Engines
- Installing the Collector on macOS
- Installing the Collector for a POC
- Updating the Collector
- Installing the Mobile Bridge
- Installing the Finder
- Importing and replacing certificates
- Hierarchizing your infrastructure
- Adding users
- Planning for disaster recovery

Related references

- Hardware requirements
- Connectivity requirements
- Reference architectures

Hardware requirements

Hardware requirements

Nexthink Appliance

The Appliance consists of a Linux-based 64-bit operating system and all the packages needed to run one of the server components of Nexthink: the Portal or the Engine. The Portal and the Engine must be installed in separate physical or virtual machines, except for some small setups, where they can share the same appliance. When installed in virtual machines, hardware requirements may vary depending on the load of the infrastructure.

Nexthink officially supports the following virtualization platforms:

- **VMware ESXi** versions 5.5 and later.
- **Hyper-V** on Windows Server (only VMs of type *Generation 1*).

But both the Portal and the Engine may run on any other virtualization platform of your choice. Beware that some versions of popular virtualization platforms may impose particular limits on the number of CPUs and amount of RAM that you can assign to a virtual machine. In installations with many devices, the possible maximum values may not reach the specified requirements. Likewise, in virtualized environments with high load, the performance of IO operations may not be sufficient for the Portal or the Engine to write to disk normally. In case of doubt, please contact Nexthink Customer Success Services to validate your virtualized setup.

In all cases, your servers must be powered by 64-bit compatible processors (AMD64 or Intel 64 -not Itanium- architecture). The vast majority of AMD and Intel processors currently available in the market comply with this requirement. Because of the high memory bandwidth demands of the Portal and the Engine, the installation on machines with a NUMA (Non-Unified Memory Access) architecture is not supported.

For installations on the cloud, the required types of Azure or AWS compute nodes are indicated.

Note: Appliances bound to the Nexthink Cloud offering (SaaS) are under the exclusive control of Nexthink and do not necessarily conform to the hardware requirements listed below.

General considerations

The following definitions apply to all tables of requirements detailed below, for both virtual or physical appliances:

CPU cores

> Number of CPU cores required by the Appliance. The reference model for a CPU core in a physical Appliance is a single CPU core of an Intel Xeon E5-2695 v3 @ 2.3 GHz. Fewer CPU cores may be required when using newer or faster CPUs in an Appliance. Likewise, depending on the measured performance of a specific setup and its particular CPU models, Nexthink may ask customers to increase the number of CPU cores in the Appliance to keep system usability up to acceptable levels. Always validate with Nexthink Customer Success Services in case of doubt.

Memory

The amount of RAM required by the Appliance. As for the type of RAM, the minimal requirement for all configurations is DDR3-1600 with data rate of 1600 MT/s. Dedicate RAM exclusively to the Nexthink Appliance and avoid using shared memory, which may negatively impact the performance.

Many aspects of the usage of Nexthink and your infrastructure can affect the performance and requirements of your appliances: the total number of devices and their actual activity; the number of defined metrics, services, categories, scores, and remote actions; the type of hypervisor, the load of other VMs, the vCPU to pCPU ratio and the IOPS, when running on a virtual appliance. All of them have an impact on the performance and the amount of data that can be kept on the Nexthink appliances.

As the number of possible combinations is unmanageable, always validate your settings with the help of Nexthink Customer Success Services in case of doubt.

Portal requirements

To help you size the Appliance for hosting the Portal, we define a metric called *complexity*. The complexity of a setup, along with the number of licensed devices, gives you an idea of the computation power required by the Portal for that particular setup:

```
complexity = entities * hierarchies * (max_levels + 2)
```

Where:

- **entities** is the total number of entities across all Engines;
- **hierarchies** is the total number of hierarchies;
- **max_levels** is the number of user-defined hierarchy levels for the hierarchy which has the largest number of levels (excluding both the root level and the entity level).

If you have already defined your hierarchies (in a pre-production environment, for instance), find these numbers in the Portal by logging in as administrator and

navigating to **ADMINISTRATION Hierarchies**:

Note the total number of entities (1) and the number of levels in the selected hierarchy (2) in the example above: 33 entities and 2 levels (**Region** and **Country**). If you have defined more than one hierarchy, select the hierarchy with the highest number of levels to use this number as value for **max_levels** in the formula.

According to the size and complexity of your setup, the hardware requirements of the Appliance that hosts the Portal are the following:

Max devices	**Max complexity**	**Memory**	Disk 500 metrics 1000 metrics	Details (90 days) 500 metrics 1000 metrics	**CPU cores**	**Network**
150 k	60 000	59 GB	1 TB 2 TB	1 TB 2 TB	8	1 Gbps
100 k	40 000	41 GB	600 GB 1.2 TB	700 GB 1.4 TB	6	1 Gbps
50 k	12 000	23 GB	300 GB 600 GB	450 GB 900 GB	6	100 Mbps
20 k	4 000	17 GB	200 GB 400 GB	220 GB 440 GB	4	100 Mbps
10 k	2000	13 GB	100 GB 200 GB	120 GB 240 GB	4	100 Mbps
5 k	2 000	12 GB	60 GB 120 GB	60 GB 120 GB	2	100 Mbps

- Ask Nexthink Support for setups with more than 150k devices.
- The Portal requires at least 10 MB/s of disk throughput.
- The total number of entities across all Engines is limited to 8 000.

- The table shows the disk space required for a default maximum of 500 enabled metrics in green and for an absolute maximum of 1000 enabled metrics in red. The relation between disk space required and number is basically linear, so that doubling the number of metrics requires the double of disk space. Nexthink recommends to increase the default maximum number of enabled metrics gradually.

The quantities in the **Details** column correspond approximately to the additional disk space required to store 90 days of historical details of count metrics. Add the value in the **Disk** column to the value in the **Details** column to get the total disk space required. For more information, see the article about data retention in the Portal.

Once you have an Appliance that meets the requirements of the Portal, configure the Portal to allocate your hardware resources and make the most out of them.

Virtualized Portals

The same hardware requirements stated above apply to Portals in virtualized environments. The recommended types of virtual machines for installing the Portal on the supported cloud platforms are the following:

Configuration		Cloud platform	
Max devices	**Max complexity**	**Azure**	**AWS**
150 k	60 000	Standard_E8s_V3	r5d.2xlarge
100 k	40 000	Standard_DS13_V2	r5d.2xlarge
50 k	12 000	Standard_DS4_v2	r5d.2xlarge
20 k	4 000	Standard_E4s_v3	r5.xlarge
10 k	2000	Standard_DS3_V2	r5.xlarge
5 k	2 000	Standard_DS3_V2	r5.large

For more information about the requirements of cloud platforms, see the installation instructions on Azure and AWS.

Engine requirements

The following table holds the hardware requirements of the Engine for a few representative configurations:

Max events	Max devices	Max entities	Memory	Disk	CPU cores	Network	Disk throughput

200 M	10 k	500	26 GB	250 GB	14	1 Gbps	SSD
100 M	10 k	100	18 GB	200 GB	10	1 Gbps	25 MB/s
50 M	3 k	100	8 GB	100 GB	5	100 Mbps	10 MB/s

- Get in contact with Customer Success Services if you need to go over the limit of 200 million events.
- SSD drives are required for setups with more than 100 million events not only because of their high throughput but also because of their faster random accesses, which are critical for the correct functioning of the Engine. These are the reference specifications that your disk setup should match:
    - Block size for single operation: 4 Kb
    - IOPS: 20 000 (**random** access)
    - Bandwidth: 80 MiB/s
    - Latency: 0.1 ms
- If you install the Collector in servers, take into account for the sizing of the Engine that a single server is roughly equivalent to 20 normal devices.
- The indicated number of cores include 20 simultaneous Finder users. If more than 20 users access Nexthink Engine simultaneously, 1 additional core is required for each 5 users (up to a maximum of 24 cores).
- Tests under controlled conditions have demonstrated that the Engine is capable of dealing with up to 100 *normalized* Finder users when run on a 24 cores appliance (20 users for the first 8 cores + 16 cores * 5 users per core).
- A *normalized* user is characterized for querying the Engine once every 25 seconds with a query that takes 10% of a core dedicated to the Engine. If Finder users deviate too much from this behavior, the number of supported users may vary drastically. Note as well that any other kind of query to the Engine (such as queries to the Web API) reduces the number of supported users.
- The maximum number of supported mobile devices for all Engine configurations is 5 000.
- For Engines with Nexthink Act and more than 8000 devices, allocate an extra half GB of RAM (already accounted for in the table above).

Because memory consumption in the Engine highly depends on the usage of Nexthink, the most difficult hardware requirement of the Engine to adjust is the RAM. Once you have made a rough estimation of your hardware needs for the Engine, fine tune your RAM requirements by computing their value with the help of the following formula:

```
RAM [bytes] = #Events * 60 +
              #Devices * 500,000 +
```

```
#Entities * #Services * 80,000 +
3 GB (system and proxy) +
0.5 GB (for Nexthink Act if #Devices > 8000)
```

Remember the maximum values per Engine for each parameter that you can enter in the formula:

```
#Events = 200 million
#Devices = 10,000
#Entities = 500
#Services = 100
```

Virtualized Engines

The same hardware requirements apply to Engines in virtualized environments. The following table completes the representative configurations in the table above with the recommendations for virtual and cloud platforms:

Configuration		Virtual platform	Cloud platform	
Max events	**Max devices**	**MHz allocation to resource pool**	**Azure**	**AWS**
200 M	10 k	from 1 to 2 Ghz per vCPU	Standard_F16s_v2	c5.4xlarge with SSD gp2
100 M	10 k	from 1 to 2 Ghz per vCPU	Standard_F16s_v2	c5.4xlarge with SSD gp2
50 M	3 k	from 1 to 2 Ghz per vCPU	Standard_F8s_v2	c5.2xlarge with HDD sc1

The number of cores per socket must be one for vCPUS. Regarding the allocated MHz, it is fine to go with the lower value of the range if there is proper monitoring of the infrastructure in place. In case of performance issues, Support will ask you the monitoring information (CPU ready ratio, co-stop, RAM usage, ...) and will most probably ask you to increase your settings.

Regarding memory requirements, as the Engine is an in-memory database, it really depends on the way the hypervisor will address memory overcommitment with our appliance OS and Engine process. If the hypervisor can find and consolidate memory pages with identical content from the Engine VMs on the same host, it could be ok to overcommit. Again, in case of performance issues, Support will ask you the monitoring information about memory usage and overcommitment and may ask you to increase your settings.

Because the Portal mainly identifies the Engine appliances through the MAC address of their network cards for licensing purposes, it is important that the MAC address of your virtual appliances does not change with time. Use static assignment of MAC addresses on all your virtual appliances to avoid licensing issues, especially when rebooting the machines.

For more information about the requirements of cloud platforms, see the installation instructions on Azure and AWS.

Running Nexthink on a single appliance

For very small installations, the Portal and the Engine can run on the same physical or virtual appliance.

Max devices	1 000
Max complexity	2 000
Events	20 M
Memory	19 GB
Disk capacity	120 GB
Disk write speed	10 MB/s
CPU cores	6
Network	100 Mbps

Nexthink traffic redirection service

The Collector traffic redirection service (`nxredirect`) is a tool included in the Engine appliance that resends activity information (UDP traffic) received from the Collectors to one or more additional Engines. Optionally, the redirection service is able to anonymize sensitive Collector data on the fly.

The hardware requirements of `nxredirect` depend on the service being run alongside the Engine or in an appliance where the Engine has been stopped:

Nxredirect is run alongside the Engine
: The maximum number of supported devices is 10 000 without anonymization, or 5 000 if anonymization is switched on. The hardware requirements of the Engine apply (see table above). No additional hardware is needed.

Nxredirect is run independently (i.e. the Engine has been stopped)
: The maximum number of supported devices depends heavily on anonymization being switched on or off, ranging from 5 000 up to

350 000 devices.

Assuming an average traffic per device as indicated in the product overview of the Collector, the hardware requirements of `nxredirect` are as follows:

	Max devices	Anonymization	CPU cores	Memory	Disk
Nxredirect + Engine	5 000	On	Engine reqs	Engine reqs	N/A
	10 000	Off			
Nxredirect alone	5 000	On	2	5 GB	N/A
	350 000	Off			

External backups

The disk space requirements given for the Appliance already take into account the amount of space needed to keep up to ten internal backups of either the Portal or the Engine.

In the case that you activate external backups, Nexthink recommends you to reserve the following quantities of external storage, depending on the size of your setup. The figures indicate the file size for each individual backup.

Nexthink Portal

The backup size for the Portal depends on the number of devices, the complexity, the amount of history and the number of widgets and reports. We recommend regularly monitoring the used capacity and adapting it based on actual needs.

Max devices	External backup size
150 k	50 GB
100 k	30 GB
50 k	15 GB
20 k	10 GB
10 k	5 GB
5 k	3 GB

Nexthink Engine

The disk requirements for the backup of the Engine are more predictable than those of the Portal and only depend on the number of events stored in the Engine.

Max events	External backup size	Network throughput
200 M	16 GB	30 MB/s
100 M	8 GB	25 MB/s
50 M	4 GB	15 MB/s

Data Enricher

The following content applies exclusively to the Nexthink Cloud offering.

For the Cloud offering of Nexthink, the Windows Server that runs the Data Enricher requires the following hardware:

Number of items	CPU cores	Memory	Min network bandwidth
• 100 k - 500 k users • 20 k - 100 k destinations	4	8 GB	25 Mbps
• 20 k - 100 k users • 10 k - 20 k destinations	4	4 GB	25 Mbps
• 1 k - 20 k users • 1 k - 10 k destinations	2	4 GB	25 Mbps

Mobile Bridge

To collect information from mobile devices synchronized via ActiveSync with Microsoft Exchange, the Mobile Bridge uses a Remote PowerShell connection to your Exchange Client Access server.

Install the Mobile Bridge on a dedicated Windows Server 2008 R2 or later. The hardware requirements for the Mobile Bridge are those same ones recommended by Microsoft for installing their operating system. The Mobile Bridge is compatible with Exchange 2010 SP2 or 2013.

Nexthink Collector

	Without Web & Cloud	With Web & Cloud

Disk	35 MB	
Network card	Any, wireless or wired	
Average network bandwidth	100-150 bps	150-250 bps

Nexthink Finder

Starting from Nexthink V6.3, the Finder supports high DPI screens. When setting DPI scaling in Windows, the Finder adapts its size properly.

Memory	4 GB system memory, at least 2 GB available
Disk capacity	50 MB
CPU	2 cores, 2 GHz
Network	100 Mbps recommended

Certified Hardware List

Nexthink V6 appliances include a Linux-based operating system that is derived from the freely distributed sources of a major North American Enterprise Linux vendor. This vendor maintains a list of supported hardware that has been tested and is certified to work with its Linux distribution. To help you choose your hardware for your appliances (the Portal and one or more Engines), verify that it is in the following list:

- Certified Hardware List (Red Hat link)

Related tasks

- Planning for disaster recovery
- Allocating resources for the Portal
- Hierarchizing your infrastructure
- Redirecting Collector traffic

Related references

- Data retention
- Server support
- Collector overview
- Hardware requirements - Installing Windows Server 2008 (Microsoft link)
- Exchange 2013 System Requirements (Microsoft link)

Connectivity requirements

Overview

Find the connectivity requirements of every Nexthink product in the reference tables below. You can configure some of the products to use either a secure or a non secure channel for specific services (see the column **Reason**). Depending on their configuration, note that you may require to allow connections through a different port number.

Starting from V6.19, if rule-based Collector assignment is turned on, the TCP channel of the Collector also connects to the Portal. Collectors use this connection to ask for their assigned Engine. From V6.20 on, if you change the default port number of the Collector TCP channel, modify accordingly the port number where the Portal is listening.

Starting from V6.21, the Collector no longer requires a separate UDP channel to send end-user analytics to the Engine. Instead, end-user analytics, as well as coordination data and updates, may be optionally transmitted through the TCP channel. If you change the default port numbers that the Collector uses for communicating with the Engine, change as well the default port numbers in the Engine through the Web Console. Starting from V6.24, the default is to use TCP port 443 for all Collector communications in on-premises setups, although the use of a custom TCP port (default 8443) and the UDP channel are still allowed.

For each connection, the tables indicate the transport protocol used. When an application protocol handles the connection over the transport layer, the name of the application protocol precedes the name of the transport protocol.

First, find in this overview two diagrams:

- A diagram with the connections and default ports that are common to all Nexthink Appliances, regardless of the Appliance hosting the Portal, the Engine or both.
- A diagram with the default ports of the Portal and Engine Appliances separately, as well as the connections with other components.

Common connections of the Appliance

Connections between Portal, Engine and other components

Connections required for rule-based Collector Assignment

Starting from V6.19, the following additional connections are required if rule-based Collector assignment is turned on. Federate your appliances before activating rule-based Collector assignment.

The connectivity between Engines through TCP and UDP ports 8301 is optional, as the consensus protocol behind rule-based Collector assignment uses these connections to implement a feature that is actually not required by Collector assignment. If communication through TCP and UDP ports 8301 is blocked between Engines (by internal firewalls, for instance), the underlying consensus protocol will write failed connection messages to its log file:

```
/var/nexthink/nxconsul/logs/nxconsul.log
```

You can safely ignore these error messages.

Engine

In the following table, we describe the different ports that must be open on the Engine appliance to communicate seamlessly with the other Nexthink components and with standard network services.

Port Number	Protocol	Direction (IN/OUT)	Reason	Domai
22	SSH / TCP	IN	Secure shell connection to the CLI	
	SSH / TCP	IN OUT	Appliance federation	
25	SMTP / TCP	OUT	Mail server for notifications	
53	DNS / UDP	OUT	Resolving destination names by reverse IP	
99	HTTPS / TCP	IN	Administration through the Web Console	
123	NTP / UDP	OUT	Time synchronization	0.centos.pool.ntp.org 1.centos.pool.ntp.org

				2.centos.pool.ntp.org
389	LDAP / TCP	OUT	Connection to Active Directory (non secure)	
443	WebSocket / TCP	IN	Collector TCP channel to the Engine (on-premises default)	
	WebSocket / TCP	IN	User connection from the Finder (Nexthink Cloud only)	
	HTTPS / TCP	IN	Audit Trail API connection from the Portal	
	HTTPS / TCP	IN	Access to the Web API	Only for Engines on the
	HTTPS / TCP	OUT	Connection to the Application Library	application library v application library v
HTTPS / TCP	OUT	Connection to automatic updates	updates v6.nexthink.com updates centos v6.nexthink.com	
636	LDAPs / TCP	OUT	Connection to Active Directory (secure)	
999	UDP	IN	Optional: Collector analytics	
	TCP	IN	User connection from the Finder (on premises only) or the Portal	
1671	HTTPS / TCP	IN	Access to the Web API	Only for Engines on pren
7000 7001 7002 7003	TCP	OUT	Communication channels with the Portal	
8300	TCP	IN OUT	Communication with Portal for Collector assignment	
8301	TCP & UDP	IN OUT	Communication with Portal and peer Engines for Collector assignment	
8443	WebSocket / TCP	IN	Collector default custom / Nexthink Cloud TCP channel to the Engine	
10402	TCP	OUT	Additional communication with Portal for Collector assignment	
11031	HTTPS / TCP	OUT	Communication with the Mobile Bridge	

Portal

In the following table, we describe the different ports that must be open in the Portal appliance to communicate seamlessly with the other Nexthink components.

Port	Protocol	Direction	Reason	Domains

Number		(IN/OUT)		
22	SSH / TCP	IN	Secure shell connection to the CLI	
	SSH / TCP	IN OUT	Appliance federation	
25	SMTP / TCP	OUT	Mail server for notifications	
53	DNS / UDP	OUT	Lookup name of AD servers	
80	HTTP / TCP	IN	Access to the Portal (non secure)	
88	TCP & UDP	OUT	Kerberos authentication of AD users	
99	HTTPS / TCP	IN	Administration through the Web Console	
	HTTPS / TCP	OUT	Centralized administration of the Engine	
123	NTP / UDP	OUT	Time synchronization	0.centos.pool.ntp.org 1.centos.pool.ntp.org 2.centos.pool.ntp.org
389	LDAP / TCP	OUT	Connection to Active Directory (non secure)	
443	HTTPS / TCP	IN	Access to the Portal (secure)	
	WebSocket / TCP	IN	User connection from the Finder	
	WebSocket / TCP	IN	Collector TCP channel to the Portal (on premises default)	
	HTTPS / TCP	IN	Installation and updates of the Finder from the Portal	Portal address
	HTTPS / TCP	IN	API of remote actions	Portal address

	HTTPS / TCP	OUT	Connection to the Online License mechanism	license.nexthink.com
	HTTPS / TCP	OUT	Connection to the Application Library	alib.nexthink.com application library v5.nexthink.com application library v6.nexthink.com
	HTTPS / TCP	OUT	Connection to automatic updates	updates v6.nexthink.com updates centos v6.nexthink.com
636	LDAPs / TCP	OUT	Connection to Active Directory (secure)	
999	TCP	OUT	Connection to the Engine	
7000 7001 7002 7003	TCP	IN	Communication channels with the Engine	
8100	HTTP / TCP	OUT	Send license information to Local License Manager	
8300	TCP	IN OUT	Communication with Engines for Collector assignment	
8301	TCP & UDP	IN OUT	Communication with Engines for Collector assignment	
8443	WebSocket / TCP	IN	Collector default custom / Nexthink Cloud TCP channel to the Portal	
10402	TCP	IN	Additional communication with Engines for Collector assignment	

Local License Manager

The Local License Manager resides in the same machine as the Portal.

Port Number	Protocol	Direction (IN/OUT)	Reason
8100	HTTP / TCP	IN	Get license information from the Portal

Mobile Bridge

The Mobile Bridge needs to connect to the Exchange CAS to get mobile information. In turn, it offers a REST interface for the Engine to use to retrieve the collected information.

Port Number	Protocol	Direction (IN/OUT)	Reason
80	HTTP / TCP	OUT	Communication with Exchange (non secure)
443	HTTPS / TCP	OUT	Communication with Exchange (secure)
11031	HTTP / TCP	IN	REST interface for the Engine

Finder

In the following table, we describe the different ports that must be opened on the computers running the Finder to communicate seamlessly with the other Nexthink components.

Port Number	Protocol	Direction (IN/OUT)	Reason	Domains
25	SMTP / TCP	OUT	Send email in case of error	
80	HTTP / TCP	OUT	Connection to the documentation web site	doc.nexthink.com
	HTTP / TCP	OUT	Verification of security certificates	ocsp.verisign.com
443	WebSocket / TCP	OUT	User connection to the Portal	
	WebSocket / TCP	OUT	User connection to the Engine (Nexthink Cloud only)	

	HTTPS / TCP	OUT	Installation and updates of the Finder from the Portal	Portal address
	HTTPS / TCP	OUT	Support telemetry	alib.nexthink.com
	HTTPS / TCP	OUT	Connection to the Library	library.nexthink.com
999	TCP	OUT	User connection to the Engine (on premises only)	

Collector

In the following table, we describe the different ports that must be opened on the computers running the Nexthink Collector to send data seamlessly with the Nexthink Engine.

Port Number	Protocol	Direction (IN/OUT)	Reason
999	UDP	OUT	Optional: Collector UDP channel to the Engine
443	WebSocket / TCP	OUT	Collector default (on premises) TCP channel to the Engine and, if rule-based Collector assignment is turned on, to the Portal
8443	WebSocket / TCP	OUT	Collector default custom / Nexthink Cloud TCP channel to the Engine and, if rule-based Collector assignment is turned on, to the Portal

Applies to platforms:

In addition, starting from V6.19, Windows Collector components call a Windows API method once every 24 hours that triggers a connection for client to domain controller operations through TCP port 135. Ephemeral TCP ports in the range 49152-65535 are used for service response.

Applies to platforms:

Data Enricher

The following content applies exclusively to the Nexthink Cloud offering.

The Windows Server that runs the Data Enricher requires the following communication channel to reach the Nexthink Cloud. The ports for connecting to Active Directory through a secure or insecure channel are configurable (Active Directory default port numbers are shown).

Port	Protocol	Direction	Reason	Domain

Number		(IN/OUT)		
53	DNS / UDP	OUT	Resolving destination names by reverse IP	
389	LDAP / TCP	OUT	Connection to Active Directory (non secure)	
443	HTTPS / TCP	OUT	Send AD and DNS data	agora.*reg*.nexthink.cloud (*reg* is the availability region of the customer)
636	LDAPs / TCP	OUT	Connection to Active Directory (secure)	

Related tasks

- Federating your Appliances
- Installing the Data Enricher

Related references

- Changing the default ports in the Appliance

Software requirements

Portal and Web Console supported operating systems

Windows
macOS

Other combinations of browsers and operating systems may be suitable for both the Portal and the Web Console, but they have not been thoroughly tested. Some elements of the graphical interface may appear unusable or disproportionate in size in unsupported combinations of browsers and operating systems.

Portal supported browsers

Microsoft Edge 38 and later
Mozilla Firefox 68 and later
Google Chrome 30 and later

Microsoft Internet Explorer 11 support is ending soon. From January 1, 2021, some features may still work, but they are not guaranteed to work in the long term (read more).

Web Console Supported Browsers

Microsoft Edge 79 and later.
Mozilla Firefox 68 and later.
Google Chrome 30 and later.

Known issues

Support for Microsoft Edge prior to version 79.0.309 and Internet Explorer 11 is ending. While some features may still work at the moment, they are not guaranteed to work in the long term.

Finder supported operating systems

Windows 10 - 64 and 32 bits
Windows 8.1 - 64 and 32 bits

Windows Collector supported systems

Desktop

Windows 10 - 64 and 32 bits
Windows 8.1 - 64 and 32 bits
Windows 7 SP1 - 64 and 32 bits - requires OS update (see prerequisites below)

Server

Windows Server 2019 - 64 bits and Windows Server, version 1809
Windows Server 2016 - 64 bits and Windows Server, version 1709
    Editions: Standard, Essential, and DataCenter.
    Installation options: desktop experience and core.
Windows Server 2012 and 2012 R2 - 64 bits
Windows Server 2008 R2 - 64 bits
    Requires OS and .NET Framework updates

Prerequisites

- The targeted device must be able to directly or indirectly reach the Engine and, optionally, the Portal through the network.
- To install the Collector using their MSI, the Windows installer (msiexec) version 3.0 or higher is required.
- The installation of the Collector 6.24.1.0.x on Windows 7 (32 and 64 bit) and Windows Server 2008 R2 (64 bit) devices requires update Microsoft KB 3033929 to be previously applied.
- On Windows Server 2008 R2, the Nexthink Act module requires to install .NET Framework 3.5 SP1.
- The Collector auto-update feature only works when the Nexthink Appliances have access to the Internet.

Compatibility with the Engine

The latest Engine accepts traffic from the latest version of the Collector and all its previous versions. Before upgrading to the latest release of the Collector, upgrade your Engines first.

Mac Collector supported systems

macOS 10.15 Catalina
macOS 10.14 Mojave
macOS 10.13 High Sierra

Prerequisites

The targeted device must be able to directly or indirectly reach the Engine and, optionally, the Portal through the network.

Compatibility with the Engine

The latest Engine accepts traffic from the latest version of the Collector and all its previous versions. Before upgrading to the latest release of the Collector, upgrade your Engines first.

End of support for Internet Explorer 11

As of January 1, 2021, Nexthink will no longer support Internet Explorer 11 (IE11) or Microsoft Edge Legacy to access the content provided by the Nexthink Portal. After that date, the Portal user interface might not function correctly when using Internet Explorer 11 or Microsoft Edge Legacy.

This does not affect the ability of the Nexthink Collector to monitor activity or web traffic of Internet Explorer 11 or Microsoft Edge Legacy.

When will this happen?

Event	Date
Announcement of the removal of support for IE11 and Microsoft Edge Legacy	16.10.2020
Support ends	01.01.2021*

** We are still collecting feedback and this date may change. If you have any comments, please get in contact with us via one of the means at the bottom of this page.*

What does this mean?

Currently, the Nexthink Portal supports the use of Internet Explorer 11 or Microsoft Edge Legacy as a web browser. As of January 1, 2021, this will no longer be the case. Users of Internet Explorer 11 and Microsoft Edge Legacy might find that the Nexthink Portal does not work correctly, or at all. Microsoft Edge Legacy is any version of Microsoft Edge which uses the EdgeHTML browser engine. This was deprecated in favor of a Chromium-based browser engine as of version 79.0.309 of Microsoft Edge.

As an important distinction, this change does not affect the ability of Nexthink to monitor Internet Explorer 11 or Microsoft Edge Legacy activity on endpoints.

Why is Nexthink removing support for Internet Explorer 11 and Microsoft Edge Legacy?

In 2015, the Edge browser was released by Microsoft. Internet Explorer 11 no longer receives major updates to support functionality that allows modern and complex web applications such as Nexthink to work. In January 2020, Microsoft Edge received an upgrade to use a Chromium-based browser engine, and legacy versions no longer receive major updates. As a result, there is an increasing gap between the capabilities of Internet Explorer 11/Microsoft Edge Legacy and other browsers.

An increasingly large amount of effort is required to make the Nexthink Portal work for Internet Explorer 11 and Microsoft Edge Legacy, an effort that could be better spent developing useful features, reliability, and functionality that Nexthink users expect.

Who is affected?

Anyone who uses Internet Explorer 11 or Microsoft Edge Legacy to access the Nexthink Portal content.

How should I prepare for this?

Use a more modern web browser like Google Chrome, Mozilla Firefox, or modern Microsoft Edge.

Who should I contact if I have questions?

Contact Nexthink Support or your customer success manager for more information.

Reference architectures

Overview

Installing Nexthink requires taking architectural decisions with respect to the location of the Nexthink components and their connectivity. The choice of a particular architecture depends mainly on the geographical distribution of the assets and the network topology of an organization. Whereas global organizations have assets distributed all over the world, with regional offices typically interconnected through dedicated lines or VPN technology, local organizations have all or most of their assets placed in a single location and connected to a single LAN. The appropriate architecture for each type of organization will thus be different, although some basic architectural principles stay the same for all kinds of installations.

To help you choose the right architecture for your organization, consider the following factors and possible scenarios:

1. Location and connectivity of Appliances
    - ♦ Appliances are placed in one location on premises (recommended).

        - ◊ Appliances can connect to the Internet (recommended online installation).
        - ◊ Appliances have no access to the Internet (offline installation).
    - ♦ Appliances are in several geographically dispersed locations on premises (not recommended).
    - ♦ Appliances are located in external data centers (cloud installation).
2. Location, connectivity and total number of Collectors
    - ♦ Collectors in the intranet.
    - ♦ Roaming Collectors in the Internet (home office, travelling, etc).
        - ◊ Connected through VPN or similar (including Microsoft Direct Access or Always On VPN).
        - ◊ With no VPN connection.
3. Data anonymization
    - ♦ Anonymized analytics.
    - ♦ GDPR compliance.
4. Integrations
    - ♦ Integrations inside the intranet (e.g. SCCM, SMTP, Active Directory)
    - ♦ Integrated software in the Internet (e.g. cloud services such as ServiceNow).

While an exhaustive description of all the possible combinations is beyond the scope of this article, we present below a set of reference architectures on which you can base your own deployment. Choose the reference architecture that suits best to your specific setup.

Basic architectural principles

Once the hardware requirements for running Nexthink are met, the quality of the network connections between the different Nexthink components mainly determines the overall performance and responsiveness of a Nexthink setup. For architectural purposes, we can classify these connections according to the communicating Nexthink components:

- Nexthink visualization and query tools with the Nexthink Appliances:
    - ♦ Finder to Portal and Engines.
    - ♦ Browser (web front-end) to Portal.
- Among Nexthink Appliances themselves:
    - ♦ Portal to Engine.
    - ♦ Engine to Engine.
- Collector with Nexthink Appliances:
    - ♦ Collector with Engine (UDP and TCP connection)

    - ♦ Collector with Portal (TCP connection, if rule-based Collector assignment is used)

With the introduction of the Cross-Engine features in V6.19, a good connectivity between the Finder and the Nexthink Appliances and between the Appliances themselves has become fundamental to offer Nexthink users a satisfactory experience. Apply thus the following recommendations, especially when using the Cross-Engine features:

- Place all your Nexthink Appliances in the same data center. If this is not possible, ensure that the connectivity between the Portal and the Engines is equivalent to that of a local network.
- Ensure a good connectivity between the Finder and the Nexthink Appliances, both Portal and Engine.

The rationale for these recommendations in Cross-Engine scenarios is the following:

- All the queries from Finder that require an answer from multiple Engines are routed through the Portal, which gathers and merges the responses of every Engine.
- The slowest of the connections between the Portal and the Engines determines the overall query time (the Portal waits up to 3 minutes for the Engines to respond).
- Finder users will face responsiveness issues and suffer long waiting times if the communication with the Portal and the Engines is not fluid, which leads to a poor user experience.

On their part, the connections between the Collectors and the Appliances are much less demanding in terms of network bandwidth and latency. Therefore, the main concern is to ensure that Collector data reach their intended destination through the network:

- Ensure that Collectors can reach their correspondent Engine and, if rule-based Collector assignment is used, that Collectors can reach the Portal as well.
    - ♦ Use either VPN technology or traffic redirection for Collectors that are not directly connected to the same network as the Appliances.
    - ♦ Configure firewalls and proxies appropriately so they do not block the Collector communications.
- Avoid fragmentation of UDP traffic, as it may cause significant loss of Collector data.

Global organizations

Global organizations extend over several locations in different countries. To create a private network over a wide area, the local area networks of a global organization are connected through dedicated lines or, more often, through VPN technology. A *Virtual Private Network* (VPN) enables devices and servers in distant places to connect through shared or public networks as if they were all directly connected to a single local network.

The reference architecture for a global organization looks thus as follows:

The key points of the reference architecture for global organizations are the following:

- All Nexthink Appliances, whether physical or virtual, reside in the same data center.

- Multiple Engines are required, as global organizations typically deploy a large number of Collectors.
- Regional offices are interconnected to form a single private network (through VPN in the example figure).
- The IT department (Finder and Portal users) is preferably located next to the Nexthink Appliances to have good connectivity.
    - Using the Finder from a distant regional office is still possible, although the Finder may lose responsiveness if connectivity is not good (Finder icon displayed dimmed in the example figure).
- Remote Collectors connect to the global private network through the Internet (using VPN client technology in the example figure).

Appliances in different locations

If for some reason you really need to have Appliances distributed among different locations, place the Portal in the office with the highest number of end-user devices or Engines (generally these two numbers go hand in hand), or with the highest number of Finder users.

The connection between offices should offer enough bandwidth and low latency for Nexthink users to work comfortably with the Finder or with the Portal front-end while connected to the Engines or the Portal located in another office. Remember though that this is not the recommended architecture for global organizations, especially in the case that Cross-Engine features are enabled.

Regional and Local organizations

For regional organizations with several data centers, preferably place your Nexthink Appliances in the data center where most of your Finder users work. For local organizations that do not extend over several locations, all Nexthink

Appliances are naturally placed together in the same data center.

Depending on the total number of deployed Collectors, install one or several Engines and one Portal on separate physical or virtual appliances and federate your slave Appliances (Engines) with the master (Portal).

Small local organizations

For setups with fewer than one thousand end-user devices, it is possible to host both the Portal and the Engine in a single appliance, as depicted below.

Remote or roaming devices

When the end-user devices and the Nexthink Appliances are in the same intranet, every machine can directly reach each other over the network: Collectors can talk to their assigned Engine and, in turn, Engines can communicate with the Portal because they all reside in the same intranet.

The fact that the intranet is implemented as a Local Area Network (LAN) in a single office or as a Wide Area Network (WAN) extending over several regional offices is probably important for network performance, but irrelevant to our discussion. It does not matter either if the Nexthink Appliances are deployed on physical or virtual servers. The most important property for a simplified deployment is to have direct connectivity among computers, as it is the case in an intranet.

Regardless of the size of an organization, it is common these days to have employees working from remote locations. The reasons may vary: home office, commuting, visiting customers, etc. When an end-user device is outside the corporate network, the Collector running on the device may lose the connectivity to its assigned Engine, as the Engine is usually not reachable through the Internet because of security reasons. If the Collector cannot reach the Engine, activity information is irremediably lost while the device is roaming.

Running VPN client software

One way for the Collector to not lose the connectivity with the Engine is to run VPN client software on the roaming device so that it always stays connected to the corporate network, even when the device is out of the office. Of course, the use of VPN client software in your roaming devices requires that you have a VPN infrastructure ready in your corporate network. Establishing a VPN connection is the preferred solution to deal with roaming devices if you already use VPN technology.

For instance, if you have Microsoft DirectAccess (for Windows 7 clients) or Always On VPN (for Windows 10 clients) or any other VPN technology, benefit from it to keep the Collector connected to the Engine on roaming devices. Remember to configure your Nexthink setup to support DirectAccess if you use this particular solution.

See the diagram on global organizations for an example of roaming devices connected through VPN.

Forwarding UDP Collector traffic

As an alternative to VPN technology, configure a Nexthink Appliance to forward Collector traffic to your Engines. The redirection service is a feature in the Engine Appliance that lets you forward UPD traffic from the deployed Collectors to one or more instances of the Engine, optionally anonymizing sensible data on the fly.

To forward Collector traffic from the Internet to your corporate network, place the additional Engine Appliance in the DMZ and use it exclusively for redirection.

Because the Collector is configured to point to a specific Engine, the DNS name of the Appliance that performs the redirection in the Internet must match the DNS name of the Engine in the corporate intranet. It is therefore mandatory for this case to use DNS names and not IP addresses to configure the External DNS name of the Engine and the Collector assigned Engine. In this way, regardless of the device being inside the corporate intranet or out in the Internet, the Collector is pointing to the correct DNS name of the Engine.

In Nexthink, the last IP address of a device is determined by the source address of the UDP datagrams that it sends to the Engine. While roaming without a VPN, however, a device is usually behind a NAT router that hides the private IP address assigned to the device. Moreover, the redirection service can modify the source IP address in the UDP datagrams of roaming devices, so they can be redirected to the intranet without being rejected. This has implications on how devices are assigned to entities and Engines depending on whether devices are roaming or not. Contact the Customer Success Services of Nexthink to find out the best configuration possible for your particular case.

Starting from V6.21, the redirection service can deal with data sent through either the UDP or the TCP channel of the Collector, but it exclusively handles the end-user data part of the TCP channel. Therefore, features such as Engage, Act or the automatic updates, which are sent through the TCP channel, do not work for roaming devices with the redirection service. If you need these features to work on roaming devices, prefer a VPN solution or configure a TCP reverse proxy as indicated in the next section.

Forwarding TCP Collector traffic

To enable features such as the Engage or Act modules and the automatic updates on roaming devices that cannot connect through VPN, enable the TCP channel between the Collectors and the Engine by installing a reverse proxy in the same Appliance that redirects Collector traffic.

Starting from V6.21, if all your Collectors send all their data through the TCP channel only, you do not require the redirection service (`Nxredirect`) to be running on the Appliance, as no UDP traffic is sent and the reverse proxy configuration is able to redirect the full TCP channel of the Collector.

Connection to online services

Independently of your particular setup and the reference architecture that you choose, the deployed Nexthink Appliances must connect to the online services provided by Nexthink to receive updates, get security information about binaries and domains, and validate their licenses.

Moreover, the Appliances optionally connect to NTP servers to synchronize their clocks. By default, the Appliances that run the Portal or the Engine connect to official CentOS NTP servers. Change the configuration in the Web Console if you prefer to synchronize with NTP servers that are geographically closer to your Appliances, or even located within your own corporate network.

In addition to the Nexthink Appliances, the device that runs the Finder must have access to the Nexthink online services to download library packs from the Nexthink Library.

Offline Nexthink Appliances

In setups with special security concerns, Nexthink Appliances have no direct connection to the Internet. To update your offline Appliances, mirror the Nexthink repository in a server under your control and then apply the updates to the Appliances.

Because mirroring the Nexthink repository requires a formal agreement with Nexthink, contact your Nexthink representative or Customer Success Services to proceed with this method and get technical assistance.

Related tasks

- Federating your Appliances
- Redirecting Collector traffic
- Redirecting the Collector TCP channel
- Setting the names of the Engines
- Enabling Cross-Engine Finder features

Related references

- Connectivity requirements
- Support for DirectAccess

Installing Portal and Engine Appliances

Installing the Appliance

Installing the Appliance on a Physical Server

To install the Appliance on a physical server, enter the BIOS and modify the following settings:

1. Power on the server and enter the BIOS setup. This is usually done by pressing down [F2] or [DEL] keys before the computer attempts to boot the operating system. Explore the exact method to enter BIOS setup in your user manual.
2. In the system settings, set the date and time of the system to match the current UTC (Coordinated Universal Time). Time precision is important to ensure consistency of data in the system. In order to keep time precision, you may configure the Appliance to use NTP servers as described in Network Parameters section below.
3. Insert your copy of the the Nexthink V6 Installation DVD into your DVD Drive.
4. Go to boot sequence (or boot order) settings and select the CD/DVD Drive to be the first device in the list of bootable devices. The system will boot from the Nexthink Installation DVD next time.
5. Exit BIOS setup saving your changes.

Installing the Appliance on Virtual Server

To install the Appliance on a virtual machine, the exact installation steps will depend on your virtualization platform. Here, we assume that you are familiar with the creation and configuration of virtual machines on your virtualization platform. If this is not the case, please take your time to learn how to use it. Independently of the virtualization platform that you are using, perform the following operations:

1. Create a new virtual machine and select CentOS 7 as the guest operating system.
    - If CentOS 7 is not an available option in your virtualization platform, select a generic Linux 2.6 (or higher) 64-bit operating system as guest.
2. Insert the Nexthink V6 Installation DVD into the DVD drive of the host

machine or, alternatively, copy the ISO image of the Nexthink V6 Installation DVD to a file system accessible to the host machine.
3. Indicate your virtualization platform to use the DVD drive or the copied ISO image for booting the your newly created virtual machine for the first time.
4. Start the virtual machine.
5. Explore the user manual of your virtualization platform to find out how to synchronize the clock of the virtual machine to the clock of the host, if the host has precise timing. Alternatively, use NTP servers as described in Network Parameters section below.

Finishing the Installation

When on of the previous steps have been completed, the installation is identical for both physical or virtual servers. If everything went well, your system will now boot from the Nexthink V6 Installation DVD and you should see the Nexthink V6 splash screen.

1. Press [ENTER] or wait for the 6 seconds timeout.
2. After the splash screen, the End-User License Agreement is displayed. Accept to proceed
3. In the following screen, you are warned that your hard disk will be erased during the installation procedure and that all data on it will be lost. Ensure that you do not have any valuable data on the hard disk and type Yes to proceed.
4. The installation procedure whether you want to configure the network. If you wish to configure the network, a small dialog box will open where you may change the default IP address, subnet mask and default gateway. If you choose not to configure your network, the Appliance will take these default values. These can be modified as described in Network Parameters section below.
5. Select the type of keyboard that is attached to your computer. By default, US keyboard is highlighted.
6. Choose a secure password for the root user, ensure that it is kept in a safe place or use another method to make sure that it is not forgotten.
7. The installation procedure is formating your hard disk and installing the necessary software packages.
8. Once the installation has finished, remove the Nexthink V6 Installation DVD from the DVD drive or detach the ISO image from your virtualization platform and reboot the system.

Appliance Configuration

Appliance Network Parameters

1. If nothing has been specified during the installation, the Appliance server default IP address is 192.168.0.99 with a netmask of 255.255.255.0. To access its web-based configuration page, configure your computer to have the static IP address on the same subnet as the Engine default IP address, that is, 192.168.0.0/24. If you modified the network configuration of the server during the installation of the Engine, use your modified configuration values instead of the defaults and then the following steps.
2. With a cross-over cable, connect the first Ethernet port on the server to the Network Interface Port (NIC) on the client computer. Alternatively, if there is a switch between the client PC and the server, connect both devices to it.
3. Test the network connectivity between the client computer and the server using ping command. Execute the ping test using the default IP address of the Engine: 192.168.0.99. If the ping test fails with the first Ethernet port, repeat the test by connecting to each available Ethernet port on the server. By default, the first Ethernet port is used for testing the network connections, but some servers may behave differently.
4. Using a web browser and type the following URL in the address field, using the HTTPS protocol: https://192.168.0.99:99. Note that if you just type the IP address 192.168.0.99:99, your web browser will use HTTP by default and you will get a blank page. Ensure that you specify the HTTPS protocol.
5. Accept the SSL certificate from Nexthink. Depending on the web browser, you may have a warning about the certificate. Accept it and you will arrive at the Console welcome screen:
    1. Enter the Login Username as *admin* and Password also as *admin* to connect to the Web Console for the first time
    2. You are prompted to change the admin password of the Web Console. Type in the old password *admin* and the new password twice. Click **Save changes**.
    3. You are also prompted to change the password of the *nexthink* account (the account for logging in to the CLI of the Appliance via SSH). Type in the old password *123456* and the new password twice. Click **Save changes**.
        - ◊ This is the range of supported characters for changing the password the first time:

          `a-z 0-9 ~%.:_-`
        - ◊ Note that if you have previously changed the default SSH password of the Appliance through the CLI, you will not be

able to change the password again at this point if your precedent password includes any unsupported character.

4. Click the **Appliance** tab at the top of the window.
5. Select the **Network interfaces** section from the left-hand side menu.
6. Click **EDIT** to configure the detected interfaces. Select the type of address, whether **Static** (recommended) or DHCP, the IP address, the subnet mask, and the default gateway.
7. Click **OK**. A message confirms the configuration of the network interface.
8. Select the **Network parameters** section from the left-hand side menu.
9. Enter or modify the External DNS name, Internal DNS name, Hostname, Domain name, DNS servers (IP addresses), Timezone, and NTP time servers.
10. Click **Save changes**.

When applying the new settings, the connection to the Web Console is lost. To reconnect to the Web Console, type in the configured External DNS name in your browser.

Appliance Proxy configuration

Software installation and updates, as well as the access to the Nexthink Library, use the HTTP and HTTPS protocols. If the Appliance needs to go through a proxy for HTTP/S connections, configure the Proxy settings:

1. Log in to the Web Console as admin.
2. Click the **Appliance** tab at the top of the window.
3. Select the **Cloud services** section from the left-hand side menu.
4. Under **Internet proxy**, tick the box **Enable internet proxy**. Fill out the configuration settings for the proxy that appear below.
    - **HTTP proxy**: Name or IP address and port number of the proxy server.
    - **Authentication method**: Choose None, Basic, NTLM, or Digest. Note that software updates only support the Basic method (or None, if authentication is not required). If an authentication method is chosen:
    - Type in the credentials **Username** and **Password**.
5. Optional: Click the **TEST** button to check that the Proxy chain is working and that the supplied parameters are correct. The Appliance tries to connect to the Nexthink Updates.
6. Click **Save changes** to store the Proxy data permanently.

If you are required to manually configure the proxy to access the host where the Nexthink Updates reside, the URLs accessed by the Engine are:

- updates-v6.nexthink.com on TCP port 80 (HTTP) or 443 (HTTPS).
- updates?centos?v6.nexthink.com on TCP port 80 (HTTP) or 443 (HTTPS).

If you are required to manually configure the proxy to access the host where the Application Library resides, the URLs accessed by the Engine, the Portal, and the Web Console are:

- application-library-v5.nexthink.com on TCP port 443 (HTTPS).
- application-library-v6.nexthink.com on TCP port 443 (HTTPS).

To test the connectivity of the Portal appliance to the Application Library:

1. Log in to the CLI of the appliance that hosts the Portal.
2. Test the proxy configuration running the following script:
   /var/nexthink/portal/rsquery/proxyConfig.py --test
   - If you get an error beginning with **com.nexthink.common.rest.RestException**, detect if it was a certificate problem with the following command (which ignores certificates):
     /var/nexthink/portal/rsquery/proxyConfig.py --test --ignore-ssl-problems true
   - If you get an answer finished by the word **Success!**, the test was successful
3. Optional: To see other options of the script, type in:
   /var/nexthink/portal/rsquery/proxyConfig.py --help

See the connectivity requirements for a complete list of the ports and domains that need to be available for communication in the different Nexthink components.

Nexthink Engine & Portal installation

Installation can be done either by using our online repository or, if the Appliance is not connected to the Internet, by uploading an offline installation package.

1. Log in to the Web Console as admin.
2. Click the **Appliance**
3. Select the **Installation** section from the left-hand side menu. This item is only available if you have installed neither the Engine nor the Portal in the

Appliance yet.
4. Select if you want to use the Nexthink online repository or a file.
5. In **Installation protocol**, tick the box **Enable https** to connect to the repository using a secure connection. In case of connection problems, try unticking the box. Click **Save**.
6. In **Repository selection**, choose the type of installation:
    - Select **Use Nexthink online repository** to download the software from the web.
        - Optional: Click **TEST WEB ACCESS** to verify the access to the online repository.
    - Select **Upload Nexthink repository from file** to install the software from an offline installation file previously downloaded from Product Downloads. In the desired release page, look for the download of the Nexthink installation package (tgz file).
        - Click **CHOOSE FILE** to browse your local file system and select the installation file.
7. In **Software selection**, choose whether to install the Engine, the Portal or both. Note that it is not recommended to install the Engine and the Portal on the same Appliance, except for small setups.
8. Accept the license agreement.
9. Click **Install**.

Please note that the installation process can be started only once. This page is no longer available after the first installation. Go to the **Update** page for upgrading your Appliance instead.

Related references

- Connectivity requirements

Installing the Appliance on Azure

To install the Appliance on Microsoft Azure:

1. Check the hardware requirements of the virtual machines to request in Azure according to the size of your setup.
2. Select the latest Nexthink release in Product Downloads:
    1. Download the VHD image for Azure of the Nexthink Appliance.
3. Follow the installation instructions.

The operations described in this article should only be performed by a Nexthink Engineer or a Nexthink Certified Partner.

If you need help or assistance, please contact your Nexthink Certified Partner.
Related references

- Azure hardware requirements

Related tasks

- Azure installation instructions

Installing the Appliance on AWS

To install the Appliance on Amazon Web Services (AWS):

1. Check the hardware requirements of the virtual machines to request in AWS according to the size of your setup.
2. Select the latest Nexthink release in Product Downloads:
    1. Download the image for AWS of the Nexthink Appliance.
3. Follow the installation instructions.

The operations described in this article should only be performed by a Nexthink Engineer or a Nexthink Certified Partner.

If you need help or assistance, please contact your Nexthink Certified Partner.
Related references

- AWS hardware requirements

Related tasks

- AWS installation instructions

Installing the Appliance on OTC

To install the Appliance on Open Telekom Cloud (OTC):

1. Check the hardware requirements of the virtual machines to request in OTC according to the size of your setup.
2. Select the latest Nexthink release in Product Downloads:
    1. Download the image for OTC of the Nexthink Appliance.

3. Attach and format a data disk to your VM before installing Nexthink.
4. Follow the installation instructions.

The operations described in this article should only be performed by a Nexthink Engineer or a Nexthink Certified Partner.

If you need help or assistance, please contact your Nexthink Certified Partner.
Related references

- OTC hardware requirements

Related tasks

- Attaching a data disk to your VM in OTC
- OTC installation instructions

Managing Appliance accounts

Overview

There are three accounts that let you manage your Appliance:

- **Nexthink Console account**: To log in to the Web Console of the Appliance. The Web Console lets you install the Nexthink software and configure most of the available settings.
- **SSH Nexthink account**: Support account used for logging in to the command line interface of the Appliance. Needed for advanced operations not available through the Web Console and for federating slave Appliances.
- **Portal remote management account**: Used by Portal administrators to centrally perform simple management operations on the Appliances connected to the Portal.

Changing the password of the management accounts

Upon first use, the Web Console requires you to change the password of the admin account for the Web Console itself, as well as the password of the SSH support account *nexthink* for the command line interface.

To change any of the passwords of the management accounts subsequently:

1. Log in to the Web Console as admin.
2. Click the **Appliance** tab at the top of the window.
3. Select the section **Accounts** from the left-hand side menu.
4. Choose the management account:
    - Under **Nexthink Console account**, change the admin password of the Web Console:
        1. Type in the old password (default **admin**).
        2. Type in the new password twice.
        3. Click **SAVE CHANGES**.
    - Under **SSH Nexthink account**, change the password of the CLI user *nexthink*:
        1. Tick the box to **Enable SSH Nexthink account** for the Appliance to support CLI access.
        2. Type in the old password (default **123456**).
        3. Type in the new password twice.
        4. Click **SAVE CHANGES**.
    - Under **Portal remote management account**, set the password to allow the centralized management of the Appliance from the Portal:
        1. Tick the box to **Enable Portal remote management account** for the Appliance to support management from the Portal.
        2. Type in the new password twice.
        3. Click **SAVE CHANGES**.

The **Notifications** setting at the bottom of the **Accounts** section is not really an account for managing the Appliance. Instead, it holds a list of email accounts for receiving notifications from the Appliance.

The operations described in this article should only be performed by a Nexthink Engineer or a Nexthink Certified Partner.

If you need help or assistance, please contact your Nexthink Certified Partner.
Related tasks

Sending email notifications from the Appliance

Setting the names of the Portal

Overview

The Appliance that hosts the Portal (the master) is identified by two lists of names. The names can be one or more DNS names, one or more IP addresses, or a mix of the two.

External IP/DNS name
: The list of names and IP addresses of the Portal as seen both by the Finder and by the web browsers that connect to the front-end of the Portal. If you replace the default certificates, these names must be included as SANs in the server certificate of the Portal.
Warning: When providing multiple names, keep in mind that rule-based Collector assignment addresses the Portal by the first specified external DNS name only.

Internal IP/DNS name
: The list of names or IP addresses of the Portal as seen by the slave Appliances. This field must be correctly set before federation.

The two names can in fact be the same if the Portal offers the same interface both internally and externally.

Note that the name of the Portal used in email digests is configured elsewhere.

Setting the DNS names of the Portal

To specify the fully qualified domain names (or IP addresses) of a Portal:

1. Log in to the Web Console of an Appliance hosting the Portal as admin.
2. Click the **Appliance** tab at the top of the Web Console.
3. Select **Network parameters** from the left-hand side menu.
4. Under the **DNS** section:
    1. Type in the external name of the Portal in **External IP/DNS name** (e.g. *myportal.example.com*).
        - ◊ To specify more than one name, separate each name or IP from another with a space character (e.g. *myportal.example.com 192.0.2.10*). Remember that only the first external name is used by rule-based Collector assignment to communicate with the Portal.
    2. For the **Internal IP/DNS name**, either:
        - ◊ Tick the option **same as external DNS name** to use the same name set as the external interface.
        - ◊ Type in an internal name different from the external name.
    3. Type in the hostname of the Portal in **Hostname** (e.g. *myportal*).
    4. Type in the domain part of the Portal name in **Domain** (e.g. *example.com*)

5. Optional: Type in the addresses of up to four name servers in **DNS servers**.

5. Click **Save changes** to store your changes.

Related tasks

- Federating your Appliances
- Importing and replacing certificates
- Sending email notifications from the Appliance

Setting the names of the Engines

Overview

The Appliance that hosts an Engine (a slave) is identified by two lists of names. The names can be one or more DNS names, one or more IP addresses, or a mix of the two.

External IP/DNS name
: The list of names (FQDN) and IP addresses of the Engine as seen by the Collectors, by the Finder, and by clients of the Web API. If you replace the default certificates, these names must be included as SANs in the server certificate of the Portal.
Warning: When providing multiple names, keep in mind that rule-based Collector assignment addresses each Engine by the first specified external DNS name only.

Internal DNS name
: The list of names (FQDN) and IP addresses of the Engine as seen by the master Appliance. This field must be correctly set before federating the Engine.

The two names can in fact be the same if the Engine offers the same interface both internally and externally.

Setting the names of an Engine

To specify the fully qualified domain names (or IP addresses) of an Engine:

1. Log in to the Web Console of an Appliance hosting the Engine as admin.
2. Click the **Appliance** tab at the top of the Web Console.
3. Select **Network parameters** from the left-hand side menu.

4. Under the **DNS** section:
    1. Type in the external name of the Engine in **External IP/DNS name** (e.g. *myengine.example.com*).
        ◊ To specify more than one name, separate each name or IP from another with a space character (e.g. *myengine.example.com 192.0.2.11*). Remember that only the first external name is used by rule-based Collector assignment to communicate with each Engine.
    2. For the **Internal IP/DNS name**, either:
        ◊ Tick the option **same as external DNS name** to use the same name set as the external interface.
        ◊ Type in an internal name different from the external name.
    3. Type in the hostname of the Engine in **Hostname** (e.g. *myengine*).
    4. Type in the domain part of the Engine name in **Domain** (e.g. *example.com*)
    5. Optional: Type in the addresses of up to four name servers in **DNS servers**.
5. Click **Save changes** to store your changes.

Related tasks

- Federating your Appliances
- Importing and replacing certificates
- Assigning Collectors to Engines

Specifying your internal networks and domains

To help the Engine make the difference between network traffic inside your organization and network traffic destined to external entities, specify your internal networks and domains from the Web Console.

This configuration is specific to each Engine. If you have several Engines installed, set the internal networks and domains for each one of them.

Specifying the internal networks

To specify the subnetworks that the Engine must recognize as belonging to your organization:

1. Log in to the Web Console hosting the Engine as admin.

2. Click the **Engine** tab at the top right corner of the page and select **Internal networks & domains** from the left-hand side menu.
3. At the bottom of the table entitled **Internal network configuration**, click **ADD INTERNAL NETWORK** to add a new internal network to the table.
    1. In the dialog that shows up for the new internal IP network, specify:
        - ◊ The subnetwork base address in the field **Network**.
        - ◊ The subnetwork mask in the field **Mask**.
    2. Click **OK**. The Engine restarts immediately and the button shows a spinning wheel until the new network is effectively added.
4. Repeat the operation for as many internal networks as you need to specify.
5. Optional: Click the link **Edit** at the right of the network entry in the table to edit its contents. A change in an existing network triggers an Engine restart.
6. Optional: Click the link **Delete** at the right of the network entry in the table to remove the entry. Deleting an existing network triggers an Engine restart.

Specifying the internal domains

Specifying the internal domains is only useful if you have purchased the Web and Cloud product. You need to write down only those domains that are hosted in servers outside your internal networks, so they are still considered *internal* web traffic even though they can be managed by an external organization. Domains served from your internal network are naturally considered internal.

The Engine never compacts domains identified as internal and it never sends these domains to the Application Library for detecting threats, since they are trusted.

To specify your internal domains:

1. Log in to the Web Console as admin.
2. Click the **Engine** tab at the top right corner and select **Internal networks & domains** from the left-hand side menu.
3. Write down the list of domains inside the text box under the title **Engine internal domains** at the bottom of the page. Use the wildcards **?** and ***** to replace one or several characters of the domain name and separate each domain in the list by a space. For instance:
    ***.example.com *.nexthink.com *.nexthink.ch**
4. Click **Save changes** to make your changes permanent and restart the Engine.

Related tasks

- Reporting the URL of HTTP web requests

Federating your Appliances

Overview

Starting from Nexthink V6.6, Appliances are organized around a new master / slave architecture called a *federation*. The Appliance hosting the Portal functions as master, while the Appliances hosting the Engines work as slaves.

When installing a new Appliance or when updating an Appliance from V6.5 (or previous) to V6.6 (or higher), the Appliance enters what is called compatibility mode. In compatibility mode, Appliances do not profit from the benefits of federation. The main advantages of federating your Appliances include:

- Centralized configuration.
- Centralized and automatic updates of Appliances and Collectors.
- Readiness for upcoming features.

Starting from V6.17, because of the security constraints introduced by automatic Appliance hardening, federating your Appliances is *mandatory* to enable the real-time communication between the Portal and the Engines.

In the case of small setups which include both the Portal and the Engine in the same Appliance, federation is automatic. In any other case, to take advantage of centralized configuration and updates and get ready for future improvements, federate your Appliances.

Federating an Appliance

Before federating a slave with a master Appliance, they must satisfy the following pre-requisites:

- The Appliance to be federated is indeed a slave Appliance (Engine only).
- The slave Appliance is not federated yet.
- The master and the slave Appliances share the same version.
- A bi-directional communication channel exists between the master and the slave Appliance.

- The master Appliance is able to reach the slave Appliance using the internal DNS name specified for the slave.
- The slave Appliance is able to reach the master Appliance using the internal DNS name specified for the master.

To federate an Appliance:

1. Log in to the Web Console of the master Appliance (the Portal) as admin.
2. Click the **Appliance** tab at the top of the Web Console.
3. Select **Federated appliances** from the left-hand side menu. This option is only available if there is no Engine installed in the master Appliance.
4. Click the button **ADD APPLIANCE** at the bottom of the page. A dialog to add the Appliance shows up.
    1. Type in the DNS name or IP address of the slave Appliance in **Internal DNS name**. This name must match the internal name that you specified for the slave Appliance.
    2. Type in the password of the SSH Nexthink account in the slave Appliance as **Password**.
    3. Specify the settings of the slave that you want to control from the master in **Settings to Centralize**. Tick zero or more of:
        ◊ **Cloud Services**
        ◊ **Mail server**
        ◊ **Privacy**
        ◊ **External backup**
    4. Click **OK** to federate the slave Appliance. The Engine in the slave Appliance is automatically restarted to apply the new configuration.

Federation process

During federation, the master and slave Appliances exchange their SSH public keys to enable secure bi-directional communication. In addition, the federation creates a public key infrastructure (PKI) to make the TCP connection between the Collectors and the slave Appliances trustworthy through TLS:

1. The master Appliance generates a Root Certificate, its associated private key (not shown in the figure), and a Customer Key (an *ad hoc* cryptographic key for the slave Appliances to authenticate Collectors) during its installation.
2. When you federate a slave Appliance, the Customer Key of the master is mirrored at the slave.
3. Additionally, the federation process issues a Server Certificate for the slave Appliance based on its External DNS name and signed with the private key of the Root Certificate in the master.

4. When generating the Collector installer or the MST, download both the Root Certificate and the Customer Key from the master Appliance and provide them as parameters to the installation, as explained in the instructions to install the Windows Collector.

After federation, the Collector authenticates a slave Appliance by using the Root Certificate to validate the Server Certificate presented by the Appliance as part of the TLS handshake. In its turn, the slave Appliance accepts the connection from the Collector only if the Collector has the same Customer Key as the Appliance itself. Therefore, you must always supply the correct Customer Key to the Collector during its installation:

If you replace the generated Server Certificates in the slave Appliances by your own certificates, do not provide the generated Root Certificate when installing the Collector. By not supplying the Root Certificate, the Collector falls back to the Windows Trusted Root Certificates Store for validating your certificates instead.

Note that the communications protected by the PKI are not related to device information, but to the centralized update and other upcoming features. Collectors use a different mechanism to secure the communication of device info to the Engines via UDP.

As for the centralized settings, the configuration files of the master Appliance are

mirrored at the slave. Thus, in the slave Appliance, it is no longer possible to change the centralized settings, which are displayed as read-only in the Web Console.

Appliance management and connection to Engines from the Portal

Two management tasks in the Portal overlap to some extent with the features of federating your Appliances:

- Connection to the Engines
- Appliance management

Regarding the connection to the Engines, you still need to connect your Portal to the Engines, even after federation, for the Portal to be able to collect data.

As for Appliance management, it is still available in the Portal, but two of its features have been disabled because they are overridden by federation:

- SMTP configuration (overridden by the **Mail server** centralized settings).
- Central update (overridden by the centralized update of federated Appliances).

Related tasks

- Setting the names of the Portal
- Setting the names of the Engines
- Managing Appliance accounts
- Sending email notifications from the Appliance
- Connecting the Portal to the Engines
- Importing and replacing Certificates
- Installing the Collector on Windows

Related references

- Compatibility mode
- Appliance Hardening

Connecting the Portal to the Engines

Overview

For the Portal to compute and display data in its widgets, connect it to the Engines that receive and organize the end-user data coming from the Collectors.

Starting from V6.17, federate your Appliances before connecting the Portal to the Engines and distributing the number of licensed devices among Engines. Otherwise the Portal will be unable to fully communicate with the Engines because of Appliance hardening.

Adding a new Engine

To connect the Portal to an Engine:

1. Log in to the Portal as central administrator.
2. In the top menu **ADMINISTRATION**, select the **Engines** dashboard under the section **SYSTEM CONFIGURATION**.
3. Click the plus sign that is located in the top right corner of the widget **Engines Management**. The dialog to add a new Engine shows up.
4. Type in the IP address or DNS name of the appliance that hosts the Engine in the **Address (IP or hostname)** field.
5. In the **Port** field, type in the port number that the Engine uses to communicate with the Finder and the Portal.
6. Optional: In the field **Description**, write down a brief sentence to help you distinguish the new Engine that you want to connect to the Portal from other Engines.
7. Click **Ok**.

After completing the procedure, the **Engines** dashboard displays the new Engine as a row in the table of connected Engines. The row displays the name, address, description, version and timezone of the Engine. However, since the connection is not yet established, a red dot appears in the first column of the row and the actual name, version and timezone of the Engine are not available yet.

Establishing the connection

To establish the connection and get information from the Engine:

1. In the table of Engines find the chain and pencil icons that are placed to the right of the **Name** column.
2. Click on the chain icon to establish the connection with the Engine. The red dot turns to yellow and then to green, to indicate that the connection is

now established. The widget fills in the name, version and timezone of the Engine with the information that the Engine itself sends.

Repeat the operation described above for any other Engine that you want to connect to the Portal.

Once the connection is established, the Portal collects information from the Engine in a regular basis. While the Engine connection is working, you cannot edit the parameters of the Engine and so the pencil icon in the row that holds the information about the Engine is disabled. Otherwise, if the Portal cannot establish a connection to the Engine, the dot in the beginning of the row stays red, which means that you probably did not set the parameters of the Engine correctly. In this case, click the pencil icon, edit the connection parameters of the Engine as explained above and try to establish the connection again by clicking the chain icon.

Similarly, if the appliance of an Engine changes its configuration and the modifications make the connection with the Portal fail, the dot will turn to red as well. To recover the connection:

1. Click the chain icon (displayed as a broken chain now) to unlink the Engine from the Portal and be able to edit the modified parameters.
2. Edit the parameters of the Engine as we just explained above.
3. Click again the chain icon and wait for the red dot to turn to green.

Related tasks

- Federating your Appliances

Related references

- Appliance hardening

Configuring session performance storage

Overview

The session performance data is only stored if you enable it using the Portal. By default, it is disabled. When activated, the necessary settings will be automatically configured on all connected Engines. It is not possible to restrict the change to a single Engine or a subset of them. Changing the parameters will

not trigger a restart of the Engines.

Session performance storage management

To configure the storage policy, connect to the Portal as a central administrator. In the main navigation bar on the left, click on the administration icon and select **Engines** under the **System configuration** section.

Check the box next to **remote sessions** if you want to store additional informations related to the remote sessions connecting to a server.

Check the box next to **local sessions** if you want to store additional informations related to the remote sessions using the console session.

Impact on data retention

Activating the option to store that supplementary data on the Engine will have an impact on the amount of events stored on it, and thus the overall data retention. You should expect a typical impact of less than 5%. So for an Engine configured to store 200 million events, when full, up to 10 million of them could be linked to

the session performance monitoring.

Related references

- Session performance
- Data retention in the Engine

Setting up a software license

Once you have installed all the appliances (the Portal and one or more Engines), request a software license to make the whole system work properly. Before requesting a license, read carefully the licensing terms and make sure that you have the following information readily available:

- Total number of devices to monitor (Windows and Mac OS).
- Total number of Mobile devices to survey.
- Number of servers.
- Validity period of the license:
    - Start date.
    - Expiration date.
- Desired optional modules, along with their own validity period (start and end dates):
    - Nexthink Act.
    - Nexthink Engage.
    - Nexthink Integrate.
    - Nexthink Enhance.
    - Nexthink Web and Cloud (now integrated into the basic offer).
- Type of license required:
    - Online.
    - Offline.

The validity period of the optional modules cannot exceed the validity period of the license.

To request more information about the licensing process or for any question related to the license of your specific setup, please send an email to:

.

Determining the activation mode of your license

The activation of the license depends on the connectivity of your appliances to the Internet.

- Request an **online** license if your appliances connect to the Internet. Online licenses are easier to set up and more flexible for updating than offline licenses. Nexthink recommends to use an online license whenever possible.
- Request an **offline** license only when your appliances cannot connect to the Internet.

Ordering and activating a new license

Once you issue a Purchase Order, you will receive a new license activation key from the Sales operations department of Nexthink by email.

To activate your new license:

1. Open the Portal and log in as central administrator.
2. In the **ADMINISTRATION** menu, select **Licenses** under the **SYSTEM CONFIGURATION** section.
3. Enter the activation key.
4. Select your mode of activation for the license:
    - Online activation.
    - Offline activation.
5. Allocate the number of licensed end-user devices (Windows and Mac), servers and Mobile devices among your Engines.
6. Finish the activation of the license:
    - If you requested an online activation of the license:
        1. Click **Apply** and you are done.
    - If you requested an offline activation of the license:
        1. Click **next** to go on with the activation.
        2. Click **Download license file** to get an encrypted file holding your license information.
        3. Go to https://sign-license.nexthink.com/ to get your signed license file
        4. Upload the signed license file in Portal

Updating or modifying an existing license

At some point in time, you may want to modify an existing license for different

reasons: change the number of devices, extend the expiry date, etc. Both our Licensing and Support teams handle license update requests; however, each team manages different aspects of the process. Use the table below to appropriately direct your request:

Case	Licensing	Support
Modify the number of licensed endpoints, servers, or mobile devices.		
Add a new module.		
Refresh the license after adding a new module or modifying the number of licensed devices.		
Transfer licensed devices to another license.		
Reallocate licensed devices among different Engines within the same license.		
Change activation mode of the license: online or offline.		
Reactivate non-expired license in *out-of-grace* state (usually because of inactivity).		
Renew an expired module or license.		
Disable Support Telemetry.		
Inability of Portal to communicate license info (connectivity issue between Nexthink components).		
Inability to log in to the Finder with error message: **User authentication failed (no license available)**.		

Stakeholders receive a notification after a license is modified in the Central License Manager of Nexthink. Find more information in the description of the email notification after the modification of a license.

Concurrent management of the license

Beware that more than one central administrator may access the license management dashboard in the Portal at the same time. In this case, the Portal displays appropriate warning messages if the concurrent modification of the license might lead to inconsistencies.

Related references

- Licensing terms
- Software components

Sending email notifications from the Appliance

Mail server settings

For the Engine and the Portal to send alert notifications and dashboard digests via email, configure the mail server (SMTP) settings of the Appliance through the Web Console. Configure first the mail server settings of the master Appliance:

1. Log in to the Web Console of the Appliance that hosts the Portal (the master) as admin from a web browser:
   https://<Name_or_IP_address_of_Appliance>:99
2. Click the **APPLIANCE** tab at the top of the window.
3. Select the **Mail server** section from the left-hand side menu.
4. Tick the option **Enable mail server** and fill out the form:
   - **SMTP server**: The name or IP address of the mail server, followed by the port number (usually, 25).
   - **Sender email address**: The email account to use for sending the notifications on behalf of the Engine or the Portal.
   - **Username** and **Password**: The user credentials to provide if the mail server requires authentication.
   - Tick the box **Enable TLS** if your mail server requires encrypted communication. The Appliance only supports STARTTLS as the mechanism to establish an encrypted mail channel.
   1. Optional: Verify your mail server settings in **Send test email**. Click the button **SEND** to post a test message to the recipients listed in the **Accounts** section of the left-hand side menu, under **Notifications** (see below).
5. Click **Save changes** to make your changes permanent.

In a master / slave setup, the Engine resides in the same Appliance as the Portal and shares the same mail server settings. In usual setups with one or several Engines hosted in separate Appliances, you have two options for configuring the mail server of the slave Appliances:

- Centralize the mail server settings of the slave Appliance during federation. In this way, the slave Appliance (the Engine) takes the mail server configuration of the master Appliance (the Portal).
- Log in to the Web Console of the slave Appliance and configure its mail server settings as previously shown for the master Appliance. This is only possible if you have not centralized the mail server settings of the slave Appliance yet; in which case, the **Mail server** section becomes read-only in the Web Console of the slave Appliance.

Appliance notifications

The Appliance sends notifications via email to a list of selected recipients with information on the status of updates and backups:

- Appliance update
  - Update available
  - Update completion
  - Update error
- Backup
  - External backup
  - Backup error

To set the list of recipients for the Appliance notifications:

1. Log in to the Web Console of the Appliance that hosts the Portal (the master) as admin from a web browser:
   https://<IP_address_of_Appliance>:99
2. Click the **APPLIANCE** tab at the top of the window.
3. Select the **Accounts** section from the left-hand side menu.
4. Under **Notifications**, type in the list of **Email addresses** that will receive the notifications. Separate the email addresses from each other by a comma.
5. Click **Save changes** to make your changes permanent.

Set the address of the Portal for the links in email digests

Users of email digests can click on some parts of the digest to open the Portal and display the appropriate dashboard; that is, the dashboard that contains detailed info about the clicked part (metric or service).

For the links in the digest to correctly point to the Portal, set the base address of the Portal (DNS or IP) in the Web Console:

To set the base address of the Portal for the links in the email digests:

1. Log in to the Web Console of the Appliance that hosts the Portal from a web browser as admin:
   https://<Name_or_IP_address_of_Appliance>:99
2. Click the **Portal** tab at the top of the window.
3. Select the **General** section fromt the left-hand side menu.
4. Under **Parameters**, type in the name or IP address of the Portal in **Portal address**.

5. Click **SAVE CHANGES** and wait for the Portal to restart.

Note that the Finder also uses this address for detecting and installing updates and that it is required for drilling-down from the Portal to the Finder.

Related tasks

- Federating your Appliances
- Managing Appliance accounts
- Installing the Appliance
- Receiving email digests
- Receiving alerts
- Updating the Finder
- Drilling-down to the Finder

Related references

- System alerts

Allocating resources for the Portal

Adapt the configuration of the Portal to your available hardware resources in order to maximize their utilization and optimize performance. To that end, edit the configuration file `startup.properties` in the Portal and set the appropriate memory values depending on your hardware resources and size of your installation.

Starting from V6.24, Nexthink runs a single-JVM on all setups (previous **SMALL** mode, which is implicit now). Adapt the amount of reserved memory to the size of your setup, according to the table below. The **Total memory** size corresponds to the actual memory installed, according to the specified minimum hardware requirements for the Portal appliance. If you are using a single appliance for both the Engine and the Portal, divide the memory installed by two (hardware requirements for a single appliance) to get an estimation of the **Total memory** for the Portal:

Max devices	Total Memory	Configuration
150k	59 GB	MAX_MEMORY=50G
100k	41 GB	MAX_MEMORY=34G
50k	23 GB	MAX_MEMORY=16G

20k	17 GB	MAX_MEMORY=10G
10k	13 GB	MAX_MEMORY=6G
5k	12 GB	MAX_MEMORY=6G

- Ask Nexthink Support for setups with more than 150k devices.

To edit the configuration file of the Portal and set the values that fit your hardware:

1. Log in to the CLI of the Portal appliance.
2. Stop the Portal:
   ```
   sudo systemctl stop nxportal
   ```
3. Make a copy of the sample configuration file to use it as the current configuration file:
   ```
   sudo -u nxportal cp \
   /var/nexthink/portal/conf/startup.properties.sample \
   /var/nexthink/portal/conf/startup.properties
   ```
4. Edit the configuration file with the appropriate values from the table above:
   ```
   sudo vi /var/nexthink/portal/conf/startup.properties
   ```
5. Restart the Portal:
   ```
   sudo systemctl start nxportal
   ```

For example, in an installation with 45 000 devices, look up in the table above and find that, for a maximum of 50k devices, you must allocate 16 GB of memory.

In that case, change the values of those parameters in the `startup.properties` configuration file of the Portal. The file should look like this:

```
MAX_MEMORY=16G
```

For large installations, please contact Nexthink for instructions on how to properly allocate resources for the Portal. You may also need to increase the number of connections to the Portal database.

Related references

- Hardware requirements
- Support FAQ: Maximum number of connections for PostgreSQL

Installing the Collector

Installing the Collector on Windows

Overview

You must install the Collector in all the Windows devices from which you want to get end-user analytics. Depending on the number of devices and their geographical location in your corporate network, the installation of the Collector may be a technically challenging task.

For small setups or particular cases, you may opt for installing the Collector individually on each device. For medium to large setups, the installation of the Collector usually requires the use of automated deployment tools in practice.

To keep your Collectors up-to-date, either rely on the same deployment tools that you used for their installation or on the automated update proposed by Nexthink.

Applies to platforms:

Prerequisites

You need:

- One or more Windows devices on which to install the Collector.
- The Nexthink Collector Installer (recommended) or the Collector MSI packages.
- The Customer Key and Root Certificate of the master Appliance. These are essential to enable the complementary TCP connection of the Collector with the Engine. Read this article if you need to install the Collector as part of a POC, before having installed the definitive master Appliance.
- Optional: A third-party automated deployment tool. Instructions on how to install the Collector via SCCM 2007, SCCM 2012, and GPO (Active Directory) are provided.

Find the Nexthink Collector Installer and the Collector MSI packages inside the Collector zip file available in the Product Downloads page of Nexthink:

1. Open your favorite web browser.
2. Navigate to the official Nexthink documentation web site:

doc.nexthink.com.

3. Click **Product download** in the top right corner of the main documentation page, above the search tool.
4. In the **Nexthink Help Center**, click the **Product Downloads** section.
5. Sign in with your customer credentials.
6. Click the first entry of the **Latest V6 releases** list.
7. Optional: Click the link to the release notes of the Collector to learn about the new features and bug fixes.
8. Under **Download links**, find the **Collector** section.
9. Click to download the **Collector package for Windows**.
10. Optional: Verify your download with the provided SHA-256 hash.
11. Extract the contents of the downloaded **collector-6.x.x.x.zip** file.
12. Find the Nexthink Collector Installer in the **Installer\Collector** folder. This is the recommended tool for generating an executable that embeds the Collector MSIs and the custom configuration options in a single bundle to easily deploy the Collector. Two versions are provided:
    - **NEXThink_Collector_Installer_Silent.exe** (recommended), to generate silent Collector installers. When deploying the Collector on a device, a silent installer renders the installation procedure unnoticeable to the end user of the device.
    - **NEXThink_Collector_Installer.exe** (for debugging), to generate installers that open a command-line window.
13. Find the Collector MSI file, **NEXThink_Collector.msi**, in the folders:
    - **x64\signed** (64-bit version)
    - **x86\signed** (32-bit version).

Download the Customer Key and default Root Certificate from the master Appliance:

1. Log in to the Web Console of the master Appliance as admin.
2. Select the **Appliance** tab at the top of the Web Console.
3. Click **Collector management** in the left-hand side menu.
4. Under **Collector default certificates** at the bottom of the page, click the buttons **DOWNLOAD CUSTOMER KEY** and **DOWNLOAD DEFAULT ROOT CERTIFICATE** to download, respectively, the text files holding the Customer Key and the default Root Certificate of the master Appliance (the latter is required only if you did not replace the certificate for the TCP communication channel of the Appliances with the Collector).

You need to know:

- The DNS name or IP address of the Appliance that hosts the Engine or, if rule-based collector assignment is turned on, of the Appliance that hosts

the Portal. The name or IP address must match the External DNS name of the Engine or the External DNS name of the Portal, respectively.
- TCP port number for the connection of the Collector with the Appliances (default 443).
- Optional: UDP port number where the Engine is listening for the Collector (default 999, but prefer TCP-only).

Deploying the Collector using the Nexthink Collector Installer

The Nexthink Collector Installer is a tool that helps you deploy the Collector by producing a standalone executable file that holds the MSI files of both the 32-bit and 64-bit versions of the Collector. Therefore, you can use the same executable to install the Collector on any device that runs a supported version of Windows.

To generate the executable, use the graphical interface of the Nexthink Collector Installer to set the installing options of the Collector:

1. Double-click the appropriate Nexthink Collector Installer executable file for generating either:
    - A silent installer: **Nexthink_Collector_Installer_Silent.exe** (recommended).
    - A installer that opens a command-line window: **Nexthink_Collector_Installer.exe**.
      In either case, the following dialog shows up:

2. Specify the configuration settings of the Appliance that will receive Collector information under **Nexthink Appliance settings**. If the rule-based collector assignment is turned on, use the settings of the Portal; otherwise, use the settings of the Engine:
    - **Address**: DNS name or IP address of the Appliance (it must match the External DNS name of the Portal or the Engine).
    - **Data over TCP**: Tick the option to convey end-user data through the TCP channel of the Collector (default), instead of the UDP channel.
    - **Ports (TCP)**: Port number on which either the Portal or the Engine listens to the TCP channel of the Collectors.
    - **Ports (UDP)**: Port number on which the Engine listens to the UDP channel of the Collectors. Only enabled if **Data over TCP** is not checked.
    - **DNS**: Tick the option **Prefer IPv6** if you want the Collector to favor the use of IPv6 over IPv4 to communicate with the Appliances when the name of the Appliances resolve to both IPv6 and IPv4 addresses.
3. Set the **General settings**. Optionally tick the box for any additional setting. In particular:
    1. Installation options which are not shown in the dialog take their

default value. Check **Install configuration tool** to modify them later with the Collector Configuration Tool. In the case of an update, the values of the non-visible settings are preserved from the previous installation.

2. Check the option **Web and Cloud Data** if you have purchased the Web and Cloud product. Furthermore, click the button **Settings** to the right of this option to open a dialog where you can list the domains for which you want to store the full URL paths of web requests. That is, for every web request that falls under one of the specified domains, the Collector reports the full URL path to the Engine and not just the domain.
3. Check the option **Report print jobs and printers** to enable print monitoring. Starting from V6.18, reporting print information is disabled by default.
4. Check the option **Use the assignment service** if you activated the rule-based assignment of Collectors.
5. Select the activation of the **Engage** features depending on the target operating system:
    - ◊ **Enable except on servers**: the Collector launches campaigns only on devices that do not run a server OS.
    - ◊ **Enable on all devices**: the Collector launches campaigns on devices with any type of operating system.
    - ◊ **Disable**: the Collector ignores campaigns.
6. Select the **Script execution policy** for remote actions that the Collector will run on the device:
    - ◊ **Signed by a trusted publisher or by Nexthink** (default): the Collector runs on the device only those remote actions with a PowerShell script that is signed either by Nexthink or by a company whose certificate is listed in the Trusted Publishers certificate store.
    - ◊ **Signed by a trusted publisher**: the Collector runs on the device only those remote actions with a PowerShell script that is signed by a company whose certificate is listed in the Trusted Publishers certificate store.
    - ◊ **Disabled**: the Collector runs no remote action on the device.
    - ◊ **Unrestricted**: the Collector runs any remote action on the device, regardless of the digital signature of its script.
7. Optional: Type in an integer number (0 to 2147483647) as **Collector tag** to identify the group of Collectors generated with the installer. The Collector tag is visible in the Finder and is useful for defining the entities to build up hierarchies.
8. Optional: For more flexibility in the identification of Collectors, type in a label (max 2048 characters) as **Collector string tag** to identify

the group of Collectors generated with the installer. The Collector string tag is visible in the Finder and is useful for defining the entities to build up hierarchies.

9. Choose the files that protect the non-traffic information (TCP connection) of the Collector to the Engine:
    - ◊ **Customer Key**: Click **Browse** to select the file that holds the Customer Key of the master Appliance.
    - ◊ **Root CA**: Click **Browse** to select the file that holds the default Root Certificate of the master Appliance. If you leave this field empty, the Collector assumes that you replaced the server certificates in the Engine and falls back to using the Windows Trusted Root Certification Authorities Store for verifying the certificates presented by the Engine (the slave) Appliance.
10. Finally, specify a couple of directories:
    - ◊ **Ouptut directory**: Click **Browse** to select the folder where to create the executable files for installing and uninstalling the Collector.
    - ◊ **(Optional) Logs directory**: Type in the network place where the Collectors deployed with this method will write their installation logs.

4. Optionally provide the custom proxy settings of the Collector under the **Proxy configuration** section:
    1. Tick **Automatic proxy** to provide a PAC file for the automatic configuration of the proxy settings.
        - ◊ **PAC address**: type in the URL of the PAC file.
    2. Tick **Manual proxy** to manually provide the proxy settings.
        - ◊ **Proxy address**: Type in the FQDN or IP address of the proxy.
        - ◊ **Proxy port**: Type in the port number where the proxy is listening.
5. Click **Create**, three files are generated:
    - ♦ **NEXThink_Collector[*engine_address*].exe**: Executable file to install the Collector.
    - ♦ **NEXThink_Collector_Uninstaller[*engine_address*].exe**: Executable file to uninstall the Collector.
    - ♦ **NEXThink_Collector[*engine_address*].exe.txt**: Text file with the list of the settings used to create the executable installer.
6. Click **OK** to close the dialog.

Nexthink Collector Installer error codes

The installer executable returns one of the following values:

- 0: Success; reboot is not required.
- -1: Success; reboot is required.
- 3: Failure; Collector installation has started but msiexec failed.
- other: Failure; the actual value corresponds to a Windows Internal error code.

The uninstaller executable returns one of the following values:

- 0: Success; reboot is not required.
- -1: Success; reboot is required.
- 1: Success; Collector was not found (nothing was uninstalled).
- 3: Failure; Collector uninstallation has started but msiexec failed.
- other: Failure; the actual value corresponds to a Windows Internal error code.

Remember that rebooting is usually not required when installing the V6 Collector. The installer requires you to reboot a device only if you are upgrading from the V5 Collector or if the Control Panel extension of the Collector was running during installation.

Deploying the Collector through SCCM

In this section, learn how to deploy the Collector over groups of end-user devices using Microsoft System Center Configuration Manager. The instructions assume that you are a systems administrator with a basic understanding of the Windows operating system and deploying enterprise software, and that you are familiar with SCCM. The present documentation covers the deployment of the Collector with SCCM 2012 and SCCM 2007. For other versions of SCCM, the procedure may be slightly different. Please refer to the section on deploying software packages in the user manual of your specific version to deploy the Collector.

The deployment of the Collector SCCM requires you to provide an executable file that is responsible for the actual installation of the Collector in your devices. To generate this executable, use the Nexthink Collector Installer described above.

Deploying the Collector through SCCM 2012

Create a collection of devices:

1. Click the Windows **Start** button, go to the **Microsoft System Center 2012** program group, and run the Configuration Manager console.
2. In the **Assets and Compliance** workspace on the left-hand side of the main panel, righ-click **Device Collections** and select **Create Device Collection**.
3. On the **General** page of the **Create Device Collection Wizard**, specify the following fields:
    - **Name**: Type in a unique name for the collection.
    - Optional **Comment**: Type in a meaningful comment (e.g. *Deployment for the Nexthink Collector*).
    - Optional **Limiting collection**: Click **Browse** to select a collection that puts a limit on the members of the current collection or select **All systems** in order not to limit the current collection.

Create a boundary and add it to a boundary group:

1. In the Configuration Manager console, go to the **Administration** workspace.
2. Right-click **Boundaries** and select **Create Boundary**.
3. Define the boundary by setting additional restrictions on the target devices in which to push the software installation (e.g. by IP address range).
4. Right-click **Boundary Groups** and select **Create Boundary Group**.
5. Type in a name for the group.
6. Add the previously created boundary to this group.
7. Optional: Verify that you added the correct number of devices to the group by looking at the value in the column **Member Count**.

Create the application to install:

1. In the Configuration Manager console, go to **Software Library** workspace.
2. Right-click **Applications** and select **Create Application**.
3. Choose the option "Manually specify the application information".
4. Specify the location and name of the application (in our case, the name should be **Nexthink_Collector_Installer_Silent.exe**). The new application is added to the list of available applications.

Now the new Application should appear in the list. When you click it, there is a Deployments tab at the bottom of the window. Later this tabs will show a list of deployments of this application to different device groups.

Distribute and deploy the Application:

1. In the list of **Applications**, right-click the previously created Collector application and select **Distribute Content**. The distribution wizard opens.
    1. Confirm the correct executable file of the installer (**Nexthink_Collector_Installer_Silent.exe**).
    2. As **Content Destination**, select **Distribution Point**.
    3. Specify the shared folder that holds the installer.
2. Optional: Survey the distribution process from the Configuration Manager console.
    1. In the main panel, navigate to **Monitoring** > **Distribution Status** > **Content Status**.
    2. Click the application that you have just distributed. If you see **Success** and a green colored graph below, you can now deploy the application.
3. Back in the **Software Library** workspace, navigate to **Applications**.
4. Right-click the Collector application and select **Deploy**.
    1. Select the collection of devices that you created earlier.
        ◊ If you cannot see your collection in the list, switch from "User Collections" to "Device Collections".
    2. Check that the distribution point is correct and click **Next**.
    3. Set **Action** to **Install** and **Purpose** to **Required** and click **Next**.
    4. Set the **Schedule** to an appropriate moment for starting the deployment (e.g. **As soon as possible**) and click **Next**.
    5. Tick **Software installation** and click **Next**.
    6. Accept the default options for the rest of the wizard.
5. Optional: Check the status of your deployment in the **Deployments** tab at the bottom of the window.

To verify the deployment on a client device:

1. Log in to the client device and wait for the pop-up notification about installation of new software.

To speed up this process a little bit, you can manually force the software deployment evaluation cycle in the SCCM client:

1. Open the **Control Panel**.
2. Navigate to **Configuration Manager** and click the **Actions** tab.
    1. Choose **Machine Policy Retrieval & Evaluation Cycle** and click **Run Now**.
    2. Choose **Application Deployment Evaluation Cycle** and click **Run Now**

To debug the deployment process and see its log files, check the following:

- On the server machine, open the SCCM utility to view log files:
  ```
  C:\Program Files\Microsoft Configuration
  Manager\tools\cmtrace.exe
  ```
- The server logs are stored in:
  ```
  C:\Program Files\Microsoft Configuration Manager\Logs\
  ```
- On the client machine, the logs are stored in one of these three paths:
  ```
  C:\Windows\CCM\
  C:\Windows\ccmsetup\
  C:\Windows\ccmcache\
  ```

If your deployment is not successful, check the following troubleshooting points:

- In the Configuration Manager console, navigate to **Administration** > **Site Configuration** > **Servers and Site System Roles** and choose your server. In the table below, right-click **Distribution point** and select **Properties**. In the **Boundary Groups** tab, verify that boundary group that you previously created is listed in the **Boundary Groups list**. If not, add it to the list.
- In the Configuration Manager console, navigate to **Software Library** > **Applications**. Right-click the Collector application and select **Properties**. Verify the following points:
  - In the **Distribution Settings** tab, make sure that the option **Distribute the content for this package to preferred distribution points** is ticked.
  - In the **Content Locations** tab, make sure that your distribution point (the path to your shared folder) is in the table. If not, add it, and click **Redistribute**.
- If the remote installation fails with error code 0x87d00324 (displayed in Software Center on the client machine), the installation itself was in fact successful and the Collector should be running. It is the mechanism for detecting the installation of the application which failed. In this case, check the detection criteria:
  1. In **Software Library**, right-click the deployed Collector application and select **Properties**.
  2. In the tab **Deployment Types**, double-click the installer script in the list **Detection Method**.
  3. Check if the detection method is configured correctly. Since you are using the Collector installer executable, the detection should be done by registry key.

Deploying the Collector through SCCM 2007

Create a collection of devices:

1. Click the Windows **Start** button, go to **All Programs**, select **Microsoft System Center**, select **Configuration Manager 2007** and then select **ConfigMgr Console**.
2. In the **Configuration Manager Console**, navigate to **System Center Configuration Manager** / **Site Database** / **Computer Management**.
3. Right-click **Collections**, and then click **New Collection**.
4. On the **General** dialog box of the **New Collection Wizard**, enter a name for the collection.
5. Click **Next** and click the computer icon, which opens the **Create Direct Membership Rule Wizard**. Click **Next**.
6. On the **Search for Resources** dialog box:
    1. Click the **Resource** class drop-down menu and select **System Resource**.
    2. Click the **Attribute name** drop-down menu and select **Name**.
    3. In the **Value** field enter **%**, and then click **Next**.
7. On the **Collection Limiting** dialog box, click the **Browse...** button, select **All Windows Workstation or Professional Systems**, and then click **Next**.
8. On the **Select Resources** dialog box, select the check box for each of the targeted computer resources.
9. Click **Next**, and then on the **Finished** dialog box, click **Finish**.
10. On the **Membership Rules** dialog box of the **New Collection Wizard**, click **Next**.
11. On the **Advertisements** dialog box, for now, do not assign an advertisement because it is not yet created.
12. Click **Next**. On the **Security dialog** box, accept the defaults, click **Next**, and then click **Close**.

Create a package source directory:

1. Click on the **Start** button, go to **All Programs**, select **Microsoft System Center**, select **Configuration Manager 2007** and then select **ConfigMgr Console**.
2. Navigate to **System Center Configuration Manager** / **Site Database** / **Computer Management** / **Software Distribution**.
3. Right-click **Packages**, point to **New**, and then click **Package**.
4. On the **General** dialog box of the **New Package Wizard**, enter the:
    - Name
    - Version
    - Manufacturer
    - Language
5. Click **Next** and do the the following:
    1. Select **This package contains source files**.

    2. Click the **Set** button and then enter the path for the location of the source files in the **Source directory** field.
6. Click **Next** and accept the default settings on all of the following dialog boxes: Data Access, Distribution Settings, Reporting, and Security.
7. On the **Wizard Completed** dialog box, click **Close**.

To use a server as a distribution point for providing packages to distribute packages to your client computers, first designate a site system as a distribution point. To select a distribution point for the newly created package:

1. Right-click **Distribution Points**, click **New Distribution Points**, click **Next**, and then click the check box for the distribution point.
2. Click **Next**. Upon completion of the **New Distribution Points Wizard**, click **Close**.

Create a program with setup and install parameters:

1. Right-click **Programs**, point to **New**, and then click **Program**.
2. On the **General** dialog box:
    1. Enter a name for the package in the **Name** field.
    2. In the **Command line** field, browse and select the executable file that you have previously generated with the Nexthink Collector Installer.
    3. In the **Run** field, click the drop-down menu and select **Hidden**.
    4. In the **After running** field, verify the default of **No action required** is selected.
3. Click **Next** and accept the defaults on the **Requirements** dialog box.
4. On the **Environment** dialog box:
    1. Click the **Program can run** drop-down box and select **Whether or not a user is logged on**. This will enable Run with administrative rights for the Run mode.
    2. Leave the default for **Drive mode** to **Runs with UNC name**.
5. Click **Next**.
6. On the **Advanced** dialog box, select the check box for **Suppress program notifications**, and then click **Next**.
7. To view the **Summary** dialog box, click **Next**.
8. To finish the process of creating the new program, click **Next**, which will then display the **Wizard Completed** dialog box.
9. To exit from the **New Program Wizard**, click **Close**.

Verify that the package was installed on the distribution point, by:

- Navigating to System Center Configuration Manager, Site Database, Computer Management, Software Distribution, Packages, Collector package name, Package Status, Package Status.
- Checking the status changing from Install Pending to Installed.

Create the advertisement:

1. Right-click **Advertisements**, point to **New**, and then click **Advertisement**.
2. On the **General** dialog box of the **New Advertisement Wizard**:
    1. Enter a name for the advertisement in **Name** field.
    2. Click the **Browse** button for the **Package** field, and click on the package you want to advertise and then click **OK**.
    3. Click the **Browse** button for the **Collection** field, click on the collection.
3. Click **OK**, and then click **Next**.
4. On the **Schedule** dialog box, enter the date and time in the Advertisement start time fields for when the advertisement will begin, and then click the asterisk button for **Mandatory Assignments**.
5. On the **Assignment Schedule** dialog box, click the **Schedule** button and enter the same date and time that you previously entered in the **Advertisement start time** fields on the **Schedule** dialog box.
6. To return to the **Schedule** dialog box, click **OK**.
7. Accept the default values on the **Distribution Points**, **Interaction**, **Security**, and **Summary** dialog boxes.
8. Upon successful completion of the **New Advertisement Wizard**, click **Close** on the **Wizard Completed** dialog box.

On the client, wait for the next Machine Policy Retrieval & Evaluation Cycle.

Test your results:

1. Go to a the target PC that is member of the Collections you have created to deploy.
2. Click **Start, Run** and type **eventvwr.msc** to show the event viewer message.
3. In the left pane of the **Event Viewer**, select **Application**, you will see some source events **MsiInstaller** logged as a Success Audit event.
4. If you get any error, see the error log generated in **C:\Windows\Temp\Msi.log**.

Optionally, remove the package:

1. Open the **Systems Management Server console** and expand the Package that contains the Collector.
2. Open the **Program Properties** dialog box, and then on the **General** tab:
    1. In the **Command line** field, browse and select the Collector file.

The package will be removed silently at the next Machine Policy Retrieval & Evaluation Cycle from the end-user device.

Deploying the Collector through GPO

In this section, learn how to deploy the Collector over large groups of end-user devices using a standard Windows technology such as Active Directory Group Policy. The instructions assume that you are a system administrator with a basic understanding of the Windows operating system and the deployment of enterprise software.

Create a distribution point:

1. Log in to the server as an Administrator user.
2. Create a shared network folder.
3. Set permissions on this folder in order to allow access to the distribution package.
4. Copy the MSI of the Collector to the shared folder.
5. Generate transform files (MST) for controlling the options passed to the MSI for installation. Use the Orca utility from Microsoft to generate the MST, for instance.
6. Copy the generated MST to the shared folder.

Create a Group Policy Object:

1. Click on the **Start** button, go to **All Programs**, select **Administrative Tools** and then select **Active Directory Users and Computers**.
2. Right-click your domain name in the console tree, select **New** and click **Organizational Unit**.
3. In the **New Object** dialog box, type a descriptive name for the new organizational unit, and then click **OK**.
4. In the right panel, select **Computers** and click on the computer that you want add to your Organizational Unit.
5. Drag and Drop these computers in the name of the Organizational Unit created. In the right panel, select **Nexthink_Collector_Deploy**, you will see all the computers tied to your Organizational Unit.
6. Click on the **Start** button, go to **All Programs**, select **Administrative Tools** and then select **Group Policy Management**.

7. Right-click your domain name in the console tree and select **Create a GPO in this domain**, and **Link it here...**.
8. In the **New GPO** dialog box, type a descriptive name for the new policy, and then click **OK**.

Assign an MSI package:

1. Click on the **Start** button, go to **All Programs**, select **Administrative Tools** and then select **Group Policy Management**.
2. Right-click your GPO name and select **Edit...**.
3. On this **Group Management Editor**, expand **Computer Policies, Software Settings and Software Installation**, select **New** and then click **Package...**.
4. In the **Open** dialog box, browse to the distribution point you created for the Nexthink Collector during the distribution point.
5. Select the MSI file containing the Collector installer you want to deploy, and then click **Open**.
6. In the **Deploy Software** dialog box, select **Advanced**, and then click **OK**.
7. In the **Properties** dialog box for the package you created.
    1. Click the **Deployment** tab, and then select **Uninstall** this application when it falls out of the scope of management.
    2. Click **Advanced** on the **Deployment** tab, choose **Ignore language when deploying this package**, uncheck the option **Make this 32-bit X86 application available to Win64 machines**, and then click **OK**.
    3. On the **Modifications** tab, specify any modification transforms you want to apply when the package is installed by clicking **Add** and then opening each transform from its network location.
    4. On the **Security** tab, verify the name(s) of any computer(s) to which you are assigning software.
    5. Click **OK** to close the Properties dialog box.
8. In the **Group Policy** dialog box, expand **Computer Configuration, Administrative Templates, and Windows Components**.
    1. In the **Windows Components** folder, select **Windows Installer**.
    2. Select **Always install with elevated privileges**.
        1. Select **Properties**.
        2. In the **Always install with elevated privileges Properties** dialog box, click the **Setting** tab, select **Enabled**, and then click **OK**.
9. In the **Windows Installer** panel of the **Group Policy** dialog box, right-click **Logging**, and then select **Properties**.
    1. In the **Logging Properties** dialog box, on the **Setting** tab, select **Enabled**.

2. Then, in the **Logging** text box, type **iweaprcv**.
3. Click **OK** to close the **Logging Properties** dialog box.

10. In the **Group Policy** dialog box, click **File**, and then click **Exit**.

Note: The GPO must be propagated to the Active Directory Global Catalog and then to the individual computers. For this reason, allow 5-10 minutes before restarting the computers to which you are assigning the Nexthink Collector, or plan to restart the client computers twice before the system policies are synchronized.

Test your results:

1. Go to a the target PC that is member of the OU you tied the policy to.
2. Click **Start, Run** and type **gpupdate /force**.
3. A logoff or a restart message will appear: type **Y** and Enter.
4. When you restart, you should see the message **Installing Nexthink Collector...** for about a minute depending on the speed of your network and pc.
5. Click **Start, Run** and type **eventvwr.msc** to show the event viewer message. In the left pane, select **Application**, you will see some source events **MsiInstaller** logged as a Success Audit event.
6. If you have some errors, go to *C:Windows/Temp/Msi.log* and see the error log generated.

Sometimes you may need to redeploy a package (for example when doing an upgrade). To redeploy a package:

1. Click the **Start** button, go to **Programs**, select **Administrative Tools** and then select **Group Policy Management**.
2. Right-click your domain name in the console tree and select the **Properties** context menu.
3. Go to the **Group Policy** tab, select the object you used to deploy the package and click **Edit**.
4. Expand the **Software Settings** element (per-user or per-machine) which contains the deployed package.
5. Expand the **Software Installation** element which contains the deployed package.
6. Right-click the package in the right pane of the **Group Policy** window.
7. Select the **All Tasks** menu and click **Redeploy application**.
8. Click the **Yes** button for reinstalling the application wherever it is installed
9. Close the **Group Policy Management Editor**, click **OK** and exit the Group Policy Management.

Note that the GPO must be propagated to the Active Directory Global Catalog and then to the individual computers. For this reason, allow 5-10 minutes before restarting the computers to which you are assigning the Collector, or plan to restart the client computers twice before the system policies are synchronized.

Optional, remove a package:

1. Click on the **Start** button, go to **Programs**, select **Administrative Tools** and then select **Group Policy Management**.
2. Right-click your domain name in the console tree and select the **Properties** context menu.
3. Go to the **Group Policy** tab, select the object you used to deploy the package and click **Edit**.
4. Expand the **Software Settings** element (per-user or per-machine) which contains the deployed package.
5. Expand the **Software Installation** element which contains the deployed package.
6. Right-click the package in the right pane of the **Group Policy** window.
7. Select the **All Tasks** menu and click **Remove**.
8. Select from the following options:
    - Immediately uninstall the software from users and computers.
    - Allow users to continue to use the software but prevent new installations.
9. Click the **OK** button to continue.
10. Close the **Group Policy Management Editor**, click **OK** and exit the Group Policy Management.

Note that the GPO must be propagated to the Active Directory Global Catalog and then to the individual computers. For this reason, allow 5-10 minutes before restarting the computers to which you are assigning the Collector, or plan to restart the client computers twice before the system policies are synchronized.

Installing the Collector on a single device

Use the Collector MSI package to install the Collector either in interactive mode or in silent (also known as *unattended*) mode. In the latter case, no user interaction is required once the installation process is started.

This method of installing the Collector individually in every device is very tedious for large setups. Therefore, we only recommend it for proofs of concept or testing purposes.

To install the Collector in interactive mode:

1. Double-click the Nexthink Collector MSI file (**NEXThink_Collector.msi**) to start the installation program.
2. After reading the welcome message, click **Next**.
3. Fill out the form of installation settings:
    - **Appliance Name or IP address**: Type in the DNS name or IP address of the Appliance (the Portal, if rule-based collector assignment is turned on; the Engine, otherwise). Respectively, this setting must match either the External DNS name of the Portal or the External DNS name of the Engine.
    - **Appliance TCP Port**: Type in the port number on which either the Portal or the Engine listens to the TCP channel of the Collector. Default is 443, but alternatively you can select a custom TCP port that does not require root permissions (that is, a port number above 1024). Starting from V6.24, the UDP channel is no longer proposed for this installation method.
    - **Customer Key**: Copy to this field the contents of the Customer Key file previously downloaded from the master Appliance. For instance:
        1. Open the Customer Key file in the Notepad.
        2. Press **Ctrl+A** to select all the text.
        3. Press **Ctrl+C** to copy the text.
        4. Back in the **Customer Key** field, press **Ctrl+V** to paste the copied key.
    - **Root CA**: Copy to this field the contents of the Root Certificate file previously downloaded from the master Appliance. Use the same method described for copying the Customer Key.
        - ◊ If this field is left empty, the Collector assumes that you replaced the server certificates in the Engine and uses the Windows Trusted Root Certificates Store to validate the TCP channel connection to the Appliances. Replacing certificates is required if communicating via the default TCP port 443.
    - Optional: Tick the box **Install Control Panel Extension**.
        - ◊ It is best practice not to install the Control Panel Extension when deploying the Collector on a production environment. The Collector Control Panel Extension is an optional tool for assistance during troubleshooting or testing phase. It is not required by the Collector to work properly.
4. Click **Next**.
5. The installer is now ready; click **Install** to begin the actual installation.
6. Click **Finish** to close the installation once it has completed.

Starting from V6, the Collector does not require rebooting the device after installation or upgrade. Once the MSI has successfully installed the Collector,

you are requested to reboot the device only if you are upgrading from V5 or if you were running a Collector component during installation (usually, the Control Panel extension).

Command-line silent installation

To install the Collector in silent mode: Use *msiexec.exe* on the command line to install the Collector in silent mode. The executable program **msiexec.exe** comes pre-installed with every Microsoft Windows operating system. Custom parameters are provided directly on the command-line and they are not saved from one installation to another. Therefore, this is not the recommended method to install the Collector, since commands are prone to typos. For a single installation, prefer the graphical installation method instead. For larger deployments with automated tools, we recommend to use MSI Transforms.

The mandatory parameters are:

- **DRV_IP**: the IP address or DNS name of the Appliance that hosts either the Engine or, if rule-based collector assignment is turned on, the Portal.
    - ♦ It must match the External DNS name of the Engine or the External DNS name of the Portal, respectively.
- **DRV_PORT**: port number on which the Engine listens to the UDP channel of the Collector.
- **CRD_PORT**: port number on which the Appliance (Portal or Engine) listens to the TCP channel of the Collector.
- **CRD_KEY**: the Customer Key downloaded from the master Appliance.
- **CRD_ROOT_CA**: the Root Certificate downloaded from the master Appliance. Leave empty if you replaced the certificates in the Engine and want the Collector to use the Windows Trusted Root Certificates Store for validating its TCP connection with the Engine (the slave) Appliance.

Here is an example of an unattended installation:

1. Type in the command line:
   msiexec.exe /qn /i Nexthink_Collector.msi DRV_IP=192.168.84 DRV_PORT=999 CRD_PORT=8443 CRD_KEY=<Your_Key_Here> REBOOT=ReallySuppress
2. Wait for the installation process to complete.

The MSI now installs by default on any kind of Windows device, be it a laptop, a desktop, or a server.

For a comprehensive list of available options for the Nexthink Collector, see the Collector MSI Parameters reference.

See also: Windows Installer (msiexec.exe) Command-Line options Reference.

Deploying the Collector within a Windows Master Image

When including the Collector in a Windows Master Image, remove the UID that the Collector may have generated to identify the device.

To ensure the removal of the device UID:

1. Log in to the Windows device with the Collector as a user with administrative rights.
2. Press the Windows button [ ] on your keyboard.
3. Type in **cmd** to make the Command Prompt application show up as a result of the search.
4. Right-click the Command Prompt icon to open a context menu.
    1. Select **Run as administrator** from the menu.
5. From the prompt, stop the Collector by typing in:
    ```
    nxtcfg.exe /stop
    ```
6. Remove the UID of the device:
    ```
    reg delete "HKLM\SYSTEM\CurrentControlSet\Services\Nexthink Coordinator\params" /v uid
    ```
    1. To the question **Delete the registry value uid (Yes/No)?**, answer by typing in **yes**.

The system will respond with either a confirmation that the operation was successful or with an error message if the Collector did not generate the UID yet.

Interaction of the Collector with other software

To get valuable information from a device, part of the Collector needs to run in privileged mode as a kernel driver. Contrary to user applications, the programs that run in privileged mode can access the memory and the hardware of the device directly. Typically, these are the programs that control the peripherals in your device; such as the mouse, the keyboard, the hard disks or the network card. Other special programs, like antivirus software, may also need to run in privileged mode, at least partially. Errors in programs that run in privileged mode are not protected by the process isolation provided by the operating system and, therefore, they may result in system failure. Moreover, since all of these programs share the same memory space, a misbehaviour of one of them can destabilize all the others.

The Collector has been carefully designed and thoroughly tested to avoid any kind of program errors. It has also been engineered following the best practices for the development of kernel drivers, behaving as a good citizen with respect to the other drivers that are loaded into the system. Still, in some very rare cases, an elusive programming error may defeat our rigorous testing process or a misbehaving third-party driver can trigger a fault in the Collector. In these unfortunate situations, the Collector may become unstable and possibly lead to a system failure in the device of the end-user.

CrashGuard Protection

To protect you against driver misbehavior, keep your Windows drivers up to date. Older versions of Windows drivers often contain more bugs that can lead to instabilities. If a third-party driver consistently destabilizes the Collector, the *CrashGuard* protection of the Collector helps you prevent your device from crashing again and again.

The CrashGuard mechanism increases a counter each time a session is abruptly terminated. An abrupt termination of a session is either a system crash (bluescreen) or a hard reset or power off. If a subsequent session is terminated correctly, the counter is reset to zero. Otherwise, the counter is increased again. When the counter reaches a given number of consecutive abruptly terminated sessions, the Collector cancels the loading of its drivers at system startup. Of course, the CrashGuard protection is only effective if it is really the Collector or its interaction with misbehaving software that is causing the system to crash.

On the other hand, if some end-users have acquired the bad habit of shutting their devices down unsafely, they may inadvertently induce the CrashGuard protection mechanism to wrongly deactivate the loading of the Collector by always ending up their sessions abruptly. To prevent these users from abusing the CrashGuard protection mechanism, optionally specify a CrashGuard protection time interval that starts being measured at system startup and after which the abrupt termination of a session does not increase the CrashGuard counter. This measure is effective if misbehaving users shut down their devices only after the protection time interval has elapsed; that is, if the protection time interval is shorter than their working day. The default value for the protection time interval is four hours.

Additionally, specify a reactivation time to load back a Collector that was previously deactivated by the CrashGuard protection. Take preventive measures during the period of deactivation to fix the issue that made the device crash so often. The Collector is loaded back again once the reactivation time has elapsed, on the next boot of the device.

Specify the parameters that control the behavior of the CrashGuard as arguments to the MSI of the Collector:

DRV_CRASHGUARD
: Maximum number of consecutive abruptly terminated sessions after which the Collector cancels the loading of its drivers. Default is five sessions.

DRV_CGPI
: The time interval in minutes after system startup during which the CrashGuard protection mechanism reckons abrupt terminations as genuine crashes. Default is 240 min (4 hours).

DRV_REACTIVATION
: The waiting time in hours from the moment that the loading of the Collector is deactivated because the CrashGuard counter reached its maximum specified count until the moment it is reactivated. Default is 168 hours (7 days).

Modify the CrashGuard protection time interval with the Collector configuration tool on active devices after installation.

In any case, if you suspect that there is a compatibility problem between any of the drivers loaded into the end-user devices of your company and the Collector, please contact Nexthink Support.

The operations described in this article should only be performed by a Nexthink Engineer or a Nexthink Certified Partner.

If you need help or assistance, please contact your Nexthink Certified Partner.

Related tasks

- Installing the Collector for a POC
- Setting the names of the Engines
- Federating your Appliances
- Importing and replacing Certificates
- Nxtcfg - Collector configuration tool
- Create or delete a Group Policy object
- SCCM 2007 POC Installation Guidelines

Related references

- Operating systems supported by the Collector
- Updating the Collector

- Collector MSI Parameters reference
- Windows Collector proxy support
- Microsoft Windows Server 2003/2008 Group Policy home page
- System Center Configuration Manager

Installing the Collector on macOS

Overview

Nexthink distributes the Collector for macOS as a disk image (**.dmg**) file with the following contents:

- A package (**.pkg**) file for installing the Collector from a graphical user interface.
- The application **csi.app** for installing the Collector from the command line interface.
- A reporter shell application that gathers system information in the case that you find any issue when running the Collector on macOS.
- An uninstaller application to remove the Collector when it is no longer needed.

After the installation, as a sanity check, optionally verify the status of the TCP connection between the Collector and the Engine.

Starting from V6.17, the Mac Collector runs in user mode and it does not need to ask the user for explicit permissions to install any kernel extension. The fact of running in user mode comes with the added benefit of making unnecessary to reboot your macOS devices after updating or uninstalling the Collector.

Applies to platforms:

Prerequisites

You need:

- One or more macOS devices where to install the Collector.
- The Nexthink Collector disk image (`Nexthink_Collector_<version>.dmg` file).
- The Customer Key and Root Certificate of the master Appliance. These are essential to enable the complementary TCP connection of the Collector with the Engine. Read this article if you need to install the Collector as part of a POC, before having installed the definitive master

Appliance.
- Optional: A third-party automated deployment tool.

Find the Nexthink Collector image file in the Product Downloads page of Nexthink:

1. Open your favorite web browser.
2. Navigate to the official Nexthink documentation web site: doc.nexthink.com.
3. Click **Product download** in the top right corner of the main documentation page, above the search tool.
4. In the **Nexthink Help Center**, click the **Product Downloads** section.
5. Sign in with your customer credentials.
6. Click the first entry of the **Latest V6 releases** list.
7. Optional: Click the link to the release notes of the Mac Collector to learn about the new features and bug fixes.
8. Under **Download links**, find the **Collector** section.
9. Click to download the **Collector package for Mac**.
10. Optional: Verify your download with the provided SHA-256 hash.
11. Click the downloaded file `Nexthink_Collector-<version>.dmg` file.
12. Find the package file (`Nexthink_Collector-<version>.pkg`) and the csi app inside the image file.

Download the Customer Key and default Root Certificate from the master Appliance:

1. Log in to the Web Console of the master Appliance as admin.
2. Select the **Appliance** tab at the top of the Web Console.
3. Click **Collector management** in the left-hand side menu.
4. Click the buttons **DOWNLOAD CUSTOMER KEY** and **DOWNLOAD DEFAULT ROOT CERTIFICATE** to download, respectively, the text files holding the Customer Key and the default Root Certificate of the master Appliance (the latter is required only if you did not replace the certificate for the TCP communication channel of the Appliances with the Collector).

You need to know:

- The DNS name or IP address of the Engine (as specified as External DNS name of the Engine in the Web Console).
- TCP port number for the connection of the Collector with the Appliances (default 443).
- Optional: UDP port number where the Engine is listening for the Collector (default 999, but prefer TCP-only).

Graphical installation

To install the Collector on macOS using the graphical interface:

1. Double-click the provided disk image file to mount it into your filesystem and see its contents.
2. Double-click the package file `Nexthink_Collector_<version>.pkg` and the installer starts with the introduction.
3. Click **Continue** to proceed with the installation.
4. In the step **Personalization**, configure first the settings of the **Nexthink Appliance** to which the Collector will connect:

    - **Name or IP address**: Type in the DNS name or IP address of the Appliance (the Portal, if rule-based collector assignment is turned on; the Engine, otherwise). Respectively, this setting must match either the External DNS name of the Portal or the External DNS name of the Engine.
    - Optional (recommended) **Data over TCP**: Tick to send all Collector data through the TCP channel. Ticked by default.
    - **TCP port**: Type in the port number on which the Appliance listens for TCP connections from the Collector. Default is 443. A custom TCP port must be in the non-privileged range (port number above 1024).
    - **UDP port**: Type in the port number on which the Appliance listens for UDP connections from the Collector. Only enabled if **Data over TCP** is not checked.
5. Still on **Personalization**, configure the proxy settings of the Collector:
    - Tick **Automatic proxy** for the Collector to take its configuration from a proxy auto-configuration (PAC) file.

1. In **PAC address**, type in the URL of the file that determines the proxy to use.

- Tick **Manual proxy** for the Collector to use the following proxy settings:
  1. **Address**: Type in the FQDN or IP address of the proxy.
  2. **Port**: Type in the port number where the proxy is listening for connections.

6. In a second step, configure the other settings of the Collector:

- **Customer Key**: Copy and paste the contents of the the file that holds the Customer Key of the master Appliance.
- **Root CA**: Copy and paste the contents of the file that holds the default Root Certificate of the master Appliance. If you leave this field empty, the Collector assumes that you replaced the server certificates in the Engine and falls back to using the Keychain Access for verifying the certificates presented by the Appliance. You must replace the certificates to communicate via the default TCP port 443.
- Optional **Collector tag**: Type in an integer number (0 to 2147483647) that identifies a group of Collectors. The Collector tag is visible in the Finder and is useful for defining the entities to build up hierarchies.
- Optional **Collector string tag**: Type in a label (max 2048 characters) that identifies a group of Collectors. The Collector string tag is visible in the Finder and is useful for defining the entities to build up hierarchies.
- Optional: Tick **Assignment service** if you activated the rule-based assignment of Collectors.
- Optional: Tick **Nexthink Engage** to activate the features that let you engage with the end user via campaigns (requires the

purchase of the Nexthink Engage product).

- Optional: Select the execution policy of scripts included in remote actions:
  - **Disabled** (default): the Collector runs no remote action on the device.
  - **Unrestricted**: the Collector runs any remote action on the device, regardless of the digital signature of its script.
  - **Trusted publisher**: the Collector runs on the device only those remote actions with a Bash script that is signed by a Mac *identified developer*.
  - **Trusted publisher or Nexthink**: the Collector runs on the device only those remote actions with a Bash script that is signed either by Nexthink or by a Mac *identified developer*.

7. Click **Continue** to proceed.
8. In the step **Destination select**, the installer program shows the local paths in the system where it is going to install the different components of the Collector. Keep the default paths and click **Continue**.
9. The **Installation Type** step informs you about some details of the installation process, including the amount of space that the program is going to occupy on disk. Click **Install** to begin with the actual installation.
10. The installer shows the progress of the installation and it finishes with a summary message. If the installation was successful, click **Close** to terminate the procedure.

Command line installation

The command line installation lets you install the Collector even when you have access to a computer only through the shell of macOS. Using the command line installation, you can thus install the Collector either locally or remotely through an *ssh* connection.

Execute the *csi* application provided with the disk image. To mount the disk image into the file system:

1. After downloading the image file from Product Downloads, pick one of the following options:
   - If you are installing the Collector in a remote computer:
     1. Copy the image file to the remote computer:
        ```
        scp Nexthink_Collector_<version>.dmg
        <username>@<address>:
        ```
     2. Log in to the remote computer:
        ```
        ssh <username>@<address>
        ```
   - If you are installing the Collector in the local computer:

1. Change the directory to the one where you downloaded the image file.

2. Mount the image file:

```
hdiutil mount Nexthink_Collector_<version>.dmg
```

Once with the image file mounted into the filesystem of the target Mac computer, install the Collector from the command line:

1. Change the directory to the path of the *csi* application:

```
cd
/Volumes/Nexthink_Collector_<version>/csi.app/Contents/MacOS
```

2. Type in the following command and provide, as arguments:
   - **address**: the FQDN or IP address of the Appliance.
   - **port**: the port number of the UDP channel in the Appliance.
   - **tcp_port**: the port number of the TCP channel in the Appliance.
   - **rootca**: the path to the Root Certificate.
   - **key**: the path to the Customer Key file
   - **(Optional) engage**: whether to enable the Engage campaigns or not (default is *disable*).
   - **(Optional) data_over_tcp**: whether to enable the sending of all data over the TCP channel (default is *enable*).
   - **(Optional) use_assignment**: whether to enable automatic collector assignment (default is *enable*).
   - **(Optional) ra_execution_policy**: whether to enable the Act remote actions or not with the following options:

     disabled (default)
     The Collector runs no remote action on the device.
     unrestricted
     The Collector runs any remote action on the device, regardless of the digital signature of its script.
     signed_trusted
     The Collector runs on the device only those remote actions with a Bash script that is signed by a Mac *identified developer*.
     signed_trusted_or_nexthink
     The Collector runs on the device only those remote actions with a Bash script that is signed either by Nexthink or by a Mac *identified developer*.
   - **(Optional) proxy_pac_address**: provide the URL of a PAC address for automatic configuration of proxy settings.
   - **(Optional) proxy_address**: provide the FQDN or IP address of a proxy for manual configuration of proxy settings.
   - **(Optional) proxy_port**: provide the port number where a proxy is

listening for connections for manual configuration of proxy settings.

- **(Optional) tag**: integer number (0 to 2147483647) to identify an individual or batch installation of Collectors.
- **(Optional) string_tag**: label (max 2048 characters) to identify an individual or batch installation of Collectors.
  For instance:

```
sudo ./csi -address <appliance_address>
-port <appliance_udp_port> -tcp_port <appliance_tcp_port>
-rootca <root_certificate_file> -key <customer_key_file>
-engage enable -data_over_tcp disable
-proxy_pac_address <pac_URL>
-proxy_address <proxy_FQDN_or_IP> -proxy_port <port_number>
-use_assignment disable -tag 1000 -string_tag Preproduction
```

Enterprise deployment

The Collector supports its installation in an enterprise environment being based on either:

- Imaging
- Mobile Device Management (MDM)

Choose the method that better suits your needs depending on your deployment workflow.

Uninstalling the Collector

To uninstall the Collector, execute the *uninstaller* script that is provided with the image file. Assuming that you have mounted the image file into the filesystem of the computer where the Collector is installed:

1. From the shell, type in the following command:

```
sudo /Volumes/Nexthink_Collector_6.x.x/uninstaller
```

The operations described in this article should only be performed by a Nexthink Engineer or a Nexthink Certified Partner.

If you need help or assistance, please contact your Nexthink Certified Partner.
Related tasks

- Querying the status of the TCP connection of the Collector

Related references

- Operating systems supported by the Collector

Deploying the Collector with AirWatch

Overview

If your company happens to use VMware products for Unified Endpoint Management, take advantage of AirWatch to deploy the Nexthink Collector on both your macOS and Windows devices.

Applies to platforms:

Installing and enrolling the AirWatch agent

This guide assumes that you have already installed the AirWatch agent on your macOS and Windows devices and that you have enrolled them. If this is the case, skip this step.

Otherwise, to manually install the AirWatch agent on a particular device, look for the application either in the Apple or in the Windows store, or download it from https://awagent.com/, and follow the installation instructions. Once the agent is installed, an enrollment window shows up. To enroll the device, if you know the server details:

1. Click the **Server Detail** button under **Authenticate with**.

2. Type in the address of the AirWatch server.
3. Enter the name of the group in which to enroll the device. A second login screen shows up.
4. Enter your AirWatch account username and password.
5. Optional: Click **Connectivity status** in the left-hand side pane of the **Status** menu to verify that the agent is connected to the server.

If successful, the AirWatch agent is now enrolled and ready to be controlled remotely.

Deploying the Collector

To deploy the Collector with AirWatch, you need to be an AirWatch administrator. Follow these steps in the administration console:

1. Create an Organization Group.
    1. Login to the AirWatch administration console in the AirWatch server.
    2. Go to **Groups & Settings ? Groups ? Organization Groups ? Organization Group Details**.
    3. Click **Add Child Organization Group**.
    4. Fill in the properties of the group.
    5. Click **Save**.
2. Add devices to the Organization Group.
    1. Go to **Devices ? List view**.
    2. Select the devices that you have previously enrolled on which to install the Collector.
    3. Click **More Actions ? Change Organization Group**.

4. Enter the name of the group to which you want to assign the device.
5. Click **Save**.

3. Create an action to be executed on the remote devices.
    1. Go to **Devices ? Staging & Provisioning ? Components ? Files/Actions** in the left-hand side pane.

    2. Click **Add Files/Actions**.
    3. Select the Operating system of the target devices.
    4. Fill in the name and enter the name of the Organization Group that you created above
    5. Click the **Files** tab.
    6. Click **Add Files** to upload files that will be copied to the target devices when running the Action.
        - ◊ For Windows devices, upload an executable file that you have previously generated with the Nexthink Collector Installer,
        - ◊ For macOS devices:
            1. Upload the `csi.app` file that you can find inside the installer package for installing the Collector from the command line.
            2. Upload the customer key and root-ca files that you can get from the Web Console, as explained in the prerequisites for installing the Collector on macOS.

1. In a second window that shows up, enter the path where the installer files will be downloaded on the remote device.

7. From the **Add Files/Actions** menu, click the **Manifest** tab.
8. Click **Add Action** under **Install Manifest**.
9. Select **Run** from the dropdown list of Actions to perform.
10. Under **Command line and Arguments to run**, type in the commands by always specifying the full path to the referred files:
    - ◊ For Windows devices, type in the command to execute the installer.

      ```
      C:\tmp\installer.exe
      ```

    - ◊ For macOS devices, type in the following two commands, the first to unzip the csi application file and the second to run the installer. Do not add 'sudo' before the commands, since all specified commands are run as root by default:

      ```
      /usr/bin/unzip -o -a /tmp/csi.app.zip -d /tmp
      chmod +x /tmp/csi.app/Contents/MacOS/csi
      /tmp/csi.app/Contents/MacOS/csi -address
      <engine_ip> -port 999 -tcp_port 8443 -key
      <customer_key_file> -rootca <root_ca_file>
      ```

11. Click **Save**.

4. Create a Product to deploy the Action at the target device.
    1. Go to **Devices ? Staging & Provisioning ? Product List View**.
    2. Click the **Add Product** button.
    3. Enter the name of the Product.
    4. Optional: Type in a description of the Product.
    5. Select the group that you created earlier in the **Managed By** field.
    6. Select the same group under **Assigned groups**.
    7. Click the **Manifest** tab.
    8. Click **Add** and a dialog shows up.
        1. From the list **Actions to perform**, select **Install Files/Action**.
        2. From the list **Files/Actions**, select the Action that you have previously created.
    9. Click **Save**. You will be prompted to Activate or Deactivate your Product.
        - ◊ Click **Activate** if you want to run the execution immediately.
        - ◊ Click **Deactivate** if you want to run it later.
5. Run the existing Action.

1. Go to **Devices ? Staging & Provisioning ? Product List View** to get a table of existing Products that indicates whether the Product is active (green light) or not (red light).
    - ◊ To run a Product that is not active, click the red light to turn it green. and it will be executed immediately
    - ◊ To re-run an active Product:
        1. Select the Product that you want to execute by clicking the radio button at the beginning of the row in the table.
        2. On the top of the table, click **More Actions ? Reprocess**
6. Optional: Check the status of executed Action.
    1. Go to **Devices ? List View**
    2. Select the desired target device to inspect.
    3. Click the tab **More**.
    4. Under **Products**, see the history of executed Actions and their status.

Related tasks

- Installing the Collector on Windows
- Installing the Collector on macOS

Installing the Collector for a PoV

Overview

The procedure to install the Collector for a PoV greatly depends on whether you are opting for either the on-premises offering or the Cloud offering of Nexthink.

Applies to platforms:

On-premises PoV

Starting from V6.6 of the Windows Collector and V6.16 for the Mac Collector, the installation of the Collector requires two additional parameters from the master Appliance:

- The Customer Key.
- The Root Certificate.

These parameters ensure the security of the TCP communications of the Collectors with the Appliances. In the context of a proof of value (PoV) however, it is customary to deploy a few Collectors before having installed the master Appliance. As the master Appliance is needed to generate both the Customer Key and Root Certificate, it is not possible though to install the Collectors before having one master Appliance ready.

To solve this problem, the following method lets you to create a Customer Key and a Root Certificate from an *ad hoc* master Appliance and later transfer the same Customer Key and Root Certificate to the actual master Appliance that the customer will use in production.

Generating a Customer Key and Root Certificate in the ad hoc Appliance

To generate the Customer Key and Root Certificate:

1. Set up a Nexthink Appliance including both the Portal and the Engine in an environment that you control. To avoid possible conflicts, preferably install the same version of the Appliance that will later be used in production.
    - ♦ You can use, for instance, the Appliance distributed with the official Nexthink Demo kit.
2. Download the script for generating a new Customer Key and Root Certificate: gen_rck.sh.
3. Copy the script to your controlled Appliance using any SCP tool.
4. Log in to the CLI of the Appliance.
5. Execute the script as root and verify in the output message that a new Root Certificate and Customer Key are generated:
   ```
   sudo sh gen_rck.sh
   ```
6. Open a web browser and log in to the Web Console of the Appliance as admin.
7. In the **Appliance** tab, select the **Collector management** section on the left-hand side menu.
8. Under **Collector default certificates** at the bottom of the page, click the button **BACKUP CERTIFICATE AND KEY** to get a backup of the generated Root Certificate and Customer Key. The backup file has the name **root-ca-backup.tgz**. You will later use this file to transfer the Root Certificate and Customer Key to the production Appliance.

Installing the Collectors

After generating the Root Certificate and Customer Key, use them to install the Collectors for your PoV:

1. Open a web browser and log in to the Web Console of the Appliance as admin.
2. In the **Appliance** tab, select the **Collector management** section on the left-hand side menu.
3. Look for the section **Collector default certificates** at the bottom of the page.
    1. Click the button **DOWNLOAD CUSTOMER KEY** to get the file **Nexthink-customer-key.txt**.
    2. Click the button **DOWNLOAD DEFAULT ROOT CERTIFICATE** to get the file **Nexthink-root-ca.txt**.
    3. Click **Yes** in the dialog that shows up to confirm the download.
4. Use the downloaded files for installing the Collectors by means of any of the available methods.

When installing the Collectors, use the appropriate name or IP address to point to your controlled Appliance.

Deploying the Customer Key and Root Certificate in the production Appliance

Once your PoV has been successfully completed and the customer has installed the definitive master Appliance to be used in production, deploy the generated Root Certificate and Customer Key in the production Appliance:

1. Copy the backup file **root-ca-backup.tgz** to the master Appliance using any SCP tool.
2. Download the script for deploying the Customer Key and Root Certificate: deploy_rck.sh.
3. Copy the script to the master Appliance using any SCP tool.
4. Execute the script as root, passing the backup file as argument.
    ```
    sudo sh deploy_rck.sh root-ca-backup.tgz
    ```
5. Open a web browser and log in to the Web Console of the master Appliance as admin.
6. In the **Appliance** tab, select the **Network Parameters** section on the left-hand side menu.
7. Type in the **External DNS name** and the **Internal DNS name** of the master Appliance.
8. If the Portal and the Engines are hosted in different Appliances (the master Appliance is not in a master / slave configuration itself):
    1. In the Appliance tab, select the **Federated appliances** section on the left-hand side menu.
    2. Remove all Engines from the list of federated appliances (if any) by repeatedly clicking the **Delete** link to the rightmost side of each

entry.
3. Log in to the Web Console of the Appliance hosting one of the Engines that you want to federate as admin.
4. In the **Appliance** tab, select the **Network Parameters** section on the left-hand side menu.
5. Type in the **External DNS name** and the **Internal DNS name** of the slave Appliance (Engine).
6. Repeat the previous three steps for every Engine that you want to federate.

9. Back to the Web Console of the master Appliance, select the **Collector management** section on the left-hand side menu.
10. Click the button **GENERATE CERTIFICATE** that is displayed in red.
11. If your Engines reside in separate slave Appliances, federate them now:
    1. Select the **Federated appliances** section on the left-hand side menu.
    2. Click **ADD APPLIANCE** to add a new slave and provide the necessary information.

Cloud PoV

The following content applies exclusively to the Nexthink Cloud offering.

In a Nexthink Cloud setup, the installation of the Collector for a PoV does not differ much from the installation of the Collector in production. It is only the quantity of Collectors deployed which may vary.

Contrary to the on-premises case, there is no need to generate the Customer Key, as it is provided by Nexthink. In turn, there is no need to provide a Root Certificate, as the connection between the Collectors and the Nexthink Cloud is secured by certificates signed by a public CA.

For a PoV in a Nexthink Cloud setup, you just need to provide appropriate values when installing the Collectors:

As Nexthink Appliance settings, use the following:

- **Address**: Type in the FQDN of your Cloud platform as given to you by Nexthink. The name is of the form `[customer].nexthink.cloud`.
- **Data over TCP**: Enable the transmission of Collector data over TCP.
- **Ports (TCP)**: Type in port number **8443**.

As general settings, use the following:

- **Use the assignment service**: Enable the rule-based assignment of Collectors to Engines.
- **Script execution policy**: Signed by a trusted publisher or by Nexthink
- **Customer key**: Select the file that Nexthink provided to you with your unique Customer Key.
- **Root CA**: Leave it empty.

Related tasks

- Installing the Collector in Windows
- Assigning Collectors to Engines

Assigning Collectors to Engines

Overview

Every Collector must know the address of its assigned Engine to work properly. The address of the Engine was thus typically embedded in the installer of the Collector when generating the installer itself. Hence deploying the Collector on a setup with multiple Engines required to generate a different installer for every group of Collectors that were assigned to a same Engine.

Once installed, each Collector could be reassigned to any other Engine by means of the Collector Configuration tool or the Control Panel extension, when available.

With rule-based Collector assignment turned on, generate instead a single installer that provides the address of the Portal (and not of the different Engines)

to all Collectors. Upon first connection, each Collector receives the address of its assigned Engine from the Portal. Thereafter the Collector can start sending end-user information to the Engine as usual:

In subsequent connections to the Engine, each Collector checks whether it is assigned to the same or to another Engine and switches Engines accordingly. Hence, in a migration scenario, if a Collector is already connected to an Engine in legacy mode and you turn rule-based assignment on, you do not need to configure the Portal address for that Collector to receive its first assignment, as the Collector will directly receive the assignment from the Engine instead.

The Portal manages the assignment process thanks to a configurable set of rules. By modifying the set of rules, dynamically reassign Collectors to different Engines. The rules define the assignment of Collectors not only to Engines, but also to entities, which constitute the base to organize your devices in hierarchies. Therefore, note that activating rule-based Collector assignment overrides the conventional method to assign entities to devices.

In addition to the connectivity between Collectors and Portal, the process of assigning Collectors requires extra connections between the Portal and the Engines. Review the connectivity requirements for rule-based Collector assignment.

Regarding platform compatibility. Both Windows Collectors V6.19 and above and Mac Collectors V6.21 and above support rule-based assignment of Collectors.

The entity assignment specified by the same rules is valid for any version of both Windows and Mac Collectors, even if they do not have a working TCP connection.

Applies to platforms:

Support for rule-based Collector assignment

Starting from V6.20 and unless otherwise specified (see exceptions below), rule-based Collector assignment is the preferred method to assign devices to Engines in new installations. Please contact Nexthink Customer Success when upgrading a previous version of Nexthink with legacy assignment of Collectors.

The operations described in this article should only be performed by a Nexthink Engineer or a Nexthink Certified Partner.

If you need help or assistance, please contact your Nexthink Certified Partner. Rule-based Collector assignment supports rules with conditions on the following device fields:

- **Last IP address**
- **Last local IP address**
- **Name**
- **Distinguished name reported by Collector**
- **AD site**
- **Collector tag**
- **Collector string tag**

Ensure that you do not have any roaming devices that switch between Engines and whose assignment is based exclusively on the Last IP address of the device. Starting from V6.24, using the Last local IP address of the device instead solves the issue.

Incompatibilities

When the conditions depend on the IP address of the device, rule-based Collector assignment is incompatible with:

- ◊ The local IP address beta feature, which is deprecated from V6.24 and superseded by the Last local IP address field.
- ◊ The redirection of Collector traffic. If you are running the redirection service and you want to implement rule-based Collector assignment on your setup, contact Nexthink Customer Success Services to verify the compatibility of both techniques for your particular setup.

Managing certificates

Because Collectors can now communicate with the Portal through their TCP connection, if you installed a custom certificate to protect the TCP

communication of the Collectors with the Engine, you need to upload the custom certificate to the Portal appliance again.

Writing the assignment rules

Express the rules for Collector assignment in a CSV file that has the tabular structure of the following example:

entity_rule	Engine	Entity	Field1	Pattern1	Field2	Pattern2
yes	France	Lyon	ad_site	Lyon-?	DN	*OU=MAR*
no	France	Paris	local_ip	192.168.10.0/24	name	FR*
yes	USA	Los Angeles	collector_tag	200		
no	USA	New York	name	US*	ip	10.100.0.0/16

Each column in the CSV file means the following:

entity_rule (case sensitive)
: When set to **no**, the rule assigns the device to both an Engine and an entity.
When set to **yes**, the rule assigns the device to an entity, filtered by Engine.

Engine (case insensitive)
: The name of the Engine to be assigned, as specified in the connection of the Portal to the Engines.

Entity (case sensitive)
: The name of the entity to be assigned.

Field1
: The name of the first device field on which to base the assignment.

Pattern1
: The condition pattern that the first specified field must match for assigning the designated Engine and entity to the device.

Field2
: The name of the second device field on which to base the assignment.

Pattern2
: The condition pattern that the second specified field must match for assigning the designated Engine and entity to the device.

The columns **Field1** and **Field2** support the following values:

ip
: The last IP address of the device. As pattern for the field, specify either:

- ◊ A single IP address in dot-decimal notation. For example: 192.168.10.1.
- ◊ A subnet in CIDR notation. For example: 192.168.10.0/24.

local_ip
The last local IP address of the device; that is, its IP address in the local network. Specify patterns in the same way as for the **ip** field.

name (case insensitive)
The name of the device.

collector_tag
The tag number assigned to the Collector during its installation. Only the exact number is matched.

collector_string_tag
The label assigned to the Collector during its installation. Supports pattern matching.

dn
The Distinguished Name of the device as reported by the Collector. The device must be part of a domain in Active Directory.

- ◊ The format of the distinguished name reported by the Collector is the standard sequence of **attribute**=*value* elements connected by commas, from the most specific to the most general attribute. For instance: `CN=ex01,OU=Computers,DC=example,DC=org`
- ◊ In contrast, when retrieved by the Engine from Active Directory, the Finder displays the distinguished name field of a device (or of a user) as a similar sequence in reverse order, with elements connected by a forward slash. The same example dn would give: `/DC=org/DC=example/OU=Computers/CN=ex01`

ad_site
The Active Directory Site in which the device is located. A site represents one or more TCP/IP subnets.

The **name**, **collector_string_tag**, **dn** and **ad_site** fields support character pattern matching. To define the associated string pattern, use the following wildcards:

- ◊ ? : Substitutes a single character.
- ◊ * : Substitutes zero or more characters.

To write the CSV file for the rules, use UTF-8 text encoding (max 20 MB). Avoid writing the rules on editors that create a BOM character at the beginning of the file (e.g. Notepad), as this results in a *header error* when uploading the file to the Portal.

After the header, write each rule in a new line, optionally enclose each item in double quotes (needed to escape special characters), and use the semicolon as delimiter. For instance, the CSV file of our example would look as follows:

```
"entity_rule";"Engine";"Entity";"Field1";"Pattern1";"Field2";"Pattern2"
"no";"France";"Paris";"local_ip";"192.168.10.0/24";"name";"FR*"
"yes";"France";"Lyon";"ad_site";"Lyon-?";"dn";"*OU=MAR*"
"no";"USA";"New York";"name";"US*";"ip";"10.100.0.0/16"
"yes";"USA";"Los Angeles";"collector_tag";"200";;
```

For a rule to be satisfied, conditions on both Field1 and Field2 must be fulfilled, that is, both patterns must match. The rule precedence is established from top to bottom, but their processing depends on whether Collectors have the **use_assignment** flag enabled:

- If rule-based assignment is enabled in the Collector:
    1. Non entity-only rules (i.e. **entity_rule = no**) are processed first, assigning the device to a particular Engine.
    2. All rules that contain that particular Engine (e.g. **France**) are processed afterwards, assigning the device to the appropriate entity.
- If rule-based assignment is disabled in the Collector:
    1. The Collectors have their Engine assigned during installation and it does not change.
    2. All rules that contain the Engine to which the Collector was assigned are processed, assigning the device to the appropriate entity.

Entity-only rules allow catch-all default cases to prevent assignments to the empty (-) entity from occurring; especially for sticky, roaming, or VPN devices.

To upload the resulting CSV file to the Portal:

1. Log in to the Portal as central administrator.
2. From the **ADMINISTRATION** menu at the top, select **Collectors** under **SYSTEM CONFIGURATION**.
3. Click **Add new ruleset** under the **Rulesets** section.
    1. Fill in the fields in the dialog:
        ◊ For **RULE NAME**, type in a unique name for the rule.
        ◊ Optional: In **DESCRIPTION**, describe the purpose of the rule.
        ◊ Under **CSV FILE**, click **Upload new file** and select your previously created CSV file.

Simulating Collector assignment

For a seamless transition from manual to rule-based Collector assignment and to avoid unexpected assignments, simulate your rules in the Portal. The simulation of Collector assignment rules lets you see what the effect of applying the rules would be, without actually changing the currently assigned Engines and entities.

- **Warning**: If you are migrating from legacy to rule-based assignment, do not switch on rule-based assignment in the Collectors yet. Wait until rule-based assignment is fully activated. Otherwise, functions that depend on the TCP connection of the Collector, such as Engage, Act or Updates, will not work properly.

The simulation of assignment rules is thus specially useful in two situations:

- When switching from manual to rule-based assignment.
- When testing a new set of rules.

Therefore, before completely switching to rule-based assignment, try the so-called *simulation mode*. In simulation mode, Collector assignment is still manual, but the Portal lets you simulate the effect of assignment rules. Safely evaluate the readiness of your setup and the convenience of switching to rule-based assignment thanks to simulation mode.

To activate simulation mode:

1. Log in to the Web Console of the master appliance as administrator.
2. In the **Appliance** tab, select **Collector management** from the left-hand side menu.
3. Select **Manual Collector Assignment (legacy) but simulation of rule-based assignment is enabled**.
4. Click **SAVE**. Saving your changes restarts the Portal and all Engines.

To simulate your own assignment rules, write the rules in a CSV file and upload it to the Portal as explained in the previous section. From the same **Collectors** page in the Portal, find your newly created ruleset on the list and click **simulate**. The results of the new assignment are displayed.

Discovering your Collectors

While in simulation mode, see the list of Collectors that have a working TCP connection (UDP-only Collectors are not visible here yet):

1. Log in to the Portal as central administrator.
2. From the **ADMINISTRATION** menu at the top, select **Collectors** under **SYSTEM CONFIGURATION**.
3. See the list of devices whose Collector is communicating with the Portal under **Devices**. Only the first 100 devices are displayed.
4. Optional: Click **CSV** to get the full list of devices.

The columns of the table offer you current information about the assignment status of the Collectors. These columns include the properties of the Collector: **Device name**, **Collector version**, **IP address** (the last one), **AD Site**, **Distinguished name** and **Collector tag**. In addition, find other columns that are useful to know about the assignment status of the Collector:

Last seen (UTC)
: The time of the last assignment request received from the Collector.

Last seen on
: The IP address and port of the appliance where the Collector was last seen.

Assigned Engine
: The Engine to which the Collector was told to report.

Assigned entity
: The entity to which the Collector currently belongs.

Roaming since (UTC)
: Time at which the device started roaming.

Error message

Last error in the assignment of the Collector.

Activating Collector assignment

To definitively activate rule-based Collector assignment:

1. Log in to the Web Console of the master appliance as central administrator.
2. In the **Appliance** tab, select **Collector management** from the left-hand side menu.
3. Select **Rule-based Collector Assignment**.
4. Click **SAVE**. Saving your changes restarts the Portal and all Engines.

Beware that, once you activate rule-based Collector Assignment, there is no way to revert to manual assignment except by reinstalling the whole product. Indeed, once in rule-based mode, the other options disappear from the Collector management page in the Web Console:

Current active rules are then enforced, but remember that you can always simulate new rules before applying them. To apply a new set of rules, write and upload a new CSV file to the Portal, as explained in the previous section, and click **activate** from its entry on the list of rules in the **Collectors** page of the Portal. In the definitive rule-based mode, the **Collectors** page displays all Collectors, including those that do not have a working TCP connection, but a UDP only connection.

After activating rule-based Collector assignment in the Web Console, note that TCP connection related functions such as Engage, Act or Updates will not work on a particular Collector until the Collector has been properly assigned to an Engine. Use a configuration management tool to enable rule-based assignment in all your Collectors by setting the USE_ASSIGNMENT property to *enable*.

Specifying the Collector Assignment port

Once you have activated rule-based Collector assignment, specify the TCP port where each Engine advertises the assignment service:

1. Log in to the Web Console of each Engine as central administrator.
2. In the **Appliance** tab, select **Collector management** from the left-hand side menu.
3. Under **Collector communication**:
    - Select **TCP port 443 for Collector communication** when Collectors are configured to communicate via the default TCP port 443.
    - Select **Custom TCP port for Collector communication** when Collectors are configured to communicate via a custom TCP port.

4. Click **SAVE**.

Reassignment

Once the Portal has successfully assigned an Engine to a Collector, the Collector receives new assignment information from its assigned Engine, which in turn receives it from the Portal. The Collector asks for new assignment information every time that its TCP connection to the Engine is interrupted.

In addition, starting from V6.25, Collectors can be dynamically reassigned to another Engine every time that the assignment rules change. The distribution of reassignment messages to Collectors may take up to 60 minutes.

On the other hand, to limit the number of undesired Engine switches, the reassignment of a device that changes its properties (e.g. IP address) is delayed for a default of 10 days after a new rule takes over. This is especially important for devices that change frequently of subnet, such as the laptops of roaming users. Learn how to configure the *stickiness* of devices to their assigned Engines in the article about the assignment of roaming Collectors.

Assignment failure scenarios

Assignment bound to wrong IP address in Appliance

In Nexthink Appliances with multiple network interfaces, it may happen that rule-based assignment is bound to the incorrect network interface. Usually, rule-based assignment should bind to the same IP address to which the specified **Internal DNS Name** the Portal. However, during a fresh installation of the Portal, rule-based assignment may choose the wrong interface while the Internal DNS name is still not configured.

To know whether the communication between the Portal and the Engines is working for rule-based assignment, activate first simulation mode or rule-based assignment from the Web Console, as explained above, and then:

1. Log in to the CLI of the master Appliance (i.e. the Portal).
2. Query the rule
   ```
   nxconsul members
   ```
3. Verify that the output lists all the Appliances, Portal and Engines, within your setup.
4. Check the **Status** column of all entries display **alive**.

If you do not see the full list of Appliances or they are not all alive, edit the configuration file of `nxconsul`. Still from the CLI of the master Appliance:

1. Open the configuration file for editing:
   ```
   sudo vi /var/nexthink/nxconsul/conf/base-config.json
   ```
2. Verify that the first line of content holds the IP address that rule-based assignment should use:
   ```
   "advertise_addr": "172.19.2.47",
   ```
3. If this is not the case, edit the IP address.
   1. Move the cursor to the beginning of the IP address with the keys **h-j-k-l** (respectively, move left, up, down and right).
   2. Type **i** for inserting.
   3. Repeatedly press **Delete** to erase the wrong IP address.
   4. Type in the correct IP address.
   5. Press **Esc**.
4. To save your changes and exit, type in:
   ```
   :wq
   ```
5. Restart `nxconsul`
   ```
   sudo systemctl restart nxconsul
   ```

Changes in the IP address of the Appliance

Once the master Appliance configures its IP address for rule-based assignment, an alteration of its actual IP address (because of a network change, for instance) does not automatically update the configured IP address. This results in rule-based assignment malfunction because `nxconsul` is unable to elect the master Appliance as leader.

To verify that `nxconsul` cannot elect a leader:

1. Log in the CLI of the master Appliance.
2. In file `/var/nexthink/nxconsul/logs/nxconsul.log`, look for the message:
   [ERR] agent: failed to sync remote state: No cluster leader

To manually update the IP address of the Appliance:

1. Open the configuration file for editing:
   ```
   sudo vi /var/nexthink/nxconsul/conf/base-config.json
   ```
2. Look for the line that holds the old IP address of the Appliance. For instance:
   ```
   "advertise_addr": "172.19.2.47",
   ```
3. Insert the new IP address.
   1. Move the cursor to the beginning of the IP address with the keys **h-j-k-l** (respectively, move left, up, down and right).
   2. Type **i** for inserting.
   3. Repeatedly press **Delete** to erase the old IP address.
   4. Type in the new IP address.
   5. Press **Esc**.
4. To save your changes and exit, type in:
   ```
   :wq
   ```
5. Restart `nxconsul`
   ```
   sudo systemctl restart nxconsul
   ```

Collector failing to connect to assigned Engine

When a Collector gets a new assignment, the Collector tries to establish a TCP connection to its assigned Engine using the first external DNS name specified for the Engine as destination. If the Collector is unable to contact the assigned Engine after three retries, the Collector resorts to its previously assigned Engine (or to the Portal, in case of first assignment) to report the failure and waits for a new assignment on standby. While on standby, the Collector sends no activity traffic.

To see if any assignment of a Collector to an Engine has failed:

1. Log in to the Portal as central administrator.
2. From the **ADMINISTRATION** menu at the top, select **Collectors** under **SYSTEM CONFIGURATION**.
3. Locate the list of assigned Collectors under **Devices**.
4. Optional: In the column **Message**, look for list entries with the error message:
   Last assignment failed: engine *[IP address]* tcp port *[number]*: ...
5. Because only 100 devices are displayed, click **CSV** at the bottom right to get the full list of devices and look for the error message in the same column of the exported CSV file.

Related tasks

- Installing the Windows Collector
- Importing and replacing certificates
- Hierarchizing your infrastructure
- Redirecting Collector traffic
- Assignment of roaming Collectors
- Connecting the Portal to the Engines
- Setting the names of the Portal
- Setting the names of the Engines

Related references

- Connectivity requirements

Related concepts

- Entity

Assignment of roaming Collectors

Overview

When a device is roaming, the changes to its network connection usually trigger a change of its IP address and Active Directory Site. If your rules to assign Collectors to a particular Engine depend on these properties of the device, a roaming device may switch to a another Engine every time that it connects to a

different network.

Avoiding the switching of devices from one Engine to another has the following advantages:

- Improvement of data organization, because a device always sends data to the same Engine.
- Economy of licenses, because devices do not consume additional licenses on other Engines.

To avoid repeated switching between Engines of roaming devices, set a reassignment delay (stickiness) value, so that a device can switch from its assigned Engine only after the delay has elapsed.

Roaming devices that connect to a VPN may also have suffer from assignment issues if two different locations reuse the same VPN subnetwork. In the last part of this article, we consider the VPN case in detail.

Configuring the Collector stickiness

To set the stickiness in number of days:

1. Log in to the CLI of the master appliance.
2. Optional: Verify the current stickiness with the command below. If no value was previously set, the command returns an error message because the key does not exist, which implies default stickiness:
   ```
   nxconsul kv get config/nxassignment/standard
   ```
3. Type in the following command, where **N** is the delay in number of days that must elapse before the system considers that a device is roaming:
   ```
   nxconsul kv put config/nxassignment/standard \
   nxassignment.stickiness.number-of-days=N
   ```

Stickiness special values:

- The default stickiness value is 10 days.
- A stickiness value of 0 days triggers a change of Engine on every network reconnection, according to the assignment rules.

Determining when a device is roaming

A device may change from assigned Engine because:

- The properties (name, IP address, AD site or Collector tag) of the device change and a different assignment rule applies.
- An administrator activates a different set of assignment rules.

The system considers that a device is roaming in the first case only, that is, when the properties of the device change. Therefore, the stickiness value is enforced only when a device is roaming and not when modifying the assignment rules.

Note however that devices that are regarded as roaming devices are not necessarily moving from one network to another. For instance, modifying the name of a device in a way that triggers a switch of assigned Engine as a result also makes the device to be treated as a *roaming* device.

See whether a particular device is considered to be roaming (in the broad sense of the word defined above) from the *Collectors* page of the Portal.

Entity assignment of roaming devices

When rule-based assignment is based on the IP address of devices, sticky roaming devices usually get the empty entity (**-**) assigned.

The reason is that a sticky roaming device that switches networks gets a different IP address, but keeps pointing to the same Engine. If this new IP address does match any of the IP rules in the original Engine, which is usually the case when networks are partitioned to differentiate entities, the device is assigned to the default empty entity.

In general, to determine the entity that holds a roaming device, the properties of the device are matched against the rules of the Engine that is receiving its traffic.

Roaming devices on Virtual Private Networks

In some scenarios, two different locations in a corporate network may reserve the same subnet addresses for VPN access. For instance, let us imagine a set of assignment rules that try to map a single VPN subnet address to two different entities or Engines:

Engine	Entity	Field1	Pattern1
France	Paris	ip	192.168.10.0/24
Switzerland	Nyon	ip	192.168.10.0/24

As the VPN subnet is the same for both locations, when exclusively using the IP address as condition in the rules to assign Collectors, devices will always be

assigned to the Engine in France, because the first rule takes precedence over the second.

To solve this issue, add a second rule based on a different field other than IP address to disambiguate. Of course, this solution requires that you can identify devices by that other field:

Engine	Entity	Field1	Pattern1	Field2	Pattern2
France	Paris	ip	192.168.10.0/24	name	FR-*
Switzerland	Nyon	ip	192.168.10.0/24	name	CH-*

Alternatively, if no other field can help distinguish your devices, use the stickiness value to keep devices on the VPN assigned to the same Engine. This solution works if roaming devices are usually connected to the office network and they only move to VPN access from time to time, where the period on the VPN is lower than the delay specified as stickiness. In this way, they will never switch the Engine, although you will never see them in the entity assigned to the VPN either.

Related tasks

- Assigning Collectors to Engines

Collector MSI parameters reference table

Applies to platforms:

Mandatory parameters

Option Name	Default value	Description
DRV_IP	-	Engine IP or DNS name.
DRV_PORT	-	Engine port number for the UDP channel.
CRD_KEY	-	Customer Key of the Engine Appliance.
CRD_ROOT_CA	-	Root CA of the Engine Appliance -leave blank if not using the Nexthink PKI.

Optional parameters

Option Name	Default value	Description
CRD_PORT	443	Engine port number for the T

DATA_OVER_TCP	enable	Send end-user data over the channel of the Collector (defa from V6.24) instead of the tra channel. Set the option to *dis* convey end-user data ov
CFG_INSTALL	1	Install the Nxtcfg tool for char configuration of the Collector command line. 1: install, 0: do
CPL_INSTALL	0	Install the Collector Control P extension. 1: install, 0: do not
DRV_ACTIVATE_DMP	0	Specifies whether the target s should be configured for gene memory dumps in case of ST message (System crash). Its 0 (disabled), 1 (full memory d (kernel memory dump) and 3 minidump). The recommende (kernel memory dump). • This is a non-reve setting: it will not l back to its initial v uninstalling Collec • The MSI package change the syster for a less verbose dump setting (e.g setting is to gener memory dumps a DRV_ACTIVATE_ set to 3 (memory minidump), no act performed)
DRV_BFBD	0	Delay in seconds during initia the driver before the Collecto sending UDP packets to the E Maximum value: 240 (4 min)
DRV_CRASHGUARD	5	Specifies the maximum Crash count that the Collector can r canceling the loading of the d boot-time. If set to 0, the Cras feature will be disabled
DRV_DESC	0	Delay Engine communication Creation : To avoid having th blocked by certain firewalls, t

		socket layer is created during initialization steps. [1: enable
DRV_LOGSIZE	32	Addition of log rotation when DRV_LOGMODE for the logg for value: 1 -> 512 (MB)].
DRV_REACTIVATION	168	Reactivate the Collector once time (in hours) has elapsed a reached the maximum Crash The max value is 8766 hours Default is 7 days (168 hours).
DRV_TAG	0	Integer number to identify and installations of the Collector. values range from 0 to 21474
DRV_STRING_TAG	-	Character string to identify an installations of the Collector. values are any string with up characters.
DRV_LOGMODE	0	Specifies the logging mode. P values are 0, 1 and 2, meanin Verbose and Debug, respecti (Debug) is not recommended
DRV_DWEF	0	Disables Windows enumerate functionality. Possible values set to 1, the Collector does no Windows freeze or hung prob (This will result in the Fir displaying any informatio "application not respond
DRV_CGPI	240	?CrashGuard Protection Inter It is the time interval since bo minutes) after which a dirty re not increase the CrashGuard hours (240 min).
DRV_MSS	1224	Maximum size of the UDP pa transfers between the Collect Engine. Allowed values range to 16384.
DRV_PKGI	1	Period, in hours, in whic Collector checks for new packages and updates a them. Allowed values ra to 24: • 0 - Never report p

		• 1 - Report packag Collector initialize every 1 hour. • 2-24 - Report pac minutes after Coll initializes and the to 24 hours.
DRV_WEB_AND_CLOUD_DATA	1	Gather Web and Cloud inform Default value is 1 to gather a data (only if you have purcha and Cloud product). Set to 0 recording the web connection
DRV_WEB_AND_CLOUD_HOSTS	-	List of comma separated hos which to send the full URL of request. Requires the Web a product and the parameter **DRV_WEB_AND_CLOU** to be set to 1.
PRINTING	disable	Disable print notifications. Sta version 6.18 the Collector do printing information by defaul option to *enable* to report activity and to *disable* to it.
DRV_DSPS	1	Disable SMB print notification from version 5.2.8.0, the Coll not report SMB prints by defa option to 0 to enable SMB pri (requires **PRINTING** enab to 1 to disable it.
DRV_PREFERIPV6	0	Favor IPv6 over IPv4 (or vice communicating with the Engi the DNS lookup of the name Engine resolves to both IPv6 addresses, prefer IPv6 when IPv4 when set to 0.
CUSTOM_SHELLS	0	Enable the reporting of user l and user interactions in virtua embedded (kiosk mode) envi Set to 1 to enable.
RA_EXECUTION_POLICY	signed_trusted_or_nexthink	Execution policy of remo Possible values: • unrestricted

		• signed_trusted • signed_trusted_o • disabled
USE_ASSIGNMENT	disable	Disable the rule-based Collec Assignment. Starting from ve the Collector can be dynamic assigned to Engines and enti on a set of configurable rules option to *enable* to activat rule-based assignment i Collector.
ENGAGE	enable_except_on_server_os	Enable the features to engag end-users either on all device those that are not running a s Possible values: • enable_except_o • enable • disable
PROXY_PAC_ADDRESS	-	URL of the PAC file to autom configure the proxy settings
PROXY_ADDRESS	-	The FQDN or IP address of t manual proxy configuration
PROXY_PORT	-	The port number where the p listening in manual proxy con
APP_START_TIME_WHITELIST	chrome.exe,firefox.exe,iexplorer.exe, winword.exe,excel.exe,powerpnt.exe, outlook.exe,onenote.exe,onenoteim.exe, onenotem.exe,lync.exe,zoom.exe	The list of applications whose duration is measured

Windows parameters

Option Name	Default value	Description
ARPNOREMOVE	-	Setting the ARPNOREMOVE property disables the Add or Remove Programs functionality in Control Panel that removes the product. For Windows 2000, this disables the Remove button for the product from the Add or Remove Programs in Control Panel. For earlier operating systems, this has the effect of removing the product from the list of installed products on the Add or Remove Programs in Control Panel.
ARPNOREPAIR	-	Set the ARPNOREPAIR property to disable the Repair button in the Programs Wizard.

ARPSYSTEMCOMPONENT	-	Setting the ARPSYSTEMCOMPONENT property to 1 using the command line or a transform prevents the application from being displayed in the Add or Remove Programs list of Control Panel.
ARPNOMODIFY	1	Setting the ARPNOMODIFY property disables Add or Remove Programs functionality in Control Panel that modifies the product. For Windows 2000, this disables the Modify button for the product in Add or Remove Programs in Control Panel. On earlier operating systems, clicking the Add or Remove Programs button uninstalls the product rather than entering the maintenance mode wizard. **Note**: the Collector MSI package does not support this feature. ARPNOMODIFY must bet set to 1.
REBOOT	-	The REBOOT property suppresses certain prompts for a restart of the system. An administrator typically uses this property with a series of installations to install several products at the same time with only one restart at the end. The ForceReboot and ScheduleReboot actions inform the installer to prompt the user to restart the system. The installer can also determine that a restart is necessary whether there are any ForceReboot or ScheduleReboot actions in the sequence. For example, the installer automatically prompts for a restart if it needs to replace any files in use during the installation. You can suppress certain prompts for restarts by setting the REBOOT property as follows. **REBOOT = Force** Always prompt for a restart at the end of the installation. The UI always prompts the user with an option to restart at the end. If there is no user interface, and this is not a multiple-package installation, the system automatically restarts at the end of the installation. If this is a multiple-package installation, there is no automatic restart of the system and the installer returns ERROR_SUCCESS_REBOOT_REQUIRED.

		REBOOT = Suppress Suppress prompts for a restart at the end of the installation. The installer still prompts the user with an option to restart during the installation whenever it encounters the ForceReboot action. If there is no user interface, the system automatically restarts at each ForceReboot. Restarts at the end of the installation (for example, caused by an attempt to install a file in use) are suppressed. **REBOOT = ReallySuppress** Suppress all restarts and restart prompts initiated by ForceReboot during the installation. Suppress all restarts and restart prompts at the end of the installation. Both the restart prompt and the restart itself are suppressed. For example, restarts at the end of the installation, caused by an attempt to install a file in use, are suppressed.

Starting from V6, the Collector is usually able to upgrade without the need to reboot the device. Only when migrating from V5 or when the target device interferes with the installation process (for instance, by running the Collector Control Panel extension during installation), a reboot is necessary. Set the REBOOT option in these cases to specify your choice.

For instance, if you do not want your devices to reboot right away after a V5 to V6 migration, set REBOOT=ReallySuppress. As a drawback, if you set this option, the upgrade to V6 will not be complete until the end-users reboot their devices.

In unattended execution mode, all choices are silently accepted. For example, if REBOOT=Force, the computer will automatically be rebooted after the MSI package installation.

Casing of properties

Always specify the names of the parameters (the properties) of the MSI in capital letters. If you include the properties with lower case letters in an MST, they will be considered private properties and you will not be able to modify them later from the command-line.

The operations described in this article should only be performed by a Nexthink Engineer or a Nexthink Certified Partner.

If you need help or assistance, please contact your Nexthink Certified Partner.

Nxtcfg - Collector configuration tool

Overview

Nxtcfg is a small console application to read and modify the configuration parameters of the Collector. Ensure that you run Nxtcfg with administrator privileges.

Installation

By default, the Nxtcfg tool is installed along with the Collector when installing the Collector MSI. Once the Collector is installed, the Nxtcfg tool is located under `C:\Windows\System32\nxtcfg.exe`.

The Collector MSI version used determines the nxtcfg version installed (Windows 32-bit or 64-bit system).

If not required, add the option CFG_INSTALL=0 to the MSI command line, when installing the Collector.

Options

Option	Description	Example
/start	Start the Collector.	nxtcfg.exe /start
/stop	Stop the Collector.	nxtcfg.exe /stop
/restart	Restart the Collector.	nxtcfg.exe /restart
/g	Get the value of a particular configuration parameter from the Collector.	nxtcfg.exe /g ip
/s	Set the value of one or more configuration parameters of the Collector.	nxtcfg.exe /s ip=192.168.0.1 udp_port=999
/l	List all the configuration parameters of the Collector with their current values.	nxtcfg.exe /l
/d	Dump all the configuration parameters of the Collector and their corresponding values to a file.	nxtcfg.exe /d C:\temp\collector.cfg

Configuration parameters

The modification of some of the parameters requires to restart the Collector for the change to take effect. Rebooting the device forcefully restarts all Collector components as well. For each parameter, this is specified by the values in the column **Restart required** of the parameters table:

- **No**: No reboot or component restart required.
- **Yes**: Collector restart or device reboot required.

Parameter	Description	Default value	Range	Rest requ
ip	IP address or DNS name of the target Appliance.	-	-	No
udp_port	Port number for UDP communication with the target Appliance.	-	[1 - 65535]	No
tcp_port	Port number for TCP communication with the target Appliance.	-	[1024 - 65535]	Yes
tag	Optional integer number to identify the installation.	0	[0 - 2147483647]	No
string_tag	Optional label to identify the installation.	-	any string (max 2048 chars)	No
cgpi	*CrashGuard Protection Interval Value*. It is the time interval since boot (in minutes) after which a dirty reboot does not increase the CrashGuard.	240 min	-	Yes

logmode	Logging mode • 0 - Silent • 1 - Verbose • 2 - Debug (not recommended for production)	0	[0 - 2]	No
logsize	Maximum size of log file when logging is enabled. Logs are rotated after the maximum is reached.	32 MB	[1 - 512] MB	Yes
printing	Disable or enable print monitoring function.	disable	[disable - enable]	Yes
dsps	Disable (1) or enable (0) SMB print monitoring subfunction (requires **printing** enabled).	1	[0 - 1]	Yes
iops	Enable (1) or disable (0) IOPS monitoring functionality.	0	[0 - 1]	Yes
dwef	When set, the Collector does not report application freezes nor hungs.	0	[0 - 1]	Yes
mss	Maximum size, in bytes, of the UDP packets sent from the Collector to the Engine.	1224 B	[1000 - 16384] B	Yes
pkg_interval	Period, in hours, in which	1	[0 - 24]	Yes

	the Collector checks for new installed packages and updates and reports them: • 0 - Never report packages. • 1 - Report packages after Collector initializes and then every 1 hour. • 2-24 - Report packages 45 minutes after Collector initializes and then every 2 to 24 hours.			
wme	When set, the Collector reports Web and Cloud data.	1	[0 - 1]	Yes
wm_domains	List of domains for which to report the URL of web requests.	-	Comma separated domain names.	No
prefer_ipv6		0	[0 - 1]	No

	When set, the Collector prefers IPv6 to communicate with the Engine when the name of the Engine resolves to both IPv6 and IPv4 addresses.			
custom_shells	When set, enable the Collector to report user logon events and user interactions in virtualized and embedded (kiosk mode) environments.	0	[0 - 1]	No
execution_policy	The security policy to apply when executing scripts of remote actions.	signed_trusted_or_nexthink	• disabled • signed_trusted • signed_trusted_or_nexthink • unrestricted	No
customer_key	The Customer Key of the master Appliance.	-	Path to text file with cryptographic key.	Yes
root_ca	The Root Certificate of the master Appliance.	-	Path to text file with root certificate.	Yes
assignment_status	The status of the assignment of the Collector to a particular Engine.	disabled	• disabled The Collector is not using the Assignment Service. • standby The Collector is waiting the assignment to an Engine. • assigned The Collector is assigned to an Engine.	No (read

use_assignment	Instruct the Collector whether to use or not the Assignment Service.	disable	• disable Instruct Collector not to use the Assignment Service. • enable Instruct Collector to use the Assignment Service.	Yes (c when enabli
engage	Activation of the features to engage with the end-user.	enable_except_on_server_os	• enable_except_on_server_os Activate the features to engage with the end-user, except on devices that run an OS of the server type. • disable Deactivate the features to engage with the end-user. • enable Activate the features to engage with the end-user.	Yes
data_over_tcp	Send Collector data over TCP	disable	• disable Send end-user data over the UDP channel. • enable Send end-user data over the TCP channel.	Yes
proxy_pac_address	URL of the PAC file to	-		Yes

	automatically configure the proxy settings.		• <url> The URL of the PAC file • "" Do not use a PAC file	
proxy_address	FQDN or IP address of the proxy for manual configuration of proxy settings.	-	• <FQDN or IP address> The address of the proxy. • "" Do not use manual proxy settings.	Yes
proxy_port	Port number where the proxy is listening for manual configuration of proxy settings.	-	• <port number> The port number of the proxy. • "" Do not use manual proxy settings.	Yes

Nxtcfg in Remote Actions

Because the Collector communicates with the Engine when running remote actions, it is not recommended to change some of the configuration parameters of the Collector from a remote action. Thus, refrain from modifying any of the following configuration parameters when running `nxtcfg.exe` within a remote action:

- ip
- tcp_port
- customer_key
- root_ca

For the same reason, do not stop or restart the Collector with `nxtcfg.exe` in a remote action. Directly stopping or restarting the Collector abruptly ends the communication between the Collector and the Engine, resulting in the Collector losing its state in respect of the remote action. Once the Collector is up again, it starts the execution of the remote action from the beginning, potentially creating an infinite loop.

Instead, if you need to modify some Collector settings from a remote action that require a restart, use the following code in the script of the remote action:

- To stop the Collector:

```
Stop-Service -Name "Nexthink Service" -Force
Stop-Service -Name "nxtrdrv5" -Force
Stop-Service -Name "nxtrdrv" -Force
```

- To start the Collector:

```
Start-Service -Name "nxtrdrv"
Start-Service -Name "nxtrdrv5"
Start-Service -Name "Nexthink Service"
```

Setting the Customer Key and Root Certificate

The Collector uses the Customer Key and Root Certificate to validate the identity of the slave Appliance (Engine) and securily communicate with it via TLS. If any of these security parameters change in the Appliance (e.g. moving from pre-production to production environment), you must change the configuration in your Collectors accordingly.

The parameters **customer_key** and **root_ca** are special in the sense that they do not admit a direct value as argument, but a path to a text file holding the actual value of the Customer Key or the default Root Certificate, respectively. To download the Customer Key and the default Root Certificate from the master Appliance, follow the same method described for installing the Collector:

1. Log in to the Web Console of the master Appliance as admin.
2. Select the **Appliance** tab at the top of the Web Console.
3. Click **Collector management** in the left-hand side menu.
4. Under **Collector default certificates** at the bottom of the page, click the buttons **DOWNLOAD CUSTOMER KEY** and **DOWNLOAD DEFAULT ROOT CERTIFICATE** to download, respectively, the text files holding the Customer Key and the default Root Certificate of the Appliance.
    - ♦ Only use the default Root Certificate of the master Appliance if you did not replace the certificates for the TCP connection of the Collector with the Appliances.

To set the Customer Key and Root Certificate downloaded from the master Appliance, type in the following (assuming that you placed the downloaded files in the root directory of your C: drive): `nxtcfg.exe /s customer_key="C:\Nexthink-customer-key.txt" root_ca="C:\Nexthink-root-ca.txt"`

When listing the **customer_key** and **root_ca** parameters with the /**l** option of Nxtcfg, neither the full Customer Key nor the full Root Certificate are displayed. Instead, only the first few characters of both the configured key and certificate are shown. These characters are usually enough to identify the key or the certificate, while keeping the list of Nxtcfg parameters readable.

The operations described in this article should only be performed by a Nexthink Engineer or a Nexthink Certified Partner.

If you need help or assistance, please contact your Nexthink Certified Partner.
Related tasks

- Installing the Collector
- Collector proxy support
- Reporting the URL of HTTP web requests

Inspecting the connection status of the Collector

Overview

For the whole Nexthink solution to work properly, it is essential that Collectors can communicate with their target Nexthink Appliances. To easily find out the status of the connection between a Collector and its assigned Nexthink Appliance, the Collector registers all connection attempts in the standard log of the underlying operating system.

- The Windows Collector writes the messages to the Windows Event log.
- The Mac Collector writes the messages to the system log.

Connection messages

When writing log messages to the Windows Event Log, the Windows Collector classifies the messages as **Application** logs and always tags them with the same source and task category:

Source

Nexthink Collector.

Task Category

Connection events.

Find below the complete list of messages and the other tags that the Collectors, both Windows and Mac, can write to the log. The parts of the message in cursive are replaced by actual data:

Message	Level (Windows)	Event ID (Windows)	Process (macOS)	Type (macOS)
Connection to Nexthink Appliance has been established [appliance host: *host_name:port*][resolved appliance IP: *ip_address*]	Information	259	nxtcoordinator	Info
Connection to Nexthink Appliance can not be established: TCP connection failure: *error* [appliance host: *host_name:port*][resolved appliance IP: *ip_address*]	Error	256	nxtcoordinator	Error
Connection to Nexthink Appliance can not be established: Protocol error [appliance host: *host_name:port*][resolved appliance IP: *ip_address*]	Error	256	nxtcoordinator	Error
Connection to Nexthink Appliance can not be established: Websocket error: Certificate issue [appliance host: *host_name:port*][resolved appliance IP: *ip_address*]	Error	256	nxtcoordinator	Error
Connection to Nexthink Appliance can not be established: Websocket error: Host not found [appliance host: *host_name:port*][resolved appliance IP: *ip_address*]	Error	256	nxtcoordinator	Error
Connection to Nexthink Appliance can not be	Error	256	nxtcoordinator	Error

established: Websocket error: *error* [appliance host: *host_name:port*][resolved appliance IP: *ip_address*]				
The Nexthink Appliance URL does not have a valid format [appliance host: *host_name:port*][resolved appliance IP: *ip_address*]	Error	256	nxtcoordinator	Error
Received an error message from Nexthink Appliance: *error_code* error message. Message content: *error_message*	Error	256	nxtcoordinator	Error
Failed to send UDP packet [appliance host: *host_name:port*][resolved appliance IP: *ip_address*]	Error	256	nxtcoordinator	Error
Failed to send data, size data_size ; WSAGetLastError: *error* [appliance host: *host_name:port*][resolved appliance IP: *ip_address*]	Error	256	nxtcoordinator	Error
Failed to create socket with preferIpv6ForHost [ip_address]. Reason: *error* [appliance host: *host_name:port*][resolved appliance IP: *ip_address*]	Warning	257	nxtsvc	Warning

Querying the status of the TCP connection of the Collector

Windows Collector

To query the status of the TCP connection between the Collector and the Appliance for a particular device, run the Collector Configuration Tool on the device with the following option:

```
nxtcfg.exe /g tcp_status
```

Applies to platforms:

The TCP connection of the Collector is typically established between the Collector and the Engine appliance. Starting from V6.19 however, if rule-based Collector assignment is turned on, the TCP connection of the Collector is first established with the Portal appliance. Then, once it receives the coordinates of its assigned Engine, the Collector closes the connection to the Portal and establishes it with the Engine.

The Collector Configuration Tool queries the the Nexthink Coordinator service, which is the component of the Collector that is responsible for the TCP connection.

The following table shows the possible output messages along with their description. The messages that start with a timestamp are retrieved from the Coordinator service. On the other hand, messages that start without a date come directly from the Collector Configuration Tool:

Message	Definition
[ERROR] Nexthink Coordinator service is not installed	The Coordinator is not found in the device. No TCP connection is established.
[ERROR] Nexthink Coordinator service can not be queried	The Coordinator is momentarily not able to respond to the query.
[ERROR] Unknown	There was an unknown error while trying to get the status of the TCP connection from the Coordinator.
[INFO] Nexthink Coordinator service is stopped	The Coordinator service is installed but not running. No TCP connection is established.
[MM/DD/YY hh:mm:ss] [INFO] Initializing connection	The Coordinator is starting and attempting to establish a TCP connection with the Appliance.
[MM/DD/YY hh:mm:ss] [INFO] Connected	The Coordinator established a TCP connection with the Appliance.
[MM/DD/YY hh:mm:ss] [INFO] Disconnected	The Coordinator was disconnected from the Appliance and is actively waiting a new assignment.
[MM/DD/YY hh:mm:ss] [ERROR] Protocol failure	There is a version mismatch between the Collector and the Appliance.
[MM/DD/YY hh:mm:ss] [ERROR] Customer Key issue	There is a mismatch between the Customer Key in the Collector and that of the Appliance.
[MM/DD/YY hh:mm:ss] [ERROR] Certificate issue	There is an issue with the certificate validation. Probable causes:

	• The root CA used to sign the certificate was not deployed with the Collector installer or the root CA was not added to the Windows *Trusted root certification authorities*. • The address of the Appliance that is configured in Collector does not match the address defined in the subject of the certificate.
[MM/DD/YY hh:mm:ss] [ERROR] Host not found	The DNS name or IP address of the Appliance configured in the Collector designates a host that is not found in the network.
[MM/DD/YY hh:mm:ss] [ERROR] TCP connection failure: <number>	Returns an error code from the underlying implementation that indicates the reason for the failure. The most common error is *[ERROR] TCP connection failure: 0*. It indicates that the host exists, but the Collector cannot connect to the indicated TCP port.

Mac Collector

To query the status of the TCP connection between the Mac Collector and the Appliance, open the command-line interface on your macOS device and type in:

```
cat /Library/Application\ Support/Nexthink/config.json | grep -i
tcp-status
```

Applies to platforms:

Related references

- Components of the Collector

Reporting the URL of HTTP web requests

If you have purchased the Web and Cloud product, you may set up the Collector to send the URLs of those HTTP web requests that the end-users address to a selected group of domain names. By default, for every web request, the Collector only reports the domain name inside the request to the Engine (and not the full

URL) to keep the amount of generated network traffic low and avoid flooding the Engine with lots of URLs. Nevertheless, when the Collector is allowed to report the URLs of just a few web requests, the generated traffic still remains reasonably low, while you may benefit from this additional information to define services based on particular URL paths or investigations that include conditions on URLs of web requests.

Learn in this chapter how to specify the list of domain names for which the Collector must report the URLs of the HTTP requests that are addressed to them from the devices of the end-users.

Accepted syntax for the list of domains

Independently of the method chosen to configure the Collector, the accepted syntax for specifying domains is the same. The allowed characters to write domain names are a subset of the ASCII character set that comprises:

- The range of letters from **a** to **z** and from **A** to **Z**.
- The digits from **0** to **9**.
- The symbols **.** (dot) and **-** (hyphen).
- The symbols **:** (colon) and / (slash).
- The symbol ***** (star) to substitute zero or more characters.

Let us see some examples of domain names and how are they interpreted by the Collector:

www.example.com	Matches all HTTP requests addressed to www.example.com
http://www.example.com	Same as above: matches HTTP requests to www.example.com
example.com	Matches all HTTP requests to example.com
http://example.com/index.html	Matches the same as example.com (the URL path after the host name is ignored)
*.example.com	Matches any prefix before the first dot (e.g. www.example.com and ftp.example.com, but not example.com)
*example.com	Matches any prefix (e.g. www.example.com, ftp.example.com, example.com, another-example.com)
***example.com	Same as above (multiple consecutive stars count as one)
ftp.example.com	Matches all HTTP requests addressed to ftp.example.com (Note that the protocol is HTTP and not FTP)
ftp://ftp.example.com	**Error:** only HTTP scheme is allowed
https://example.com	**Error:** only HTTP scheme is allowed

-example.com	**Error:** domain names cannot begin or end with a hyphen
*	**Error:** the *match all* star pattern is not allowed alone

Configuring the list of domains in the Collector

Specify the list of the domains for which the Collector reports the URLs of web requests either before or after deploying the Collector:

- Before deploying the Collector:
  - Passing parameters to the MSI.
  - Using the Nexthink Collector Installer.
- After deploying the Collector:
  - Using the Nexthink Collector Configuration Tool.
  - Changing the value of a registry key.

Beware that if you use the Updater to deploy the Collector, many parameters of the MSI, and the list of domains in particular, cannot be set at installation time and are not saved between updates. For every automatic update of the Collector, you must reapply the settings after deployment.

Passing parameters to the MSI

Specify the list of domain names by setting the value of the parameter **DRV_WEB_AND_CLOUD_HOSTS** when you install the Collector using its MSI file. The value supplied must be a comma separated list of the domains with the syntax defined in the previous section.

This option requires the parameter **DRV_WEB_AND_CLOUD_DATA** to be set to 1 (its default value) for the Collector to gather web related information.

Using the Nexthink Collector Installer

If you use the Nexthink Collector Installer to deploy the Collector, specify the list of domains for which you want to get the full URLs in the **Web And Cloud Settings** dialog that appears when you click the **Settings** button:

In the case that you are updating the Collector, the new settings replace any previously configured list of domains.

Using the Nexthink Collector Configuration Tool

If you have already deployed the Collector, use the Nexthink Collector Configuration Tool to modify the list of domains for which to report full URLs accessed from a particular device. This requires the presence of the Nexthink Collector Configuration Tool in the device; which is installed along with the Collector by default, unless you set the MSI option CFG_INSTALL to 0.

Execute the tool with administrator privileges and specify the list of domains as a parameter in the command line with domains separated by commas:

```
C:\Windows\System32\nxtcfg.exe /s wm_domains="csv_list_of_domains"
```

Setting the value of a registry key

The list of domains for which to report full URLs is saved in the registry under:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\nxtrdrv\params\hosts
```

If you change the value of this variable, the Collector detects its modification and applies the changes accordingly. If an error is detected in the syntax of a domain, the error is logged but the service just skips to the next domain in the list. Under high load, the Collector can miss the modification of the environment variable and you must reboot to force the change. For this reason, this method is recommended only for testing in pre-production environments.

For debugging purposes, it is allowed in this case to use the *match all* star pattern: *****. This is the only exception to the rule and it may help you detect connectivity problems in a particular device.

Technical and security limits

By using any of the described methods, you can specify up to a maximum of 20 domains. The Collector limits the length of a URL to a maximum of 1024 characters. In the rare case of processing a URL longer than 1024 characters, the Collector truncates it to the first 1024 characters.

Note that the feature is only available for HTTP and not for HTTPS web requests. Due to TLS encryption, it is not possible to get the URLs of HTTPS requests. Moreover, reporting the exact URL of an HTTPS request might incur in a security or privacy breach.

In the same sense, the Collector never reports the *query string* part of a URL, that is, the optional list of parameters used by web applications that is placed at the end of the URL after a question mark. Query strings often carry sensitive information such as login names and passwords.

Related tasks

- Creating a service
- Specifying URL paths of web-based services
- Installing the Collector

Related references

- Collector MSI parameters reference table
- Nexthink Collector Configuration tool

Auditing logon events

For Nexthink to report accurate logon times and logon durations, especially in the case that you use roaming user profiles in your Windows setup, configure the audit of logon events in all your devices. You can do so with the help of Active Directory by applying a GPO to the domain of your devices.

Enabling the audit of logon events

To enable the audit of logon events:

1. Open the **Group Policy Management Console**.
2. Right-click the domain node of your devices and select the option **Create a GPO in this domain, and Link it here...**. A dialog to create the new GPO shows up.
3. Type in the name of the GPO. For example, *Logon Audit Policy*.
4. Click **OK** and the new GPO appears in the tree.
5. Right-click the newly created GPO and select the option **Edit...**. The console displays the settings for the GPO.
6. Expand the node **Computer Configuration** and navigate to **Windows Settings** / **Security Settings** / **Local Policies** / **Audit Policy**.
7. Double-click the policy **Audit logon events**.
8. Check the **Success** and, optionally, the **Failure** options.
9. Click **OK** to save your changes.
10. Run the command **gupdate** /**force** to update the GPO.

The devices in the specified domain now record the logon events in the Security log.

Overwriting or clearing events from the Security log

After you activate the audit of logon events, make sure that the Security log of Windows always has enough space to save new logon events. Set the properties of the Security log to perform an appropriate action when the maximum size of the log is reached:

- **Overwrite events as needed (oldest events first)**. *Recommended*.
- **Archive the log when full, do not overwrite events**.
- **Do not overwrite events (Clear logs manually)**.

Use the preferred first option to avoid problems with the size of the Security log.

If you choose the last option and the Security log runs out of space, you may no longer be able to log in to the device. Indeed, if the Security log is full and events are not overwritten, trying to write an audit logon event to the log fails, making the whole login procedure fail as well.

Related references

- Boot and logon duration

Viewing user interactions in virtualized and embedded environments

Overview

Because of the non-standard user logon process in Citrix XenApp and embedded (kiosk mode) Windows, the Collector is neither able to report user logons nor user interactions by using its default detection mechanism when running on these systems.

When installing the Collector in Citrix XenApp or in a Windows device running on kiosk mode, make sure that you set the *custom shells* option. This option tells the Collector to detect user logon events and interactions by means of an alternative mechanism.

To enable this special mode in the Collector, use either the **CUSTOM_SHELLS** MSI parameter during the installation of the Collector or the **custom_shells** option of the Collector Configuration Tool after it has been installed.

If you happen to install the Collector in a Citrix XenApp server, read carefully the following section.

Session termination in Citrix XenApp

Because of a known limitation of Citrix XenApp, in some cases a session may fail to close even after the user has gracefully logged off.

When a user logs in, the Collector spawns the `rundll32.exe` process. To avoid leaving sessions active and waste resources, ensure that Citrix is able to close this process when the user logs off and terminate the session:

1. Log in to the Citrix XenApp server as administrator.
2. Locate the following key in the registry editor:
   `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Citrix\wfshell\TWI`
3. Add the **rundll32.exe** process to the list of processes in the key value **LogoffCheckSysModules**:

Related tasks

- Collector MSI parameters reference table
- Nxtcfg - Collector configuration tool

Engage notifications on macOS

Applies to platforms:

On macOS, the applications that want to inform users about a particular issue send a notification to the standard notification system. In turn, the notification system displays a popup on the screen of the device that transmits the appropriate message to the end user.

From the version 6.28.1.198, the Collector does not rely on the standard notifications anymore. So the system will not prompt the user to choose between allowing or disallowing the delivery of notifications and the campaign will start.

During the campaign creation using the Finder, the system informs the users about the behavior described above when the **Skip notification** box is not ticked.

That information is located at the bottom of the campaign editor.

Related tasks

- Receiving Engage campaigns
- Creating a campaign

Collector remote and Cloud connectivity

Redirecting and anonymizing Collector traffic

Overview

To forward Collector traffic from remote devices, duplicate traffic for redundancy purposes, or anonymize end-user data on the fly, redirect the Collector traffic from the slave Appliance (Engine) to other Engines.

Configure the redirection service `nxredirect` that runs on the slave Appliance to forward (and optionally anonymize) the traffic received from the Collectors to other Engines of your choice.

While the Collector traditionally sends end-user data to the Engine through its UDP channel, starting from V6.21, the Collector may alternatively send end-user data through its TCP channel. The redirection service has been therefore adapted to handle end-user data received via either the UDP or the TCP channels. When sent through the TCP channel though, end-user data is intertwined with data from Engage campaigns, Act remote actions, and software updates. Because the redirection service processes end-user data only, the `nxproxy` component on the Engine Appliance must first receive all the TCP data and split them between end-user data, which will feed the redirection service, and data from Engage, Act, or updates.

To redirect the full TCP channel, with or without optional end-user data, read the article about TCP redirection.

Configuring the redirection service

To configure the redirection service:

1. Log in to the CLI of the slave Appliance.
2. Open or create the configuration file of the redirection service:
   ```
   sudo vi /etc/nexthink/nx_redirect.conf
   ```
3. Press **i** to insert text.
4. Write some redirection rules (see below for examples).
5. Press **Esc** to stop inserting data.
6. Save your changes and exit:
   ```
   :wq
   ```
7. Restart the service:

```
sudo systemctl restart nxredirect
```

8. For the redirection service to automatically start after every system boot:

```
sudo systemctl enable nxredirect
```

Depending on the redirection service receiving TCP or UDP data, the rules to control the redirection have a different syntax. Note however that you can combine the two types of rules if the redirecting Appliance receives Collector data from both channels. The difference in the syntax of the rules and other possible differences in the configuration of the Appliance are detailed below.

Enabling communication through the redirected ports

When redirecting traffic to other (non local) Appliances, the redirection ports are usually not the standard ports of the Appliance, as listed in the connectivity requirements. Therefore, the hardening of the Appliance blocks the communication through the redirected ports by default.

To allow the redirection of Collector traffic from one slave Appliance to another, enable the additional TCP or UDP ports used for redirection from the Web Console of the slave Appliance that receives the redirected traffic.

Redirection of end-user data via UDP

To obtain the maximum performance out of the redirection service, if all the Collectors in a setup send end-user data through the UDP channel, redirect Collector traffic by means of a slave Appliance where the Engine is preferably stopped.

The redirection service can handle the UDP traffic from 5 000 up to 350 000 devices, depending on the available hardware and setup, as anonymization and a running Engine have an impact on the performance.

Stopping the Engine service

To use the Engine appliance exclusively for redirection, stop the Engine service and disable it so it does not restart on every reboot of the appliance:

1. Log in to the CLI of the Engine.
2. Stop the Engine:

```
sudo systemctl stop nxengine@1
```

3. Permanently disable the Engine service:

```
sudo systemctl disable nxengine@1
```

Disabling the Engine removes the memory and CPU consumption of the Engine process from the redirecting appliance.

Writing redirection rules for the UDP channel

The following lines are a sample configuration of the redirection service for the UDP channel:

```
listenraw port=999
[dst=192.168.0.25:997,192.168.0.26:997 send]
```

1. The first line tells the nxredirect service to listen to the traffic received by all interfaces on UDP port 999.
2. The second line sends the received Collector packets to port 997 of the Engines with IP addresses 192.168.0.25 and 192.168.0.26.

Avoiding firewalls blocking redirected traffic from remote Collectors

To distinguish traffic from different Collectors, the redirection rule described above keeps the original IP address of the device that sent the Collector packet. However, when redirecting traffic from remote devices, keeping the original IP address of devices may not be possible.

In a setup for redirecting the Collector traffic of remote devices, the redirection Appliance is placed in a DMZ and the firewall between the DMZ and the corporate intranet usually blocks packets coming from the Internet. All redirected packets from remote devices, on which the IP address of the source is left unchanged, are thus typically blocked by the firewall in the DMZ.

To prevent the firewall from blocking remote traffic, include a **src** parameter in your redirection rules that replaces the original source IP address of devices by the IP address of the redirection Appliance in the DMZ, for example:

```
listenraw port=999
[src=172.16.2.10:999 dst=192.168.0.25:997,192.168.0.26:997 send]
```

In this way, traffic from remote devices passes through the DMZ firewall, at the cost of losing original source IP address information. Starting from V6.24, however, the Collector additionally reports the local IP of the device, which is

more reliable than the source IP address. Indeed, the source IP address of a device can be manipulated in transit by any router doing network address translation (NAT) before reaching the target Engine.

Redirection of end-user data via TCP

If all or some of the Collectors in a setup send their data through the TCP channel only, configure the `nxproxy` component to split the TCP data between Engage, Act, or updates data, which is directly sent to the Engine that runs on the same Appliance, and end-user data, which is sent to the redirection service.

Configuring Nxproxy

To configure `nxproxy` to split the TCP channel and send end-user data to the redirection service:

1. Log in to the CLI of the slave Appliance.
2. To edit the configuration file of `nxproxy`, type in:
   ```
   sudo vi /var/nexthink/nxproxy/conf/nxproxy.conf
   ```
3. Press **i** to insert text.
4. To redirect end-user data to port 11301, instead of the default port 11300 (to which the Engine listens), type in:
   ```
   nxproxy.tcp-collector.engine.grpc-address="localhost:11301"
   ```
5. Press **Esc** to stop inserting data.
6. To save your changes and exit, type in:
   ```
   :wq
   ```
7. Back to the CLI, restart `nxproxy` by typing in:
   ```
   sudo systemctl restart nxproxy
   ```

Writing redirection rules for the TCP channel

The following lines are a sample configuration of the redirection service for the TCP channel. In the forwarding Appliance (*Slave Appliance 1* in the figure), type in:

```
[listen_grpc='localhost:11301'
   [send_grpc=localhost:11300]
   [send_grpc=203.0.113.10:11301]
]
```

1. The first line tells the nxredirect service to listen to the traffic received locally from `nxproxy` on TCP port 11301.
2. The second line sends the received Collector packets to the local Engine.
3. The third line sends the received Collector packets to TCP port 11301 of the slave Appliance with IP address 203.0.113.10.

In the receiving Appliance (*Slave Appliance 2* in the figure):

1. Enable the receiving port TCP 11301 from the Web Console to prevent the firewall that runs on the Appliance from blocking the redirected traffic.
2. Add the following lines to the configuration file of nxredirect to send the data from port 11301 in the public interface back to TCP port 11300, where the local Engine is listening:

```
[listen_grpc='203.0.113.10:11301'
   [send_grpc=localhost:11300]
]
```

Anonymizing redirected traffic

For generic data analysis purposes, you may want to have access to all the data in an Engine related to services, connections, executions, etc. without necessarily associating them to a particular person or group of people. That is, you may want to analyze the data collected while keeping users, devices, and printers anonymous.

To have a redundant Engine that holds all significant data while hiding sensitive information about users, devices, and printers, redirect traffic to that Engine with

anonymization turned on. To anonymize Collector traffic, precede the redirection rule (be it a TCP or a UDP redirection rule) by the **anon** keyword and specify the encryption key of your choice.

For instance, to anonymize the data controlled by an UDP rule:

```
listenraw port=999
[anon=encryption_key dst=192.168.0.27:998 send]
```

And to anonymize the data controlled by a TCP rule:

```
[listen_grpc='localhost:11301'
   [anon=encryption_key send_grpc=localhost:11300]
]
```

When anonymizing Collector traffic, some fields of the device, the user, and the printer objects are encrypted, other fields are randomized, and others are removed.

Fine-grained control over anonymization

By default, anonymization takes an all-or-nothing approach. When anonymization is turned on, the values of all the fields listed in the tables below are actually modified.

In some situations, however, you may be interested in preserving the original values of some of those fields. To exclude a particular field or set of fields from being anonymized, add a list of comma separated exceptions to the **anon** rule in the configuration of the redirection service. For example, to anonymize neither the names of users nor the names of devices in the TCP channel and send the result to the local Engine, type in:

```
[anon=key,noUserName,noDeviceName send_grpc=localhost:11300]
```

For each anonymizable field, find the keyword to turn off its anonymization under the **Exception** column of the tables below.

Device anonymization

Device	Field	Action	Exception
Properties	SID	Randomized	noDeviceSId

	Name	Encrypted	noDeviceName
	AD Site	Removed	noDeviceDS
	Distinguished name reported by Collector	Removed	noDeviceDN
Network	Last IP address	Replaced by Engine IP	noDeviceIP
	IP addresses	Replaced by Engine IP	noDeviceIP
	MAC	Randomized	noDeviceMacs
	Group name	Encrypted	noDeviceGroupName
Operating system	Local Administrators (groups)	Encrypted	noUserGroupName
	Local Power Users (groups)	Encrypted	
	Windows license key	Removed	noWindowsKey
Hardware	BIOS serial number	Removed	noBiosSN
	Chassis serial number	Removed	noChassisSN
	Device product version	Removed	noProductSN
	Device UUID	Removed	noUuidSN
Active Directory	Distinguished name	Not retrieved	-

Note that a change in the encryption key implies a duplication of the devices. If you are redirecting to an existing Engine, remember to erase the database to avoid duplications.

User anonymization

User	Field	Action	Exception
Properties	SID	Randomized	noUserSId
	Name	Encrypted	noUserName
Active Directory	Distinguished name	Not retrieved	-
	Full name	Not retrieved	-
	Department	Not retrieved	-
	Job title	Not retrieved	-

Printer anonymization

Printer	Field	Action	Exception
Properties	Name	Encrypted	noPrinterName

Print Job anonymization

Print Job	Field	Action	Exception
Properties	Document name	Removed	noPrintJobDocName
User	User name	Removed	noPrintJobUserName

Note that the document name of a Print Job is not stored in the Engine and, therefore, not visible in the Finder. Nevertheless, the Collector sends the document name through the network and it is stored when Collector traffic is captured. Hence the possibility to anonymize it.

Combining anonymization with rule-based assignment of Collectors

Combining anonymized Collector traffic with rule-based Collector assignment is only possible if Collectors are configured to send traffic via TCP only.

For rule-based assignment to work, prevent the anonymization of device fields that are present in the assignment rules. Apply the fine-grained control technique over anonymization described above to any of these fields:

- Last IP address?
- Name?
- Distinguished name reported by Collector?
- AD site?

Related tasks

- Redirecting the Collector TCP channel
- Getting feedback from your end-users
- Scenarios for remote actions
- Updating the Collector
- Assigning Collectors to Engines

Related references

- Connectivity requirements
- Hardware requirements
- Reference architectures
- Appliance Hardening

Redirecting the Collector TCP channel

Overview

Because Nexthink Appliances installed on premise are usually not accessible through the Internet, only the Collectors running on devices inside the corporate network or with VPN access can reach their assigned Appliance. Thus, the Collectors of roaming devices without VPN access cannot directly communicate with their Appliance.

To forward Collector traffic from roaming devices to their assigned Appliances, run a redirection Appliance on a DMZ that faces both the Internet and your corporate network, so that the Appliance is reachable by the roaming devices while able to deliver traffic to your internal Appliances at the same time:

- If all Collectors are configured to send end-user data through the TCP channel, run a reverse proxy that redirects all the TCP traffic.
- If some Collectors send end-user data through the UDP channel, run the redirection service to forward the UDP traffic and a reverse proxy to redirect the TCP traffic.

Note that, although the redirection service is able to forward TCP traffic since V6.21, the service only handles the end-user data portion of the TCP channel (not Engage, Act, assignment, or updates data) for redundancy or anonymization purposes. Thus, avoid the redirection service to forward TCP traffic from the Collectors of roaming devices. Instead, run a separate reverse proxy on the Appliance for the TCP channel. The Engine Appliance already includes a version of the popular web server Nginx. Find in this article how to configure Nginx as reverse proxy for the TCP channel of the Collector.

As a prerequisite, remember that roaming devices will only be able to reach their assigned Engine both inside and outside the corporate network if the name of the Engine configured in the Collector is valid as well for the redirection Appliance when the device is on the Internet. Please contact Customer Success Services to get guidance on how to configure the DNS service for that purpose.

Running Nginx on the Engine Appliance

To redirect the TCP channel of the Collector, we assume that you already have an Engine Appliance up and running where the Engine service has been stopped to exclusively use the Appliance for redirecting Collector traffic. Remember that you only need to run the redirection service in addition to Nginx on the redirection

Appliance if any of the roaming Collectors sends end-user data through UDP.

Ensure that `nginx` is running on your redirection Appliance by logging in to the CLI of the Appliance and running the following command:

```
sudo systemctl status nginx
```

If the service is not running, first enable it with the following command to make `nginx` start on every boot of the Appliance:

```
sudo systemctl enable nginx
```

Start the service after properly configuring it, as shown in the instructions below.

Transfer TLS certificates and private key to the redirection Appliance

To secure the connection between the Collectors and `ngninx`, copy the certificates and private key that protect the TCP channel of the internal Appliances to the redirection Appliance (we name the certificate and key files in the same way as in the article on replacing certificates).

To transfer the TLS certificates and private key of an Engine:

1. Copy the certificate and key files to the redirection Appliance:
    - If you are using a custom certificate to protect the TCP connection between Collectors and the Engine:
        1. Copy the server certificate (`slave.crt`), the intermediate certificate or chain of certificates (`intermediate.crt`), if any, and the private key (`slave.key`), all in PEM format, to the `nexthink` account of the redirection Appliance using your favorite SCP tool.
        2. Log in to the CLI of the redirection Appliance.
        3. Concatenate the server and intermediate certificates. If you do not have an intermediate certificate, the command will give you a warning that you can safely ignore:
            ```
            cat slave.crt intermediate.crt >
            slave_bundle.crt
            ```
    - If you are using the default certificates generated by the Appliance:
        1. Log in to the CLI of the Engine.
        2. Get the keystore in the Engine that holds the certificates and key to protect the TCP channel.
            ```
            keytool -importkeystore \
            -srckeystore
            /var/nexthink/nxproxy/keystore/keystore.jks \
            ```

```
-destkeystore certs_key.p12 -deststoretype
PKCS12
```
When prompted for the destination keystore password and for the source keystore password, always type in **nexthink**.

3. Get the bundle of certificate files and the private key from the keystore. Provide **nexthink** as password to each command:
```
openssl pkcs12 -in certs_key.p12 -nokeys -out
slave_bundle.crt
openssl pkcs12 -in certs_key.p12 -nocerts
-nodes -out slave.key
```
4. Clean the certificate and key files off additional attributes:
```
sed -ni -e '/-BEGIN CERTIFICATE-/,/-END
CERTIFICATE-/p' slave_bundle.crt
sed -ni -e '/-BEGIN PRIVATE KEY-/,/-END PRIVATE
KEY-/p' slave.key
```
5. Copy the certificate and key files to the redirection Appliance using your favorite SCP tool.
6. Log in to the CLI of the redirection Appliance.

2. Change the permissions of the certificate and key files:
```
sudo chmod 444 slave_bundle.crt
sudo chmod 400 slave.key
```
3. Move the certificate and key files to a standard location:
```
sudo mv slave_bundle.crt /etc/pki/tls/certs
sudo mv slave.key /etc/pki/tls/private
```

If you are using automatic Collector assignment, repeat the same steps described above to transfer the TLS certificates of the Portal to the redirection Appliance. To that end, log in to the CLI of the Portal instead of the CLI of the Engine when indicated and replace:

- `slave.crt` by `master.crt`
- `slave_bundle.crt` by `master_bundle.crt`
- `slave.key` by `master.key`

Read the considerations on the automatic assignment of roaming Collectors.

Depending on how you configured your DNS for Collectors to reach both their assigned Engine (or the Portal in the case of automatic Collector assignment) and the redirection Appliance, your certificates may require to be issued to multiple subjects (i.e. multiple domain names). For default certificates, multiple subjects can be specified when setting the external names of the Engine or when setting the external names of the Portal. In the case of custom certificates that specify multiple subjects, a single server certificate may be used to protect all of the Engines and the Portal at the same time: `master.crt` and `slave.crt`

would be the same certificate in this case. Please contact Customer Success Services if you need guidance on DNS configuration.

Configuring Nginx as WebSocket reverse proxy

The TCP channel of the Collector actually uses WebSocket as its application level protocol. To configure `nginx` as a reverse proxy for the WebSocket protocol and thus redirect TCP Collector traffic, the Appliance includes an additional folder to store Nexthink-related configuration:

1. Log in to the CLI of the Appliance.
2. Create an additional configuration file for `nginx` in the folder reserved by the Appliance. The name is not important as long as the extension is `.conf`:
   ```
   sudo vi /var/nexthink/nxnginx/conf.d/nxcollector-ws.conf
   ```
3. Press **i** and insert the content detailed below.
4. Press **Esc** to stop inserting text.
5. Type in the sequence **:wq** and press **Enter** to save the changes and exit the editor.
6. Stop and disable the proxy service in the Engine to avoid interference with the redirection of the TCP channel:
   ```
   sudo systemctl stop nxproxy
   sudo systemctl disable nxproxy
   ```
7. Restart `nginx`:
   ```
   sudo systemctl restart nginx
   ```

Add a **server** directive to the configuration file for each Appliance that needs redirection. In the example below, the first server entry configures the redirection for the Portal (for Collector assignment), while the second entry configures an Engine:

```
map $http_upgrade $connection_upgrade {
    default upgrade;
    ' '      close;
}

server {
    listen {Redirection_Appliance_IP}:443 ssl;
    server_name {Portal_DNS_FQDN};

    ssl_certificate /etc/pki/tls/certs/master_bundle.crt;
    ssl_certificate_key /etc/pki/tls/private/master.key;

    location / {
        proxy_pass https://{Portal_Private_IP}:443;
        proxy_http_version 1.1;
```

```
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection $connection_upgrade;
        proxy_read_timeout 1w;
    }
}

server {
    listen {Redirection_Appliance_IP}:443 ssl;
    server_name {Engine_DNS_FQDN};

    ssl_certificate /etc/pki/tls/certs/slave_bundle.crt;
    ssl_certificate_key /etc/pki/tls/private/slave.key;

    location / {
        proxy_pass https://{Engine_Private_IP}:443;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection $connection_upgrade;
        proxy_read_timeout 1w;
    }
}
```

Substitute the following keywords for actual values:

{Redirection_Appliance_IP}

The IP address of the redirection Appliance as seen from the Internet, if the Appliance is directly facing the Internet. Alternatively, if the redirection Appliance is inside a DMZ subnetwork managed by a router that does the address translation (NAT), this is the IP address of the redirection Appliance inside the DMZ.
For example: `listen 203.0.113.10:443 ssl;`

{Portal_DNS_FQDN}

The Fully Qualified Domain Name of the Portal that returns the IP of the redirection Appliance when resolved by the Internet.
For example: `server_name portal.example.com;`

{Engine_DNS_FQDN}

The Fully Qualified Domain Name of the Engine that returns the IP of the redirection Appliance when resolved by the Internet.
For example: `server_name engine1.example.com;`

{Portal_Private_IP}

The IP address of the Portal as seen from inside the corporate network.
For example: `proxy_pass https://192.168.0.100:443;`

{Engine_Private_IP}

The IP address of the Engine as seen from inside the corporate network. For example: `proxy_pass https://192.168.0.99:443;`

The Internet names of the Portal and the Engine (respectively, **Portal_DNS_FQDN** and **Engine_DNS_FQDN**) must both resolve to the same IP address when resolved by a roaming device: the address of the redirection Appliance. The names themselves must be different for the reverse proxy to forward the traffic to the proper destination.

The example illustrates the use of the default TCP port 443 for the communication of the Collector with the Engine and the Portal. If you configured the Collectors and the Appliances to communicate through a different TCP port, substitute as well the default port number 443 for your own (the default custom port number is TCP 8443).

For each additional Engine that needs redirection, transfer the TLS certificates and add a new server entry to the configuration file, as described above.

Related tasks

- Installing the Appliance
- Redirecting collector traffic
- Assigning Collectors to Engines
- Importing and replacing certificates
- Setting the names of the Engines
- Setting the names of the Portal
- Changing the default ports in the Appliance

Related references

- Reference Architectures
- Nginx (external)

Support for DirectAccess

Overview

Microsoft DirectAccess is a technology that provides remote connectivity to devices equipped with Windows 7 and higher operating systems. Similar in concept to a traditional virtual private network (VPN), DirectAccess allows users to securely access network resources inside the intranet of their organization

when connected to the Internet. Unlike traditional VPN connections, which usually require explicit user action to be initiated and terminated, DirectAccess is transparent to the end user and automatically connects to the intranet of the company when needed.

DirectAccess relies on clients and applications that support the IPv6 stack. It encapsulates the traffic to route it through the Internet and, once it reaches the intranet, a companion technology transforms the IPv6 addresses into IPv4 if needed; that is, if the intranet uses IPv4 internally, which is usually the case.

Impact on Nexthink

Since DirectAccess requires client applications to use IPv6, three Nexthink products are impacted when a set of devices in your organization connect to the corporate network via DirectAccess: the Collector, the Engine, and the Finder.

Collector

The Collector must be able to send information to the Engine from devices that connect to the intranet of their organization through DirectAccess. Therefore, the Collector must use IPv6 to send its information. In addition, the Collector must be able to capture network information of those applications running on devices connected through DirectAccess, which also use the IPv6 stack.

When installing the Collector in a DirectAccess environment, check the option **Prefer IPv6** when running the Collector installer, or the MSI parameter **DRV_PREFERIPV6**, for the Collector to use IPv6 rather than IPv4 to send information. You can equally modify the value of this setting when the Collector is already installed with the help of the Collector configuration tool by adjusting the value of the parameter **prefer_ipv6**.

Engine

The Engine must be able to detect Collector traffic coming from DirectAccess and translate the received IPv6 addresses to their IPv4 counterparts within the intranet. To identify Collector traffic, the Engine needs to know the IPv6 subnetwork used by DirectAccess.

By default, the Engine identifies and translates IPv6 addresses in the subnet fda9:11e5:84fa::/48. If you use a different subnetwork, configure the Engine as in the following example, substituting the DirectAccess prefix given for your own:

1. Stop the Engine

```
sudo systemctl stop nxengine@1
```

2. Configure the IPv6 subnet:

```
sudo nxinfo config -s
"direct_access.prefix=fda9:11e5:84fa::/48"
```

3. Restart the Engine

```
sudo systemctl start nxengine@1
```

Finder

The Finder must be able to connect to both the Portal and the Engine even when run from a device connected to the corporate network via DirectAccess. In the case of the Finder, no additional configuration is needed, but you must use DNS names in the login dialog to resolve the address of the Portal, because the dialog does not support IPv6 addresses.

Windows Collector proxy support

The following content applies exclusively to the Nexthink Cloud offering.

Overview

The devices in a corporate network typically connect to the Internet through a proxy server instead of using a direct connection. A *proxy server* or *proxy* forwards the requests of client applications that run on the corporate devices to the servers that run on the Internet, as if the proxy itself initiated the requests. Then the proxy sends the responses from the servers back to the clients. By acting as intermediary, a proxy server can provide varied functionality such as content filtering (for improved security) or content caching (for better performance).

Thus, in a Nexthink Cloud setup, Collectors inside a corporate network that is equipped with a proxy server are usually required to send their traffic through the proxy to reach the Nexthink Cloud. In this article, learn about the different types

of proxies and configurations supported by the Collector.

Applies to platforms:

Supported types of proxies

The Collector supports the following types of proxies:

- **HTTP** (web) proxy
- **SOCKS5** proxy

Collector should work out of the box with *transparent proxies* as well. Transparent proxies automatically intercept network traffic that goes from the corporate network to the Internet, so that clients are not aware that their traffic is traversing a proxy.

Authentication

The Windows Collector supports Integrated Windows Authentication (IWA).

Because the Collector runs as a service under the local system, if a configured proxy requires NTLM authentication, the Collector responds to the challenge with the computer's identity.

Supported proxy configurations

Proxy settings may appear at different levels:

System level
: Settings apply to all users and applications on the device.

User level
: Settings apply to all applications that a user runs.

Application level
: Settings apply only to the application itself.

Because the Collector runs as a Windows service, it can read the proxy settings specified both at the application level (its own custom configuration) and at the system level, but not at the user level. Therefore, the Collector supports the methods described below to configure its proxy settings.

Microsoft Windows HTTP Services (WinHTTP)

The WinHTTP interface is meant to be used by server applications and system services such as the Collector. It provides proxy settings at the system level and

its configuration is stored in the Windows Registry:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet
Settings\Connections
Value: WinHttpSettings
Type: Binary
```

To read the proxy settings on a device, type in from the comand prompt:

```
netsh winhttp show proxy
```

To configure the proxy settings on a device, use the **set proxy** command. Type in the following to display usage and options:

```
netsh winhttp set proxy /?
```

The Windows Collector V6.24 already supported WinHTTP services to get the proxy settings. If you are migrating from V6.24, proxy settings will remain unchanged on V6.25 and later.

Starting from version 6.27.2, the Collector takes into account the bypass list set via WinHTTP or Internet Explorer Options in system context. As the Collector is a system component, the user-specific settings do not have any impact on it.

Some important remarks regarding the syntax used for the bypass list:

- The bypass entry for the Nexthink server must match the value you used during the Collector configuration. If you used the DNS name, then you must use it in the bypass list. Same goes for the IP address, if you used it for the Collector configuration then you must use it in the bypass list.
- As stated in the Microsoft documentation, the bypass list can contain wildcards. Make sure to remember that something like **.nexthink.com* is correct but *nexthink.** is not supported. The Nexthink Collector supports as well the following kind of entries *192.168.212** or *192.168.** or *192.** but other products installed on the system might not.

Microsoft Windows Internet (WinINet) API

The WinINet API was designed to give interactive desktop applications access to standard Internet protocols such as HTTP or FTP. Applications such as Internet Explorer get their proxy configuration via WinINet. This configuration is visible from the Internet Properties dialog of the Control Panel:

1. Press the **WinKey**.
2. Type in **Internet Options** and press **Enter**. The **Internet Properties** dialog shows up.
3. Select the **Connections** tab.
4. Under the section **Local Area Network (LAN) settings**, click the button **LAN settings**

5. Choose how WinINet should configure the LAN settings (which include the proxy settings), tick either:
    - **Automatically detect settings**, to use Web Proxy Auto-Discovery (WPAD) protocol.
    - **Use automatic configuration script**, to get a PAC file from the specified URL in **Address**. A *proxy-auto-config (PAC)* file is a JavaScript file with a single function that determines which proxy should be used for each client connection.
    - **Use a proxy server for your LAN**..., to manually configure the proxy settings.

By default, WinINet provides proxy settings at the user level; therefore, the Collector cannot read them. To make them readable by the Collector, promote the WinINet proxy settings to system level by setting the **ProxySettingsPerUser** value to 0 in the Windows Registry:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows\CurrentVersion\Internet
Settings
Value: ProxySettingsPerUser
Type: Binary
Data: 0
```

Or by setting the following GPO:

```
Computer Configuration\Administrative Templates\Windows
Components\Internet Explorer\
Make proxy settings per-machine (rather than per user)
```

Web Proxy Auto-Discovery (WPAD)

The *Web Proxy Auto-Discovery (WPAD)* protocol is a method to set the proxy settings automatically by leveraging the DHCP and DNS protocols. WPAD uses discovery methods in DHCP and DNS to find out the URL of a PAC file, in much the same way as WinINet gets its LAN settings when automatic detection is enabled.

Collector custom configuration

Specify the proxy settings of the Collector at the application level during installation, with the Collector Installer or with the MSI parameters, or later with the Collector configuration tool.

There are basically two options for custom configuration:

- **Automatic proxy**: Get a PAC file from the specified URL to automatically determine the proxy settings. A *proxy-auto-config (PAC)* file is a JavaScript file with a single function that determines which proxy should be used for each client connection.
- **Manual proxy**: Provide manually the address and port of the proxy.

Collector logic to choose proxy settings

The Windows Collector selects one of the methods to get its proxy settings on a trial and error basis. The Collector tests the validity of each method in sequence: the first method that yields a set of proxy settings which let the Collector connect to the Nexthink Cloud is retained. Methods are tried in the following order until one of them is successful:

1. Collector custom configuration.
    1. If enabled, automatic proxy configuration through PAC file.
    2. If enabled, manual proxy configuration.
2. Direct connection without proxy.
3. Microsoft Windows HTTP Services (WinHTTP).
4. Microsoft Windows Internet (WinINet) API.
    - ♦ Only if settings are valid at the system level (per device).
        1. Manual configuration.
        2. PAC file referenced by URL (automatic configuration script).
5. Web Proxy Auto-Discovery (WPAD).

Mac Collector proxy support

The following content applies exclusively to the Nexthink Cloud offering.

Overview

The devices in a corporate network typically connect to the Internet through a proxy server instead of using a direct connection. A *proxy server* or *proxy* forwards the requests of client applications that run on the corporate devices to the servers that run on the Internet, as if the proxy itself initiated the requests. Then the proxy sends the responses from the servers back to the clients. By acting as intermediary, a proxy server can provide varied functionality such as content filtering (for improved security) or content caching (for better performance).

Thus, in a Nexthink Cloud setup, Collectors inside a corporate network that is equipped with a proxy server are usually required to send their traffic through the proxy to reach the Nexthink Cloud. In this article, learn about the different types of proxies and configurations supported by the Collector.

Applies to platforms:

Supported types of proxies

The Collector supports the following types of proxies:

- **HTTP** (web) proxy
- **SOCKS5** proxy

Collector should work out of the box with *transparent proxies* as well. Transparent proxies automatically intercept network traffic that goes from the corporate network to the Internet, so that clients are not aware that their traffic is traversing a proxy.

Supported proxy configurations

The Mac Collector supports the configuration of proxy settings at the application level; that is, it does not get the proxy settings from the system, but from its own custom configuration. Provide the proxy settings to the Mac Collector during installation.

Collector custom configuration

To specify the proxy settings of the Collector at the application level during installation, select either one or both of the following methods:

- **Automatic proxy**: Get a PAC file from the specified URL to automatically determine the proxy settings. A *proxy-auto-config (PAC)* file is a JavaScript file with a single function that determines which proxy should be used for each client connection.
- **Manual proxy**: Provide manually the address and port of the proxy.

Collector logic to choose proxy settings

The Mac Collector selects one of the methods to get its proxy settings on a trial and error basis. The Collector tests the validity of each method in sequence: the first method that yields a set of proxy settings which let the Collector connect to the Nexthink Cloud is retained. Methods are tried in the following order until one of them is successful:

1. If selected, try automatic configuration.
    - ♦ Retrieve the PAC file, execute and determine the proxy.
2. If selected, try manual configuration.
3. Direct connection without proxy.

Related tasks

- Installing the Collector on macOS

Installing the Data Enricher

Installing the Data Enricher

General configuration file

Sample general configuration file for the Data Enricher: `general.conf`

```
[GENERAL]
# Path and name for the log file.
# Default: /ProgramData/Nexthink/nxdataenricher/log/nxdataenricher.log
log_file = /ProgramData/Nexthink/nxdataenricher/log/nxdataenricher.log

# Setup the log level desired. For production environments is strongly
recommended to setup "info" value.
# Options available:
# - CRITICAL
# - ERROR
# - WARNING
# - INFO
# - DEBUG
# More information about log levels here
https://docs.python.org/3.7/library/logging.html#levels
# Default: INFO
log_level = INFO

# Set logging format.
# Default: %(asctime)s %(process)s %(levelname)s
%(filename)s:%(lineno)d [-] %(message)s
log_format = %(asctime)s %(process)s %(levelname)s
%(filename)s:%(lineno)d [-] %(message)s

[NEXTHINKCLOUD]
# Please refer to
https://doc.nexthink.com/Documentation/Nexthink/latest/InstallationAndConfiguration/Ins
# in order to obtain more information about the Data Enricher
installation process
# and/or this file in particular.

# Nexthink Cloud endpoint to send and retrieve data from Nexthink.
# Example: endpoint = https://agora.[region].nexthink.cloud
endpoint = https://<host>

# The client ID will be used alongside the client secret to retrieve a
token.
oauth_client_id = <oauth_client_id>
```

```
# The client secret will be used alongside the client ID to retrieve a
token.
oauth_client_secret = <oauth_client_secret>

# Proxy configuration.
proxy_enabled = False
# Example: proxy_server = http://proxy.nexthink.com:3128
proxy_server = <schema>://<host>:<port>
# Only Basic Authentication or no authentication are supported. Options
available:
# - None
# - Basic
proxy_auth_type = None
proxy_user =
proxy_password =

# Allows service to check for self signed certificate in Nexthink Cloud.
# False means the self-signed certificate or a bad certificate will be
not checked.
# True means the self-signed certificate or a bad certificate will be
checked.
# Default: True
verify_cert = True

# Maximum number of entries to be updated in Nexthink in a single
request.
update_batch_size = 10000
```

AD configuration file

Sample configuration file for the Data Enricher: `enricher_nxad.conf`

```
[NXAD]
partial_refresh_enabled = True
partial_refresh_frequency = 60

# Enables/disables the full refresh functionality.
# When enabled this will refresh all engine users info at the time
configured in full_refresh_time parameter.
# Attention! This can be a very heavy load job, please configure the
execution time accordingly.
full_refresh_enabled = False

# You may configure a refresh on a daily or weekly basis.
# For a daily full refresh, enter only the local time in a 24 hour
format, for example 23:30.
# For a weekly full refresh, add the desired weekday before the time,
for example: Sat 23:30.
# The possible values are: Mon, Tue, Wed, Thu, Fri, Sat and Sun.
```

```
# Examples:
#   Daily full refresh at 3:17 AM
#   full_refresh_time = 3:17
#   Weekly full refresh on Fridays at 4:55 PM
#   full_refresh_time = Fri 16:55
full_refresh_time = Mon 23:30

# Comma separated list of ActiveDirectory attributes to be excluded.
# Options available (corresponding Nexthink Finder's name in
parenthesis):
# - distinguishedName (Distinguished name)
# - sAMAccountName (Name)
# - title (Job title)
# - department (Department)
# - displayName (Full name)
# - l (Locality name)
# - c (Country code)
# - ou (Organizational unit name)
# - physicalDeliveryOfficeName (Location)
# Example: excluded_attributes = title, department
excluded_attributes =

# Size of the search batches in AD.
search_batch_size = 1000

# Servers configuration:
# Use format "server_<x>.<property> = <value>", replace <x> for a
number and <property> by its name,
# and <value> by the desired one, like in the following example. You can
create as many servers as required.
#
# server_1.name: string (The generic name for the server. Example: if
set to "nexthink.ch",
  usernames in Finder will be shown as user@nexthink.ch.)
# server_1.address: string (Server IPv4 or FQDN address)
# server_1.port: int (Server port, usually 389 used)
# server_1.bind_dn: string (Bind DN account. Example:
"CN=user,CN=users,DC=company,DC=local")
# server_1.bind_password: string (Bind DN account password)
# server_1.base_dn: string (Start point for directory searches.
Example: "DC=company,DC=local")
# server_1.scope: string (One of the following values: base, onelevel or
subtree)

server_ad1.name = <name>
server_ad1.address = <ip_or_fqdn>
server_ad1.port = <port>
server_ad1.use_ssl = <True/False>
server_ad1.bind_dn = <CN=user,CN=Users,DC=company,DC=local>
server_ad1.bind_password = <pass>
server_ad1.base_dn = <DC=company,DC=local>
server_ad1.scope = <base/onelevel/subtree>
```

DNS configuration file

Sample configuration file for the Data Enricher: `enricher_nxad.conf`

```
[NXDNS]
partial_refresh_enabled = True
partial_refresh_frequency = 60

# Enables/disables the full refresh functionality.
# When enabled this will refresh all engine destinations info at the
time configured in full_refresh_time parameter.
# Attention! This can be a very heavy load job, please configure the
execution time accordingly.
full_refresh_enabled = False

# You may configure a refresh on a daily or weekly basis.
# For a daily full refresh, enter only the local time in a 24 hour
format, for example 23:30.
# For a weekly full refresh, add the desired weekday before the time,
for example: Sat 23:30.
# The possible values are: Mon, Tue, Wed, Thu, Fri, Sat and Sun.
# Examples:
#   Daily full refresh at 3:17 AM
#   full_refresh_time = 3:17
#   Weekly full refresh on Fridays at 4:55 PM
#   full_refresh_time = Fri 16:55
full_refresh_time = Mon 23:30

# Comma separated list of DNS servers.
# Example: servers = 8.8.8.8, 8.8.4.4
servers = <server1>,<server2>
# Timeout for each DNS server in seconds
max_dns_server_timeout = 0.5
# Percentage of allowed DNS server errors, measured with respect to the
total number of destinations
max_perc_dns_server_errors = 35
```

Installing the Event Connector

Installing the Event Connector

Overview

The purpose of the Nexthink Event Connector is to send Nexthink real-time analytics in a third-party application like ServiceNow, Splunk or Azure Data Lake Gen2 for its consumption in a highly configurable way.

Software Requirements

Supported operating systems

A separated CentOS 7 host machine is required for the deployment.
A Nexthink Appliance is recommended

Nexthink dependencies

Nexthink Engines and Portal 6.20 or later.
Integration license.

RPM dependencies

During the installation, the event connector rpm requires the appliance to be connected to the Internet to download and install the dependencies from the official repositories of CentOS.

Documentation and installation file

Installation guide, troubleshooting guide, and RPM are available on Nexthink Community

Related references

- Nexthink Event Connector documentation on Community

Installing the Finder

Installing the Finder

Installing the Finder from the Portal

The recommended way for a single user to install the Finder is to download its installer from the Portal and execute it. Users must have been allowed access to the Finder in their profile to follow this procedure. The procedure installs the Finder in a per-user context, meaning that:

- Standard (non-administrator) Windows users can perform the installation.
- The Finder is only available to the user that installed it and not to other users of the same machine.
- The Finder is able to subsequently perform automatic updates when needed and to simplify mandatory upgrades.

To install the Finder from the Portal:

1. Open a web browser and log in to the Portal from a machine that runs an operating system supported by the Finder.
2. Click your username in the top right corner to display your user options.
3. Select **Install Nexthink Finder** from the drop-down menu. Note that this option is only available for those users who have been granted access to the Finder in their profile. A dialog shows up.
4. Choose the version of the Finder that you want to download. Depending on the architecture of your machine, choose between:
    - ♦ 64-bit version (recommended): if you have a machine that runs a a 64-bit version of Windows (x64 architecture).
    - ♦ 32-bit version: if you have a machine that runs a 32-bit version of Windows (x86 architecture) or if your computer has less than 4 GB of RAM.
5. Click **Download**.
6. Once the installer has finished downloading, run it and allow the program to make changes to your computer (the procedure to run the installer may be slightly different depending on your web browser).
    - ♦ Right after installation, if the installer finds out that an older per-machine version of the Finder was present in the machine, it launches the uninstallation program:
        - ◊ If the user has or is able to obtain administrator privileges, the per-machine version of the Finder is uninstalled after the

user authorizes the program to execute.
    - ◊ If the user does not have administrator privileges or skips the step, the two versions of the Finder will coexist in the machine.
7. After finishing the installation, the Finder opens automatically.

Remember to set the correct Portal address for the Finder to open the right session after installation.

Installing the Finder from Product Downloads (not recommended)

In Product Downloads, you can find alternatives to the installation from the Portal. Use these alternatives only when you have a particular well-founded reason for it (e.g. installation on Citrix environments), as these downloads do not provide automatic updates and, therefore, they may be more difficult to maintain. These alternatives use a per-machine installer.

Running the installer in a per-machine context means that:

- Administrator privileges are required to perform the installation.
- The Finder is installed for all users of the machine.

To manually install the Finder from Product Downloads:

1. In the **Product Downloads** page, select the first entry of the **Last V6 releases**.
2. In the **Download links** section, find the links for the Finder.
3. Choose one set of downloads for the Finder, depending on the architecture and memory of your computer:
    - ♦ Per-machine installer:
        - ◊ 64-bit version EXE: recommended for computers with more than 4 GB of RAM running a 64-bit version of Windows.
        - ◊ 32-bit version EXE: suitable for 32-bit versions of Windows or for computers with less than 4 GB of RAM (even when running a 64-bit version of Windows).
4. Run the per-machine installer by double clicking in the downloaded file.

In its turn, the stand-alone executable version of the Finder is no longer proposed as a download. Automatic updates and easy upgrades offer a superior solution to the problem of frequently connecting to different versions of the Portal.

Silent installation

To install the Finder without user interaction, run the installer with the silent option. For instance:

```
finder-setup-x64-machine.exe -silent
```

Personalized installation

To automatically open the correct session after installation, emulating thus the behavior of the installation from the Portal, modify the name of the installer executable file before running it on the target device.

Provide the name of the user encoded in Base64 format and the address of the Portal as configured for sending the email digests. For example, rename the installer as:

```
finder-setup-[YWRtaW4=@portal.aonnetworks.com].exe
```

Where:

- **YWRtaW4=** is the Base64 encoding of the user name *admin*.
- **portal.aonnetworks.com** is the address of the Portal.
- Note that they are separated by the @ sign.

In case that you have the Windows authentication of users enabled in your setup, provide only the address of the Portal:

```
finder-setup-[portal.aonnetworks.com].exe
```

When executed, and after installation, the Finder authenticates the current user by their Windows credentials. If no corresponding session is available, which is usually the case if the Finder is installed for the first time, the Finder asks the user to create one session that uses Windows authentication.

NET Framework requirement

Modern versions of Windows receive the latest version of the .NET Framework through Microsoft Updates. The .NET Framework 4.6 or higher is required to install and run the Finder.

In a non-updated Windows device, either update your operating system via Windows Updates (recommended), or download and install first the .NET

Framework 4.6 (or higher) from Microsoft.

Security certificates

Upon the first execution of the Finder, you may experience some warnings related to security certificates. Certificates ensure that the communication among Nexthink components is safe. Refer to the sections about logging in to the Finder and the replacement of security certificates for more information.

Related tasks

- Updating the Finder
- Logging in to the Finder
- Importing and replacing Certificates
- Sending email notifications from the Appliance

Enabling Cross-Engine Finder features

Overview

Starting from V6.19, the Cross-Engine Finder features let you target multiple Engines at the same time or, in some use cases, target an Engine that is different from the one to which the Finder is connected.

The Cross-Engine features impact the following Finder areas:

- Investigations
- User view
- Device view
- Service view
- Search
- Auto-complete

Because the Cross-Engine features for search and auto-complete are especially demanding in terms of connectivity and system load, they can be enabled separately.

Prerequisites

The connectivity between Nexthink components and the total number of devices determine the Cross-Engine features that can be enabled in your setup. Find

below the configurations and values required to enable each level of the Cross-Engine features:

- The Portal and the Engines are located in the same data center or have equivalent connectivity.
- There is a good connectivity between Finder and Portal to enable:
  Basic features
  A bandwidth of **10 Mbps** between Finder and Portal allows for the basic Cross-Engine features to be enabled (investigations, device view, user view, and service view across multiple Engines).
  Search
  A bandwidth of **50 Mbps** or higher between Finder and Portal allows for the intermediate set of Cross-Engine features to be enabled (the basic features plus the search).
  Auto-complete
  A bandwidth of **100 Mbps** or higher between Finder and Portal,
  - where the total number of devices is below 50 K, allows for the full set of Cross-Engine features to be enabled (the basic features plus the search and the auto-complete).
  - where the total number of devices is above 50 K, contact Nexthink Support.

Keep the Cross-Engine features disabled if the prerequisites are not met.

Enabling the Cross-Engine Finder features from the Web Console

To enable the Cross-Engine Finder features:

1. Log in to the Web Console of the master Appliance as administrator.
2. Click the **FINDER** tab at the top of the window.
3. In the **General** section, tick one of the following options under **Cross-Engine Finder**:

- **Disabled**: To disable all Cross-Engine features. This is the default option.
- **Enable investigations, device view, user view and service view across multiple Engines**: To enable the targeting of multiple Engines when executing investigations or when analyzing the user and service views, as well as the ability to see the device view of devices that reside in another Engine.
- **Enable investigations, device view, user view, service view and search across multiple Engines**: To enable the search over all Engines in addition to the basic Cross-Engine features already present in the previous option.
- **Enable investigations, device view, user view, service view, search and auto-complete across multiple Engines**: To enable word completion from every Engine in addition to the search and the basic Cross-Engine features already present in the previous option. Limit the data retrieved for word completion with the additional settings of this option.

4. Click **SAVE** to permanently store your choice. The Portal is restarted if you modified the setting.

Limiting auto-complete data

The Finder uses auto-complete data to give you smarter search results and let you write your investigations faster and with higher confidence. However, loading too much auto-complete data may have an impact in the responsiveness of the Finder, specially during logon.

To keep the amount of retrieved auto-complete data under control, adapt the following settings when you activate the full set of Cross-Engine features:

While auto-complete data is always retrieved for all users and devices, filter the auto-complete data retrieved for other objects by most recently seen:

- **Domains seen in the last (number of days)**
- **Other objects seen in the last (number of days)**

Adjust the number of days for domains and other objects:

- Increase the value to get more auto-complete data.
- Decrease the value if you notice a negative impact of auto-complete data in the responsiveness of the Finder.

Individually disabling the Cross-Engine Finder features

Even when the Cross-Engine features are globally enabled in the Web Console, you may want to individually disable these features on some specific computers that run the Finder.

Disabling the Cross-Engine features can improve the performance of the Finder in environments with limited network bandwidth for users that do not need multi-Engine data, either because they usually work exclusively with one Engine or because their view is restricted to one Engine only.

To disable the Cross-Engine features for a single user, modify a value in the registry:

1. On the computer where the Finder is installed, press **Win+R** to open the Run box.
2. Type in **regedit** and press **Enter** to launch the Registry Editor.
3. Browse the Windows registry in the Registry Editor and select the key **HKEY_CURRENT_USER\Software\Nexthink**.
    - ♦ If the value **CrossEngineDisabled** does not exist in the key:
        1. Right-click the area on the right-hand side of the Registry Editor that holds the list of values for the key.

2. Select **New -> DWORD (32-bit) Value** from the context menu.
3. Type in **CrossEngineDisabled** as the name of the value.

4. Right-click the value with the name **CrossEngineDisabled** to change its data.
5. Select **Modify...** in the context menu. The dialog to edit the value shows up.
6. Set the value of the field **Value data** to 1 in the dialog.
7. Click **OK**.

Setting the maximum number of Cross-Engine investigation results in the Finder

When the Cross-Engine features are enabled in the Finder, the **List (all entities)** view lets you examine the results of an investigation over all available Engines. Because the Finder receives answers from multiple Engines, which are gathered by the Portal, the potential number of results are higher than those of a single Engine. By default, the **List (all entities)** view displays a maximum number of 10 000 results.

To modify the maximum number of results displayed in the **List (all entities)** view of the Finder:

1. Log in to the CLI of the master Appliance.
2. Optional: If the Portal has no explicit configuration file, that is, if `portal.conf` does not exist in folder `/var/nexthink/portal/conf`, create it by copying the defaults from the sample configuration file:
   ```
   sudo -u nxportal cp
   /var/nexthink/portal/conf/portal.conf.sample \
   /var/nexthink/portal/conf/portal.conf
   ```
3. Edit the configuration file of the Portal:
   ```
   sudo vi /var/nexthink/portal/conf/portal.conf
   ```
4. Add the following line at the end of the configuration file. Change the default value of 10 000 results to the desired value:
   ```
   globalconfig.portal.finder-selector.max-rows = 10000
   ```
5. Save your changes and exit the `vi` editor:
   ```
   :wq
   ```
6. Restart the Portal for the new limit to apply:
   ```
   sudo systemctl restart nxportal
   ```

Individually modifying the maximum number of Cross-Engine results in the Finder

Besides the global configuration, each Finder user may individually set the maximum number of Cross-Engine results displayed in the **List (all entities)** view.

To change the maximum number of Cross-Engine results in the **List (all entities)** view for a single user, modify a value in the registry:

1. On the computer where the Finder is installed, press **Win+R** to open the Run box.
2. Type in **regedit** and press **Enter** to launch the Registry Editor.
3. Browse the Windows registry in the Registry Editor and select the key **HKEY_CURRENT_USER\Software\Nexthink**.
    - ♦ If the value **CrossEngineMaxResults** does not exist in the key:
        1. Right-click the area on the right-hand side of the Registry Editor that holds the list of values for the key.
        2. Select **New -> DWORD (32-bit) Value** from the context menu.
        3. Type in **CrossEngineMaxResults** as the name of the value.
4. Right-click the value with the name **CrossEngineMaxResults** to change its data.
5. Select **Modify...** in the context menu. The dialog to edit the value shows up.
6. Set the value of the field **Value data** to the number of maximum desired results.
7. Click **OK**.

Related tasks

- Navigating the results of an investigation

Expanding the time frame of investigations in the Finder

Because of the large number of events that an Engine stores, investigations that iterate through activities or events may have a high computational cost for the Engine.

An investigation iterates through activities or events because of either one of the following reasons:

- The investigation retrieves activities or events. For example, an investigation that lists all the executions that ran on a particular device during the last hour.
- The investigation retrieves objects, but it does so under one or several of the following circumstances:
    - A condition on activities or events. For example, an investigation that lists the devices where a package was removed (uninstallation events) during the last day.
    - The computation of at least one aggregate that depends on activities or events and that is not pre-calculated for the full period available in the Engine. For example, an investigation that lists the devices with an outgoing network traffic bigger than 10 MB during the last hour.
    - A forced time frame restriction. For example, an investigation that lists the users with a time frame of *last 1 day* returns only the users that were active that last day.

These investigations do not admit the **Full available period** time frame because they could take too long to execute completely. In fact, to avoid long and costly computations in the Engine, the time frame of activity-related investigations is limited to a maximum of 7 days by default.

To circumvent the 7 days limit for investigations in the Finder, you need to manipulate the Windows registry. After removing the limit, the Finder allows you to query the Engine with investigations whose time frame spans up to the maximum number of days available in the Engine. Beware however that investigations with very long time frames may require more computation power from the Engine, rendering it less responsive and potentially impacting other users of the Finder, so you should handle this feature with care:

1. In the computer where the Finder is installed, press **Win(key)+R** to display the run dialog.
2. Type in **regedit** as the program to open in the dialog and press **Enter**. The Registry Editor opens.
3. Browse the Windows registry in the Registry Editor and select the key **HKEY_CURRENT_USER\Software\Nexthink**.
    - If the key does not exist, create it by right-clicking the **Software** folder:
        1. Select **New -> Key** from the context menu.
        2. Type in '*Nexthink* as the name of the new key.
        3. Right-click the area on the right-hand side of the Registry Editor that holds the list of values for the key.

4. Select **New -> DWORD (32-bit) Value** from the context menu.
5. Type in **Remove7DayLimit** as the name of the value.

4. Right-click the value with the name **Remove7DayLimit** to change its data.
5. Select **Modify...** in the context menu. The dialog to edit the value shows up.
6. Set the value of the field **Value data** to 1 in the dialog.
7. Click **OK**.

This method changes the value of the registry key in one computer only. Alternatively, you can use GPO to impose the same value for the registry key in all the computers where the Finder is installed.

The operations described in this article should only be performed by a Nexthink Engineer or a Nexthink Certified Partner.

If you need help or assistance, please contact your Nexthink Certified Partner.
Related tasks

- Editing the options of an investigation

Enabling Finder access to the Library

Overview

The Finder has access to the Library for importing ready-made content into your Nexthink setup.

Look for official content packs in the Nexthink Library catalog.

Access to the Library

Starting from 6.12, the Finder connects to the Library via HTTPS and not HTTP by default. The address of the Library is centralized in the Portal. The Finder gets the address of the Library when connecting to the Portal at user login.

The default address of the Library is `https://library.nexthink.com`. To provide your own content library, change this address in the configuration file of the Portal:

1. Log in to the CLI of the appliance that hosts the Portal.

2. Optional: If the Portal has no configuration file yet, that is, if `portal.conf` does not exist in folder `/var/nexthink/portal/conf`, create it by copying the defaults from the sample configuration file:
   ```
   sudo -u nxportal cp
   /var/nexthink/portal/conf/portal.conf.sample \
   /var/nexthink/portal/conf/portal.conf
   ```
3. Edit the Portal configuration file:
   ```
   sudo vi /var/nexthink/portal/conf/portal.conf
   ```
4. Add the following line:
   ```
   globalconfig.finder-content.library-url =
   "your_library_url"
   ```
5. Save your changes and exit:
   ```
   :wq
   ```
6. Restart the Portal to apply your settings:
   ```
   sudo systemctl restart nxportal
   ```

Provide the URL of your alternate content library either with the HTTPS or the HTTP scheme. For example:

- `https://mylibrary.example.com`, to connect through port 443 and a secure communication.
- `http://mylibrary.example.com`, to connect through port 80 and an insecure communication.

Deprecated access configuration

The following registry key in the device that runs the Finder is deprecated:

```
Computer\HKEY_CURRENT_USER\Software\Nexthink\LibraryUrl
```

Starting from 6.12, the Finder ignores the value of this registry key and only obeys to the address provided by the Portal to access the Library.

Related references

- Nexthink Library

Finder proxy support

The following content applies exclusively to the Nexthink Cloud offering.

Overview

The devices in a corporate network typically connect to the Internet through a proxy server instead of using a direct connection. A *proxy server* or *proxy* forwards the requests of client applications that run on the corporate devices to the servers that run on the Internet, as if the proxy itself initiated the requests. Then the proxy sends the responses from the servers back to the clients. By acting as intermediary, a proxy server can provide varied functionality such as content filtering (for improved security) or content caching (for better performance).

Thus, in a Nexthink Cloud setup, Finders inside a corporate network that is equipped with a proxy server are usually required to send their traffic through the proxy to reach the Portals and Engines in the Nexthink Cloud. In this article, learn about the different types of proxies and configurations supported by the Finder.

Supported types of proxies

The Finder supports the following types of proxies without user authentication:

- **HTTP** (web) proxy

The Finder should work out of the box with *transparent proxies* as well. Transparent proxies automatically intercept network traffic that goes from the corporate network to the Internet, so that clients are not aware that their traffic is traversing a proxy.

Supported proxy configurations

Being a desktop application, the Finder gets its proxy configuration through the Microsoft Windows Internet (WinINet) API. The WinINet API was designed to give interactive desktop applications access to standard Internet protocols such as HTTP or FTP. Other notable applications such as Internet Explorer get their

proxy configuration via WinINet as well. This configuration is visible from the Internet Properties dialog of the Control Panel:

1. Press the **WinKey**.
2. Type in **Internet Options** and press **Enter**. The **Internet Properties** dialog shows up.
3. Select the **Connections** tab.
4. Under the section **Local Area Network (LAN) settings**, click the button **LAN settings**

5. Choose how WinINet should configure the LAN settings (which include the proxy settings), tick either:
    - **Automatically detect settings**, to use *Web Proxy Auto-Discovery (WPAD)* protocol, a method to set the proxy settings automatically by leveraging the DHCP and DNS protocols. WPAD uses discovery methods in DHCP and DNS to find out the URL of a PAC file.
    - **Use automatic configuration script**, to get a PAC file from the specified URL in **Address**. A *proxy-auto-config (PAC)* file is a JavaScript file with a single function that determines which proxy should be used for each client connection.
    - **Use a proxy server for your LAN**..., to manually configure the proxy settings.

Finder logic to connect through a proxy

To choose whether to use a proxy or not, the Finder follows the steps below:

1. Calls the WinINet API to determine the proxy to use for connecting to the Portal via WebSocket with the URL:
   `wss://<portal_address>:443`
2. Connect to the Portal either:
    - Through a proxy server, if the call to WinINet returned a proxy.
    - Via a direct connection, if the call to WinINet returned no proxy.

3. If the connection fails, whatever the means to acces the Portal, the Finder calls WinINet again, but this time with an HTTPS scheme in the URL:
   `https://<portal_address>:443`
4. If the connection is successful, the proxy may ask for user authentication (only Basic authentication is currently supported).
   - ♦ If a proxy dialog shows up asking for username and password, enter your credentials.
5. The Finder should be now connected to the Portal.

The Finder follows the same logic to connect to the Engine. Note that the proxies to access the Portal and the Finder might be different depending on how the system was configured.

Updating from V6.x

Updating the Appliance

Overview

Starting from Nexthink V6.6, the Appliance offers a simplified auto-update mechanism that requires minimal intervention. For a stricter control over the moment of the update, manual updates are still possible. Whether automated or manual, not only does the new update mechanism update all your Appliances at once, but also provides updates for the Finder and the Collector.

When updating from Nexthink V6.5 or a previous version, update of your Appliances as usual, using either the online or the offline update described below. At the end of the process, your updated Appliances get into an intermediate state called compatibility mode. In compatibility mode, you can still work and update your Appliances individually. Federate your Appliances to enable the new update mechanism, along with many other advantages.

When updating from Nexthink V6.16 or a previous version to V6.17 or above, the Appliance undertakes a process of content centralization, during which most of the content that was previously local to each Engine becomes shared by all Engines. The centralization of content offers users a unified experience across all Engines when connecting to them with the Finder, without the need to manually export and import content from one Engine to each other.

Automatic updates

The automatic update of the Appliances helps you maintain your Nexthink software up-to-date in a centralized and comfortable way. For V6.6 and later, this is the preferred method to update your Appliances. Choose the day of the week and the hour of the day when updating your Portal and Engines is more convenient for you. The automatic update requires your Appliances to be federated.

To enable the automatic update of your Appliances:

1. Log in to Web Console of the Appliance that hosts the Portal (the master) as admin. In your browser, type the URL https://<portal.dns.or.ip>:99.
2. Click the **Appliance** tab at the top of the window.

3. Select the section **Update** from the left-hand side menu.
4. Tick the box **Automatically update Nexthink Appliance and installed components**, the subsection **Update on** shows up below with a couple of selection lists.
    1. Select the day of the week when you want to do the update. Choose the default value **any day** if you do not have a preferred day.
    2. Select the hour of the day when you want the update to start. Choose the default value **any time** if you do not have a preferred time.

At least one week must have passed between the availability of the update and the actual update of your Appliances. For instance, if you selected your preferred day of the update to be on Friday, and the update is available since Wednesday, the actual update will take place on Friday of the next week.

Online update

If your appliances have access to the Internet, this is the recommended method to update Nexthink whenever automatic updates are not enabled:

1. Log in to the Web Console of the Appliance to update as administrator. In your browser, type the URL https://<appliance.dns.or.ip>:99.
    - In V6.6 or later, if your Appliances are federated, log in to the Appliance that hosts the Portal to centrally manage the update process of all your Appliances (Portal and Engines).
2. In the section **Appliance**, select the tab **Update**. If your Appliances are federated, this tab is available in the Portal Appliance only.
3. Optional: Click the circular arrows in the **Last check for update** row to see if there is a new system update or any update of the installed Nexthink components: Portal, Engine or Web Console. If there is any update available, it is displayed in the cell on the right hand side. For each released component, find here a direct link to its release notes.
4. Optional: Check the box **Enable** of the **Automatic update** row to get the updates from the Nexthink repository as soon as they are published.
5. Optional: Press the button **Start connectivity test** to verify your connection to the Nexthink repository (updates.nexthink.com). If the repository is reachable, a message of success is displayed.
6. Click the button **Start update** to trigger the update process. By the end stages of the update, the Web Console shows its new user interface.
7. Wait for the message **Everything is up-to-date**. The update of the Appliance has been completed.

Some updates require rebooting the Appliance to be complete. Refer to the chapter on rebooting the Appliance below for more information.

Offline update

The Appliance relies on *yum* to manage the upgrade of its components. When the appliance is connected to the Internet, the Web Console instructs the yum utility to get the upgrades from the Nexthink repository. In the case that your appliances are not connected to the Internet, you must download the offline update package and, if there is any system update, the Appliance ISO. You must then manually update the Appliance using yum from the command line.

If the Appliance ISO of a particular version of Nexthink is not yet available for download, but the offline update package is already downloadable and you need to install it, ensure at least that you update your appliances to the latest available ISO (usually the ISO of the previous version) before updating the rest of the Nexthink components.

The Appliance ISO contains the operating system, the Web Console, other auxiliary packages, and the security updates for the Appliance; whereas the offline update package is a tgz file that holds the Nexthink components: Portal, Engine, Finder, and Collector. For updating each one of your appliances offline, follow the steps below.

Applying system updates

To manually update the system packages of each Appliance, using yum and the Appliance ISO:

1. Attach the Appliance ISO to the physical or virtual system that hosts.
2. Log in to the command line interface (CLI) of the Appliance.
3. Mount the ISO with the following commands:
   ```
   sudo mkdir -p /media/cdrom
   sudo mount -t iso9660 /dev/cdrom /media/cdrom
   ```
4. Update the system packages (ignore any message about already installed packages):
   ```
   sudo rpm -Uvh /media/cdrom/CentOS/centos-release-*.rpm
   sudo yum --disablerepo=* --enablerepo=c7-media --nogpgcheck
   \
   --exclude=nxconsole update
   ```
5. Wait for the operation to finish and then disconnect the ISO from the system using the following command:
   ```
   sudo umount /media/cdrom
   ```

If the system updates include a modification of the kernel of the operating system, you need to reboot the Appliance to load the new kernel. Refer to the chapter on rebooting the Appliance below.

Updating Engine, Portal and Web Console

To manually update the Nexthink components of each Appliance:

1. Connect to the corresponding Appliance to update with your favorite SCP client and copy the offline update package (tgz file) to `/home/nexthink/`. Make sure that you copy the offline *update* package and not the offline *installation* package. The latter is designed for a clean install only, not for an update.
2. Untar the offline update package:
   ```
   tar -xzvf Nexthink-offline-update-6.x.tgz
   ```
3. Ensure that the installation script is executable:
   ```
   sudo chmod a+x install_Nexthink_v6.sh
   ```
4. Run the installation script:
   ```
   sudo ./install_Nexthink_v6.sh
   ```
5. Log in to the Web Console as administrator.
6. Check that the update was correctly completed by verifying the versions of the installed components in the **Information** tab of the **Appliance** section.

Verifying the running version of the Appliance

To verify the version of the Appliance and of the installed components that are currently running on your machine:

1. Log in to the Web Console.
2. In the **APPLIANCE** tab, select **General** from the left-hand side menu.
3. Find the versions of the installed components on the table under **Versions**.

The versions displayed in the Web Console must match the versions advertised in the Release Notes of the corresponding update. Check the update process otherwise.

Rebooting the Appliance

Usually, you do not need to reboot the Appliance after an update. In the case of system updates that install a new kernel for the operating system, however, it is necessary to reboot the Appliance to load the new kernel. This condition will be made clear in the release notes of the update.

To reboot the Appliance after an update:

1. Log in to the Web Console as administrator.
2. In the **Appliance** section, select the **General** tab.
3. Under **Status**, click the button **REBOOT APPLIANCE**.
4. To the question **Are you sure you want to reboot the Appliance?**, answer by clicking **OK**.

Related tasks

- Federating your Appliances
- Compatibility mode
- Content centralization when updating the Appliance

Content centralization when updating the Appliance

Overview

When updating the Appliance to V6.17 or later from a previous version of Nexthink, a set of elements that were previously local to each Engine become centralized:

- Investigations
- One-click investigations
- Investigation-based alerts
    - My alerts
    - Global alerts

The centralization of content offers users a unified experience with the Finder across all Engines. Users find their own set of investigations and alerts no matter the Engine to which they connect from the Finder, without having to manually replicate their content on every Engine. Therefore, centralization specially benefits Finder users that work in a multi-Engine setup. For Finder users that work in a single-Engine setup, the user experience remains the same after centralization.

The centralization process

Centralization takes place automatically during the upgrade of the Appliance to V6.17 or later. As usual, the Appliance that runs the Portal gets upgraded first. Subsequently, each Engine gets upgraded. As soon as an Engine restarts, it tries to reconnect to the Portal. When an upgraded Engine re-establishes connection with the Portal, the Portal retrieves the content to be centralized from the Engine. As other Engines connect in turn, their content is added to the Portal. For its part, the Portal sends a replica of the centralized content to all the already connected Engines. In case of the presence of duplicated or similar content in more than one Engine before centralization, the Portal adopts different merge strategies that are discussed below.

At the end of the process, all Engines are synchronized with respect to user-defined content. The same user sees therefore the same content regardless of the Engine to which the Finder is connected. However, different users may see different content. Only the owner (the creator) of an investigation, a one-click, or a user-specific alert is able to see it and manage it from the Finder.

Centralization runs to completion in around one minute for large multi-Engine setups. Measurements in test environments indicate a duration of the centralization process of approximately 15 seconds for a setup with 20 Engines, 100 users and 2 000 items to centralize. Note however that this duration highly depends in the quantity of total content to centralize and, specially, in the quality of the connection between Portal and Engines. If for any reason an Engine is disconnected from the Portal, its content will not be centralized until it is connected back again.

Merge strategies

Some of the content to be centralized may reside in more than one Engine because a user recreated or copied it to several Engines. Besides, part of the copied items may have been modified in a few Engines. For instance, the definition of an investigation in one Engine may be different from the definition of a similar investigation with the same name in another Engine, because the time frame, conditions, or display fields were changed.

Thus, centralization classifies similar items according to three criteria:

- The folder path where the items are located in the Finder.
- The name given to the items.
- The definition of the items.

Based on the previous criteria, centralization considers three cases for merging repeated or similar content:

Real duplicates
: Items that are located in the same folder, share the same name, and have the same definition.

False duplicates
: Items that are located in the same folder, share the same name, but have a different definition.

Rest of items
: Items that do not fall into any of the two previous categories.

Let us illustrate how the different centralization strategies work with an example that covers the three cases. Although the example features the tree of investigations of a user in two different Engines, the principles described equally apply to trees of one-clicks and alerts and to setups with more than two Engines.

Real duplicates

Real duplicates are merged into a single item during centralization. Here, **Investigation 2** is found in **Folder A** in the tree of investigations of both **Engine 1** and **Engine 2**, and we assume that the investigation holds the same definition in both Engines.

False duplicates

False duplicates are detected during centralization and copied into a separate subfolder. In the example, we find **Investigation 1** in **Folder A** of the tree of investigations in both **Engine 1** and **Engine 2**, but the definition of this investigation is different on each Engine. From the screen capture of the example itself, we can deduce that **Investigation 1** on **Engine 1** is different from **Investigation 1** on **Engine 2** because of their associated icon: while the former

is an investigation on devices, the latter is an investigation on users. They just happen to be located in the same folder and have the same name.

Assuming that the content of **Engine 1** was centralized first, **Investigation 1** from **Engine 1** is placed directly into **Folder A** of the merged investigation tree. On the other hand, because of the path and name collision, **Investigation 1** from **Engine 2** is copied to a subfolder **Engine 2** within **Folder A**, as depicted in the figure.

Rest of items

Items that do not collide in name and folder path simultaneously are just copied with their own name to the same location in the merged investigation tree as they had in their original Engine before centralization.

Status of alerts

Because centralization gathers alerts from all connected Engines, each Engine ends up with the same or more investigation-based alerts than it had before upgrading. To avoid exceeding the limit of enabled alerts per Engine during

upgrade, alerts stay enabled only on the Engines where they were originally enabled. That is, the definition of alerts are replicated on all Engines, but not their status. Thus, alerts that appear on an Engine as a result of the centralization process are disabled by default. This is valid both for global alerts and for user-specific alerts (those in the section **My alerts** of each user). Similarly, newly created alerts are only enabled on the Engines in which they are created.

To enable a replicated alert on a particular Engine:

1. Log in to the Finder.
    1. Connect to the specific Engine on logon.
2. Go to the **Settings** section on the left-hand side accordion of the main window.
3. Choose the type of alerts to see from the drop-down list:
    - Select **Global alerts** to see the alerts that are global to all users (only users with the right permissions can manage global alerts).
    - Select **My alerts** to see the alerts that are specific to your user account.
4. Right-click the name of the alert and select **Enable on current Engine**.

Remember that enabling an alert will fail if the limit of enabled alerts has been reached on that Engine.

Monitoring the centralization process

Following the centralization from the Portal

To follow the centralization process from the Portal:

1. Log in to the Portal as a central administrator.
2. Click the **ADMINISTRATION** drop-down menu at the top of the window.
3. Select the **Engines** dashboard.
4. Look at the status line at the bottom of the table with the list of connected Engines.

The status line holds a list of the Engines for which centralization is still pending:

Troubleshooting centralization

If the centralization process is unable to complete in an Engine, the Portal displays the following message in the status line:

- **Content centralization failed for the following *N* Engine(s): <List of Engines>**

To retry the centralization process on an Engine where it failed:

1. Log in to the Portal as a central administrator.
2. Click the **ADMINISTRATION** drop-down menu at the top of the window.
3. Select the **Engines** dashboard.
4. Click the chain icon to disconnect the failed Engine from the Portal.
    1. Wait until the connection status light turns red.
5. Click again the chain icon to connect back the Engine to the Portal.
    1. Wait for the Portal to attempt the centralization again.

If the problem persists, please contact Nexthink Support.

Logging in to the Finder during upgrade

As explained above, centralization is a fast process. However, in the unlikely event that a user tries to log in to the Finder while the centralization procedure is going on, the Finder displays the following message to ask the user to log in later:

Related tasks

- Creating an investigation-based alert

- Local and shared content

Updating the Collector

Overview

Starting from V6.6, Collectors are able to update themselves in coordination with the update of your Appliances. However, to migrate your Collectors from V6.5 or previous to V6.6, you still need to use conventional methods:

- Run the executable generated with the Collector Installer on each device.
- Use the MSI of the Collector, either manually on each device or using your favorite deployment tool.

Note that the Updater (V6.5 or previous) is deprecated and it is not able to update your Collectors to V6.6.

The Collector for Mac devices does not have an update feature yet. Update it either manually or using your preferred deployment tool.

Applies to platforms:

Updating the Collector with the Collector Installer

To update the Collector using the Nexthink Collector Installer:

1. Generate the executable with the options to update the Collector as described in the installation instructions.
2. Run the generated executable in the device with the old Collector.
    - The installation options which are not visible in the Collector Installer get their values from the replaced Collector. That is, the options that you do not set in the Installer are preserved from your previous installation.
    - If the deprecated Updater is detected to be present in the device, the installer program uninstalls it.

Updating the Collector with the MSI

To update the Collector using the Collector MSI:

1. Remove the Updater (version 6.5 or previous), if present, from the devices where you want to update the Collector.

- Failing to do so results in subsequent attempts to install the Collector by means of the MSI being unsuccessful.

2. Perform an interactive or unattended installation of the Collector in the device, as described in the installation instructions, or use your favorite tool to deploy the MSI of the Collector.
   - Only if you do an interactive installation while the old version of the Collector is running, the following message shows up at the beginning of the installation:

1. Click **OK** to proceed with the rest of the installation steps. Rebooting the device is not required.

Uninstalling the Collector interactively

To update the Collector, it is not necessary to first uninstall the previous version. If nevertheless you decide to remove the Collector by means of the **Add and Remove Programs** feature of Windows, a somewhat misleading message shows up at the end of the uninstallation process:

You can safely click **OK** and ignore the warning. A reboot is actually not required.

Automatic updates

As previously said in the overview, note that the automatic update of the Collector only works in those devices where you have already installed a Collector 6.6 or higher. Devices with Collector 6.5 and previous will not get updated by the new auto-update mechanism. The rest of the sections assumes

that the version of the installed Collectors is 6.6 or higher.

When automatic updates are enabled, the installed Collectors are updated in two waves. Choose when to update the Collectors by assigning each device to an update group:

Pilot
: The device is updated during the first wave. Use a small group of pilot devices as early adopters to confirm that the new version integrates well within your infrastructure.

Main
: The device is updated during the second wave, after pilot devices. Put the majority of your devices into this group.

Manual
: The device is not automatically updated. Use this group only for special devices that should not be updated automatically.

To assign an update group to a set of devices:

1. Log in to the Finder as the main admin.
2. Execute an investigation on devices.
3. From the results of the investigation, select the device or devices that you want to update (to select all devices in the list of results, press **Ctrl+A**).
4. Right-click the selected devices.
5. Select **Edit...** from the context menu. The **Edit device** dialog shows up.
    1. In the **Nexthink Collector update group**, at the bottom of the dialog, select one of the three possible options: pilot, main, or manual.
    2. Click **Apply**

After assigning your devices to an update group, enable the automatic update of the Collector.

To enable the automatic update of the Collectors:

1. Log in to the Web Console of the Appliance that hosts the Portal as admin.
2. Click the **Appliance** tab at the top of the window.
3. Select the **Update** section from the left-hand side menu.
4. Under **Nexthink Collectors**, tick **Automatically update pilot Collectors** to enable the auto-update of the Collector in those devices that you assigned to the pilot group.

- In **Target version**, choose whether to update the pilot group to the latest available version of the Collector (recommended) or to one of the versions stored in the Appliance.
- In **Speed**, decide how fast you want the Collectors to be updated. Note that the faster the update, the more bandwith devices will require from the network to download the new version of the Collector. Choose between:
    - **expedite**, for updating all pilot Collectors in about one day.
    - **normal**, for updating all pilot Collectors in about one week.

5. Tick **Automatically update main Collectors** to enable the auto-update of the Collector in those devices that you assigned to the main group. Note that the automatic update of the pilot group is mandatory for automatically updating the main group.
    - In **Target version**, choose to update to the same version used for the update of the pilot group (recommended) or to a previous version stored in the Appliance. Select as well the moment to trigger the update of the main group. The best practice is to leave a sensible test period to the devices in the pilot group (e.g. two weeks) before updating the rest of the Collectors.
    - In **Speed**, choose between **expedite** or **normal** as you did for the update of the pilot group. Note however that you will usually assign the vast majority of your devices to the main group, requiring many more downloads than the pilot group. It is therefore recommended to use the value **normal** as the speed for updating the main group.

The automatic update of the Collectors is independent of the automatic update of the Appliance. Even if the automatic update of the Appliance is turned off, the Collectors are updated as described in this article, as long as your master Appliance is connected to the Internet and able to reach the Nexthink updates site, and only if the updated Collector version is still compatible with your installed Appliances.

Discovering devices

To keep track of the devices in your network that do not have the Collector installed yet, use the **Device discovery** tab in the Finder.

A device that does not have the Collector installed never sends information to the Engine. Therefore, the Engine ignores the existence of the device. To inform the Engine about all the devices in your network, including those that the Engine may not be aware of, create collections of devices in the **Discover** tab. You can create collections of devices based on:

- The information in Active Directory.
- The contents of a CSV file.

To create a collection of devices based on Active Directory, make sure first that you have configured the Active Directory server settings in the Engine:

1. Log in to the Finder as the main admin.
2. Select the **Device discovery** section in the left hand side accordion.
3. Right-click the title or the empty area of the **Discover** tab.
4. Select the option **Create collection from AD...** in the context menu.
5. Review how to locate your devices in the Active Directory and fill in the blanks in the dialog:
    1. Set a name for the collection in the field **Name**.
    2. Write in the field **Include DN** a query pattern to retrieve all the devices whose Distinguished Name matches the pattern. You can use the wildcards *****, to substitute for zero or more characters, and **?**, to substitute for one character, in your query.
    3. Optional: If your query pattern above includes some devices (or other AD objects) that you want out of the collection, specify them in **Exclude DN** with another query and tick the check box to the left to activate the exclusion.
6. Click OK to create the collection.

If you do not have Active Directory available to your Engine, but you have other means to get a list of all the devices in your network, you can still create a collection of devices in the **Discover** tab by providing a CSV file. The CSV file must hold at least two values per entry:

- The NetBIOS name of the device.
- The IP address or DNS name of the device.

To create a collection from a CSV file:

1. Right-click the title or the empty area of the **Discover** tab.
2. Select the option **Create collection from CSV...** in the context menu.
3. Choose a CSV file from your filesystem in the dialog that opens. A wizard guides you through the import of the CSV.
4. In step 1 of the wizard:
    1. Select the encoding, the delimiter character and the text qualifier (character used to delimit text values) of the CSV file.
    2. Optional: Click **Show file** to see the actual CSV file and help you decide what are the correct options.
    3. Click **Next**.

5. In step 2 of the wizard:
    1. Give a name to the collection that you are creating in the field **Collection name**.
    2. In **Column selection**, pick the two columns from the CSV file that hold the Netbios name of the device and the IP address or DNS name (hostname). To guide you with the selection, the values of the first entry in your CSV file are displayed in the lists.
    3. Optional: Click **Back** to correct the options that you chose in step 1 of the wizard if you realize that you set something wrong.
    4. Click **Import**.
6. The wizard reports the number of devices successfully added to the collection from the CSV file. In case of error, click **Show details** to see the reasons for not importing all the entries from the file.
7. Click **OK** to end the wizard.

In the **Discover** tab, every collection of devices displays its total number of devices to the right of its name. Additionnaly, each collection is divided into two disjoint groups of devices that also show their number of devices:

- **Without Collector**: those devices that do not have the Collector installed.
- **With Collector**: those devices that have the Collector installed.

To get a list of the devices in the collection or in any of the groups, double-click the collection or the group in the **Discover** tab. The groups get updated at the same time as the Engine detects if the Collector is installed in or uninstalled from the devices in the collection.

Related tasks

- Installing the Collector
- Importing data from Active Directory

Related references

- Collector MSI parameters reference table

Viewing Collector deprecated fields

Starting from V6.6, the fields that relate to the update of the Collector with the deprecated Updater are effectively deprecated as well. You may still need to take a look at these fields if you have old versions of the Collector in your infrastructure and you want to know about their exact update status.

To be able to see these deprecated fields in both the Finder and the Portal, set this value in the Windows Registry of all the computers that have the Finder installed:

1. On the computer where the Finder is installed, press **Win+R** to open the Run box.
2. Type in **regedit** and press **Enter** to launch the Registry Editor.
3. Browse the Windows registry in the Registry Editor and select the key **HKEY_CURRENT_USER\Software\Nexthink**.
    - ♦ If the value **DeprecatedFieldsVisible** does not exist in the key:
        1. Right-click the area on the right-hand side of the Registry Editor that holds the list of values for the key.
        2. Select **New -> DWORD (32-bit) Value** from the context menu.
        3. Type in **DeprecatedFieldsVisible** as the name of the value.
4. Right-click the value with the name **DeprecatedFieldsVisible** to change its data.
5. Select **Modify...** in the context menu. The dialog to edit the value shows up.
6. Set the value of the field **Value data** to 1 in the dialog.
7. Click **OK**.

For the user to distinguish them easily, these Collector fields appear in a separated section in the Finder and with the suffix **(deprecated)** in the Portal.

Applies to platforms:
Related references

- Data-model changes in V6.6

Updating the Finder

Overview

Whenever you log in to the Finder, the Finder checks the compatibility between its own version and the version of the Portal to which it connects. If the version of the currently installed Finder is compatible with that of the Portal and there is no new version of the Finder available for download, the Finder connects normally. Otherwise, several scenarios are possible:

- The Finder is compatible with the Portal, but there is a new version of the Finder available for download.
- The Finder is no longer compatible with a newer Portal and you must upgrade it.
- The Finder is not compatible with an older Portal and you must downgrade it.

Depending on the Finder being installed from the Portal or from the per-machine installer, the procedures to update the Finder are different. Prefer installations from the Portal whenever possible, as they let standard Windows users update the Finder easily and, in most cases, automatically. In their turn, per-machine installations require administrator privileges in Windows and more manual intervention. Use the per-machine installer only when required (e.g. installation on Citrix environments).

Let us examine the different update procedures in detail in the next sections.

Automatic updates

If you install the Finder from the Portal, the Finder automatically updates itself without notice whenever there is a new version of the Finder that does not break the compatibility with the Portal:

1. Log in to the Finder. While connecting to the Portal, the Finder detects that there is a new version available and downloads its installer program.
2. Upon exiting the application, the Finder silently launches the installer in the background to update itself.
3. Optional: when you open the Finder again, choose an action depending on the result of the update:
    - If the update succeeded, a notification appears at the top of the window indicating the new version number.
    - If the update failed, a notification appears at the top of the window to inform you that something went wrong. Click the link **Open detailed log...** to help you troubleshoot your update problems.

In a per-machine installation, the Finder does not update itself automatically and it does not inform you of the availability of a new version in the case of a minor change; that is, a change that does not break the compatibility between the Finder and the Portal. You can nevertheless download and install the new version of the Finder from the Product Downloads page, as usual.

The automatic update of the Finder is independent of the automatic update of the Appliance. Even if the automatic update of the Appliance is turned off, the Finder

is still updated as described in this article, as long as your master Appliance is connected to the Internet and able to reach the Nexthink updates site.

Mandatory upgrades and downgrades

When a change actually breaks the compatibility between the currently installed Finder and the Portal, the Finder is said to require an either an upgrade or a downgrade.

Mandatory upgrades usually occur when a major version of the product is released. In their turn, mandatory downgrades are rarer: downgrades appear only when you try to connect to an older version of the Portal. A typical downgrade scenario would consist of a pre-production environment where you install a new version of Nexthink. If you try to connect the new Finder to the Portal in your production environment, which is still running an older version of Nexthink, it will ask for a downgrade.

To execute a mandatory upgrade or downgrade of the Finder:

1. Log in to the Finder. While connecting to the Portal, the Finder detects that it is incompatible with the version of the Portal.
    - If you installed the Finder from the Portal, a dialog shows up, indicating that the Finder requires an upgrade (or downgrade) and displaying version information.
        1. Click **Upgrade** (**Downgrade**) to start the download and installation process.
        2. Once the upgrade (downgrade) has finished, the Finder restarts and restablishes the same session that you used to log in.
        3. Optional: Click the temporary link **See what's new...** that shows up at the top of the window to open the release notes for the new version of the Finder.
    - If you installed the Finder in a *per-machine* context, you simply get an error message.
        1. Go to the Product Downloads page.
        2. Download and run the appropriate installer for your computer architecture.

Setting the Portal address for Finder updates

The Finder relies on the configuration of the Portal address for performing automatic and mandatory updates. The provided address is used to connect to the Portal, detect new versions, and download them for installation.

To configure the Portal address for Finder updates:

1. Log in to the Web Console of the Appliance hosting the Portal from a web browser as admin:
   https://<IP_address_of_Appliance>:99
2. Click the **Portal** tab at the top of the window.
3. Select the **General** section fromt the left-hand side menu.
4. Under **Parameters**, type in the name or IP address of the Portal in **Portal address**.
5. Click **SAVE CHANGES** and wait for the Portal to restart.

Note that this address is required for drilling-down from the Portal to the Finder and it is also used for building the links to the Portal of the email digests.

Related tasks

- Installing the Finder
- Drilling-down to the Finder
- Sending email notifications from the Appliance

Security and user account management

Importing and replacing Certificates

Overview

To protect sensitive information against eavesdropping, all the communications between client applications (e.g. web browsers) and Nexthink components, as well as most of the communications between pairs of Nexthink components themselves, are encrypted. The majority of these communications are protected by combining the *Transport Layer Security* (TLS) protocol with a *Public Key Infrastructure* (PKI) scheme based on the X.509 standard, which uses digital certificates. A few others, most notably the transmission of device data from the Collector to the Engine, use different protection mechanisms.

This article focuses on the communications that are protected by TLS and a PKI scheme. More specifically, the article details how to replace the default digital certificates in the product by your own set of certificates when needed.

Understanding how TLS and PKI works

Although the reader is assumed to be familiar with the technologies behind TLS and PKI, let us briefly review them here for applying them later to the configuration of Nexthink. This review of how TLS and PKI work does not pretend to be exhaustive. It covers only the basic concepts and the most common cases that are considered relevant to the configuration of the Nexthink product. For an authoritative document on TLS, refer to the RFC 5246. For an authoritative document on PKI, refer to the ITU-T standard X.509. Find as well other sources of information about PKI and TLS on well-respected Internet sites, such as the Certificate Authorities Council or the support pages of the main web hosting and security companies.

A PKI is built around digital certificates. Certificates are just computer files that, in a typical client-server model, help clients ensure the authenticity of the server and protect the privacy of the communication. To that end, certificates rely on a pair of cryptographic keys that are mathematically linked: a private key, which identifies the server and must never be disclosed, and a public key, which is included unencrypted in the certificate itself and, therefore, is disclosed to everyone that gets the certificate. Content that is encrypted using the private key can only be decrypted using the public key and viceversa.

According to the TLS protocol, a client that wishes to connect to a server receives the certificate from the server. Once the identity of the server has been established by means of its certificate (see below), the client and the server negotiate the parameters of the secure connection. Thanks to the mathematical properties of their public and private keys, client and server are able to privately agree on the encryption algorithms to be used during the session and exchange randomly generated new keys for these algorithms. Some of the encryption algorithms that TLS negotiates may get old and become less secure with time, as the computation power of modern computers increases or vulnerabilities are found. To define the lists of allowed encryption algorithms for each server component in Nexthink, see the article on the Security Settings in the Appliance.

To understand how a client authenticates a server through its certificate, let us discuss first how server certificates are issued. An entity that issues certificates is called a *Certificate Authority* (CA). A CA owns a public and a private key with the properties described above. When a CA issues a new certificate to a particular subject, the CA *signs* the certificate with its own private key (signing is encrypting a hash of the certificate text), thus generating a digital signature that is attached to the certificate. A digital certificate holds thus the following important information:

- **Issuer**: the name of the entity that issues the certificate (the CA, in our case).
- **Subject**: the name of the entity to which the certificate is granted (when applied to the Nexthink product, this is usually the DNS name of a server component).
    - ♦ It is possible to issue a certificate for multiple subjects (DNS names) at the same time.
- **Public key**: the public key of the subject entity.
- **Digital signature**: the binary result of signing the certificate with the private key of the issuer.

To protect a server component in Nexthink, request a CA to generate a certificate that uses the DNS name of the server component as subject and binds it to the public key of the server with a digital signature. This is the certificate that you will use to replace the default server certificates in Nexthink. To that end, generate a *Certificate Signing Request* (CSR) with your server component information and send it to a CA. You need this step even if you are your own CA. Usually, CAs provide you with tools to generate CSRs for them. Alternatively, use OpenSSL to generate your CSRs. Remember to specify the subjects (DNS names) for the certificate in the field *Subject Alternative Name* (SAN) of your CSR, instead of the now deprecated *Common Name* (CN) field. To generate your own CSR with OpenSSL, follow the instructions in the CAcert site, for example.

A certificate that a CA issues to identify itself is called a *root* certificate. A root certificate thus has the same Issuer and Subject. Because the Issuer and the Subject are the same, the root certificate is said to be *self-signed*: it holds the public key of the CA and it was signed with the private key of the CA. If a client trusts a CA, the client may use the root certificate of the CA to authenticate any server certificate issued by the same CA. Indeed, by using the public key of the CA in the root certificate, the client can decrypt the contents of the digital signature in the server certificate (remember that the CA signed the certificate by using its private key); thus verifying the authenticity of the server. Since clients rely on root certificates for validation, root certificates must be distributed to clients in a trustworthy way (not through a simple connection). For instance, the root certificates of publicly-trusted CAs are typically distributed embedded in the operating system, the web browser, or other trust stores of specific applications. Users must therefore assume that the root certificates included in their client software are correct; that is, users must either trust the publisher of the client software or not use the software.

Usually, CAs add several layers of security and they do not sign server certificates using the private key of the root certificate, but the private key of an intermediate certificate. This intermediate certificate is itself signed by either the private key of the root certificate or by the private key of another intermediate certificate. Intermediate certificates thus form a *chain of trust* from your server certificate up to the root certificate of the CA, which is the *trust anchor*. When using server certificates that were generated in this way, install in your server component not only the server certificate, but the whole bundle of intermediate certificates that let a client follow the full chain of validation until it reaches the root certificate. Alternatively, install in the client the intermediate certificates besides the root certificate of the CA.

When talking about CAs, people usually refer to publicly-trusted institutions, but you can generate your own certificates and become your own CA as well. The certificates generated by an individual or organization without requesting them to a publicly-trusted CA are said to be *self-issued*. Using certificates issued by publicly-trusted CAs have some advantages over the use of self-issued certificates though. See the comparison table below:

Type of certificate	Issued by a trusted CA	Self-issued
Pros	• The CA manages the security of the private key of the root certificate	• You can issue certificates freely

	• Client software trusts your server certificates by default (root certificates are already in their trusted store)	
Cons	• You pay for each issued certificate	• You must manage the security of the private key of your root certificate • Client software does not trust your server certificates by default (you must distribute your root certificates)

For the purposes of this article, you do not need to know the full TLS protocol nor the mathematics of public key cryptography. There are just a couple of things that you must absolutely keep in mind when dealing with digital certificates in Nexthink:

- Server certificates enable clients to authenticate and communicate securely with a server.
- The issuer of your server certificates can be either a publicly recognized CA or your own organization (self-issued certificates).
- The client must trust the issuer of the server certificate to accept the connection.

For example, in the case of a web browser acting as a client, if the browser is not able to authenticate the server that hosts a web site because it does not know the issuer of the server certificate (i.e. it does not have the root certificate of the issuing CA in its trusted store), you usually get a warning message informing you that the connection to that web site is not secure. Most web browsers let advanced users add an exception and proceed with the connection, clearly stating nonetheless that the identity of the server cannot be confirmed and that it might be impersonated.

Secure connections in Nexthink requiring certificates

In Nexthink, the components in the Appliance (the Web Console, the Portal, and the Engine) usually play the server role in the client-server model of communication. Each one of these components uses a server certificate to provide client applications with the means to establish a secure connection. Nevertheless, the Engine and the Portal may also play the client role in some

cases. For their part, the Finder and the Collector always behave as clients.

For each component, find below the table of all the connections that require certificates:

Client	Server
Web Browser	• Web Console • Portal • Engine (Web API / NXQL Editor)
Finder	• Portal • Engine • Library (nexthink.com)
Collector (TCP connection)	• Engine • Portal (assignment)
Portal	• Engine • Active Directory • SMTP
Engine	• Application Library (nexthink.com) • Automatic Updates (nexthink.com) • Mobile Bridge

Viewing the certificates in the Appliance from the Web Console

To quickly view the digital certificates of the server components that are in place in a particular Nexthink Appliance:

1. Log in to the Web Console of the intended Appliance from a web browser:
   `https://<Appliance_address>:99`
2. Select the **APPLIANCE** tab at the top of the Web Console.
3. Click **Certificates** in the left-hand side menu.

The applicable certificates are arranged in a table:

Replacing the certificates of the server components

In this section, learn how to replace the certificates in the server components of Nexthink that the different client applications may use to authenticate them. Let us suppose that you have obtained the following set of certificates from a publicly-trusted CA or that you have generated them yourself:

- Root certificate of the CA:
  `root.crt`
- Optional bundle of intermediate certificates (provided by the CA when the server certificates are not directly signed by the private key associated to the root certificate, which is customary):
  `intermediate.crt`
- Server certificate for the master Appliance (Portal):
  `master.crt`
- Server certificates for the slave Appliances (Engines):
  `slave.crt`

For client applications to effectively authenticate your Appliances, a name in the *Subject Alternative Name* of the master and slave certificates must match the external DNS names of the Appliances hosting the Portal and the Engines, respectively. If you also need certificate validation internally between Appliances (namely, for the connections from the Portal to the Engines), configure your corporate DNS so that the internal DNS names of the Appliances match their external names. In that way, you will not need to issue server certificates with multiple subjects.

In addition, you must own the private keys associated to the server certificates of the master and slave Appliances:

- Private key associated to the server certificate of the master Appliance:
  `master.key`
- Private key associated to the server certificate of the slave Appliance:
  `slave.key`

Although we give one generic name to the files that hold the server certificates of the slave Appliances (`slave.crt`) and their associated keys (`slave.key`), note that you will usually need a different server certificate and private key for each one of your slave Appliances, because each slave Appliance is identified as a different subject in the certificate (they have a different DNS name). The exception to this rule is if you use multiple-subject certificates.

The certificates and cryptographic keys described here are all assumed to be in **PEM format**, which is basically a Base64 (text) encoding of the binary DER format. Note that the extension of the certificate file (.crt, .cer, .pem) is not relevant, but the actual content of the file is determinant. To follow the instructions below, convert the certificate files to PEM format if this is not already the case. For instance, to convert a root certificate in DER format to PEM with the help of the `openssl` command line tool, type in:

```
openssl x509 -inform DER -outform PEM -in root.der -out root.crt
```

If you obtained your PEM certificate files from a Windows system, Nexthink strongly recommends you to convert them to Unix text format to avoid problems when chaining certificates. Once you have copied the certificate files generated in Windows to the appliance, run the following command on each one of them (substitute `win-cert.crt` and `unix-cert.crt` for the appropriate names):

```
tr -d '\r' < win-cert.crt > unix-cert.crt
```

Replacing the server certificates in the master Appliance

The supplied certificate replaces, at the same time, the default server certificates of:

- The Web Console in the master Appliance
- The Portal (which includes the Collector communication with the Portal for rule-based assignment, when using TCP port 443).

If Collectors communicate with the Portal through a custom TCP port, the default certificate that protects this channel is generated during the federation of the Appliances and it is secure enough. Optionally replace this certificate separately.

To replace the server certificates of the Web Console and of the Portal in the master Appliance:

1. Log in to the Web Console of the master Appliance.
2. In the **APPLIANCE** tab, select **Certificates** from the left-hand side menu.
3. Under the section **Certificates**, find the list of installed certificates for the Web Console and the Portal (including Collector communication, when using TCP port 443).
4. Click the button **REPLACE CERTIFICATES** below the list of certificates. A dialog shows up:
    1. Placed to the right of the word **Certificate**, click the button **CHOOSE FILE**.
    2. Select the file **master.crt** from the dialog that shows up.
    3. Placed to the right of the words **Private key**, click the button **CHOOSE FILE**.
    4. Select the file **master.key** from the dialog that shows up.
    - Optional: Tick **Use an intermediate or chain certificate** if you have a bundle of intermediate certificates.
        1. Click the button **CHOOSE FILE** that shows up when you tick the previous option.
        2. Select the file **intermediate.crt** from the dialog that shows up.
5. Click the button **Replace certificates for Portal, Collector communication (port 443) and Web Console**.

If you installed Nexthink in a single Appliance, the dialog to replace the certificates in the master Appliance also include the slave (Engine) Appliance:

Optionally replace the certificate for Collector communication, when using a custom TCP port (not the default TCP 443):

1. Log in to the Web Console of the master Appliance.
2. In the **APPLIANCE** tab, select **Certificates** from the left-hand side menu.
3. Under the section **Certificates used for the Collector custom TCP port**, find the installed certificate for the communication of the Collector when using a custom TCP port.
4. Click the button **REPLACE CERTIFICATE** below the certificate. A dialog shows up.

    1. Placed to the right of the word **Certificate**, click the button **CHOOSE FILE**.
    2. Select the file **master.crt** from the dialog that shows up.
    3. Placed to the right of the words **Private key**, click the button **CHOOSE FILE**.
    4. Select the file **master.key** from the dialog that shows up.
    - Optional: Tick **Use an intermediate or chain certificate** if you have a bundle of intermediate certificates.
        1. Click the button **CHOOSE FILE** that shows up when you tick the previous option.
        2. Select the file **intermediate.crt** from the dialog that shows up.
5. Click the button **Replace the certificate for the Collector communication on custom TCP port**.

Replacing the server certificates in the slave Appliance

The supplied certificate replaces, at the same time, the default server certificates of:

- The Web Console in the slave Appliance
- The Engine
    - The connection with the Finder and the Portal, as well as the Web API (NXQL).
- The Collector communication when using the default TCP port 443.

If Collectors communicate with the Engine through a custom TCP port, the default certificate that protects this channel is generated during the federation of the Appliances and it is secure enough. To replace it, follow the same procedure as shown above for the master Appliance, but from the Web Console of each slave Appliance.

If you intend to replace the certificate that secures the TCP communication of the Collectors with the Engine:

- Replace the default certificates **after** federating the slave Appliance.

- Leave **empty** the root certificate field when installing the Collector (or generating the Collector installer).

  Distribute instead the root certificate (**root.crt**) to the Trusted Root Certification Authorities certificate store of the Windows devices where the Collector is being installed. This step may not be necessary if the root certificate is from a publicly trusted CA and, therefore, already included in the Windows store.

To replace the server certificates of the Web Console and of the Engine in the slave Appliance:

1. Log in to the Web Console of the slave Appliance.
2. In the **APPLIANCE** tab, select **Certificates** from the left-hand side menu.

3. Under the section **Certificates**, find the list of installed certificates for the Web Console and the Engine (including Finder and Web API) and the Collector communication when using the default TCP port 443.
4. Click the button **REPLACE CERTIFICATES** below the list of certificates. A dialog shows up:
5. Under the section **Replace certificates**:
    1. Placed to the right of the word **Certificate**, click the button **CHOOSE FILE**.
    2. Select the file **slave.crt** from the dialog that shows up.
    3. Placed to the right of the words **Private key**, click the button **CHOOSE FILE**.
    4. Select the file **slave.key** from the dialog that shows up.
    - Optional: Tick **Use an intermediate or chain certificate** if you have a bundle of intermediate certificates.
        1. Click the button **CHOOSE FILE** that shows up when you tick the previous option.
        2. Select the file **intermediate.crt** from the dialog that shows up.
6. Click the button **Replace certificates for Engine, Collector communication (port 443) and Web Console** .

If you replaced the certificate that secures the TCP connection between the Collectors and the Engine (which is mandatory for Collector on TCP port 443), leave empty the root certificate field when installing the Collector. The Collector uses then the Trusted Root Certification Authorities certificate store to validate its TCP connection with the Engine. Note that you replace only the certificates in the slave Appliance and not the Customer Key. The Engine still uses the same Customer Key previously transferred from the master Appliance during the federation process to identify the Collector:

Restoring the default certificates

If the replacement of the default certificates fails or presents any issue (for instance, if the uploaded certificates are not trusted by the devices in which the Collector is installed), it is possible to sort back to the default (or previous) certificates, which are automatically backed up in the Appliances.

To list the backed up certificates of a master or slave Appliance:

1. Log in to the CLI of the Appliance.
2. List the backed up certificates with the following command:
   ```
   ll /var/nexthink/console/certificates/backup/
   ```
3. Choose the set of certificates from the list that you want to restore. The saved certificates are sorted by date and component name.

To restore the server certificate of the Web Console in a master or a slave Appliance:

1. Optional: If you used a chain of intermediate certificates, copy it to the configuration folder of the Web Console:
   ```
   sudo cp -p
   /var/nexthink/console/certificates/backup/xxxxxxxx-console-intermediate.crt
   /var/nexthink/console/etc/intermediate.crt
   ```
   1. Ensure that the file `/var/nexthink/console/etc/lighttpd-console.conf` contains the line `ssl.ca-file="/var/nexthink/console/etc/intermediate.crt"`.
2. Restore the server certificate of the Web Console:
   ```
   sudo cp -p
   /var/nexthink/console/certificates/backup/xxxxxxxx-console-certificate.pem
   /var/nexthink/console/etc/certificate.pem
   ```
3. Restart the Web Console:
   ```
   sudo systemctl restart nxconsole
   ```

To restore the certificates related to the Portal in a master Appliance:

1. Restore the server certificate of the reverse proxy:
   ```
   sudo cp -p
   /var/nexthink/console/certificates/backup/xxxxxxxx-portal-nginx.crt
   /var/nexthink/nxnginx/ssl/nginx.crt
   ```
2. Restore the private key of the reverse proxy:
   ```
   sudo cp -p
   /var/nexthink/console/certificates/backup/xxxxxxxx-portal-nginx.key
   /var/nexthink/nxnginx/ssl/nginx.key
   ```
3. Restart the reverse proxy:
   ```
   $ sudo systemctl restart nginx
   ```

To restore the certificates related to the Engine in a slave Appliance:

1. Optional: If you used a chain of intermediate certificates, copy it to the configuration folder of the Engine:
   ```
   sudo cp -p
   /var/nexthink/console/certificates/backup/xxxxxxxx-engine-intermediate.crt
   ```

```
/var/nexthink/engine/common/etc/intermediate.crt
```

1. Ensure that the file `/var/nexthink/engine/01/etc/nxengine.xml` contains the following line inside the `ssl` tag:
   ```
   <certificate_chain_file>/var/nexthink/engine/common/etc/interme
   ```
2. Restore the server certificate of the Engine:
   ```
   sudo cp -p
   /var/nexthink/console/certificates/backup/xxxxxxxx-engine-certificate.pem
   /var/nexthink/engine/common/etc/certificate.pem
   ```
3. Restore the private key of the Engine:
   ```
   sudo cp -p
   /var/nexthink/console/certificates/backup/xxxxxxxx-engine-key.pem
   /var/nexthink/engine/common/etc/key.pem
   ```
4. Restart the Engine:
   ```
   sudo systemctl restart nxengine@1
   ```

Optional: Restore the default certificate for the custom TCP connection with the Collectors in both the master and slave Appliances.

1. Log in to the Web Console.
2. In the **APPLIANCE** tab, select **Certificates** from the left-hand side menu.
3. Under the section **Certificates used for the Collector custom TCP port**, click the button **RESET TO DEFAULT CERTIFICATE** at the bottom:

Importing CA certificates into client components

When behaving as client applications, Nexthink components need access to the root certificate of the CA that signed the server certificates to be able to authenticate the server components. Learn here how to import the root certificate and, in some cases, the bundle of intermediate certificates. Therefore, we assume that you have access to the following files:

- Root certificate of the CA:
  ```
  root.crt
  ```
- Optional bundle of intermediate certificates (provided by the CA when the server certificates are not directly signed by the private key associated to the root certificate, which is customary):
  ```
  intermediate.crt
  ```

Importing CA certificates into Windows for the Collector and the Finder

To validate servers, the Finder looks for root certificates in the Windows Trusted Root Certification Authorities store. For its part, the Collector resorts to the same certificate store when the **Root CA** certificate field is left empty during its installation; that is, when you do not use the default *ad hoc* PKI of federation. The store includes by default the root certificates of all the CAs trusted by Microsoft. If you got your server certificates from a publicly-trusted CA, its root certificate is most probably already in the list. If you acted as your own CA to generate your server certificates (that is, if you self-issued the server certificates), add the root certificate to the store.

To add the root certificate to the Trusted Root Certification Authorities store (Windows 10):

1. Log in to the Windows 10 device as a user with administrator rights.
2. Download the certificate file `root.crt` to the device.
3. Type **WinKey+R** to open the Run dialog.
4. Type in *certlm.msc* and press **OK**.
5. Click **Yes** in the dialog that shows up to allow the program make changes to your computer.
6. Right-click **Trusted Root Certification Authorities** and select **All Tasks** > **Import...**.
7. The **Certificate Import Wizard** starts. Click **Next**.
8. Click **Browse** and select the `root.crt` file.
9. Click **Next**.
10. In the dialog **Place all certificates in the following store**, click **Next** to accept the proposed certificate store (**Trusted Root Certification Authorities**).
11. Verify the certificate to be imported and click **Finish**.

After importing the root certificate, the Finder is able to connect to the Portal without prompting any certificate error and Collectors installed with an unspecified root certificate can use the TCP channel to communicate with the Engine. If the server certificate in the Engine was signed using the key of an intermediate certificate, the connection of the Finder with the Engine will however issue the message **The security certificate of Nexthink Engine could not be validated**. This situation happens because the Engine does not currently manage intermediate certificates for its connections with the Finder, the Portal, and the Web API. To solve this, either repeat the previous procedure and import the file `intermediate.crt` into the **Intermediate Certification Authorities** store or ignore the message and validation altogether by clicking **Continue anyway** in the dialog.

Importing CA certificates into macOS for the Collector

Similarly to the Windows Collector, if you replaced the default PKI certificates generated during federation and you left empty the **Root CA** field when installing the Collector, the Mac Collector will look for a valid root certificate in the **Keychains** store.

If your server certificates were signed by a publicly-trusted CA, the root certificate is most probably already present in the Keychains. Otherwise, add the root certificate to the list of trusted certificates as follows:

1. Log in to the macOS device as a user with administrator rights.
2. Download the certificate file `root.crt` to the device.
3. Run the app **Finder > Applications > Utilities > Keychain Access**.
4. Select **System** on the left-hand side panel, under **Keychains**.
5. Drag the file `root.crt` and drop it on to the list of keychain items on the right-hand side of the window. A dialog shows up asking for your password to modify the keychain.
    1. Type in your password.
    2. Click **Modify keychain**. The root certificate appears as a new item with a red cross (X) on the left, meaning that it is still untrusted.
6. Double-click the new item with the red X that represents the imported root certificate. A dialog to modify the certificate settings shows up.
    1. Click **Trust** to expand the section.
    2. In the first entry of the list **When using this certificate**, select **Always trust**. All other entries change automatically to the same setting.
    3. Close the dialog. Again the system asks for your password to validate the modification.
        1. Type in your password.
        2. Click **Update Settings**. The red X disappears from the list item.
7. Exit **Keychain Access**.

This process can be automated via MDM.

Importing CA certificates for the Portal

When behaving as a client component, the Portal uses the default keystore of the JDK installed in the Appliance to validate the server certificates that it receives:

- `/usr/java/default/jre/lib/security/cacerts`

If your server components use certificates signed by a generally trusted Certification Authority (CA), you do not need to import the certificates into the Portal, because they will already reside in the keystore. On the other hand, if you are securing the connection of the Portal to your server components with self-issued certificates, import the CA certificates into the Portal with the help of a utility written for this purpose.

For instance, to import the CA certificates that were used to generate the server certificate of the Engine into the Portal:

1. Log in to the CLI of the master Appliance.
2. Copy the appropriate certificates and keys to the home directory of the nexthink account in the master Appliance by using your favorite SCP tool.
    1. Copy the root certificate `root.crt`.
    2. If necessary, copy the bundle of intermediate certificates `intermediate.crt`. Installing intermediate certificates is usually not mandatory, but you may need to install it if the server component is not able to provide them (the Engine can do it since V6.10).
3. Stop the Portal:
    ```
    sudo systemctl stop nxportal
    ```
4. Import the root certificate into the keystore:
    ```
    sudo sh /var/nexthink/portal/security/import_certificate.sh
    \
    -alias root_engine -file root.crt \
    -storepass changeit
    ```
5. If necessary, import the bundle of intermediate certificates into the keystore:
    ```
    sudo sh /var/nexthink/portal/security/import_certificate.sh
    \
    -alias inter -file intermediate.crt \
    -storepass changeit
    ```
6. Restart the Portal
    ```
    sudo systemctl start nxportal
    ```

The Engine presents the same server certificate to the Portal as to other client applications. Since the subject of this server certificate must be set to the external DNS name of the Engine, but the Portal connects to the Engine through it internal name, you either need to have the same internal and external DNS name for the Engine or a multiple-subject certificate.

The **-alias** option lets you identify the certificates that you import. For different certificates, you must choose an alias that is unique within the same keystore (`root_engine` and `inter_engine` in our example). Trying to import another certificate with the same alias results in an error. To reuse an alias, delete the

previous certificate from the keystore:

1. Log in to the CLI of the master Appliance.
2. Delete the certificate identified by the alias:
```
sudo /usr/java/default/jre/bin/keytool \
-delete -alias root_engine \
-storepass changeit \
-keystore /usr/java/default/jre/lib/security/cacerts
```

Note that the default password for the JDK keystore is **changeit** (argument to the option **-storepass**). To actually change the password of the keystore:

1. Log in to the CLI of the master Appliance.
2. Ask for password modification:
```
sudo /usr/java/default/jre/bin/keytool -storepasswd \
-keystore /usr/java/default/jre/lib/security/cacerts
```
3. You are prompted to type in the current password for the keystore and to type in twice the new password.

The Portal may be instructed to ignore certificate problems when communicating with server components. By default, the Portal ignores certificate errors when connecting to the Engine, but not when connecting to the mail or the LDAP servers.

For the Engine and the LDAP server components, there is an entry in the configuration file of the Portal (**/var/nexthink/portal/conf/portal.conf**) that controls certificate validation. To enforce validation, set the value of each entry to false. Find below the entries which correspond to the Engine and the LDAP server, with their default values:

```
globalconfig.portal.dispatcher.engine-ssl-ignore-certificate-problems=true
globalconfig.ldap.skip-ssl-certificate-validations=false
```

Importing CA certificates for the Engine

When behaving as a client component, the Engine uses the CA certificates listed in the following file to validate the server certificates that it receives:

- `/var/nexthink/engine/common/etc/ca-bundle.crt`

To add a new CA root certificate to this file, just append the certificate file to it:

1. Log in to the CLI of the slave Appliance.
2. Copy the root certificate `root.crt` to the home directory of the nexthink

account in the slave Appliance by using your favorite SCP tool.
3. Append the root certificate to the bundle of certificates from publicly-trusted CAs:

```
cat root.crt | sudo tee -a \
/var/nexthink/engine/common/etc/ca-bundle.crt > /dev/null
```

The operations described in this article should only be performed by a Nexthink Engineer or a Nexthink Certified Partner.

If you need help or assistance, please contact your Nexthink Certified Partner.
Related tasks

- Logging in to the CLI
- Federating your Appliances
- Nxtcfg - Collector configuration tool

Hierarchizing your infrastructure

Overview

To manage the complexity of a big company or organization, you usually divide it into a set of hierarchical levels. You can build hierarchies according to different criteria. For instance, if a company is spread throughout several countries, it is possible to group parts of the organization according to their geographical location. You can then arrange the locations in a hierarchy of cities, regions, countries and even continents. Other possibility is to divide the company into functional departments, such as Research and Development, Human Resources, etc. and then divide each department into units, each unit into sub-units and so forth, until you are satisfied with the decomposition. Several hierarchies may be built for the same company and coexist within it at the same time.

Nexthink hierarchies let you arrange the devices in your IT infrastructure in a way that reflects the structure of your company, with the advantage of getting results from Nexthink that directly map into the existing structure. For instance, you can quickly detect if a problem impacted every device in your company or just the computers in the department of Human Resources. Break down results from investigations, dashboard widgets and IT services according to the defined hierarchies. In addition, use hierarchies to delimit scopes of visibility for users (view domains) and administration rights over parts of the company (administration domains).

Example of a hierarchy built with mixed functional and location criteria

Specifying entities

To organize your set of devices into a hierarchy, group your devices by *entities*. Entities are logical groups of devices that make up the first level of all hierarchies. Each device belongs to at most one entity, whose name is displayed in the special device field **Entity**.

Starting from V6.19, there are two ways to assign entities to each device:

- Manual assignment mode (legacy).
- Rule-based Collector assignment mode, which assigns devices to Engines in addition to entities.

In both cases, you write a Comma Separated Value (CSV) file that specifies the names of entities and how to assign each entity to groups of devices. The format of the CSV file for manual assignment is described in the next section. See the section about rule-based Collector assignment for more information about the format of the CSV file in that mode of operation. Once rule-based assignment is activated, manual mode is overridden.

Assigning entities in manual mode

To assign entities to sets of devices:

1. Log in to the Portal as a central administrator.
2. In the **ADMINISTRATION** menu, select **Hierarchies** under **SYSTEM CONFIGURATION**.
3. Click the pencil icon in the top left corner of the **Hierarchies** panel, next to the total number of entities.
4. In the dialog that shows up, click the button **Choose file** to pick the **CSV file to import**. Once you have chosen the file, the dialog displays a **Preview** field below that shows how your CSV file will be imported. If columns are not correctly detected, modify the fields described in the next

steps.

5. Specify the character that separates the columns in the CSV file in the **Delimiter** field. By default, the delimiter is the comma character.
6. Choose the text encoding of the CSV file in the field **Encoding**. If you choose a UTF encoding, do not use an editor that creates a BOM header at the beginning of the file (e.g. Notepad). You can select one of the following text encodings:
    - ISO-8859-1 (Latin 1).
    - UTF-8.
    - UTF-16.
7. In the field **Text qualifier**, specify the type of quotes that you used to delimit text in the CSV file, if necessary.
8. Click **Ok** to import the CSV file and modify the entities. A summary of the changes carried out appears in a new dialog.
9. Click **Ok** again in the summary dialog to finish the import.

Format of the CSV file for defining entities in manual mode

The CSV file that defines the entities must have five columns per line, or six columns if you add an optional comment as the last item. Either all of the lines or none must provide a comment, although the comment of a line may be an empty string in the former case. Each line in the CSV file defines an entity that is assigned to a set of devices in a particular Engine. The entities that you specify here are the basic building blocks of the hierarchies that you will build later; therefore, they are placed at the lowest level of the hierarchies, called the *Entity* level. The Entity field of devices gets a value according to the specified rules. Each line in the CSV file holds the following items, ordered below by their position:

1. Engine name
2. Entity name
3. Entity assignment rule
4. Type of rule
5. Platform
6. Optional comment

The rules for assigning entities to devices in the CSV file are simpler than the rules for categories that you can specify in the Finder. The CSV file supports four types of rules. Choose one of them in the column **Type of rule**. Each type of rule refers to the field of the device that must match the pattern specified in the column **Entity assignment rule** for the device to belong to the entity. See below the list of types of rules and their corresponding device field:

ip
: Last IP address of the device.

name
: The name of the device.

dn
: The distinguished name of the device (an Active Directory value).

collector_tag
: The custom tag that identifies the Collector installed in the device.

The format of the pattern in the column **Entity assigment rule** depends on the type of rule that you specified to select devices:

- For an **ip** rule, specify either a single IP address in dot-decimal notation, for example 192.168.0.10, or a subnet in CIDR notation, for example 192.168.0.10/24.
- For a **name** or **dn** rule, give the name or the distinguished name of the device. You can use the wildcards **?** and ***** as substitutes for one or several characters.
- For a **collector_tag** rule, indicate the exact number used to tag the installation. Note that several Collectors can be installed in different devices using the same tag.

In the fifth column (the **Platform**), specify the kind of devices to which the rule applies. You can set it to ***** for the rule to apply to every kind of device. Otherwise, you can use the values **windows**, **mac_os** and **mobile** for the rule to apply only to Windows, Mac or mobile devices, respectively. If you want to apply a rule to a couple of platforms only, repeat the same rule using different platform values.

In a fresh installation of Nexthink, the default rule for assigning entities is the following:

- "Nexthink";"other";"*";"name";"*";"Automatically generated default entity"

That is, the default rule assigns the entity *other* to every device of the Engine called **Nexthink**, which is the default name of the Engine. From there, you can replace the content of the entity rules as explained above.

Devices that do not match any entity assignment rule are assigned the empty entity, which is represented by a dash sign (**-**) in both the Finder and the Portal.

Priority of the entity assignment rules

The order of the definitions of entities in the CSV file determines the priority of their assignment rules. Devices that match the rules of several entities are assigned to that entity whose rule appears first in the CSV file.

This is similar to the auto-tagging order of keywords when editing categories in the Finder.

One entity per Engine limitation

A single entity cannot spread among different Engines. In the CSV file, you cannot have the entity **GE** on two Engines, so the following is not valid:

```
"Engine1";"GE";"172.16.1.0/24";"ip"; "*"
"Engine2";"GE";"172.16.4.0/24";"ip"; "*"
```

Limit on the number of rules per entity

The maximum number of rules that you can specify in the CSV file for a single entity is 1000.

If more than 1000 rules are specified for one entity, the rules for that particular entity are invalid and thus ignored. All the devices that do not match any subsequent valid rule of another entity are assigned the empty entity, represented by the dash sign (**-**).

Creating a hierarchy

Once you have specified the entities that form the base of the hierarchies, you can start building your own hierarchies by adding new levels on top of the entities.

To create a new hierarchy:

1. Log in to the Portal as a central administrator.
2. In the **ADMINISTRATION** menu, select **Hierarchies** under **SYSTEM CONFIGURATION**.
3. Click the plus sign the icons displayed at the top right of the panel. The dialog to add a new hierarchy shows up.
4. Type in a name for the new hierarchy in the **Name** field.
5. Add levels to your hierarchy. See the next section for details.

6. In the choice group **Base hierarchy on**, choose between **all Engines** to create a global hierarchy or **selected Engines** to create a hierarchy that applies to a set of Engines. Note that if you create a hierarchy that applies to a set of selected Engines, you can later promote it to a global hierarchy. On the other hand, if you create a global hierarchy, it is impossible to downgrade it to a hierarchy based on a group of selected Engines.
    - If you decided to create the hierarchy for a group of **selected Engines**, select your Engines as follows:
        1. Click the **Add** button below the table of Engines. A small dialog with a list of Engines shows up.
        2. Pick an Engine from the list and click **Ok**. Repeat from the previous step until you have selected all the Engines that you wish. The selected Engines are displayed in the table.
7. Click **Ok** to finish the creation of the hierarchy.

Adding hierarchy levels

The levels of the hierarchy indicate the depth of the tree that graphically represents the hierarchy. In the example figure of the hierarchy above, there are three levels defined:

1. Entity level: The lowest level in the hierarchy. It is composed of the names of entities. Each name represents the set of the devices assigned to the entity, according to the rules in the CSV file.
2. Region level: Groups entities into different regions named after the four cardinal points (North, South, East and West).
3. Department level: Divides the company into several departments that are located in one or several regions.

The Entity level is mandatory for all hierarchies. When you create a new hierarchy, you add levels on top of the Entity level. The root node of the hierarchy is always at the central administration level, which is never defined explicitly.

To add levels to a hierarchy from the dialog to create a new hierarchy:

1. Click the **Add** button below the table of levels. A small dialog to edit the level shows up.
2. Enter the name of the level.
3. Click **Ok** to add the level to the table.
4. Repeat from the first step to create as many levels as you need.
5. Optional: Move the created levels up or down in the table by clicking the arrows that appear in the next column, to the right of the name of the level. Note that the Entity level is always the lowest level and that you cannot

move it inside the table.

There is a special level that you can use directly above the Entity level called the Engine level. This level makes a first groupment of entities per Engine. To create the Engine level, click the icon with the small Nexthink logo and the plus sign that is placed to the right of the Entity level in the table of levels of the dialog to create hierarchies. The Engine level is automatically filled by the system, which detects the entities (keywords) that are present in each Engine. For that reason, keywords must not be repeated in different Engines. At the end of the process, a new node is created at the Engine level for each Engine found in your system. Similarly to the Entity level, this level cannot be moved upwards or downwards inside the hierarchy.

To manually create the nodes for the other non-special levels, read the following section.

Building the hierarchy tree by editing the entities

Once you have finished creating a hierarchy and its levels, you need to specify nodes for every level. Nodes in one level are used to group the elements of the level below to form the hierarchy. You add nodes to a level by editing the entities of the hierarchy.

To add nodes to the levels of a hierarchy:

1. In the **Hierarchies** panel, select the entities that you want to group from the **Entity** table. Click the row that represents an entity in the table while holding the **Ctrl** or **Shift** keys down to select multiple entities.
2. Click the button **Edit selected entities** below the **Entity** table. A dialog appears with a set of text fields, where each field holds the name of the node to which the set of selected entities belong. Since this is the first time that you edit the entities, the text fields are displayed empty.
3. Type in node names for every level displayed in the dialog.
4. Click **Ok** to group the selected entities below the specified nodes in the hierarchy.
5. Click the floppy disk icon in the top right part of the **Hierarchies** panel to save your work on hierarchies.

Editing a hierarchy

To edit a hierarchy, click the pencil icon that you see at the top right of the **Hierarchies** panel. The dialogs and options for editing the hierarchy are identical to those used when you created the hierarchy.

When you edit the entities of an existing hierarchy, they may already belong to some of the nodes in the hierarchy. You can see the names of the nodes in the columns of the different levels in the **Entity** table. After selecting a group of entities and clicking the button **Edit selected entities**, you find the names of the nodes in the dialog that displays the levels of the hierarchy for the selected entities:

- If the selected entities belong to only one node at a particular level, the text field for that level displays the name of the node.
- If the selected entities belong to different nodes at a particular level, the text field for that level displays the value **[multiple]**.

With the edition of entities, you can add or remove branches from your hierarchy tree or modify it in any other way you choose. Find below a couple of examples:

Example of creating a branch

Example of moving a branch

Be careful when editing a hierarchy that has been already used for aggregating results or for defining user domains. After the edition of an existing hierarchy, a dialog called **Impact of changes** displays all the elements in the Portal that got their associated domains invalidated because of the changes in the hierarchy. Click **Continue** to carry on with the changes anyway. Alternatively, click **Cancel** to revert the changes or to re-edit the hierarchy for reducing the impact.

If you edit a hierarchy, do not forget to save your changes by clicking the floppy disk icon at the top right of the **Hierarchies** panel.

Cleaning up the hierarchy

Eventually, a hierarchy may be based on entities that are no longer used. A couple of cases may bring up this situation:

- The CSV file that defines the entities got some rules removed.
- All the devices assigned to a particular entity were removed from an Engine.
- An Engine became temporarily or definitively unreachable.

The entities that are no longer in use are not automatically removed from the system. Instead, they are respresented in the **Entity** table with an exclamation mark **!** at the beginning of the row. This indicates that the entity was not present in any Engine. You can redefine the entities and add the corresponding keywords to enable these entities again, or you can remove them if you no longer need them. To erase the unused entities:

1. Click the broom icon in the top right part of the **Hierarchies** panel. A check list of the unused entities shows up.
2. Check the box of every entity that you want to delete.
3. Click the button **Delete selected entities**.

Note that if an entity is removed and then is detected in an Engine, it will appear again in the **Entity** table, though without any values for the nodes up in the hierarchy.

Viewing hierarchies

If you have created multiple hierarchies, the **Hierarchies** panel lets you select the hierarchy that you want to view. Pick the desired hierarchy from the list that is placed as the first element in the top heading of the widget, labeled by the word **Hierarchy**, before the other icons.

To see a graphical representation of your hierarchy, click the **View current hierarchy** button. The Portal opens a new window that displays the nodes of the hierarchy as rounded boxes with their names inside organized in a tree-like structure that shows the defined levels. Depending on your browser and your security settings, you may need to enable the pop-ups for the Portal to open the new window.

Otherwise, once you select a hierarchy, you see the levels of the hierarchy with the list of nodes for each level in the upper part of the panel. In the lower part, you see the **Entity** table, with the names of the entities and the nodes that they

belong to. The entities shown in the entity table are filtered by the nodes that you select in the list of nodes of the hierarchy levels. To view all the entities, select the special keyword **All** from the list of nodes of every level. The keyword **All** means that you want to see the entities of all the nodes at that level.

Additionally, you can select the **Overview** mode. In this mode, you just see a big **Entity** table where the columns include the levels of all the hierarchies at the same time. This mode lets you quickly view all the nodes to which an entity belongs in any of the defined hierarchies.

Renaming levels and nodes

When viewing a particular hierarchy in the **Hierarchies** panel, note that there is a clickable text to the right of every level labeled **(rename)**. This text also appears to the right of the Entity level in the Entity table. Renaming a level as described below has no impact on the computed metrics, it only changes the displayed names when navigating the hierarchy levels in the dashboards.

To rename a level in your hierarchy:

1. Click the **(rename)** word to the right of the level. A small dialog to edit the name of the level shows up.
2. Type the new name for the level. The new name must not conflict with the name of any other level in the hierarchy.
3. Click **Ok** to actually rename the level.

Below the list of nodes of every level, you also find a piece of clickable text labeled **rename node** (except for the nodes of an Engine level, because these have the names of the Engines and you are not allowed to change them). Beware that renaming a node is equivalent to replacing the existing node by a new one, so all the results of the metrics grouped by the renamed node and its descendants are cleared.

To change the name of a node:

1. Select the name of the node inside the list of the level.
2. Click **rename node**. A small dialog to edit the name of the node shows up.
3. Type the new name for the node. The new name must not conflict with the name of any other node in the same level.
4. Click **Ok** to actually rename the node. Only the nodes that are part of the filter to view the hierarchy are renamed (see previous section).

Remember that:

- Renaming nodes clears the results of metrics grouped by hierarchies.
- Renaming levels does not modify any result.

Exporting and importing hierarchies

To backup and restore a hierarchy, you can export it to a CSV file or import it from a CSV file from the **Hierarchies** panel.

To export a hierarchy to a CSV file:

1. Select the hierarchy that you want to export in the list of hierarchies of the widget (the list at the top part labeled **Hierarchy**).
2. Click the icon with the arrow down and the initials **CSV** at the top right part of the widget to download the hierarchy as a CSV file.
3. Follow the instructions of your web browser to save the CSV file in the local filesystem.

To import a hierarchy from a CSV file:

1. Click the icon with the plus sign and the initials **CSV** at the top right part of the widget. The dialog to import the hierarchy shows up.
2. Click on the button **Browse** to select the CSV file to import from your local filesystem. A preview of the CSV to import is displayed according to your import options.
3. For the other options in the dialog, select the semicolon as separator character, UTF-8 as text encoding and the double quotes as text identifier if your file was generated by the Portal. Otherwise, use your own custom settings.
4. Click **Ok** to import the hierarchy.

Deleting a hierarchy

Deleting a hierarchy has a direct impact on all objects that depend on that hierarchy. Be sure to know what you are doing before deleting a established hierarchy. The following may happen when you remove a hierarchy from the system (not an exhaustive list):

- Administrators whose administration domain is based on the hierarchy are not be able to log in to the Portal.
- Objects in a view domain based on the hierarchy are visible to central administrators only.
- User accounts with a view domain based on the hierarchy see nothing because they no longer have access rights.

Related tasks

- Creating categories and keywords
- Assigning Collectors to Engines

Related concepts

- Hierarchy
- Category

Adding users

Overview

Right after installation, the only user that exists in the system is the first and main central administrator or *admin* user. The admin user has unrestricted access to all data available in both the Portal and the Finder. Moreover, the admin user is able to create and modify all kinds of content in the system, including dashboards, investigations, categories, alerts and user accounts.

Incidentally, you may want to give other people the chance to log in to the system and use it without necessarily having all the capabilities of the admin user. The admin user can thus create accounts for other users, restrict their views on the data and limit their ability to alter content. In this section, learn how to add users to the system and control their access to the data recorded.

Prerequisites

Before defining new profiles and users, ensure that you have installed a license for the product. Otherwise, some configuration pages will not show up.

Account update considerations

Beware that changes to accounts and their permissions may not take immediate effect on logged in users.

For users logged in to the Finder or to the Portal, the user keeps the permissions before the change during the session lifetime. For users making use of Web API (NXQL), the old permissions are still in force up to five minutes after the change,

until the Engine synchronizes account information with the Portal.

Defining user roles

The *roles* attributed to a user determine the responsibilities of the user. Depending on their responsibilities, users carry out different tasks to achieve their goals. Roles let you group the items that enable users to execute their assigned tasks. When assigning roles, specify the modules that a user or group of users can see in the Portal, the investigations that they are able to run in the Finder, and the alerts of which they must be aware.

To incorporate items into a role, first create those items either in the Finder or in the Portal. It is not essential to have all the items ready before defining a role. You can start by creating the role with a few items and later edit the role to add the missing items.

To define a new role:

1. Log in to the Portal as administrator .
2. Click the **ADMINISTRATION** drop-down menu at the top of the window.
3. Select the option **Roles** to open the dashboard for editing roles.
4. Click the plus sign at the top right hand side of the dashboard to open the wizard for adding a new role.

Step 1: Adding modules

1. Type in the name of the new role in the **Name** field.
2. Optional: Click **Add module** to add an existing module of the Portal to the role. A dialog to choose the module pops up.
    1. Select a module from the list labeled **Module**.
    2. Click **Add**. The dialog closes and the selected module is added to the **Modules** list of the role.
3. Repeat the previous step to add as many modules as the role needs.
4. Click **Next** to go on with the next step of the wizard.

Step 2: Adding service-based alerts

1. Optional: Click **Add alert** to include service-based alerts to the role. A dialog to specify the alerts pops up.
    1. Select a service-based alert from the list labeled **Alert**.
    2. Optional: Click **yes** in the **Mandatory** section to force the subscription to the alert of all users with the current role. By default, the alert is not mandatory.

3. Click **Ok**.

2. Repeat the previous step to add as many alerts as the role needs.
3. Click **Next**.

Step 3: Adding investigations

1. Optional: Click **Add investigation** to share existing investigations with all users who have the current role assigned. A dialog to specify the investigation pops up.
    1. Export an investigation or a folder of investigations from the Finder to the clipboard.
    2. Paste the contents of the clipboard on the dialog of the wizard.
    3. Click **Add**. The dialog to paste the investigation closes and the investigation is added to the **Investigations** list of the role.
2. Repeat the previous step to add as many investigations as the role needs.

Step 4: Adding one-click investigations

1. Optional: Export a pack with all the one-click investigations that you want to add to the role from the Finder.
    1. Paste the pack of one-click investigations on the dialog of the wizard.
2. Click **Next**.

Step 5: Adding investigation-based alerts

1. Optional: Click **Add alert** to include investigation-based (Finder) alerts to the role. A dialog to specify the alert pops up.
    1. Export an alert or a folder of alerts from the Finder to the clipboard.
    2. Paste the contents of the clipboard on the dialog of the wizard.
    3. Click **Add**. The dialog to paste the alert closes and the alert is added to the **Alerts** list of the role.
        ◊ The syslog notification mechanism of global alerts is local to the Engine where the global alert was created and, therefore, not propagated to other Engines via roles. If you add a global alert with syslog notification enabled to a role, only the email notification mechanism is propagated to the users with that role.
2. Repeat the previous step to add as many alerts as the role needs.
3. Click **Next**.

Step 6: Adding remote actions

This step is available only if you have purchased a Nexthink Act license. Moreover, only the main admin or users with the right to edit remote actions in their profile can assign role-based remote actions to other users.

1. Optional: Click **Add remote action** to assign a remote action to the current role. A dialog shows up.
    1. Select a remote action from the drop-down list. Only remote actions which can be triggered manually are available in the list.
    2. Click **Ok** to add the remote action.
2. Repeat the previous step to add as many remote actions as the role requires.
3. Click **Finish** to end the wizard. The new role is added to the list of the **Roles** dashboard.

Defining user profiles

The *profile* of a user defines the type of user, the access rights of the user to the different domains of a hierarchy (both as a viewer and as administrator, if applicable) and to the functions of the Finder. Moreover, you can associate one or multiple roles to a profile. Thus, users are able to play any of the roles associated to their profile, along with any other possible role that you may additionally assign to them.

Profile types

There are two main types of profiles:

User
: This profile is intended for users that only have the right to view the information; both in the Portal and, optionally, in the Finder. They are able to see only the data that belongs to their view domain (a subset of the available hierarchies), possibly limited by privacy settings as well. Optionally, users can create and publish Portal modules (dashboards).

Central administrator
: Users with a *Central administrator* profile can practically do all that the main admin user does. The difference is that, while the main admin has complete visibility over all the information available, the information that central administrators can see is limited by their privacy settings. Central administrators have the rights to create and manage Portal content, create other user accounts, access all hierarchies, create and modify profiles and hierarchies, control the connections of the Portal to the Engines, and

manage the product license.

In general, an *administrator* is either the main admin user or a user with the central administrator profile.

See here the complete matrix of access rights and permissions.

To create a new profile:

1. Log in to the Portal as administrator.
2. Click the **ADMINISTRATION** drop-down menu at the top of the window.
3. Select the option **Profiles** to open the dashboard for editing profiles.
4. Click the plus sign at the top right hand side of the dashboard to add a new profile. The wizard to add a new profile opens.

Step 1: Choosing the type of account

1. Type in a name for the new profile in the field labeled **Profile name**.
2. Select one of the three types of accounts from the choice **Account type**.
    - Select **User** if the profile is intended for users without administrative tasks.
        - Optional: Uncheck the box **Allow creation of personal dashboards** to prevent users with the current profile from creating their own modules and dashboards. By default, the box is checked, allowing the users to create Portal content.
        - Optional: Check the box **Allow publication of dashboards** to enable users with the current profile to publish their own modules and dashboards, so that others can use them.
    - Select **Central administrator** to create users that can administer the whole system in the same way as the main admin user, except for the fact that you can restrict what they see in their data privacy settings.
3. In the section **Available metrics**, choose the group of metrics that users with the current profile may use to build their own dashboards and see in dashboards created by others:
    - Select **All metrics** for the user to be able to see and use any of the metrics in the system. This option is mandatory if the user must be able to edit metrics (see step 3).
    - Select **Only metrics in roles** for the user to be able to see and user only those metrics which are part of their roles; that is, metrics embedded in the modules added to their roles. This is the only option available if the user has no right to create dashboards.
4. Click **Next** to go on with the next step of the wizard.

Step 2: Set privacy settings, roles and view domain

1. Select the **Data privacy** settings for the profile:
    - **anonymous users, devices, destinations and domains**: user accounts with this profile cannot see the names of users, devices, destinations, or domains.
    - **anonymous users and devices**: user accounts with this profile can see neither the names of users nor of devices.
    - **anonymous users**: user accounts with this profile cannot see the names of users.
    - **none (full access)**: user accounts with this profile have full access to the collected data.
2. Select the roles of the profile by clicking their name in the **Role(s)** list. Use the **Ctrl** key to select several roles at the same time. The investigations, alerts, modules, etc attributed to the selected roles are inherited by the profile.
3. Specify the view domain of the profile for each defined hierarchy. Users with the current profile can only view the objects grouped in the specified domain:
    1. In the **from** field, select the highest level in the hierarchy that belongs to the view domain.
    2. In the **Node** field, either:
        - Choose the top node of the view domain from the available nodes of the level. This node and all the nodes below it belong to the view domain, down to the level specified in the next step.
        - Leave the top node undefined by choosing **--parameter--** from the list. Define the top node of the view domain individually for each user when creating their user account.
    3. In the **to** field, select the lowest level in the hierarchy that belongs to the view domain.
4. Click **Next**.

Step 3: Set Finder access

To let users with the current profile access the Finder and its different features:

1. Check the box **Finder access**.
2. Select the time zone of the user.
3. Optional: Check the box **Allow editing of application and object tags** to let users with the current profile manually modify the tags of objects in the Finder.

4. Optional: Check the box **Allow system configuration** to let users with the current profile edit categories, services, metrics, scores, and global alerts, as well as import and export content, or manually synchronize users and devices with Active Directory. You can only select this option if you gave full access to the profile in the privacy settings of the previous step.
5. Optional: Check the box **Allow editing of remote actions** to let users with the current profile add and modify Nexthink Act scripts. In addition to a Nexthink Act license, this option requires the profile to have full access to data in the privacy settings and an unrestricted view domain in at least one of the defined hierarchies.
6. Optional: Check the box **Allow API of remote actions** to let users with the current profile execute remote actions programmatically through the Nexthink Act API. In addition to a Nexthink Act license, this option requires the profile to have full access to data in the privacy settings and an unrestricted view domain in at least one of the defined hierarchies.
7. Optional: Check the box **Allow editing of campaigns** to let users with the current profile create, modify, and publish campaigns, as well as trigger manually targeted campaigns, to get end-user feedback. This option requires the profile to have full access to data in the privacy settings and an unrestricted view domain in at least one of the defined hierarchies.
8. Optional: Check the box **Allow management of Collectors** to let users with the current profile follow and control the deployment of the Collector from the Finder. Again, you can only select this option if you gave full access to the profile in the privacy settings of the previous step.
9. Set the visibility level of Web & Cloud information for the users with the current profile to either **restricted** or **full** in the list under **Web & Cloud visibility**.
10. Optional: Check the box **Access campaigns trigger API** to let users with the current profile send campaigns programmatically through the Nexthink Engage API. In addition to a Nexthink Engage license, this option requires the profile to have full access to data in the privacy settings and an unrestricted view domain in at least one of the defined hierarchies.
11. Click **Finish** to end the creation of the profile. The profile is added to the list of profiles in the dashboard.

Creating a user

After defining roles and profiles for users, create the user accounts that make use of them. To create user accounts in the Portal, either:

- Create individual user accounts manually.
- Provision user accounts from Active Directory (recommended).

Find below how to manually create a new user account. To learn how to provision user accounts to Nexthink from existing user accounts in Active Directory, see the article on provisioning user accounts from Active Directory.

Nexthink supports both internal and external management of credentials to authenticate users:

Internally managed	Externally managed	
Password based	**SSO**	**Password based**
• Portal stores the credentials	• SAML authentication • Windows authentication	• Active Directory

Because the Finder connects to the Portal, it is the Portal that holds the responsibility of authenticating users. The Portal decides whether to authenticate a user by either internal or external means based on the provided login name for that particular user:

- If the login name includes a @ character, the Portal assumes external authentication of the user. The exact external method is determined by the configuration of the Portal.
- Otherwise, the Portal authenticates the user with the internally stored credentials.

Because the login name of the users provisioned from Active Directory is in the UPN format (*username@domain*), the provisioned users are all authenticated with the help of external mechanisms such as Active Directory or SAML.

To create an individual user account:

1. Log in to the Portal as administrator.
2. Click the **ADMINISTRATION** drop-down menu at the top of the window.
3. Under **ACCOUNT MANAGEMENT**, select the option **Accounts** to open the dashboard for editing accounts.
4. Click the plus sign in the top right corner of the dashboard. The wizard to create a new user account shows up.

Step 1: Setting personal data and profile

1. Type in the name of the user:

- To use internal authentication, type in the desired account (login) name of the user in the field **Username**.
- To externally authenticate users, type in the name of the user in a format that includes the @ character in the field **Username**:
    - In the case of Active Directory or Windows authentication, type in the **sAMAccountName** of the user followed by the @ character and the DNS domain name (e.g. `jwick@example.com`). Note that this field is case sensitive. Therefore, the name of the Nexthink account must exactly match the sAMAccountName name in Active Directory.
    - In the case of SAML authentication, type in the *Name ID* of the user, as returned by the Identity Provider.

2. Type in the complete name of the user in the field **Full name**.
3. Configure the email address for sending notifications to the user in the field **Email address**.
4. Depending on the authentication method applied to the user, enter a password for the user or not:
    - If the user is internally authenticated, type in a password for the user in the field **Password** and retype it in **Password confirmation**.
        - The default minimum password length for an internally managed account is 8 characters (configurable).
    - If the user is externally authenticated, enter no password. The **Password** field becomes uneditable and displays the message **Managed externally** as soon as the **Username** includes an @ character.
5. Select the profile of the user from the list **Profile**. The user gets all the permissions, default content and roles associated to the profile.
    - If the selected profile does not define a particular top node for the view domains of the users with that profile (because the domain is parameterized), select now the top nodes of those domains individually for the current user.
6. Optional: tick the check box **Never automatically sign out this account from Portal when active** if you want to override the session timeout control configured in the Portal and never log out the user from the Portal while active. Note that having a live view on a service keeps a user active even without actual user interaction.
7. Click **Next**.

Step 2: Setting additional roles

1. Optional: If you want the user account to inherit content from one or more roles that do not belong to its assigned profile, select the desired roles

from the list **Additional roles**. Use the **Ctrl** key to select more than one. Note that the list of **Additional roles** does not display roles that already belong to the profile of the user account.
2. Click **Ok** to end the creation of the user account. The account is added to the list of accounts in the dashboard.

Related tasks

- Provisioning user accounts from Active Directory
- Enabling SAML authentication of users
- Enabling Windows authentication of users
- Setting the minimum password length for local accounts
- Controlling session timeouts in the Portal
- Setting up a software license
- Triggering remote actions via their API

Related references

- Access rights and permissions
- Active Directory Authentication

Enabling SAML authentication of users

Overview

Many organizations adopt *Identity and Access Management (IAM)* solutions to facilitate their employees access to business applications through a single corporate login. This technique is known as *Single Sign-On (SSO)* access control. SSO improves the overall security of the organization by reducing the number of passwords that employees have to remember and type in.

The *Security Assertion Markup Language (SAML)* is a standard for exchanging authentication and authorization information securely between parties, namely the *service provider* (an application that needs to authenticate users) and the *identity provider* (a system that issues assertions about user identity). SAML is widely used in organizations to implement SSO.

By leveraging SAML, let Nexthink users comfortably log in to both the Portal and the Finder (the service providers) through your existing corporate SSO solution (the identity provider).

To call the integration APIs, create dedicated Nexthink local accounts instead. Make sure that you set a minimum complexity for the password of local accounts.

The operations described in this article should only be performed by a Nexthink Engineer or a Nexthink Certified Partner.

If you need help or assistance, please contact your Nexthink Certified Partner.

Prerequisites

To enable SAML-based authentication of Nexthink users, you need:

- An IAM corporate solution that supports SAML to provide single sign-on. In this document, find the instructions to configure either *Microsoft Active Directory Federation Services (AD FS)* or *Microsoft Azure Active Directory (Azure AD)*. Other systems may require similar configuration.
    - As SAML identity provider, the IAM solution must support the *HTTP Redirect Binding* for the Portal to be able to initiate SSO.
- A proper external DNS name for the Portal (not an IP address) that matches the address of the Portal for email digests.
- Nexthink users whose authentication is externally managed; that is, whose username includes a @ character (e.g. `jwick@acme.com`). These accounts are automatically created if you provision users just-in-time with SAML.
- HTTP protocol disabled in the Portal, as combining HTTP with HTTPS may cause redirection issues in some web browsers. From the Web Console of the master Appliance, on the **Portal** tab, untick the option **Enable HTTP** under the **General > Parameters** section.

Procedure to enable SAML

To enable SAML authentication for Nexthink:

1. Enable SAML in the configuration file of the Portal and change the default options if needed.
2. Configure Microsoft AD FS, Azure AD, or any other IAM solution, as SAML identity provider.
    1. Add the Portal as *relying party* (in SAML parlance, a service provider is a special case of a relying party that, in addition to receive and accept info from other parties, consumes SAML assertions to provide a service).
    2. Specify how to issue SAML claims for the Portal to consume.

◊ In the examples shown below, the UPN of the users is mapped to the Name ID in SAML to simplify the transition from Active Directory to SAML. Other configurations are possible depending on the format in which you saved the names of the users in the Portal, as long as the names include an @ character. For instance, you can use email addresses that do not necessarily match the UPN as Name IDs.

3. Configure the Portal as a SAML service provider and link it to your identity provider.

Enabling SAML authentication in the Portal

Edit the configuration file of the Portal to enable SAML as authentication method:

1. Log in to the CLI of the appliance that hosts the Portal.
2. Optional: If the Portal has no configuration file yet, that is, if `portal.conf` does not exist in folder `/var/nexthink/portal/conf`, create it by copying the defaults from the sample configuration file:
   ```
   sudo -u nxportal cp
   /var/nexthink/portal/conf/portal.conf.sample \
   /var/nexthink/portal/conf/portal.conf
   ```
3. Edit the configuration file of the Portal:
   ```
   sudo vi /var/nexthink/portal/conf/portal.conf
   ```
4. Add a configuration line to it:
   1. Press **Shift + G** to go to the last line of the file.
   2. Press **o** to add a new line.
   3. Type in the following line:
      ```
      globalconfig.saml.enabled = true
      ```
   4. Press **Esc** and type in the following colon command to save changes an exit:
      ```
      :wq
      ```
5. Restart the Portal:
   ```
   sudo systemctl restart nxportal
   ```

To troubleshoot SAML authentication, try changing the following advanced options in the configuration file of the Portal in the same way as shown above for the option to enable SAML. Read the error logs of the Portal in `/var/nexthink/portal/log/*.err` to find out possible causes.

Option	Default value
globalconfig.saml.strict	false
globalconfig.saml.validate_current_url	false

globalconfig.saml.want assertions encrypted	true
globalconfig.saml.want assertions signed	true
globalconfig.saml.signature algorithm	"http://www.w3.org/2000/09/xmldsig#rsa sha256"
globalconfig.saml.want-messages-signed	true

Enabling SAML strict mode

When enabling **strict** mode for improved security, set the value of **validate current url** depending on your identity provider. The following combinations have been tested:

Identity provider (IdP)	validate current url
Azure AD	true
Microsoft AD FS	true
OneLogin	false

In some cases setting the value of **globalconfig.saml.want-messages-signed** to **false** allows proper SAML authentication. That parameter will instruct the Portal not to require a signed returned message from the IdP.

Configuring Microsoft AD FS as identity provider

As a reference, this example configuration of AD FS shows how to use the UPN of users in the email format as the SAML Name Identifier for the Portal.

To configure Microsoft AD FS as SAML identity provider for the Portal:

1. Log in to the Windows Server machine that runs AD FS as administrator.
2. Open AD FS management console.
3. Under **Actions**, select **Add Relying Party Trust...**. The wizard to add a trusted relying party (in this case, the Nexthink Portal as service provider) shows up.
    1. On the **Welcome** step, choose a **Claims aware** relying party.
    2. Click **Start**.
    3. On the **Select Data Source** step, select the option **Import data about the relying party from a file**.
        1. Open a web browser on the following address to download the metadata file from the Portal (replace `<Portal>` by the external DNS name of your Portal):
            `https://<Portal>/saml/metadata`
        2. Save the file `portal-sp-metadata.xml`, as proposed by the web browser.
        3. Close the web browser to return to the wizard dialog.

4. Click **Browse...** to specify the location of the file with the Portal metadata. A file dialog shows up.
5. Select the file just downloaded from the Portal and click **Open**. The file dialog closes.
6. Click **Next >**.
7. On the **Specify Display Name** step, type in a suitable name for the relying party (e.g. *Nexthink Portal*).
8. Optional: Type in any additional information about the relying party under **Notes**.
9. Click **Next >** repeatedly to skip the rest of steps in the wizard until you reach the **Finish** step.
10. Click **Finish** to complete the addition of the Portal as a trusted relying party. The wizard closes.

4. In the left-hand side tree, click **Relying Party Trusts** to get the list of trusted relying parties.
5. Right-click the trusted relying party entry that represents the Nexthink Portal just added.
6. From the context menu, select the entry to edit the policy for issuing claims:
   - In Windows Server 2016, select **Edit Claim Issuance Policy...**.
   - In Windows Server 2012, select **Edit Claim Rules...**.
7. In the **Issuance Transform Rules** tab, click **Add rule...** to map the UPN of the user to the Name ID, which the Portal matches against the username field for authenticating . The wizard to add a new transform rule for claim issuance shows up.
   1. On the **Choose Rule Type** step, select **Send LDAP Attributes as Claims** under **Claim rule template**.
   2. Click **Next >**.
   3. On the **Configure Claim Rule** step, provide the following information:
      - Under **Claim rule name**, type in a suitable name for the rule (e.g. *Map UPN to Name ID*).
      - Under **Attribute store**, select **Active Directory**.
      - Under **Mapping of LDAP attributes to outgoing claim types**, select **User-Principal-Name** as **LDAP Attribute** and **Name ID** as **Outgoing Claim Type**.
   4. Click **Finish**
8. Back to the **Issuance Transform Rules** tab, click again **Add rule...** to add a new rule to send the UPN as Name ID in the email format. The wizard to add a new transform rule shows up once more.
   1. On the **Choose Rule Type** step, select **Transform an Incoming Claim** under **Claim rule template**.
   2. Click **Next >**.

3. On the **Configure Claim Rule** step, provide the following information:
    ◊ Under **Claim rule name**, type in a suitable name for the rule (e.g. *UPN as Name ID in email format*).
    ◊ As **Incoming claim type**, select **UPN**.
    ◊ As **Outgoing claim type**, select **Name ID**.
    ◊ As **Outgoing name ID format**, select **Email**.
    ◊ Leave checked the option **Pass through all claim values**.
4. Click **Finish**.
5. Click **OK** to close the dialog to edit the claim rules.

9. Right-click again the trusted relying party entry that represents the Nexthink Portal.
10. Select **Properties** from the menu. The dialog to watch and modify the properties of the relying party shows up.
    1. Under the **Advanced** tab, select **SHA-256** as **Secure hash algorithm**.
    2. Click **OK** to close the properties dialog.

Configuring Azure AD as identity provider

To configure Azure AD as a SAML identity provider for the Portal:

1. Log in to Azure from your web browser https://portal.azure.com.
2. Click **Azure Active Directory** on the left-hand side panel.
3. Under **Manage**, select **Enterprise applications**.
4. Click the **New application** button preceded by plus sign at the top of the window.
5. Under **Add an application**, choose the **Non-gallery application** tile.
6. On the panel **Add your own application** that appears to the right, type in the name of the application (for instance, *Nexthink Portal*) inside the **Name** field.
7. Click the **Add** button at the bottom of the panel.
8. On the left sidebar of the Enterprise application that you have just created, under **Manage**, select **Single sign-on** .
9. In the page **Select a single-sign on method**, choose the **SAML** tile. The page **Set up Single Sign-On with SAML** shows up.
10. Open a new tab in the web browser.
    1. Type in the following address to download the metadata file from the Portal (replace `<Portal>` by the external DNS name of your Portal):
    `https://<Portal>/saml/metadata`
    2. Save the file `portal-sp-metadata.xml`, as proposed by the web browser.

3. Close the tab to return to the page **Set up Single Sign-On with SAML**.

11. Click the button **Upload metadata file** at the top left corner of the page.
12. Select the metadata file `portal-sp-metadata.xml` that you have just downloaded from the Portal.
13. Click the pencil icon at the top right corner of the second tile to edit the **User Attributes & Claims**. The page to edit the claims appears.
    1. Ensure that the **Name identifier value** (that is, the Name ID returned by Azure AD) is the user principal name (UPN) in the email format:
        ◊ **user.userprincipalname [nameid-format:emailAddress]**
    2. Close the **User Attributes & Claims** page.
14. On the third tile of the page **SAML Signing Certificate**, click the **Download** link associated to the last entry: **Federation Metadata XML**. You will use this file later to link the Portal to Azure AD as its SAML identity provider.
15. Click the pencil icon to the right of the same third tile to edit the **SAML Signing Certificate**:
    1. Verify that SAML assertions are signed using SHA-256 as signing algorithm. On the page that shows up, you should read:
        ◊ Signing option **Sign SAML assertion**
        ◊ Signing algorithm **SHA-256**
    2. Close the **SAML Signing Certificate** page.
16. Back to the left sidebar of the Enterprise application that represents the Portal, under **Manage**, select **Users and groups**.
17. Click the button **Add user** to add users or groups that can log in to the Portal via SAML authentication.

Configuring a generic SAML identity provider

Although it is not possible to detail the configuration instructions for every SAML identity provider available in the market, the procedure to configure a compliant provider should not differ significantly from the two reference examples shown above.

When manually configuring a generic SAML identity provider, keep the following information at hand (replace `<Portal>` by the external DNS name of your Portal):

Metadata	`https://<Portal>/saml/metadata`
Entity ID	`https://<Portal>/saml`
	`https://<Portal>/saml/withauth`

Assertion Consumer Service (ACS)	
ACS Binding	`urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST`
NameID format	`urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress`
Nexthink (SP) request binding	`HTTP-Redirect` (not present in the metadata)

Configuring the Portal as a SAML service provider

Because the Finder relies on the Portal for authentication, configuring the Portal as SAML service provider enables users to log in to both the Finder and the Portal through SSO.

To configure the Portal as service provider:

1. Log in to the CLI of the appliance that hosts the Portal.
2. Get the metadata XML file of the identity provider:
   - When using AD FS as identity provider, type in the following command by replacing `<ADFS>` by the address of your AD FS server:
     ```
     wget \
     https://<ADFS>/FederationMetadata/2007 06/FederationMetadata.xml
     ```
   - When using Azure AD as identity provider, retrieve the Federation Metadata XML file that you downloaded while configuring Azure AD as identity provider.
     1. Copy the file to the Portal appliance with your favorite SCP tool.
3. Save the file as `idp_entity_descriptor.xml` in the configuration folder of the Portal. Assuming that the original name of the file is `FederationMetadata.xml`, type in:
   ```
   sudo mv FederationMetadata.xml \
   /var/nexthink/portal/conf/idp_entity_descriptor.xml
   ```
4. Change the owner of the file:
   ```
   sudo chown nxportal:nexthink \
   /var/nexthink/portal/conf/idp_entity_descriptor.xml
   ```
5. Restart the Portal:
   ```
   sudo systemctl restart nxportal
   ```

After configuring the Portal as a SAML service provider, the selected users should be able to log in to both the Finder and the Portal using the corporate login method.

Alternate UPN suffixes in SAML authentication

When a user logs in via SAML authentication with the reference configuration for AD FS or Azure AD shown above, the UPN of the user is mapped to the Name ID returned by the identity provider. If the user logs in with an alternate UPN suffix, the identity provider returns the UPN with the fully qualified domain name as suffix.

For example, let us consider an organization that manages the following fully qualified domains:

- `us.acme.com`
- `de.acme.com`

The fully qualified name of the domains is used to define the UPN of the users:

- `jwick@us.acme.com`
- `mkahn@de.acme.com`

To simplify user access though, administrators may define an alternate UPN suffix `acme.com`, so that users do not have to memorize lengthy subdomain names and levels. With this alternate UPN suffix, the resulting account names look like this:

- `jwick@acme.com`
- `mkahn@acme.com`

Even if users log in with their alternate UPN suffix through SAML, the identity provider returns the UPN with the fully qualified domain name to the Portal instead. For instance, if user John Wick logs in to the Portal (or to the Finder) with the username `jwick@acme.com`, SAML authentication will issue a Name ID claim for user `jwick@us.acme.com`, which the Portal will then match against its stored usernames. Because the identity provider already carries out this transformation, the Portal does not try to map alternate UPN suffixes to fully qualified UPN suffixes when using SAML authentication, contrary to what it does in the case of authentication through Active Directory.

Therefore, for SAML authentication to work properly, the Portal must store usernames in the UPN format with the fully qualified domain name as suffix. Users provisioned from Active Directory are normally stored with the correct UPN.

Related tasks

- Logging in to the Portal
- Logging in to the Finder
- Setting the names of the Portal
- Sending email notifications from the Appliance
- Just-In-Time provisioning of user accounts with SAML
- Provisioning user accounts from Active Directory

Related references

- Active Directory authentication

Just-In-Time provisioning of user accounts

Overview

Manually adding users to Nexthink through the Portal may be a tedious and error prone operation, specially if you have a fair amount of users to add to your setup.

Thanks to the just-in-time (JIT) provisioning of user accounts, take advantage of the users and groups managed by your SAML identity provider to automatically create the required user accounts in the Portal when users log in for the first time.

In addition, user information is verified and access rights updated on every login. For instance, if the group membership of a user changes, the access rights of the user change accordingly.

Prerequisites

To provision users just-in-time with SAML, you need first to:

- Have an admin account in Nexthink that is not SAML authenticated (local or AD account). This admin account will be required to complete JIT SAML configuration.
- Enable SAML authentication of users in Nexthink.
- Define user profiles in Nexthink.
- Add users to your SAML identity provider and define groups of users.

Procedure and method

These are the main steps to provision users just-in-time with SAML:

1. Enable JIT provisioning of users through SAML in the Portal.
2. Instruct your SAML identity provider to convey group membership and personal information in the SAML assertions about a user.
3. Map user groups to user profiles in the Portal.

The idea is thus to assign profiles to users based on their group membership and update their personal information on every login. Depending on whether a particular user account already exists in Nexthink or not, the system does the following:

Group to profile mapping:	Successful	Unsuccessful
User account missing in Nexthink	• Create user account in Nexthink: ♦ Username = Name ID ♦ Set profile based on group mapping ♦ Set full name and email • Log the user in	• Deny access to the user
User account exists in Nexthink	• Update account in Nexthink: ♦ Update profile based on group mapping ♦ Update full name and email • Log the user in	• Deny access to the user • Deactivate / Delete user account

In case of an unsuccessful mapping of a user group to a profile, the user gets its account:

- Deactivated, if the user logged in to the system in the past.
- Deleted, if the user has never logged in to the system before.

When deactivated, a user account still keeps the data associated to it, including modules, dashboards, etc. If a deactivated user later joins a properly mapped group and is thus reprovisioned, all associated data is recovered. In turn, if a user account is deleted, it loses all its associated data.

Enable JIT provisioning in the Portal

Configure the Portal to support the JIT provisioning of users:

1. Log in to the CLI of the appliance that hosts the Portal.
2. Optional: If the Portal has no configuration file yet, that is, if `portal.conf` does not exist in folder `/var/nexthink/portal/conf`, create it by copying the defaults from the sample configuration file:
   ```
   sudo -u nxportal cp
   /var/nexthink/portal/conf/portal.conf.sample \
   /var/nexthink/portal/conf/portal.conf
   ```
3. Edit the configuration file of the Portal:
   ```
   sudo vi /var/nexthink/portal/conf/portal.conf
   ```
4. Add a configuration line to it:
   1. Press **Shift + G** to go to the last line of the file.
   2. Press **o** to add a new line.
   3. Type in the following line:
      ```
      globalconfig.saml.jit-user-provisioning = true?
      ```
   4. Press **Esc** and type in the following colon command to save changes an exit:
      ```
      :wq
      ```
5. Restart the Portal:
   ```
   sudo systemctl restart nxportal
   ```

Advanced configuration

In case that your SAML identity provider does not allow you to modify the name of the attribute in the SAML assertions that conveys the required information, add the name that identifies that piece of information system (usually a URI) to the configuration file of the Portal. There is a dedicated entry for each one of the required assertions: full name, group membership, and email.

The default values in the configuration file of the Portal support the names used by AD FS and the ones that you supply when configuring Azure AD as indicated below.

```
fullname-attribute-names =
 [ "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/name",
   "nexthink.fullname" ]
groups-attribute-names =
 [ "http://schemas.xmlsoap.org/claims/Group",
   "nexthink.groups",
   "http://schemas.microsoft.com/ws/2008/06/identity/claims/groups" ]
email-address-attribute-names =
 [ "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress",
   "nexthink.email" ]
```

Adding group membership and personal information to SAML assertions

Configure your SAML identity provider to include the group membership and the personal information (full name and email address) of users in its SAML assertions, also known as *claims* in Azure AD and AD FS.

Configuring claims in AD FS

To configure the claims in Microsoft AD FS:

1. Log in to the Windows Server machine that runs AD FS as administrator.
2. Open AD FS management console.
3. On the left-hand side panel, under **Trust relationships**, select **Relying Party Trusts**.
4. Right-click the entry that you must have previously configured to define the Portal as a relying party.
5. From the context menu, select the entry to edit the policy for issuing claims:
    - In Windows Server 2016, select **Edit Claim Issuance Policy...**.
    - In Windows Server 2012, select **Edit Claim Rules...**.
6. In the **Issuance Transform Rules** tab, click **Add rule...** to get the full name of the user in the SAML assertions. The wizard to add a new transform rule for claim issuance shows up.
    1. On the **Choose Rule Type** step, select **Send LDAP Attributes as Claims** under **Claim rule template**.
    2. Click **Next >**.
    3. On the **Configure Claim Rule** step, provide the following information:
        - Under **Claim rule name**, type in:
          `nexthink.fullname`
        - Under **Attribute store**, select **Active Directory**.

        - ◊ Under **Mapping of LDAP attributes to outgoing claim types**, select **Display-Name** as **LDAP Attribute** and **Name** as **Outgoing Claim Type**.
    4. Click **Finish**
7. Back to the **Issuance Transform Rules** tab, click **Add rule...** again to add the groups of the user to the SAML assertions.
    1. On the **Choose Rule Type** step, select **Send LDAP Attributes as Claims** under **Claim rule template**.
    2. Click **Next >**.
    3. On the **Configure Claim Rule** step, provide the following information:
        - ◊ Under **Claim rule name**, type in:
          `nexthink.groups`
        - ◊ Under **Attribute store**, select **Active Directory**.
        - ◊ Under **Mapping of LDAP attributes to outgoing claim types**, select **Token-Groups - Qualified by Long Domain Name** as **LDAP Attribute** and **Group** as **Outgoing Claim Type**.
    4. Click **Finish**
8. Back to the **Issuance Transform Rules** tab, click **Add rule...** for the third time to add the email address of the user to the SAML assertions.
    1. On the **Choose Rule Type** step, select **Send LDAP Attributes as Claims** under **Claim rule template**.
    2. Click **Next >**.
    3. On the **Configure Claim Rule** step, provide the following information:
        - ◊ Under **Claim rule name**, type in:
          `nexthink.email`
        - ◊ Under **Attribute store**, select **Active Directory**.
        - ◊ Under **Mapping of LDAP attributes to outgoing claim types**, select **E-Mail-Addresses** as **LDAP Attribute** and **E-Mail Address** as **Outgoing Claim Type**.
    4. Click **Finish**.
9. Click **OK** to close the page for editing claim rules.

Configuring claims in Azure AD

To configure the claims in Azure AD:

1. Log in to Azure from your web browser https://portal.azure.com.
2. Click **Azure Active Directory** on the left-hand side panel.
3. Under **Manage**, select **Enterprise applications**.

4. Select the Nexthink Portal application that you must have previously configured.
5. Click the pencil icon at the top right corner of the second tile to edit the **User Attributes & Claims**. The page to edit the claims appears.
6. Click the pencil icon to the right of **Groups returned in claim**. The page **Group Claims (Preview)** shows up.
    1. Choose **Groups assigned to the application**, as the groups associated to the user to be returned in the claim.
    2. Select **Group ID** as the **Source attribute** to return.
    3. Under **Advanced options**, tick **Customize the name of the group claim**.
    4. As **Name (required)**, type in:
        `nexthink.groups`
    5. Click **Save** to return to the **User Attributes & Claims** page.
7. Click the button **Add new claim** to include the full name of the user in the issued SAML assertions. The page **Manage user claims** shows up:
    1. As **Name**, type in:
        `nexthink.fullname`
    2. Choose **Attribute** as type of **Source**.
    3. As **Source attribute**, select:
        `user.displayname`
    4. Click **Save**.
8. Click the button **Add new claim** to include the email of the user in the issued SAML assertions. The page **Manage user claims** shows up:
    1. As **Name**, type in:
        `nexthink.email`
    2. As **Source attribute**, select:
        `user.mail` or `user.userprincipalname`
    3. Click **Save**.
9. Optional: Delete the claims not consumed by the Nexthink Portal.
10. Get the identifiers of the groups in Azure AD to map them to Nexthink profiles later.
    1. Back to the main page of the Azure portal, click **Azure Active Directory** on the left-hand side panel.
    2. Under **Manage**, select **Groups**. The list of active groups appears on the page **Groups - All groups**.
    3. Select one of the groups that you wish to map to a profile in Nexthink.
    4. On the left-hand side menu of the page, select **Properties** under **Manage**.
    5. In the **Properties** page, under the **General settings** section.
    6. Click the paper icon to the right of the **Object ID** field to copy the identifier of the group.

7. Paste the **Object ID** somewhere else (e.g. a text editor) and save it, so that you can reuse it later.
8. Click **Discard** at the top of the **Properties** page to go back to group selection and repeat the operation for as many groups as you need to map to profiles in Nexthink.

Mapping groups to profiles

To map the groups defined in your SAML identity provider to the profiles defined in Nexthink Portal:

1. Log in to the Portal with a local or AD admin account (see prerequisites above).
    - **Warning**: Do not try to log in through corporate single sign-on with this account! As user groups are not mapped to profiles yet, the mapping will fail and the account might be deactivated (if not local).
2. Click the **ADMINISTRATION** drop-down menu at the top of the window.
3. Select **Accounts** under **ACCOUNT MANAGEMENT**. The page to manage user accounts appears.
4. Click the button **SAML Groups** at the top of the page.
5. Click the button **Add group** to set a new mapping.
    1. Type in the name of a group in the column **AD group name**.
        - If Azure AD is your SAML identity provider, type in or paste the previously saved **Object ID** of the group.
    2. Select an available user profile from the list in the **Profile** column.
        - If the profile is parameterized, choose the view domain of the users to be imported from the **View** list in the **Profile Domain** column.
        - Additionally, if the parameterized profile is of the administration type, choose the administration domain of the users to be imported from the **Admin** list in the **Profile Domain** column.
6. Optional: Repeat the previous step to add more mappings.
7. Click OK.

At login time, the Portal grants access to all users that are members of at least one of the mapped groups. The exact permissions of the user are determined by the assigned profile.

Determining mapping precedence

Because users may belong to more than one group, the order in which you specify the mapping of the groups is important. Namely, if a user belongs to two

groups and both groups are mapped to different profiles in the Portal, the user gets assigned the profile that is mapped to the first group in the list.

Related tasks

- Enabling SAML authentication of users
- Adding users

Enabling Windows authentication of users

Overview

Windows authentication lets Nexthink users comfortably log in to both the Portal and the Finder by securely using their Windows logon information, without requiring the users to type in their credentials again (single sign-on).

For Windows authentication to work, the following prerequisites must be fulfilled:

- The Portal must have a proper external DNS name (not an IP address as name).
- The user must have been created in Nexthink as an Active Directory user.
- Multiple domain configurations are supported.
- The domain controller must run one of the following operating systems:
    - Windows Server 2019 and Windows Server, version 1809
    - Windows Server 2016 and Windows Server, version 1709
    - Windows Server 2012 R2
    - Windows Server 2008 R2

The example configuration in this article is provided for illustration purposes only. For more information on Active Directory and the command-line tools to configure it, please consult Microsoft documentation or contact Microsoft support.

The operations described in this article should only be performed by a Nexthink Engineer or a Nexthink Certified Partner.

If you need help or assistance, please contact your Nexthink Certified Partner.

Domain configuration

To let the Portal connect to your domain controller and perform the authentication of users, you require:

- A dedicated user account in Active Directory for the Portal.
- A Service Principal Name.
- The generation of a keytab file.

The Portal also acts on behalf of the Finder to perform Windows authentication; therefore, there is no need of additional configuration for the Finder. As enabling technology, the Portal makes use of Kerberos-based authentication.

For the sake of example, let us imagine that you want to enable Windows authentication within the following setup:

- Domain name: **example.com**
- External DNS name of the Portal: **portal.example.com**
- Name of the Portal account in AD: **nxtPortalSso**
- Password of the Portal account in AD: **userPassword**

Whenever any of these elements appears in the following instructions, substitute them for your own data. Pay attention to the letter case of the commands and names given in the instructions. Failing to respect the case will result in a misconfiguration of the service. For example, if the domain name is displayed as **EXAMPLE.COM** in the instructions, replace it by your own domain name in upper case.

To configure the domain controller:

1. Log in to the domain controller as administrator.
2. Click **Start** > **Administrative Tools** > **Active Directory Users and Computers**.
3. Click the node of your domain (**example.com**).
4. In the details pane, right-click the OU or CN in which to create the user account for the Portal.
5. Select **New** > **User** from the context menu.
6. In **User logon name**, type in *nxtPortalSso*. Fill in the other fields with values that let you easily identify the account as belonging to the Portal (their exact value is irrelevant).
7. Click **Next**.
8. In **New Object - User**, type *userPassword* in both the **Password** and **Confirm password** fields and set the following password properties:
    - **User cannot change the password** - true.
    - **Password never expires** - true.
9. Click **Next**.
10. In the **Account** tab of the user properties, set the following option:
    - **This account supports Kerberos AES 256 bit encryption** - true.

11. Click **Finish**.
12. Open a command line window.
13. As the Service Principal Name (SPN), use the canonical host name of the Portal (DNS A record) and not an alias (or CNAME record). To create a new SPN, type in:
```
setspn -S HTTP/portal.example.com nxtPortalSso
```
14. To generate the keytab file, type in:
```
ktpass -out .\nxtportal.keytab -princ
HTTP/portal.example.com@EXAMPLE.COM -mapUser
nxtPortalSso@EXAMPLE.COM -mapOp set -pass userPassword
-crypto AES256-SHA1 -pType KRB5_NT_PRINCIPAL
```

Portal configuration

To enable Windows authentication in the Portal:

1. Log in to the CLI of the Appliance hosting the Portal.
2. Optional: If the Portal has no configuration file yet, that is, if `portal.conf` does not exist in folder `/var/nexthink/portal/conf`, create it by copying the defaults from the sample configuration file:
```
sudo -u nxportal cp
/var/nexthink/portal/conf/portal.conf.sample \
/var/nexthink/portal/conf/portal.conf
```
3. Edit the configuration file of the Portal:
```
sudo vi /var/nexthink/portal/conf/portal.conf
```
4. Add the following lines in the Portal configuration file (again, pay attention to the letter case of the configuration settings):
```
globalconfig.sso.enabled = true
globalconfig.sso.realm = "EXAMPLE.COM"
```

```
globalconfig.sso.service-name = "portal.example.com"
```

5. In case of a multi-domain Active Directory, the following line should also be added (optional for single domain):

```
globalconfig.sso.accepted-realms="EXAMPLE.COM,CHILD.EXAMPLE.LOCAL"
```

   - The domains are listed and separated by a comma.
   - Domain names cannot contain a comma.
   - Pay attention to the letter case of the configuration settings.

6. Save your changes and exit by typing:

```
:wq
```

7. Copy the keytab file generated in the previous section to the Portal:
   1. Use your favorite SCP tool to copy the file **nxtportal.keytab** to the home directory of the nexthink account in the Portal.
   2. Log in to the CLI of the Portal and type in:

```
sudo chown nxportal:nexthink nxtportal.keytab
sudo chmod 600 nxtportal.keytab
sudo mv nxtportal.keytab
/var/nexthink/portal/conf/sso
```

8. Restart the Portal:

```
sudo systemctl restart nxportal
```

Browser configuration

To connect to the Portal using Windows authentication, the web browser must trust the URL of the Portal. According to your specific supported browser, follow one of the configuration instructions below.

Internet Explorer or Chrome

1. Open the **Control Panel**.
2. In the **Network and Internet** category, select **Internet Options** (or just click **Internet options** if you have a list view without categories).
3. In the **Security** tab, select **Local intranet**.
   1. Click the **Sites** button.
   2. At the bottom of the dialog, click **Advanced**.
   3. Under **Add this website to the zone:**, type in the DNS name of the Portal: **portal.example.com**.
   4. Click **Add** and then **Close**.
4. In the **Advanced** tab, scroll the **Settings** list.
   1. Under the **Security** section, tick the box **Enable Integrated Windows Authentication***.
5. Click **OK**.

If you do not have permissions to modify these options, contact your system administrator. Note that you need to restart the computer for the changes to take

effect.

Firefox

1. Open **Firefox**.
2. In the address bar, type in **about:config**.
3. Click the button at the bottom of the warning message to accept the risk of changing the configuration settings.
4. In the **Search** box, look for the setting **network.negotiate-auth.trusted-uris**.
5. Double-click the name of the property in the results of the search to change its value.
6. Enter the DNS name of the Portal: **portal.example.com**.
7. Click **OK**.

Related tasks

- Adding users
- Setting the names of the Portal
- Logging in to the Finder
- Logging in to the Portal

Related references

- Active Directory Authentication
- Canonical domain names for Windows authentication

Provisioning user accounts from Active Directory

Overview

Manually adding user accounts to Nexthink may be a tedious process when many users need access to the Portal and, optionally, to the Finder. If you manage your corporate user accounts with Active Directory (AD), take advantage of groups in AD to dynamically provision user accounts to Nexthink and set their permissions accordingly.

Basically, the solution is to map AD groups to user profiles in Nexthink. Then the Portal automatically provisions user accounts from the AD users that belong to those groups.

Prerequisites

The provisioning of user accounts to Nexthink works in Active Directory setups with one or multiple domains.

In the case of a setup with multiple domains, the following constraints apply:

- Each group to be provisioned must not contain users from different domains, whatever the nature of the group (local, global or universal). That is, all users in a group must belong to the same domain.
- Nexthink recommends creating dedicated global groups in each domain for the Nexthink users to be provisioned.
- In case that alternate UPN suffixes are used, please refer to the dedicated section to check the extra configuration needed.

The solution has been tested on Domain Controllers running the following versions of Windows Server:

- Windows Server 2008 R2
- Windows Server 2012 R2

Other versions may not be suitable for provisioning users.

Configuring LDAP

To provision user accounts from Active Directory, configure first the LDAP connection of the Portal to the AD servers (Domain Controllers):

1. Log in to the Web Console of the master Appliance (the Appliance that hosts the Portal) from a web browser. Replace the example by the actual address of the Portal:
   https://portal.yourcompany.com:99
2. Click the **PORTAL** tab at the top of the window.
3. Select **Active Directories** from the left-hand side menu.
4. Click the button **ADD ACTIVE DIRECTORY** to add a new AD server.
5. Fill out the form that shows up:
   - **Server name:** A generic name to identify your AD server.
   - **Server address:** The DNS name or the IP address of your Active Directory server, followed by the TCP server port (usually 389, for non-secured LDAP connection).
   - **Enable LDAP over SSL:** Optionally tick the box to use a secure connection to the AD server. If you enable SSL, import the AD server certificate into the Portal when necessary.

- **Bind DN:** The Distinguished Name of the account for connecting to the AD server. Example: CN=portalAD, OU=servers, DC=company, DC=local.
- **Bind Password:** The password that corresponds to the Bind DN account.
    - The password can include any printable ASCII character except for the less than sign, the single quote, and the double quotes: `< ' "`.
- **Users base DN:** The starting node in the AD tree for searching users. It must be an Organizational Unit.
- **Groups base DN:** The starting node in the AD tree for searching groups. It must be an Organizational Unit.
- **Scope:** Where to look for users and groups from their defined base nodes. There are three possible values:
    - **base:** Search only for entries at the base DN.
    - **onelevel:** Search for entries one level under the base DN, but not including the base DN nor any nodes at a deeper level.
    - **subtree:** Search for entries at the base DN and all levels under it.
- **Groups filter:** Use this LDAP search filter to optimize the provisioning. It is important for Active Directories having a lot of groups, as it can improve the synchronization time and resource consumption. Filters restrict the groups to be added to the portal that are listed in the Nexthink mapping screen. Please refer to the Provisioning performance optimization section for more details about how to use this feature.
- **Recursion through groups:** Untick this box to disable the recursion through groups during the provisioning, which may increase the performance of the provisioning. Only do so if advised by Customer Success Services, because the impact on provisioning needs to be tested case by case. Please refer to the Provisioning performance optimization section for more details about how to use this feature.

6. Optional: Click **TEST LDAP PARAMETERS** to check the connection with the AD server. The Portal must be running for the test to work.
7. Click on **Save changes** to save the configuration.

The Portal does not immediately update user and group information after saving the configuration. Instead, the Portal is scheduled to synchronize with the AD server every hour. Alternatively, force a synchronization with the AD server from the account management dashboard in the Portal (see how in Mapping AD groups to user profiles below).

Provisioning performance optimization

Provisioning users from Active Directory can be very resource intensive in setups with a high number of groups.

To optimize provisioning performance, consider the **Groups filter** and **Recursion through groups** parameters to limit the number of retrieved groups. If you are not familiar with LDAP search queries or with recursion through groups, please contact Nexthink Support before updating these parameters.

The default synchronization frequency is 24h. Do not increase this frequency unless there is a real business need.

Groups filter

Prerequisites

Use group filters to limit the number of groups retrieved from AD. Group filters only have an impact on the retrieved groups and not on the retrieval of members within the groups. To know more about writing filters, refer to Microsoft official documentation about Search filters. To focus on groups only, any filter added to the Web Console is logically combined with the filter `(objectClass=group)` by using the `&` operator.

For example, if you add the following filter to the **Groups filter** field:

```
!(cn=*RESTRICTED*)
```

The resulting filter used by the Portal is:

```
(&(objectClass=group)(!(cn=*RESTRICTED*))
```

Note that Microsoft Active Directory does not support extensible matching.

More examples of Group filters

1. Retrieve groups that contain either *nexthink* or *portal* in their name (partial match of group name):
   ```
   |(cn=*nexthink*)(cn=*portal*)
   ```
2. Select groups based on their distinguished names. The filter below returns just the two specified groups:
   ```
   |(distinguishedName=cn=g1,ou=admin,dc=nexthink,dc=com)(distinguishedName=cn
   ```
3. Retrieve all groups that are members of the group named *nexthinkGroups*:
   ```
   memberOf=cn=nexthinkGroups
   ```

Recursion through groups

By default, the Portal retrieves the members of a group recursively, that is, it will automatically retrieve users in nested groups. For very large AD deployments, this can lead to performance issues, in such cases it is recommended to disable recursion: untick the option **Recursion through groups** to avoid recursing through nested groups. In this case only the users that are direct members of the group will be retrieved.

Preparing your existing users

Your existing users may fall into the following two categories:

- Users authenticated by Active Directory, that is, those whose username is a UPN of the form *user@company.suffix*.
- Users not authenticated by Active Directory.

Depending on their category, and before mapping AD groups to profiles, prepare your existing users for a successful migration to the provisioning of users from AD.

Migration of users authenticated by Active Directory

For users authenticated by Active Directory who belong to any of AD groups to be mapped, the migration is straightforward. After provisioning, their Portal and Finder content is preserved, but their profile may be modified according to the AD groups to which they belong.

If a user authenticated by Active Directory does not belong to any of the AD groups to be mapped, the user continues to exist as an AD authenticated user in Nexthink. The user keeps the same content and profile as before provisioning.

Migration of users not authenticated by Active Directory

For users not authenticated by Active Directory, but by the Portal itself, convert them first to AD authenticated users. To that end, change their username to a proper UPN and proceed as in the previous case.

If a Nexthink user does not exist in Active Directory, you will not be able to supply a UPN name for the user and the migration will not be carried out. After provisioning, the user continues to exist as a Nexthink-only user.

Mapping AD groups to user profiles

Once the Portal is able to retrieve AD information on groups and users from the Domain Controller, map the groups that the Portal finds AD to user profiles in Nexthink. The Portal retrieves AD groups of any scope (domain, global, or universal) and of any type (security or distribution).

To map AD groups to user profiles:

1. Log in to the Portal as central administrator.
2. Click the **ADMINISTRATION** drop-down menu at the top of the window.
3. Under **ACCOUNT MANAGEMENT**, select the option **Accounts** to open the dashboard for editing accounts.
4. Optional: Click the button **Synchronize with AD** at the top of the dashboard to force the Portal to update the information on users and groups from the Domain Controllers. While the update process is going on, the Portal displays the message **Synchronization in progress** in place of the button.
5. Click the button **Set AD groups** at the top of the dashboard. The dialog for mapping AD groups to profiles shows up.
6. Click the button **Add group** to set a new mapping.
    1. Type in the name of a group in the column **AD group name**. As you type, a list of the possible groups to complete the name appears below. The groups are displayed in the form `groupName@domainName`. Finish typing or select one of the groups provided as a suggestion. Note that it is not possible to provision two groups that have the same name if they are in the same domain.
    2. Select an available user profile from the list in the **Profile** column.
        - ◊ If the profile is parameterized, choose the view domain of the users to be imported from the **View** list in the **Profile Domain** column.

◊ Additionally, if the parameterized profile is of the administration type, choose the administration domain of the users to be imported from the **Admin** list in the **Profile Domain** column.

7. Optional: Repeat the previous step to add more mappings.
8. Click OK.

The Portal automatically adds the users in the mapped AD groups to its own list of user accounts. Their *Username* in the Portal is the same as their account name in Active Directory (UPN of the form *user@company.suffix*).

The status and the time of the last AD synchronization are displayed at the bottom of the screen. In case of failed synchronization, see the errors in the tooltip.

Determining mapping precedence

Active Directory users may belong to more than one AD group. If you defined different mappings for the AD groups to which a particular user belongs, the first defined mapping takes precedence. That is, the order in which you define the mappings determines their priority.

See the AD group and the mapped profile of a particular user under the columns **AD group** and **Profile** in the accounts management dashboard. These fields, as the whole list of accounts, are refreshed when the Portal synchronizes with AD.

Authentication and permissions of provisioned user accounts

User accounts provisioned from AD groups naturally make use of Active Directory authentication. For all users that use this type of authentication, the Portal checks user credentials against Active Directory at each login attempt. Therefore, if a particular user is removed from AD, the user is immediately unable to log in to the Portal anymore.

On the other hand, a change of membership to an AD group may result in a different profile being assigned to a provisioned user, but only after the Portal synchronizes with AD. Since the profile determines the permissions and access rights of the user in Nexthink, the user may temporally have out-of-date access rights in force. If immediate effects are required, use manual synchronization.

Deleting and disabling provisioned user accounts

Changing the mappings of AD groups to profiles or the composition of AD groups themselves may result in some of the previously provisioned users no longer being part of the provisioning. Specifically, any of these two actions may lead to that situation:

- Removing a mapping of an AD group to a profile.
- Revoke the membership of a user to an AD group that takes part in a mapping.

Users that are left out of account provisioning after any of these operations fall into either one of these two categories:

- Users who never logged in to Nexthink.
- Users who logged in to Nexthink at least once.

Users who never logged in to Nexthink (via the Portal, the Finder, or NXQL request) are physically removed from the system, otherwise they are just *disabled*. A disabled user does not appear in the list of accounts and cannot log in. However, the configuration and content associated to a disabled user is kept in the system. If a disabled user is recreated as a result of being mapped again, the account is reactivated with all its previous configuration and content.

If you actually delete a provisioned user from the list of accounts in the Portal, by selecting the user and clicking the bin icon in the **Accounts** dashboard, all the configuration and content associated to the user is removed from the system and the user can no longer log in. However, beware that if the user still belongs to one of the mapped AD groups, the account will be recreated at the next synchronization of the Portal with the AD. If you do not want a deleted user account to reappear in Nexthink, remember to revoke its membership to any of the mapped AD groups.

Maximum number of users

The default maximum number of users in the product is 500. This limit includes both currently existing users and previously existing users that logged in to the product at least once (via the Portal, the Finder, or NXQL request) and were subsequently removed.

Provisioned users from AD groups who never logged in and were subsequently removed from the provisioning (for instance, because of a deleted mapping) are physically removed from the system and they do not take part in the counting of

users to compute the limit. On the other hand, *disabled* users do count for the limit.

If you need to overcome the limit of 500 users, please contact Nexthink Support.

Related tasks

- Adding users
- Enabling Windows authentication of users
- Importing Data from Active Directory
- Importing and replacing certificates

Related references

- Access rights and permissions
- Active Directory Authentication

Establishing a privacy policy

Overview

Nexthink privacy is built around five pillars:

Security of information: The information is collected via encrypted channels and the access to all databases is restricted.

User privileges: The privileges of a user define the subset of the devices or locations that the user can access (view domains), the rights of the user to change the configuration (administration privileges), the creation of content (dashboards) and the access to external web domains and web requests.

Anonymization: Users, devices, destinations and web domains are anonymized by default. Users need special privileges to access identity information of these objects.

Storage policy: The full set of information is collected and stored by default. However, it is possible to remove and prevent collecting devices and other information from the dataset. There is also a special policy for Web & Cloud storage that can prevent the collection of web domains.

Audit trails: Every change in the configuration settings is audited, including account edition.

Security of information

Overview of communication channels

The following schema describes the communication architecture from a high level point of view.

The table describes the communication channels used to access or transport sensitive information:

Core components			Protocol or encryption
Collector	->	Engine	UDP encrypted
Finder	<-->	Engine	TLS
Portal	<-->	Engine	HTTPS by default
Portal	<-->	Nexthink Central License Manager	HTTPS
Optional			
Shell	<-->	Appliance (Engine or Portal)	SSH
API	<-->	Engine	REST HTTPS
Active directory	<-->	Engine	SSL
Cloud Intelligence / Enhance	<-->	Engine	HTTPS
Investigation Library	<-->	Portal	HTTP
Investigation Library	<-->	Finder	HTTP
DB backup	<-->	Engine	SMB
Email	<-->	Engine	SMTP
Nexthink updates	<-->	Finder, Appliance	HTTPS, HTTP
Nexthink customer improvement program	<-->	Finder	HTTPS

All the channels that transport sensitive information are encrypted. All optional channels have to be activated or configured, apart from the shell that is set-up by default.

Collected data

Nexthink does not collect any information about the content of files, e-mail, web sites or any other content. Nexthink collects the following data:

Objects (represent real life items recognized by Nexthink)
• User • Device • Package • Application • Executable • Binary • Port • Destination • Printer • Domains
Activities (represent actions performed by Objects)
• Installation • Execution • Connection • Print job • System boot • User logon • Web request
Events (are warning or errors)
• Device warning • Device error • Execution warning • Execution error

User privileges

Accounts are based on *profiles* and *roles*.

Profiles determine the access rights of a user:

- Access to the Portal, possibly limited to a *view domain*, the right to create and publish dashboard content in the Portal, and administration rights

(management of accounts, additional content, and system configuration).
- Access to the Finder, the rights to edit applications, objects tags, categories, services and global alerts.
- Access related to web domains (Web & Cloud visibility) in the Finder. By default, users can only see the web domains that are configured in web-based services.

Roles define the default content that is available to a user in the Finder and in the Portal. Roles are assigned to users either indirectly through their profiles or directly through the user account.

- For non-administrator users, roles limit the content that can be accessed in the Portal.

Limiting the view to a domain

Devices can be grouped along a hierarchical tree. For example, a tree with three levels: Department / Region / Entities.

View Domains

A View domain represents the set of data that a user has the right to see. It is defined by a node of the hierarchy and optionally by a limit in the depth. Based on the previous example, a view domain could limit the view to a specific Department and allow the user to drill-down to the underlying Region but prevent to see the details by Entities.

Creating and publishing dashboards in the Portal

Administrators can create, publish, and manage Portal modules, which are a construct that groups dashboards.

An administrator can see and manage the modules published by any other user, where *managing* means updating or deleting a published module.

Normal users, on the other hand, can only see a module created by an administrator if the module is included in their roles. The creation and publication of modules is also restricted for normal users. Normal users can create and publish Portal modules only if they have the following options checked in their profile, respectively:

- **Allow creation of personal dashboards**
- **Allow publication of dashboards**

Normal users can see the modules published by other normal users. A normal user with the permission to publish dashboards can manage the modules created by other normal users, but not by administrators.

Of course, normal users with the right to create dashboards can manage their own personal modules; that is, the modules that they have created or that they have copied to their personal content.

Privileges for users of Nexthink Finder

For users of the Finder, select their privileges when creating the user profiles (step 4).

The privileges are related to the edition and application of object tags, the modification of the system configuration (categories, metrics, campaigns, remote actions, etc), and other features for system management.

Anonymization

Access rights to data

There are four levels of data privacy defined in the profile of the account, that specify the access rights of each account to particular pieces of information:

Access rights	Description
Anonymous users, devices, destinations, and web domains	The names of users, devices, destinations, and web domains are not visible to the account
Anonymous users and devices	The names of users and devices are not visible to the account
Anonymous users	Only the names of users are not visible to the account
None (full access)	No restrictions: all names are visible

The following table enumerates the visible attributes of **users, devices, destinations and domains** for each data privacy level.

Data Privacy Level	Users	Devices	Destinations	Domains
None (full access)	Username Distinguished Name Full Name Nexthink UID	Computer name Windows SID IP address Nexthink UID	Destination name IP address Nexthink UID	Domain name Nexthink UID
Anonymous users	*Anonymized users*	Computer name Windows SID IP address Nexthink UID	Destination name IP address Nexthink UID	Domain name Nexthink UID
Anonymous users and devices	*Anonymized users*	*Anonymized devices*	Destination name IP address Nexthink UID	Domain name Nexthink UID
Anonymous users, devices, destinations and domains	*Anonymized users*	*Anonymized devices*	*Anonymized destinations*	*Anonymized domains*

Display of anonymized UIDs

When the data privacy level enforces anonymous users, devices, destinations or domains, their UIDs are hidden from the results of an investigation as follows (example based on devices):

That is, the UID is displayed in the form **anonymized *object*** , where object is the type of retrieved object under anonymization.

Investigations using the name of the object are not possible. However, if an authorized Finder user provides the UID of an object, any user may refer to the object in an investigation through its UID.

Categories

Categories also support data privacy: a level can be set for a category so that only accounts with the same or a higher data privacy level will be able to see and use a given category. For example, if a category is created with a Data Privacy level set to "none (full access)", only Finder user accounts having a "none (full access)? level will be able to see and use this category. The privacy setting on categories applies only to the Finder.

Examples of user profiles

These are some examples of user profiles that can be configured with the current privacy features of Nexthink:

Nexthink administrator	
He is the administrator of Nexthink products within the enterprise and therefore has full access rights.	

User privileges
 	
Portal:
 	
`Administrator: yes`
 	
`Reader: all domains`
	Anonymization
	
	Portal & Finder:
	
	`none (full access)`

Dashboard creation: public Finder: Allow access, allow edition	

CIO	
He needs high level information. Therefore he will mainly use Portal as a Reader.	
User privileges Portal: Administrator: no Reader: all domains Dashboard creation: public Finder: No access, No edition	**Anonymization** Portal & Finder: anonymous users

Privacy officer	
He has the full access regarding data anonymization and can provide the User UID to other co-worker if needed.	
User privileges Portal: Administrator: no Reader: all domains Dashboard creation: public Finder: Allow access, No edition	**Anonymization** Portal & Finder: none (full access)

Security engineer
He needs full access to all data such that he can investigate any issues.

User privileges	**Anonymization**

Portal:
 	
`Administrator: no`
 	
`Reader: all domains`
 	
`Dashboard creation: public`
 	
Finder:
 	
`Allow access, allow edition`
	Portal & Finder:
	
	`none (full access)`

Network & system engineer	
He needs access regarding connection and destination but does not need to access user information.	
User privileges	**Anonymization**

Portal:
 	
`Administrator: no`
 	
`Reader: all domains`
 	
`Dashboard creation: personal`
 	
Finder:
 	
`No access, No edition`
	Portal & Finder:
	
	`anonymous users`

Support engineer	
He only needs to access user information when required and needs to ask the privacy officer for User UID.	
User privileges	**Anonymization**

Portal:
 	
`Administrator: no`
 	
`Reader: all domains`
	Portal & Finder:
	
	`anonymous users`

`Dashboard creation: no`
 	
Finder:
 	
`Allow access, No edition`	

IT project manager (transformation)	
He is only accessing information related to a specific project and only needs anonymous information.	

User privileges
 	
Portal:
 	
`Administrator: yes`
 	
`Reader: limited domains`
 	
`Dashboard creation: personal`
 	
Finder:
 	
`Allow access, allow edition`
	Anonymization
	
	Portal & Finder:
	
	`anonymous users, devices, destinations and domains`

Storage policy

Database

The following databases are used in Nexthink product:

Engine	Portal
Database (in memory) Database backup • Internal (automatic) • External (not configured by default)	Database Database backup • Internal (automatic) • External (not configured by default)

Ignoring fields

In addition to the anonymization of data, it is possible to configure the system to ignore certain data that is delivered by the collector. In this case, data are not recorded at all:

ignore_username	If this is set to true, engine will no longer store the user names and Finder will show 'Unknown' for all usernames.
user_interaction	If set to false, user interaction information will no longer be recorded (it will not be displayed in the device view and the "interaction time" aggregate will be always 0%).
ignore_windows_license	If set to true, windows license key will no longer be stored.
ignore_print_jobs	If set to true, all print jobs will be ignored.
ignore_external_ip	If set to true, destination IP address outside the specified internal networks are set to 0.0.0.0 in connections.
ignore_external_domains	If set to true, domains which are not part of the internal domains are not recorded; except for domains that are explicitly included in the definition of a web-based service.

Retention time

By default, a device is removed automatically from the Engine Database after 3 months of no activity. The retention time can be configured.

Ignoring specific devices

For each device, it is possible to restrain the collected information at the level of the Engine. The possible settings are:

- Web requests, connections and executions (by default, everything is stored)
- Connections and executions
- Executions only
- None
- Remove

For the latter case, this means that the device will be removed from Engine database if there is no activity for more than one day (i.e. the Collector was uninstalled).

In the Finder, right-click a particular device in the list view results of an investigation or in the top-left icon of its own device view and select **Edit...** :

Ignoring specific application, executables, binaries and domains

The same is possible for applications, executables and binaries. The only difference is that it is not possible to remove them, but only to stop storing the related information.

Web & Cloud

Because Web & Cloud data has a significant impact on the data retention of the Engine, there are three different settings for the storage policy of domains and web requests that let you control how they are stored.

1. Log in to the Web Console as administrator.
2. Under the **APPLIANCE** tab, select **Privacy** from the left-hand side menu.
3. In the **Web & Cloud** section, select the desired **Storage policy** from the list.

Web & Cloud storage policy		Use cases	Result
1	**none**	I don't want to store any information related to web domains.	Domains and web requests are discarded.
2	**services only**	+ I want to monitor internal or external web services like saleforce.com, office365.	Storage is discarded unless related to a configured web-based service. (*)

3	**all**	+ I want to discover all web applications used in my company. + I want to see if there are any security breach in my company	Every domain and web request is stored. But the visibility can be restricted and depends on user privileges. (*) (**)

(*) When a web-based service is created, its underlying web requests and domains are stored their visibility is unrestricted.

(**) If a web request does not belong to a defined service, its access is restricted.

Visibility for metrics

In the same way Finder users need special privileges to view web domains and web requests that are not part of a web-based service (see above), metrics have a similar setting that limits the web domains and web requests that are visible in the dashboards of the Portal.

From the Web Console, under the **Web & Cloud** section, select the **Visibility for metrics** from the list:

- **full**, to enable metrics the use of web data from any stored web request or domain (in accordance to the storage policy).
- **restricted**, to prevent metrics from using any web data that is not related to a web-based service.

Engine internal domains

Internal domains are never sent to Cloud Intelligence. To identify internal domains, the following rules apply:

- Domains with non-official TLD (top level domain)
- Domains with name corresponding to IP addresses belonging to Engine internal network.
- Domains with names matching custom rules (e.g. *.nexthink.com). These rules can be set up in the Web Console.

Excluded domains

For privacy reasons, you may want to avoid storing web requests to particular domains. For instance, a web application that collects opinions and complaints of employees about their peers and superiors requires the anonymity of the participants. However, with the right level of permissions, a user of the Finder

can easily discover who connected to the application and when, just by investigating the web requests that are addressed to the domain of the web application. To make the system ignore web requests to specific domains, add the domains to the *excluded domains* list found in the Web Console.

To add a domain to the excluded domains list:

1. Log in to the Web Console as administrator.
2. Click to the **Appliance** tab at the top of the window.
3. Select **Privacy** from the left-hand side menu.
4. Under **Web & Cloud**, add the domain to the list **Excluded domains**:
    - Separate the names of the domains with a single space character (e.g. *anonymize.nexthink.com *.example.com*).
    - You can use wildcards in the names of the domains:
        - The question mark **?** may be replaced by any single character.
        - The asterisk ***** may be replaced by any number of characters.

Audit trails

Auditing Nexthink is performed using the syslog framework. It captures actions performed with administrator rights that may impact the system. It is not a logging facility.

Only the action and who performs it is audited. The values that are set are not logged.

The complete list of audit point is available here.

Data sent to Nexthink

Nexthink Appliances automatically send non-personal data to Nexthink SA to provide value-added services to Nexthink customers. Learn how to enable or disable these services to select which data you send to Nexthink in the article about operational data sent to Nexthink.

Related tasks

- Adding Users
- Specifying your internal networks and domains

Related concepts

- Service

Related references

- Operational data sent to Nexthink
- Data retention

Disabling local accounts for interactive users

Overview

After enabling a corporate login solution for Nexthink, either via SAML or Windows authentication of users, disable local accounts for interactive users to enforce the security policies of corporate accounts.

Reserve the local accounts for API calls only.

Disabling local accounts

To disable local accounts for interactive users:

1. Log in to the CLI of the master Appliance that hosts the Portal.
2. Optional: If the Portal has no configuration file yet, that is, if `portal.conf` does not exist in folder `/var/nexthink/portal/conf`, create it by copying the defaults from the sample configuration file:
   ```
   sudo -u nxportal cp
   /var/nexthink/portal/conf/portal.conf.sample \
   /var/nexthink/portal/conf/portal.conf
   ```
3. Edit the configuration file of the Portal:
   ```
   sudo vi /var/nexthink/portal/conf/portal.conf
   ```
4. Add a configuration line to it:
   1. Press **Shift + G** to go to the last line of the file.
   2. Press **o** to add a new line.
   3. Type in the following line:
      ```
      globalconfig.portal.user.allow-local-logins = false
      ```
   4. Press **Esc** and type in the following colon command to save changes an exit:
      ```
      :wq
      ```
5. Restart the Portal:
   ```
   sudo systemctl restart nxportal
   ```

Setting the complexity and minimum length of passwords for local accounts

Overview

Even though Nexthink recommends externally managed accounts for improved security, you may still need to create local accounts for API users or for testing purposes.

To prevent central administrators from assigning weak passwords to local accounts, configure complexity criteria such as the minimum length of the password and the types of characters that the password must include. These will be verified when adding new users to the system or when updating the password of an existing user.

The default minimum length of a password is 8 characters and a minimum of three other complexity criteria must be met. These limits do not apply to accounts whose password is externally managed (SAML or Active Directory accounts).

Required types of characters in local passwords

By default, a local password must fulfil at least three out of the four following complexity criteria on the type of characters included:

- 1 uppercase letter
- 1 lowercase letter
- 1 digit
- 1 special character

Where the configurable list of special characters is:
`<space>!"#$%&'()*+,-./:;<=>?@[]^_`{|}~`

And the number of complexity criteria to fulfil is configurable as well.

Setting the minimum length and complexity criteria

To change the minimum length and other complexity criteria for passwords of local accounts:

1. Log in to the CLI of the Portal appliance.
2. Optional: If the Portal has no configuration file yet, that is, if

`portal.conf` does not exist in folder `/var/nexthink/portal/conf`, create it by copying the defaults from the sample configuration file:

```
sudo -u nxportal cp
/var/nexthink/portal/conf/portal.conf.sample \
/var/nexthink/portal/conf/portal.conf
```

3. Edit the Portal configuration file:
   ```
   sudo vi /var/nexthink/portal/conf/portal.conf
   ```
4. Press **G** to go to the end of the file.
5. Press **o** to insert a new line.
6. Type in the following line to change the default value to 12, for example:
   ```
   globalconfig.portal.user.password.min-length = 12
   ```
7. Type in the following line to change the default number of default complexity criteria that a password must fulfil to all 4, for example:
   ```
   globalconfig.portal.user.password.number-of-criteria = 4
   ```
8. Type in the following line to explicitly set the default list of special characters:
   ```
   globalconfig.portal.user.password.special-characters =
   " !\"#$%&'()*+,-./:;<=>?@[]^_`{|}~"
   ```
9. Press **Esc** to stop editing.
10. Save your changes and exit by typing:
    ```
    :wq
    ```
11. Restart the Portal to apply your settings:
    ```
    sudo systemctl restart nxportal
    ```

Related tasks

- Adding users

Protecting local accounts against brute force attacks

Overview

Externally managed accounts (SAML or Windows authentication) are already protected against brute force attacks by the mechanisms of each identity provider.

To protect local accounts against brute force attacks, a local account is blocked for fifteen minutes after five failed login attempts by default. Configure the blocking period and maximum number of failed login attempts in the Portal.

Setting the maximum login attempts and blocking period

To set the maximum number of failed login attempts and the blocking period of local accounts:

1. Log in to the CLI of the Portal appliance.
2. Optional: If the Portal has no configuration file yet, that is, if `portal.conf` does not exist in folder `/var/nexthink/portal/conf`, create it by copying the defaults from the sample configuration file:
   ```
   sudo -u nxportal cp
   /var/nexthink/portal/conf/portal.conf.sample \
   /var/nexthink/portal/conf/portal.conf
   ```
3. Edit the Portal configuration file:
   ```
   sudo vi /var/nexthink/portal/conf/portal.conf
   ```
4. Press **G** to go to the end of the file.
5. Press **o** to insert a new line.
6. Type in the following line to configure the maximum number of failed login attempts before blocking the local account. For example, to specify the default of 5 times:
   ```
   globalconfig.portal.user.max-consecutive-failed-logins = 5
   ```
7. Type in the following line to configure the blocking period. For example, to specify the default of 15 minutes:
   ```
   globalconfig.portal.user.lock-account-duration = 15m
   ```
8. Press **Esc** to stop editing.
9. Save your changes and exit by typing:
   ```
   :wq
   ```
10. Restart the Portal to apply your settings:
    ```
    sudo systemctl restart nxportal
    ```

Central administrators can see the blocked local accounts and the time until they are blocked in the Portal, by opening the dashboard **Accounts** under **ADMINSTRATION** - **ACCOUNT MANAGEMENT**.

Related tasks

- Adding users

Preventing password saving in the Finder

Overview

Saving the password of login sessions in the Finder may be a convenient feature

for users to avoid typing their password again and again. However, for security reasons, you may want to enforce a policy of making password input mandatory, especially if the users share the workstations that they use to log in to the Finder.

Starting from V6.18, it is however more convenient and secure to enable Windows authentication of Finder and Portal users. From V6.21 on, SAML authentication of users enables other corporate single-sign on options. Whenever possible, prefer a corporate single sign-on solution based on either SAML or Windows authentication of users to preventing password saving using the method described below.

Procedure

The Finder reads a key in the Windows registry to know whether to allow users to save their password or not. If the value of the key is set to 1, the Finder hides the options **Remember password** and **Sign me in automatically** in the login dialog.

To prevent users from saving their password in Finder sessions:

1. In the computer where the Finder is installed, press **Win(key)+R** to display the run dialog.
2. Type in **regedit** as the program to open in the dialog and press **Enter**. The Registry Editor opens.
3. Browse the Windows registry in the Registry Editor and select the key **HKEY_LOCAL_MACHINE\SOFTWARE\Nexthink**.
    - If the key does not exist, create it by right-clicking the **SOFTWARE** folder:
        1. Select **New -> Key** from the context menu.
        2. Type in *Nexthink* as the name of the new key.
        3. Right-click the area on the right-hand side of the Registry Editor that holds the list of values for the key.
        4. Select **New -> DWORD (32-bit) Value** from the context menu.
        5. Type in **preventUsersFromSavingPassword** as the name of the value.
4. Right-click the value with the name **preventUsersFromSavingPassword** to change its data.
5. Select **Modify...** in the context menu. The dialog to edit the value shows up.
6. Set the value of the field **Value data** to 1 in the dialog.
7. Click **OK**.

This method changes the value of the registry key in one computer only. Alternatively, you can use GPO to impose the same value for the registry key in all the computers where the Finder is installed.

Related tasks

- Logging in to the Finder
- Enabling Windows authentication of users
- Enabling SAML authentication of users

Controlling session timeouts in the Portal

Overview

To prevent Cross-Site Request Forgery (CSRF), sessions in the Portal are time-limited and protected by secure tokens.

By default, a token remains valid for 8 hours. If you are inactive for more than 8 hours while in a Portal session, your next action in the Portal will redirect you to the login page.

In turn, a session is valid for 24 hours by default. After continuously using the Portal for 24 hours without interruption, the session expires and you are forced to log in again to renew the session.

Setting the value of session timeouts

The validity time for both tokens and sessions is configurable. Remember that the longer the interval, the more vulnerable the Portal is to CSRF attacks.

1. Log in to the CLI of the Portal appliance.
2. Optional: If the Portal has no configuration file yet, that is, if `portal.conf` does not exist in folder `/var/nexthink/portal/conf`, create it by copying the defaults from the sample configuration file:
   ```
   sudo -u nxportal cp
   /var/nexthink/portal/conf/portal.conf.sample \
   /var/nexthink/portal/conf/portal.conf
   ```
3. Edit the Portal configuration file:
   ```
   sudo vi /var/nexthink/portal/conf/portal.conf
   ```
4. Type in the following line to set the value for the validity time of tokens (minimum value is 2 minutes). Use the suffix **h** to specify the time interval in hours and **m** to express it in minutes. For example, to set the period to

its default value of 8 hours:

```
globalconfig.portal.session.token-validity-period = 8 h
```

5. Type in the following line to set the value for the validity time of sessions. For example, to set the period to its default value of 24 hours:

```
globalconfig.portal.session.maximum-session-lifetime = 24 h
```

   ♦ Optional: Express it in minutes:

```
globalconfig.portal.session.maximum-session-lifetime = 1440
m
```

6. Save your changes and exit:

```
:wq
```

7. Restart the Portal to apply your settings:

```
sudo systemctl restart nxportal
```

Overriding session timeouts

Note that, when creating a user, the user may be granted the privilege of never being timed out. In that case, the values configured for session timeouts do not apply to that user.

Related tasks

- Adding users

Security settings in the Appliance

Overview

The Appliance uses standard mechanisms for authentication and security:

- Connections to the CLI of the Appliance are established through OpenSSH, which is the SSH implementation installed in the operating system of the Appliance.
- Connections to the Portal are managed by the security layer of the underlying Java implementation.
- Connections to the Web Console and the Web API of the Engine are encrypted and authenticated with TLS.

Starting from V6.17, the Appliance hardening ensures that the ciphers and algorithms negotiated by the security protocols in the Appliance are currently considered strong. Starting from V6.20, the Web Console admits TLS 1.2 only.

Legacy browsers still in use within your organization may require though the use of protocols, ciphers and algorithms that are no longer considered secure. Nexthink recommends that you update your software so that it implements the latest security mechanisms. Nevertheless, in case that you cannot easily replace your legacy browsers, find below how to configure the Portal and the Engine to support security protocols that are not strong enough to be enabled by default.

Portal secure protocols and ciphers

By default, the Portal supports TLS 1.2 as a security protocol. Most modern browsers and operating systems are able to use this protocol to secure their communications over the Internet. Associated to this protocol, the Portal also supports a default set of cipher suites (considered strong) to negotiate the security settings of a connection.

However, users of Internet Explorer in either Windows Vista or Windows XP, for instance, are limited to TLS 1.0. Therefore, if you want the Portal to support TLS 1.0, you must add it to the list of supported protocols in the configuration file of Nginx, the reverse proxy component of the Portal that handles the connections.

To change the supported protocols and cipher suites:

1. Log in to the CLI of the Appliance hosting the Portal.
2. Edit the SSL configuration file of Nginx:
   ```
   sudo vi /var/nexthink/nxnginx/conf.d/ssl.conf.overrides
   ```
3. Type in the names of the supported protocols and cipher suites in the entries:
   - `ssl_protocols`
   - `ssl_ciphers`
4. Save the file and exit by typing:
   ```
   :wq
   ```
5. Restart Nginx:
   ```
   sudo systemctl restart nginx
   ```

For instance, these are the protocols and cipher suites supported by default:

```
ssl_protocols TLSv1.2;
ssl_ciphers ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:
ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:
ECDHE-ECDSA-CHACHA20-POLY1305:ECDHE-RSA-CHACHA20-POLY1305:
DHE-RSA-AES128-GCM-SHA256:DHE-RSA-AES256-GCM-SHA384;
```

To support the protocols TLS 1.0 and/or TLS 1.1 in addition to the default

protocol TLS 1.2, substitute the entry of included protocols for:

```
ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
```

Conversely, to exclusively support TLS 1.2 for improved security, replace the entry by:

```
ssl_protocols TLSv1.2;
```

Specify the names of supported ciphers in the format understood by the OpenSSL library. See the full list of supported ciphers with the command:

```
openssl ciphers
```

Engine secure protocols and ciphers

To secure the communications through the Web API, the Engine supports by default TLS 1.2 and a set of ciphers considered strong. These security settings are also valid for the query interface with the Finder and the Portal, as well as for the LDAP and the Application Library clients.

The security settings are configurable in the **ssl** section of the configuration file `/var/nexthink/engine/01/etc/nxengine.xml`. If they are not specified, their configuration is equivalent to the following values:

```
<config>
   <engine>
      ...
      <ssl>
          ...
          <ciphers>ECDHE-ECDSA-AES256-GCM-SHA384,ECDHE-RSA-AES256-GCM-SHA384,
          ECDHE-ECDSA-AES128-GCM-SHA256,ECDHE-RSA-AES128-GCM-SHA256,
          DHE-RSA-AES256-GCM-SHA384,DHE-RSA-AES128-GCM-SHA256</ciphers>
          <protocols>tlsv1.2</protocols>
      </ssl>
      ...
   </engine>
</config>
```

To configure a different set of supported ciphers and protocols, modify each element in the **ssl** section:

ciphers

List of ciphers supported by the Engine. Specify the names of the ciphers in the format accepted by *openssl*. Separate each supported cipher either by a colon ':' or a comma ',' delimiter. To see the list of all the available ciphers that you can choose from, log in to the CLI of the Engine and type: `openssl ciphers`.

protocols

List of supported protocols, separated by comma ',' delimiters.

For instance, to support old browsers, enable protocols SSL 3.0 and TLS 1.0:

```
<ssl>
<protocols>sslv3,tlsv1,tlsv1.1,tlsv1.2</protocols>
</ssl>
```

Note that there is no need to modify the ciphers, since these protocols can use AES256-SHA and AES128-SHA, which are allowed by default.

Web Console secure protocols and ciphers

The Web Console admits clients to connect only through TLS 1.2 by default.

To change the list of protocols and ciphers in the Web Console:

1. Log in to the CLI of the Appliance that hosts the Web Console.
2. Edit the configuration file of the web server that provides the communication to the Web Console:
   `sudo vi /var/nexthink/console/etc/lighttpd.conf`
3. Locate in the file the line with the comment that indicates the start of the SSL section:
   `#### SSL engine`
4. Replace the default settings by the desired protocols and ciphers. Use the options and syntax of the underlying Lighttpd web server.
5. Save your changes and exit by typing:
   `:wq`
6. Restart the Web Console:
   `sudo systemctl restart nxconsole`

The operations described in this article should only be performed by a Nexthink Engineer or a Nexthink Certified Partner.

If you need help or assistance, please contact your Nexthink Certified Partner.
Related tasks

- Importing and replacing Certificates

Related references

- Appliance Hardening

Setting the Do Not Disturb periods between campaigns

Set both the Do Not Disturb period and the Non-negotiable Protection period between campaigns in the configuration file of the Portal.

To configure the Do Not Disturb periods:

1. Log in to the CLI of the Portal appliance.
2. Optional: If the Portal has no configuration file yet, that is, if `portal.conf` does not exist in folder `/var/nexthink/portal/conf`, create it by copying the defaults from the sample configuration file:
   ```
   sudo -u nxportal cp
   /var/nexthink/portal/conf/portal.conf.sample \
   /var/nexthink/portal/conf/portal.conf
   ```
3. Edit the Portal configuration file:
   ```
   sudo vi /var/nexthink/portal/conf/portal.conf
   ```
4. Type in the following line to set the value for the do not disturb period. Use the suffix **m**, **h**, or **d** to specify the time interval in minutes, hours, or days, respectively. For example, to set the period to its default value of 6 hours, type in:
   ```
   globalconfig.euf-service.customization.do-not-disturb-period
   = 6 h
   ```
   Type in the following line to set the value for the non-negotiable protection period. Use the suffix **m**, **h**, or **d** to specify the time interval in minutes, hours, or days, respectively. For example, to set the period to its default value of 20 minutes, type in:
   ```
   globalconfig.euf-service.customization.non-negotiable-protection-period
   = 20 m
   ```
5. Save your changes and quit the editor by typing:
   ```
   :wq
   ```
6. Restart the Portal to apply your settings:
   ```
   sudo systemctl restart nxportal
   ```

Related tasks

- Limiting the reception rate of campaigns

Data retrieval and storage

Data retention

Data retention in the Portal

Nexthink is able to keep historical data for several years in the database of the Portal. The Portal consolidates the data collected from the Engines and keeps them in permanent storage. The Portal does not store all the individual events, but the results of widget computation.

History	Number of devices
Several years (view by periods of 2 years max)	Up to 150 000 (consult for bigger setups)

Keeping historical details of count metrics

In the case of count metrics, the Portal stores the total number of objects satisfying a particular set of conditions. The Portal keeps these numbers of counted objects as regular historical data. In addition, for every count metric, the Portal stores the list of objects that contributed to the metric for the current day, week, month, and quarter. The list of objects that contributed to a count metric, along with their selected set of display fields, are known as the *details* of the metric. To see the details of a count metric, open a dashboard with KPI or table widgets representing the values of that count metric in the Portal, hover the mouse cursor over a particular value and select **Show details**.

When additional disk space is allocated, the Portal can store the details of count metrics not only for the current period (aggregated object lists for the current day, week, month, and quarter), but also for previous days, weeks, months, and quarters. To keep historical details of count metrics, make sure that you reserve some disk space for that purpose in the Portal appliance:

1. Log in to the Web Console of the Appliance that hosts the Portal as admin. Use your web browser:
   https://<appliance_address>:99
2. Click the **Portal** tab at the top of the window.
3. Select the **General** section from the left-hand side menu.
4. Under **Parameters**, choose the quantity of disk space that you want to dedicate to the storage of history details for count metrics from the list

labeled **Disk space allocated for historical data**.

The number of days of stored historical details depend on the amount of disk space reserved, the number of enabled count metrics, the number of display fields included in each metric, and the actual metric data collected each day. When the disk space dedicated to store the details of count metrics is exhausted, the Portal launches a cleanup process that deletes one or a few full days of historical details, starting from the oldest day in the saved history.

Find orientative figures for setting the disk space for historical details in the hardware requirements of the Portal.

Data retention in the Engine

In its turn, the Engine stores real-time data with a greater level of detail than the Portal. All events are kept in volatile memory until the configured maximum is reached. You can browse all the data in the Engine down to the event level with the help of the Finder.

We present in this section several tables that contain an estimation of the number of weeks that correspond to a maximum number of events stored the Engine. Once the maximum is reached, new events replace the oldest ones, which have already been consolidated in the Portal before being dismissed.

Estimations are calculated on the following basis:

- Working day of 8 hours.
- Week of 5 working days.
- Practically no activity outside working hours.

Please note that the amount of data required for mobile devices is negligible.

Without the Web and Cloud Feature

If the Web and Cloud feature is not enabled, the data retention periods are expressed as follow (in weeks):

	200M events	100M events	50M events
10 000 devices	4-6 weeks	2-3 weeks	1-2 weeks
5 000 devices	8-10 weeks	4-5 weeks	2-3 weeks

With Web and Cloud Feature but for Services Only

If Web and Cloud has been exclusively enabled for defined Services, the following data retention periods are expected (in weeks):

	200M events	100M events	50M events
10 000 devices	4 weeks	2 weeks	1 week
5 000 devices	8 weeks	4 weeks	2 weeks

With Full Web and Cloud Feature

If the Web and Cloud feature is fully enabled, the following data retention periods are expected (in weeks):

	200M events	100M events	50M events
10 000 devices	2-4 weeks	1-2 weeks	1 week
5 000 devices	4-6 weeks	2-3 weeks	1-2 weeks

Increasing data retention in the Engine

If you reckon that your current data retention period is rather short, increase it by trading off detailed information for more capacity. Gain up to 50% more history by setting a more aggressive aggregation policy in the Engine.

Related tasks

- Establishing a data retention policy in the Engine

Increasing the maximum number of metrics

Overview

For a good compromise between the number of available metrics and the consumption of resources in the Portal in terms of disk space and computation time, the default maximum number of enabled metrics in a setup is 500 metrics. In large setups, however, a total of 500 metrics is often not enough to cover all needs.

To avoid falling short of available metrics, increase the limit of maximum enabled metrics up to 1000 metrics from the Web Console.

Configure the limit of metrics on the Web Console

To increase the maximum number of metrics:

1. Log in to the Web Console of the master appliance.
2. Select the **PORTAL** tab at the top of the window.
3. Click **General** on the left-hand side menu.
4. Under **Parameters**, select the desired new maximum number of enabled metrics from the list **Max number of Metrics**:

5. If you have reserved disk space for the details of count metrics:
    1. Double the quantity of disk space stated in the field **Disk space allocated for historical data**.
6. Click **SAVE CHANGES**. To activate the new value, the Portal restarts.

When setting a new limit to the maximum number of enabled metrics, remember the following guidelines:

- Increase your current value step by step (add 100 metrics from the list each time).
- Ensure that your setup runs correctly for a week:
    - ♦ The disk space available for the Portal is sufficient.
    - ♦ The nightly computation does not extend for longer than 2 hours.

If your setup runs smoothly with the new setting and you still need more metrics, go back to the first step until you reach the absolute maximum of 1000 metrics.

Disk space requirements

The disk space requirements in the Portal grow linearly with the number of enabled metrics. See the table of Portal requirements to find the disk space required for 500 and 1000 enabled metrics.

Related references

- Hardware requirements
- Data retention

Establishing a data retention policy in the Engine

Overview

The Engine stores the real-time data that it receives from the the Collectors in the form of events. Events are very numerous and they usually take most of the memory reserved to the Engine. The types of events that occupy most of the space in memory are executions, connections, and web requests. When two or more of these events are very similar to each other and they occur in sequence, the Engine may consider that they are actually the same event. In that case, the Engine combines the data of the events and stores only one event in its database. We say then that the Engine *aggregates* the information of several events into one; thus saving memory space and resulting in a larger history for the Engine.

When you have the web monitoring feature fully enabled, you usually collect a huge number of web domains. In the same spirit of event aggregation, when two or more domain names share their highest level domains, the Engine may group them into one generalized domain by obeying specific rules. This process is known as domain *compaction* or domain *compression* and it replaces one or more of the lower level domains in the domain name by the wildcard character *****. For instance, the Engine might compact the domains **one.example.com** and **two.example.com** into ***.example.com**. Note however that those domains declared as internal or included in the definition of a web-based service are considered of special interest to you and, therefore, they are never compacted.

Learn here how to set the maximum number of events and establish the policies for both the aggregation of events and the compaction of domains in the Engine.

Setting the maximum number of events

To set the maximum number of events that the Engine can store:

1. Log in to the Web Console as admin.
2. Click the **Engine** tab at the top of the window.
3. Select the **General** section from the left-hand side menu.

4. Under **Parameters**, choose a number from the **Max stored events** drop-down list.
    - The amount of RAM available in the Engine limits the possible choices for the maximum number of events. To be able to select a high number of events, ensure that the Engine complies with the hardware requirements regarding the available memory.
5. Click **SAVE CHANGES**. Note that the Engine is restarted after saving the changes.

Setting the aggregation policy for events

Choose among four strategies of aggregation for an optimal trade-off between detailed event information and history length. The more aggressive the policy, the fewer individual (non aggregated) events are visible from the Finder.

1. Log in to the Web Console as admin.
2. Click the **Engine** tab at the top of the window.
3. Select the **General** section from the left-hand side menu.
4. Under **Parameters**, choose one of the following aggregation policies from the list labeled **Aggregation policy**:
    - **very low - normal history**, for the traditional minimal aggregation.

- **low - up to 10% more history**, for increasing the history 10% approx. while keeping most of the individual events.
- **medium - up to 80% more history**, for a more aggressive aggregation policy to increase history in the Engine up to 80%.
- **high - up to 100% more history (default)**, for the most aggressive aggregation policy to practically double the history traditionally available in the Engine. Starting from V6.19, this is the default setting for new installations and for upgraded appliances that never changed their default value.

5. Click **SAVE CHANGES**. Note that the Engine is restarted after saving the changes.

To further increase history, especially if you installed the Collector on servers, configure your Engines to apply the **aggresive** reduction of destinations to Collector traffic. An increase of up to 30% in history is expected on setups with servers (or other types of devices) that communicate in bursts with multiple destinations, at the cost of losing the individual information of every connection.

Setting the compaction level for domains

1. Log in to the Web Console as admin.
2. Click the **Engine** tab at the top of the window.
3. Select the **General** section from the left-hand side menu.
4. Under **Parameters**, choose one of the following domain compression policies from the list labeled **Domain compression**:
   - **medium (recommended)**, the default compression policy for domains with more than five levels or with repetitive (or randomly generated) subdomains. This is the recommended setting.
   - **high**, to apply a compression method to all the stored domain names according to a public list of domain suffixes.
5. Click **SAVE CHANGES**. Note that the Engine is restarted after saving the changes.

For a detailed explanation of compaction policies, see the section about compaction in the definition of domain.

Allocating extra memory for Nexthink Act

If you have purchased Nexthink Act and your Engines support more than 8000 devices each, allocate an extra half gigabyte of memory for your Engines to deal with the additional outputs from remote actions.

This extra amount of memory should be present on every Engine if you followed the hardware requirements indications.

1. Log in to the Web Console as admin.
2. Click the **Engine** tab at the top of the window.
3. Select the **General** section from the left-hand side menu.
4. Tick the option under **Allocate Extra Memory**:

5. Click **SAVE**.

Related concepts

- Event
- Domain

Related references

- Data retention in the Engine
- Hardware requirements
- Server support
- Public suffix list (external)

Storing Engine data in a secondary disk drive

In some situations, you may want the Engine to store its data in a disk drive different from the system drive:

- Little space available in the system drive.
- Faster secondary drive.

The following procedure shows you the recommended way for storing the data of the Engine in a secondary disk drive:

1. Log in to the CLI of the Appliance that hosts the Engine.
2. Create a new partition in the secondary disk using **fdisk**. For this part of the procedure, we assume that your secondary disk is a second SCSI or SATA device in the Appliance named /**dev**/**sdb**. If this is not the case, you make have to adapt the commands below to suit your specific needs. Type the following commands to create the first primary partition in the

secondary disk:

```
sudo fdisk /dev/sdb
```

`n` (for creating a new partition)

`p` (for creating a primary partition)

`1` (create the first partition)

`1` (default number for the first cylinder)

`2610` (default number for the last cylinder)

`w` (write the partition info to the disk)

3. Format your newly created partition with the ext4 filesystem:
   ```
   sudo mkfs -t ext4 /dev/sdb1
   ```
4. Stop the Engine:
   ```
   sudo systemctl stop nxengine@1
   ```
5. Rename the data folder of the Engine to keep its contents:
   ```
   cd /var/nexthink/engine/
   sudo mv 01/ 01-old/
   ```
6. Recreate the data folder of the Engine:
   ```
   sudo mkdir 01/
   ```
7. Mount the folder on the recently created partition of the secondary disk:
   ```
   sudo mount /dev/sdb1 /var/nexthink/engine/01
   ```
8. Edit the **/etc/fstab** file for the system to automatically mount the secondary drive while booting:
   ```
   sudo vi /etc/fstab
   ```
9. Add the following line to the end of the file:
   ```
   /dev/sdb1 /var/nexthink/engine/01 ext4 defaults 1 2
   ```
10. Save your changes and quit the text editor:
    ```
    :wq
    ```
11. Copy the contents of the old data folder of the Engine to the new data folder:
    ```
    sudo cp -r /var/nexthink/engine/01-old/*
    /var/nexthink/engine/01
    ```
12. Set the Engine as owner of the data folder:
13. `chown -R nxengine:nxengine /var/nexthink/engine/01`
14. Restart the Engine
    ```
    sudo systemctl start nxengine@1
    ```

Now the Engine is using the secondary disk drive as storage medium.

You can use a similar method to store the logs of the Engine in a secondary disk drive. Just mount the directory **/var/log/nexthink** on a partition of the secondary disk in much the same way as explained above for **/var/nexthink/engine/01**.

The operations described in this article should only be performed by a Nexthink Engineer or a Nexthink Certified Partner.

If you need help or assistance, please contact your Nexthink Certified Partner.

Importing data from Active Directory

The Engine provides an out the box integration with Active Directory to retrieve the following information via the Lightweight Directory Access Protocol (LDAP):

- **User**: Distinguished Name, Full name, Department, Job title.
- **Device**: Distinguished Name.

The Engine retrieves as well the following information through DNS resolution (DNS namespaces mirrors the AD domains used by an organization):

- **Printer**: Host name.
- **Destination**: Name.

This article discusses data integration from Active Directory and should not be confused with Active Directory Authentication.

LDAPv3 and Active Directory

Reference document: Active Directory LDAP Conformance provided by Microsoft.

Windows Server 2000

The Windows 2000 implementation of Active Directory is an LDAP-compliant directory supporting the core LDAPv3 RFCs available.

Windows Server 2003

Building on the foundation established in Windows 2000 Server, the Active Directory service in Windows Server 2003 is offering new LDAPv3 capabilities:

- **Transport Layer Security (TLS)** - Connections to Active Directory over LDAP can now be protected using the TLS security protocol.

- **Digest Authentication Mechanism** - Connections to Active Directory over LDAP can now be authenticated using the DIGEST-MD5 Simple Authentication and Security Layer (SASL) authentication mechanism. The Windows Digest Security Support Provider (SSP) provides an interface for using Digest Authentication as an SASL mechanism.

Windows Server 2008 and 2012

Both Windows Server 2008 and Windows Server 2012 support LDAPv3.

Other implementations

Although Nexthink officially supports Active Directory based on Windows Servers only, other LDAPv3 compliant implementations (such as OpenLDAP) should work as long as the schema in use is the same as in Active Directory.

Setting Up Active Directory Authentication

LDAP servers require an authenticated connection before they will allow queries (searches). This authenticated connection is called a bind. Most LDAPs allow an anonymous bind?where no username or password is submitted; however, others restrict searches to its members and require an authenticated username and password. An Active Directory server requires authenticated access for read-only searches, and you need to have a bind DN and the corresponding bind password. The syntax for the bind DN depends on the LDAP server itself:

NetBIOS logon name
    <domain name>\<username>
Active Directory User Principal Name (UPN)
    username@domain.name
Distinguished Name
    CN=username, OU=users, DC=domain, DC=name

The Engine supports the authenticated method using the **Distinguished Name** syntax only.

Configuring the Engine through the Web Console

1. Log in to the Web Console that is hosting the Engine from your web browser:
    https://engine.yourcompany.com:99
2. Click the **Engine** tab at the top of the window.
3. Select **Active Directories** from the left-hand side menu.
4. Click the button **ADD ACTIVE DIRECTORY** to add a new AD server.
5. Fill out the form **Add Active Directory** as follows:
    - **Server name:** The generic name for your AD server. Example: if you write ?nexthink.ch?, the usernames in the Finder will be shown as user@nexthink.ch.

- **Server address:** Enter here the IP address of your Active Directory server (we currently do not support the DNS or Netbios name) and the TCP server port (usually 389).
- **Bind DN:** The Distinguished Name. Example: CN=reflexengine, CN=applications, OU=servers, DC=company, DC=local.
- **Bind Password:** Enter the password corresponding to the Bind DN account.
- **Base DN:** The Base DN to be used as a starting point for directory searches. Base DN is usually the Organizational Unit where users are located. Example: ?OU=Users, DC=company, DC=local?.
- **Scope:** The SCOPE setting is the starting point of an LDAP search and the depth from the base DN to which the search should occur. There are three options (values) that can be assigned to the scope parameter (we strongly recommend the **subtree** scope option):
    - **base:** This value is used to indicate searching only the entry at the base DN, resulting in only that entry being returned (keeping in mind that it also has to meet the search filter criteria!).
    - **onelevel:** This value is used to indicate searching all entries one level under the base DN - but not including the base DN and not including any entries under that one level under the base DN.
    - **subtree:** This value is used to indicate searching of all entries at all levels under and including the specified base DN.

6. Optional: Click **TEST LDAP PARAMETERS** to check the connection with the AD server.
7. Click on **OK** to add the server. The Engine restarts.

Trusted Domains

Due to the technology used to query Active Directory, the Engine retrieves information from those objects belonging to the domain specified in the configuration only (see **LDAP Base DN** above). It does not follow referrals nor retrieve any information from objects in other domains, even when these other domains share a trust relationship with the configured domain.

Add as many Active Directory servers to the configuration as needed to retrieve objects from several domains.

Querying Active Directory to obtain a User's Distinguished Name

For testing purposes, we advise you to use a powerful tool from Microsoft called Active Directory Explorer. Download it from here.

Here is an example on how you can retrieve a user's DN using this tool :

1. Connect to your AD using your windows username.
2. Click on **Search > "class = User -- user" > "Attribute = sAMAccountname" > "relation = is" > "value = YOUR Windows username"**, then click on **Add**.
3. Click on **Search** to retrieve the corresponding user's DN.

Active Directory data retrieval

The Engine queries its configured LDAP servers each time that it discovers a new user or a new device.

Engines do not automatically refresh LDAP information once they have retrieved it for a particular user or device. It is however possible to force a manual update via the Finder:

1. Log in to the Finder as a user with *system configuration* permissions.
2. Click the sprocket icon in the top right corner of the Finder window.
3. Select the option **Synchronize with Active Directory...**.

The Finder schedules a synchronization with Active Directory data.

The operations described in this article should only be performed by a Nexthink Engineer or a Nexthink Certified Partner.

If you need help or assistance, please contact your Nexthink Certified Partner.

Setting the locale in the Portal

Overview

The user interface of the Portal is available in two languages: English and French. Change the locale settings in the configuration file of the Portal to choose the language for the user interface. The locale settings also determine the format of date and time expressions in the Portal.

Weeks are numbered in the Portal to identify weekly periods along the year. Depending on your location, weeks may start on a different day of the week. For instance, in some countries the week starts on Monday, whereas in other countries the week starts on Sunday. In different regions, there are different conventions as well to specify which week is the first week of the year. Configure the Portal to specify both the first day of the week and the first week of the year depending on your local conventions.

Language and date-time format

Basically, there are three possible configurations: International English, US English, and French. By default, the Portal is set to international English, which is different from US English only in the format of dates and time. In international English, days come first in dates and time is expressed in 24 hours format; whereas in US English, months come first in dates and time is expressed in a 12 hours format with the AM or PM suffix. Find examples of the differences among the three formats in the table below.

	International English	US English	French
Locale settings	en_CH en_UK	en_US	fr fr_CH
Date format	Jan '14 7 Sep 21.09.14	Jan '14 Sep 7 09/21/14	janv. 14 7 sept. 21.09.14

Time Format	14:45:12 15:00 today	02:45:12pm 3pm today	14:45:12 15:00 aujourd'hui

To set the locale in the Portal:

1. Log in to the CLI of the Portal appliance.
2. Optional: If the Portal has no configuration file yet, that is, if `portal.conf` does not exist in folder `/var/nexthink/portal/conf`, create it by copying the defaults from the sample configuration file:
   ```
   sudo -u nxportal cp
   /var/nexthink/portal/conf/portal.conf.sample \
   /var/nexthink/portal/conf/portal.conf
   ```
3. Edit the configuration file of the Portal:
   ```
   sudo vi /var/nexthink/portal/conf/portal.conf
   ```
4. Set the default locale option by typing in the following line. For example, to set the locale to French:
   ```
   globalconfig.portal.user.default-locale = "fr"
   ```
5. Save your changes and quit the editor by typing:
   ```
   :wq
   ```
6. Restart the Portal to apply your settings:
   ```
   sudo systemctl restart nxportal
   ```

Setting the first day of the week

Days are numbered from 0 (Sunday) to 6 (Saturday). To specify the first day of the week, set it as the first element of the **week-days** array, followed by the next four days, in the configuration file:

1. Log in to the CLI of the Portal appliance.
2. Optional: If the Portal has no configuration file yet, that is, if `portal.conf` does not exist in folder `/var/nexthink/portal/conf`, create it by copying the defaults from the sample configuration file:
   ```
   sudo cp -u nxportal
   /var/nexthink/portal/conf/portal.conf.sample \
   /var/nexthink/portal/conf/portal.conf
   ```
3. Edit the configuration file of the Portal:
   ```
   sudo vi /var/nexthink/portal/conf/portal.conf
   ```
4. Add the following line to set the first day of the week:
   - For example, to set day 1 (Monday) as the first day of the week, as it is common in most of Europe and other parts of the world that follow the ISO standard, type in:
     ```
     globalconfig.portal.portal.week-days = [1, 2, 3, 4,
     5]
     ```
   - Alternatively, to set the day 0 (Sunday) as the first day of the week, as it is custom in the UK and the USA, type in:

```
globalconfig.portal.portal.week-days = [0, 1, 2, 3,
4]
```

- And to set day 6 (Saturday) as the first day, as it is usual in islamic countries, type in:

```
globalconfig.portal.portal.week-days = [6, 0, 1, 2,
3]
```

5. Save your changes and quit the editor by typing:

```
:wq
```

6. Restart the Portal to apply your settings:

```
sudo systemctl restart nxportal
```

Specifying the first week of the year

Because the Portal numbers weeks to let you navigate through weekly periods, it is important for the Portal to know which week is considered to be the first week of the year in your region. The configuration setting for determining the first week of the year is related to the convention for choosing the first day of the week. It is expressed by a number that, when subtracted by the number which represents the first day of the week, indicates the latest day of the week that must belong to the new year (it has to lie in January) for the whole week to be regarded as the first week of the new year.

There are three standard values for indicating the first week of the year for different regions of the globe:

- ISO: **4**
  When substracted by 1 (Monday), it yields 3. Three days after Monday is **Thursday**.
- North American: **6**
  When substracted by 0 (Sunday), it yields 6. Six days after Sunday is **Saturday**.
- Islamic: **12**
  When substracted by 6 (Saturday), it yields 6. Six days after Saturday is **Friday**.

As an example, let us look at the transition from 2015 to 2016 for each one of the standard regions:

- In a region following the ISO standard, the first days of the new year fall in the week from Monday, Dec 28th 2015 till Sunday, Jan 3rd 2016. Since Thursday of that week is on Dec 31st 2015, it is not in January of the new year. So that is not the first week of 2016, but the last week of 2015. The first week of 2016 goes from Jan 4th till Jan 10th 2016.

- In a North American region, the week with days in both years goes from Sunday, Dec 27th 2015 to Saturday, Jan 2nd 2016. Because Saturday lies in January 2016, this week is reckoned to be the first week of the year.
- In an Islamic region, the week that marks the transition between the two years goes from Saturday, Dec 26th 2015 to Friday, Jan 1st 2016. Since Friday lies in January 2016, this week is then regarded as the first week of the year.

Note that in regions that follow either the North American or the Islamic conventions, it is enough that the last day of the week falls into January of the new year for the whole week to be the first one of the year.

To set the value for determining the first week of the year:

1. Log in to the CLI of the Portal appliance.
2. Optional: If the Portal has no configuration file yet, that is, if `portal.conf` does not exist in folder `/var/nexthink/portal/conf`, create it by copying the defaults from the sample configuration file:
   ```
   sudo -u nxportal cp
   /var/nexthink/portal/conf/portal.conf.sample \
   /var/nexthink/portal/conf/portal.conf
   ```
3. Edit the configuration file of the Portal:
   ```
   sudo vi /var/nexthink/portal/conf/portal.conf
   ```
4. Add the following line depending on the convention followed in your region to compute the first week of the year:
   - ISO:
     ```
     globalconfig.portal.portal.first-week-of-year-contains
     = 4
     ```
   - North America:
     ```
     globalconfig.portal.portal.first-week-of-year-contains
     = 6
     ```
   - Islamic:
     ```
     globalconfig.portal.portal.first-week-of-year-contains
     = 12
     ```
5. Save your changes and quit the editor by typing:
   ```
   :wq
   ```
6. Restart the Portal to apply your settings:
   ```
   sudo systemctl restart nxportal
   ```

Changing the Time Zone of the Portal

Overview

Because of the distributed nature of the Nexthink solution, the time zone of the Portal may refer to either:

- The time zone of the machine where the Portal itself is installed.
- The time zone of the Portal account in each Engine.

The local time of the Portal

Use the Web console to change the time zone of the Appliance that is running the Portal:

1. Log in to the Web Console that is hosting the Portal as admin:
   https://<appliance_ip_address>:99
2. Click the **Appliance** tab at the top of the window.
3. Select the **General** section from the left-hand side menu.
4. Under **Time**, choose the appropriate time zone from the list labeled **Timezone**, according to the place where the Portal is located.

The Portal uses the time zone of the machine where it is installed in combination with the time zone of the Portal account on each Engine to schedule the collection of data from the Engines. For more information, see Time Zones and data collection.

The time zone of the Portal account

The time zone of the Portal account determines the time shift between the Portal and each Engine and it influences both the time of data collection and the results of the computation of dashboards.

The time zone of the Portal account is set to the same value as the time zone of the admin account in all Engines. As a result, data collected from different Engines coincide in real-time, although it may correspond to different local times.

The operations described in this article should only be performed by a Nexthink Engineer or a Nexthink Certified Partner.

If you need help or assistance, please contact your Nexthink Certified Partner.
Related tasks

- Time Zones and data collection

Time Zones and data collection

Overview

The Portal collects data from the Engine once every day to compute the metrics for its dashboards and build up its history. Because collecting data from the Engine is a costly operation, the Portal is programmed by default to get the data during the night, when the activity of the Engine is supposed to be low. By default, at one o'clock in the morning, the Portal starts collecting information about the events that occurred during the past day, which usually are the 24 hours that went by from past midnight to midnight one hour ago. The whole computation process can take up to several hours, depending on the quantity of data collected and the number and complexity of the metrics to compute.

The Portal determines the *past day* individually for each Engine in the following way:

- The Portal gets the time of the last event stored in the Engine.
- The past day is the whole day before the date of the last event in the Engine (always computed in the time zone of the Portal).

For Engines with low activity, this way of determining the past day might mean that the past day is two (or more) days ago. For instance, if an Engine has no event during the current day yet (when the Portal starts the nightly computation) but the last event happened the day before, the Portal considers the past day to be two days ago for that Engine.

Special care has to be taken when the Portal and the Engine are placed in different time zones, in particular when the Portal is connected to multiple Engines. A setup with Engines placed in distant locations may lead to surprising results in the Portal if the data collection process is not well understood. One o'clock in the morning in one time zone may be two in the afternoon in another. Thus, data collection may not be triggered during the night for all Engines.

This document explains how the Portal determines when to start collecting data from the connected Engines and other issues that arise when the Portal and the Engines are placed in different time zones.

The time zone of the Portal account

The Portal connects to the Engine by means of a dedicated account. This account is unique to each Engine and is similar to the accounts of the users of

the Finder. The time zone of the Portal account matches the time zone of the admin user in every Engine.

Default behavior

By default, the time zone of the admin user (and, therefore, that of the Portal account) is configured in every Engine to have the time zone of Europe/Zurich, which corresponds to Central European Time (CET, UTC +1 hour) during the winter and Central European Summer Time (CEST, UTC +2 hours) during the summer. Therefore, from the point of view of the Portal, all Engines share the same time zone (Europe/Zurich), even when this is actually not the case.

To schedule the collection of data, the Portal computes the local time that is equivalent to 01:00 Zurich time. When the scheduled time is reached, the Portal begins to collect data from all Engines.

If you change the time zone of the admin account, a similiar scenario occurs. All the Engines automatically set the time zone of their Portal accounts to be the same as the time zone of the admin account. As a result, the Portal starts collecting data from all Engines at 01:00 according to the time zone of the admin account. As explained in the previous default case, the Portal computes the equivalent local time for scheduling the data collection.

Example

Let us illustrate the influence of time zones in the data collection with an example involving one Portal connected to two Engines. Imagine that we have a Portal installed in London, one Engine in New York and another Engine in Paris. For the sake of simplicity, we are not going to deal with daylight savings. Therefore, we assume that the Portal in London has UTC time, that the Engine in New York has UTC -5 hours and that the Engine in Paris has UTC +1 hour as their respective time zones.

Suppose that most of the devices with the Collector installed are located in Paris. It makes sense thus to have the time zone of the admin account set to Paris. This ensures that the computation occurs during the night in Paris, when most of the devices are inactive. Since the Portal account shares the same time zone of the admin account, both the Engine in New York and the Engine in Paris have the time zone of the Portal account set to Paris time.

The Portal in London triggers the computation at 01:00 Paris time, that is 00:00 London time. The Engine in Paris has its data collected as usual, from midnight one day ago to midnight one hour ago. However, for the Engine in New York the situation is different. Since its time zone has been centralized to Paris, data collection is performed from 18h last day to 18h today, coinciding in real-time with the collection of data in Paris.

Impact on users

As we said at the beginning, data collection is a costly operation. It increases sensibly the load of the Portal and the Engines while it is going on. To impact the fewer users possible, the Portal collects data during the night. However, in scenarios with multiple time zones involved, the night is not simultaneous for everyone. More users may be impacted as a result of the Portal performing data collection during local working hours.

For instance, In the previous example, where the Portal adapts to the time zone of Paris, users of the Portal in New York may experience poor response time if they try to connect to the Portal late in the evening, because data collection was started at 19:00 New York time and it can go on for a few hours.

Similarly, users of the Finder may experience a decrease in the performance of their connection to an Engine, if the Engine is being solicited by the Portal because of the data collection process.

Therefore, it is recommended to use the time zone of the Engine where most of the users of both the Portal and the Finder are located. In this way, you reduce the impact of data collection on the majority of your users.

Interpreting the results

Be careful with metrics that compute values for particular intervals of time in a day. For instance, let us consider a metric *Number of desktops with nightly activity* that is based on a *between* hours condition. The metric is supposed to return the number of desktops which had any kind of activity during the night, but we have seen that the night is not simultaneous for everybody in setups with multiple time zones.

In the example, the Engine in New York is computing from 18:00 yesterday to 18:00 today, but the Portal makes the computation with respect to the centralized time zone, which is Paris time. Therefore, the widget reports the desktops with nightly activity according to Paris time and not to New York time, even for desktops placed in New York.

Remember that the widgets in the Portal display their results with respect to the time zone used to launch the computation:

- By default, the time zone of Europe/Zurich.
- The time zone of the admin account, if you change it from Europe/Zurich to any other value.

The users of the Portal see time information in their web browser according to one of these possible time zones and it is the same time zone for all users. You should therefore not confuse the time zone of the results in the Portal with the time zone configured in the profile of the user. The time zone in the profile of the user exclusively serves to present information in the Finder, if the user of the Portal is allowed access to the Finder.

Related tasks

- Changing the nightly computation time of the Portal
- Changing the Time Zone of the Portal

Changing the data collection time of the Portal

Changing the starting time of data collection

To change the default time of data collection in the Portal:

1. Log in to the Appliance that hosts the Portal.
2. Optional: If the Portal has no configuration file yet, that is, if `portal.conf` does not exist in folder `/var/nexthink/portal/conf`, create it by copying the defaults from the sample configuration file:
   ```
   sudo -u nxportal cp
   /var/nexthink/portal/conf/portal.conf.sample \
   /var/nexthink/portal/conf/portal.conf
   ```
3. Edit the configuration file of the Portal:
   ```
   sudo vi /var/nexthink/portal/conf/portal.conf.
   ```
4. Add the following lines to the configuration file of the Portal, or modify their values if they are already present. For example, to start the data collection at 2h20:
   ```
   # Time (hour) at which collection for the previous day takes
   place
   globalconfig.portal.collector.time-to-collect = 2
   # Time (minutes) at which collection for the previous day
   takes place
   globalconfig.portal.collector.time-to-collect-minutes = 20
   ```
5. Save your changes and exit the vi editor:
   ```
   :wq
   ```
6. Restart the Portal:
   ```
   sudo systemctl restart nxportal
   ```

Note that the actual time for triggering the nightly computation depends on how you configure the time zone of the Portal and the Engines.

Changing the maximum number of days collected

Every night, the Portal usually collects data of metrics for the past day only. However, for those metrics with their last days empty of data (because they could not be computed or because their history was cleared), the Portal computes not only the past day, but the number of days configured (up to the maximum number of days available in each Engine).

To set a different number of days to go back and compute metrics with no history, add the following line to the configuration file of the Portal. For instance, to compute five days of history, type in:

```
globalconfig.portal.collector.nb-of-days = 5
```

By default, the Portal goes back **3 days** in the past to compute metrics when the data for their last days are missing. Set the configuration variable to **-1** for the historical computation to go back up to the maximum number of days available in each Engine. Remember that computing metrics for dates in the past has some limitations.

The operations described in this article should only be performed by a Nexthink Engineer or a Nexthink Certified Partner.

If you need help or assistance, please contact your Nexthink Certified Partner.

Related references

- Computing dashboard data
- Time Zones and data collection

Nightly task schedules timetable

This table summarizes the time of execution of those tasks that the different Nexthink components perform during the night, when the activity in your IT infrastructure is supposed to be low.

Little activity in the IT infrastructure is desirable because the nightly tasks require appreciable processing power from the Nexthink appliances. In particular, the execution of the tasks *Data collection* and *Engine cleanup and maintenance* may significantly reduce the responsiveness of your Engines to concurrent requests (investigations, remote actions, etc).

Some of them are configurable so you can adapt their activation to the time that suits you best.

Local time	Task	Affects	Indicative duration	Defined in
22:00	Rule-base assignment backup	Portal	< 3 minutes	`/etc/cron.d/nxassignment-crontab`
22:15	Portal backup	Portal	< 3 minutes	`/etc/cron.d/portal-crontab`
22:30	Nginx config backup	Portal	< 3 minutes	`/etc/cron.d/nxnginx-crontab`

01:00	License check	Engine	< 5 minutes	non-configurable

01:00	Data collection	Portal and Engine	minutes to hours	Parameter
				`globalconfig.portal.collector.time-t`
				in file
				`/var/nexthink/portal/conf/portal.con`

01:10	Web Console backup	Web Console	< 3 minutes	`/etc/cron.d/nxconsole-crontab`
03:45	Engine cleanup and maintenance	Engine	15 - 30 minutes	non-configurable
04:15	Engine backup	Engine	< 5 minutes	`/etc/cron.d/nxengine-crontab`

Related tasks

- Web Console backup and restore
- Portal backup and restore
- Engine backup and restore

Enabling printing support

Overview

The feature described in this article has been deprecated.
Starting from V6.18, print monitoring is disabled by default in the Collector. To enable printing support in Nexthink, either:

- Turn on print monitoring during the installation of the Collector.
- Enable print monitoring after the installation of the Collector with the help of the Collector configuration tool.

Note that print monitoring is enabled or disabled at the Collector level, and thus on a per device basis. The rest of the Nexthink components are able to deal with printing-related information as soon as it is available from any Collector.

Applies to platforms:

During Collector installation

Set the option to enable print monitoring support either by passing a parameter to the MSI or by generating the executable to install the Collector with the same parameter:

- If you install the Collector directly through its MSI:
    1. Pass the parameter **PRINTING=enable** to the MSI in the command-line.
- If you generate a executable with the Nexthink Collector Installer:
    1. Tick the option **Report print jobs and printers** when generating the executable from the Nexthink Collector Installer.

After Collector installation

This method is only available if you installed the Collector configuration tool on the device during Collector deployment by either:

- Setting the value of CFG_INSTALL to 1 (default) when running the MSI to install the Collector.
- Generating the executable to install the Collector with the option **Install configuration tool** ticked in the Nexthink Collector installer.

Start the command prompt on the device where the Collector is installed and run the Collector Configuration Tool as follows:

1. Log in to the Windows device with the Collector as a user with administrative rights.
2. Press the Windows button [ ] on your keyboard.
3. Type in **cmd** to make the Command Prompt application show up as a result of the search.
4. Right-click the Command Prompt icon to open a context menu.
    1. Select **Run as administrator** from the menu.
5. At the prompt, type in:
    ```
    nxtcfg.exe /s printing=enable
    ```

Related tasks

- Installing the Windows Collector

Related references

- Information on printers and printing

- Nxtcfg - Collector configuration tool

Ignoring specific print ports

The feature described in this article has been deprecated.
To prevent the Engine from recording print jobs that use specific print ports, list the print protocol prefixes of the ports to be ignored under the **ignored_print_ports** item of the configuration file of the Engine. Along with the print jobs, the Engine also discards the printers that are associated with them.

By default, when the element **ignored_print_ports** is not specified, this option is set in the Engine to ignore the ports with prefixes **TS** and **CLIENT**. Popular virtual environments use these print protocols to print on redirected printers. In this way, the Engine avoids recording duplicate print jobs and printers in virtual environments where the Collector is installed in both client devices and remotely accessible virtual machines.

To set the prefixes of the print protocols that the Engine must ignore:

1. Log in to the CLI of the appliance that hosts the Engine.
2. Open the configuration file of the Engine for editing:
   sudo vi /var/nexthink/engine/01/etc/nxengine.xml
3. Under **config** / **local** / **aggregation** add the following lines:
   ```
   <ignored_print_ports>
       <port_prefix>PREFIX_1</port_prefix>
       ...
       <port_prefix>PREFIX_N</port_prefix>
   </ignored_print_ports>
   ```
4. Save your changes and exit with the following command:
   :wq
5. To make your changes effective, restart the Engine:
   sudo systemctl restart nxengine@1

The operations described in this article should only be performed by a Nexthink Engineer or a Nexthink Certified Partner.

If you need help or assistance, please contact your Nexthink Certified Partner.
Related tasks

- Logging in to the CLI

Related references

- Information on printers and printing

Related concepts

- Printer
- Print job

Enabling support for SMB printers

Overview

The feature described in this article has been deprecated.
In Microsoft Windows networks, it is customary to share printers via the SMB protocol.

When the Collector has SMB printer support enabled and it is installed in a device that uses a shared SMB printer, the Collector listens to print job notifications from the SMB printer, as it does for any other kind of printer (local, TCP, or WSD). In the case of SMB printers however, after the Collector starts listening, Windows generates an excessive number of print job notifications, which are sent through the network via RPC (Remote Procedure Calls).

In environments where many devices are connected to SMB printers, this results in high CPU load and memory footprint in the print server (the device that shares the printer), combined with a massive generation of network traffic that may have a negative impact on the whole network. For this reason, SMB printer support is by default turned off in the Collector.

Support for SMB printers require the general print monitoring to be enabled in the Collector. Starting from V6.18, print monitoring is disabled by default.

Applies to platforms:

Disabling AsyncRPC calls related to printing

As a workaround, to minimize the network load, Microsoft proposes to disable asynchronous remote procedure calls related to printing either in the print server or in the client devices. Please note that the following modifications may have an impact on other tools that rely on these settings. Ensure that you know what you are doing.

To disable printing-related AsyncRPC on the print server side (the device that shares the printer):

1. Log in to the print server as a user with administrator capabilities.
2. Type **Win+R** to open the Run Box.
3. Type in **regedit** in the Run Box and press **Enter** to launch the Registry Editor.
4. If prompted by User Account Control, click **Yes** to allow changes to the PC.
5. Locate and select in the Registry Editor the following key:
   HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Print
6. From the top menu, select **Edit > New > DWORD (32-bit) Value**
   1. Enter the name for the value **DisableRpcTcp**.
   2. Enter the data for the value **1**.
7. Reboot the device

To disable printing-related AsyncRPC on client devices (the devices that use the printer):

1. Log in to the client device as a user with administrator capabilities.
2. Type **Win+R** to open the Run Box.
3. Type in **regedit** in the Run Box and press **Enter** to launch the Registry Editor.
4. If prompted by User Account Control, click **Yes** to allow changes to the computer.
5. Locate in the Registry Editor the following key:
   HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\WindowsNT\Printers
6. From the top menu, select **Edit > New > DWORD (32-bit) Value**
   1. Enter the name for the value **EnabledProtocols**.
   2. Enter the data for the value **6**.
7. Reboot the device

Since setting this value individually for every client device may be cumbersome, it is recommended to modify the registry settings of your client devices through Group Policy. Please refer to an administration manual of Active Directory for more information.

Enabling SMB printer support in the Collector

After disabling printing related AsyncRPCs, enable SMB printer support in the Collector.

To enable SMB printer support in the Collector during its installation:

- Set the parameter **DRV_DSPS** to 0 in the options to the MSI.

To enable SMB printer support in a Collector that is already installed:

- Set the parameter **dsps** to 0 using the Nxtcfg tool.

The operations described in this article should only be performed by a Nexthink Engineer or a Nexthink Certified Partner.

If you need help or assistance, please contact your Nexthink Certified Partner.
Related references

- Information on printers and printing
- Collector MSI parameters reference table
- Nxtcfg - Collector configuration tool

Related concepts

- Printer
- Print job

Changing the thresholds of High CPU warnings

Overview

High CPU warnings for devices and executions are triggered when the CPU load exceeds some default values. The default values have been chosen to detect both significant high CPU loads in a device and the particular applications that cause high CPU load during their execution.

If you receive too many high CPU warnings in your setup, up to the point that they stop being meaningful, raise the default thresholds. To change the default thresholds, edit the configuration file of the Engine:

1. Log in to the CLI of the Engine.
2. Edit the configuration file:
   ```
   sudo vi /var/nexthink/engine/01/etc/nxengine.xml
   ```
3. Change the high CPU settings inside the tag **<aggregation>** (under **<config>**, **<engine>**). See below each possible individual setting.

Repeat this operation in every Engine of your setup.

The operations described in this article should only be performed by a Nexthink Engineer or a Nexthink Certified Partner.

If you need help or assistance, please contact your Nexthink Certified Partner.

Device warnings

The deprecated device warning **High thread CPU usage (deprecated)** is triggered when the CPU load in a device exceeds 80% of a single logical processor. To change that threshold, modify the value of the following setting:

```
<machine_high_cpu>80</machine_high_cpu>
```

The device warning **High overall CPU usage** is triggered when the CPU load is above 70%, taking into account all the logical processors of the device. This threshold is independently modifiable for each type of device (laptop, desktop, or server):

```
<normalized_high_cpu_laptop>70</normalized_high_cpu_laptop>
<normalized_high_cpu_desktop>70</normalized_high_cpu_desktop>
<normalized_high_cpu_server>70</normalized_high_cpu_server>
```

Execution warnings

By default, for any process to trigger a **High thread CPU usage** warning, it has to take more than 50% of CPU load. The threshold is controlled by the following setting:

```
<process_high_cpu>50</process_high_cpu>
```

In the case of the system process, the threshold is lowered to 40%. Change the default with the following setting:

```
<system_high_cpu>40</system_high_cpu>
```

Related references

- Errors and warnings for devices and executions

Automatic restart of unresponsive Engine

Overview

The Engine periodically resets a watchdog timer to indicate that it is running correctly. When not reset, the watchdog timer expires within ten minutes by default. In consequence, if the Engine is not able to reset the timer before ten minutes elapse, the timer triggers the restart procedure of the Engine.

Internal faults or very complex queries involving millions of events may render the Engine unresponsive. In these cases, the watchdog timer forces the Engine to restart anew and thus recover from potentially blocking situations.

Changing the timeout value

To change the default value of the watchdog timer:

1. Log in to the CLI of the Engine.
2. Optional: Verify the current value of the watchdog timer by typing in:
   ```
   nxinfo config -w | grep watchdog
   ```
   - The result is the configured value of the watchdog timer in seconds. For the default value of ten minutes, the result of the command should display 600 seconds:
     ```
     <watchdog_timeout>600</watchdog_timeout>
     ```
3. Set the new value of the watchdog timer. For instance, to double the default of ten minutes and make it twenty minutes (1200 s), type in:
   ```
   nxinfo config -s tweak.watchdog_timeout=1200
   ```
4. Restart the Engine for the new watchdog value to take effect:
   ```
   sudo systemctl restart nxengine@1
   ```

Maintenance operations

Logging in to the CLI

The command line interface (CLI) of the Nexthink Appliance gives you access to a terminal where you can inspect and control every aspect of the system by using all the power of the Linux shell.

To log in to the CLI, connect to the Appliance with the help of an SSH client as the user **nexthink**.

The operations described in this article should only be performed by a Nexthink Engineer or a Nexthink Certified Partner.

If you need help or assistance, please contact your Nexthink Certified Partner.

Special operation modes for the Engine and the Portal

When operating normally, the Engine receives and processes all information coming from the Collectors and sends data to the Portal over time. For demo purposes or other special reasons, you may want to alter the normal functioning of the Engine and the Portal. In this chapter, learn how to freeze the time in the Engine and in the Portal (demo mode) or how to make the Engine store device information only and filter all other events (zero config mode).

In addition, know how to deal with the compatibility mode of Internet Explorer when browsing the Portal. Lastly, if you have the Web and Cloud module activated, learn how to configure the Engine for recording HTTP connections with extended status codes from a proxy.

Setting up demo mode

While working with the Portal, imagine that you detect an interesting occurence in your network, such as a high rate of failures in a service at a particular time of the day. You may want to share your findings with other people in your team our with management. Ideally, you would like to replay the same situation at a later time to analyze what happened at that point in time with the help of all Nexthink products. To that end, you can back up the databases of the Engine and the Portal and restore them later in other instances in demo mode.

Demo mode freezes the time of the Engine and the Portal, so they do not evolve with the passing of time. Therefore, you consistently find the same data that was present when you made the backup in both the Engine and the Portal. To prevent data loss in your production environment, you must not use the production Engine and Portal to play your demos, but dedicated instances of the Engine and the Portal that you have installed elsewhere; for instance, a virtual machine in your personal desktop.

An Engine in demo mode does not process any packet coming from the Collector nor performs any kind of activity: it does not create new events in the database, it does not notify new alerts, it does not send or retrieve information from the application library, etc.

To set up the demo mode in the Engine:

1. Log in to the CLI of the appliance that hosts the demo Engine.
2. Edit the configuration file of the Engine that is found in `/var/nexthink/engine/01/etc/nxengine.xml` and set the **mode** tag to **static_time**:
   ```
   <config>
   <engine>
   <mode>static_time</mode>
   </engine>
   </config>
   ```
3. Restart the Engine:
   ```
   sudo systemctl restart nxengine@1
   ```

The keyword **static_time** forces the Engine to freeze its internal date and time to the moment right after the end of the last event included in its database. Since the time is frozen, the Engine no longer sends real-time service information to the Portal. For the Portal to work in sync with your demo Engine, the time set in the Portal must match the time in the Engine and the Portal must receive real-time services data from the Engine.

To get the time settings from the Engine and send the data of real-time services to the Portal, take these additional steps in the Engine appliance:

1. Call the function **now** in the Engine and note down the result. The function gives you the frozen time:
   ```
   nxinfo shell -e "call now()"
   ```
2. Schedule a cron job to send real-time service data to the Portal every 10 minutes:
   1. Execute in the CLI of the Engine:
      ```
      sudo crontab -e
      ```

2. In the vi text editor that opens, type in the following line:
   */10 * * * * /usr/bin/nxinfo lua --command "monitor:send_data_to_portal()"
3. Save your changes and quit the editor with the command:
   `:wq`

After Engine configuration, set the demo mode in the Portal:

1. Log in to the CLI of the appliance that hosts the demo Portal.
2. Optional: If the Portal has no configuration file yet, that is, if `portal.conf` does not exist in folder `/var/nexthink/portal/conf`, create it by copying the defaults from the sample configuration file:
   ```
   sudo -u nxportal cp
   /var/nexthink/portal/conf/portal.conf.sample \
   /var/nexthink/portal/conf/portal.conf
   ```
3. Edit the configuration file of the Portal:
   ```
   sudo vi /var/nexthink/portal/conf/portal.conf
   ```
4. Add the following lines, where **EngineTime** is the frozen time in the Engine that you noted down previously:
   ```
   # Demo mode
   globalconfig.portal.special.demo = true
   globalconfig.portal.special.static.time = "EngineTime"
   ```
5. Save your changes and quit the editor:
   `:wq`
6. Restart the Portal:
   ```
   sudo systemctl restart nxportal
   ```

Now you have your Engine and Portal ready in demo mode. You may have to wait up to ten minutes for real-time services to receive data from the Engine though.

Stopping the time in the Engine

With the **static_time** option, the Engine selects the optimal point in time to freeze the time in the Engine for a demo. This time corresponds to the instant right after the occurrence of the last event recorded in the database of the Engine. In the case that you want to freeze the time of the Engine to a different point in time, you can do it by setting the following option in the configuration file of the Engine (*/var/nexthink/engine/01/etc/nxengine.xml*):

```
<config>
<engine>
<tweak>
<static_now>time</static_now>
</tweak>
</engine>
```

```
</config>
```

Where **time** is in the format YYYY-MM-DD**T**HH:MM:SS (e.g. 2014-01-01T18:00:00).

This option should be used with care because it can leave events that were originally in the database out of the time range of the Engine or make them too old. Use preferably the **static_time** option for your demos unless you have a very specific requirement.

Storing only device information in the Engine

This mode of operation can be used to deploy a large number of Collectors in a setup with several Engines. The deployment is done in two phases. During the first phase, all Collectors send information to one special Engine that is configured to store device information only. Then, in the second phase, Collectors are classified and definitively configured to send data to a normally operating Engine. For the details on the procedure, please contact Nexthink Customer Success Services.

This special mode of operation of the Engine is known as *zero config* mode. An Engine in zero config mode shows the following properties:

- The Engine processes and stores only device information coming from the Collectors, namely, the MAC address, IP address and SID of the devices. All activities and information related to other objects are discarded.
- Devices are created with a special storage policy called **inventory**. A device with this storage policy is never removed from the database in spite of having no events associated.
- The number of devices is not enforced by the license.
- The Engine rejects any connection from the Portal.
- The communication of the Engine with the application library is disabled.

To set up zero config mode, please contact Nexthink Support.

Dealing with compatibility mode of IE in the Portal

The Portal is usually best displayed with the latest rendering capabilities of modern browsers. When working with Internet Explorer, though, you or your organization may have set the browser to *compatibility mode* because of some legacy web applications which are key to your business and are best rendered in older versions of IE.

Even with *compatibility mode* on, web sites can still indicate Internet Explorer to use its latest rendering engine instead. To that end, they use a particular HTTP header when serving web pages. To make the Portal work in this mode, so it tells Internet Explorer to use its most recent render version:

1. Log in to the CLI of the appliance hosting the Portal.
2. Optional: If the Portal has no configuration file yet, that is, if `portal.conf` does not exist in folder `/var/nexthink/portal/conf`, create it by copying the defaults from the sample configuration file:
   ```
   sudo -u nxportal cp
   /var/nexthink/portal/conf/portal.conf.sample \
   /var/nexthink/portal/conf/portal.conf
   ```
3. Edit the configuration file of the Portal:
   ```
   sudo vi /var/nexthink/portal/conf/portal.conf
   ```
4. Add the following line,:
   ```
   globalconfig.portal.http.compatibility-mode = true
   ```
5. Save your changes and quit the editor:
   ```
   :wq
   ```
6. Restart the Portal:
   ```
   sudo systemctl restart nxportal
   ```

Recording web requests with extended connection status codes

During normal operation, the Engine ignores web requests with connection status codes between 300 and 499 by default. These extended status codes may be issued by proxies when establishing a secure connection with a server on a client request.

Starting from Engine 5.2.8, you can tell the Engine to record these connections by logging in to the CLI and typing the following command:

```
sudo nxinfo config --set \
web_monitoring_accept_proxy_extended_status_codes=true
```

Restart the Engine for the new configuration to take effect and beware that acknowledging this kind of connections may significantly increase the number of recorded web request events and, therefore, decrease your time interval for data retention.

The operations described in this article should only be performed by a Nexthink Engineer or a Nexthink Certified Partner.

If you need help or assistance, please contact your Nexthink Certified Partner.

Related tasks

- Logging in to the CLI
- Engine backup and restore
- Portal backup and restore

Changing the default ports in the Appliance

Overview

Nexthink Appliances listen to a set of default network ports to communicate with the other Nexthink components and serve external requests. Some of the default ports are configurable.

For the Engine Appliance, configure:

- The transport protocol (TCP or UDP) and the port numbers to communicate with the Collectors.
- The port number of the Web API.
- If rule-based Collector assignment is activated, specify the port for the assignment service as well.

For the Portal Appliance, configure:

- The port number on which the Portal listens to assignment requests from the Collectors, when rule-based Collector assignment is enabled.

Changing the ports in the Engine

By default, the Engine allows the communication with the Collectors through TCP port 443 only. This option requires that you have installed your own digital certificates in the Engine.

Based on how you configured the Collectors during installation, you may allow other custom TCP or UDP ports in the Engine to handle the communication between the Engine and the Collectors. Note that these options are not mutually exclusive; that is, you can have different sets of Collectors communicating with the Engine through different ports.

Remember that a single TCP connection between Collector and Engine can convey all available data: Engage, Act, updates, rule-based assignment and,

optionally, device information and activity; whereas the UDP connection between Collector and Engine solely transfers device information and activity data, requiring an additional TCP connection for full connectivity.

To change the default ports in the Engine:

1. Log in to the Web Console of the Appliance that hosts the Engine.
2. Select the **APPLIANCE** tab at the top of the Web Console.
3. Click **Security** on the left-hand side menu.
4. Under **Nexthink Ports**, find the following configurable ports:
    1. Tick **Allow Collector TCP port 443** to enable the communication with the Collectors through the default TCP port 443.
        - ◊ Remember that this option requires your own digital certificates.
    2. Tick **Allow Collector TCP port** to enable the communication with the Collectors through a custom TCP port.
        1. Type in the port number for the custom TCP connection of the Collector with the Engine. For this connection, you must use a port that does not require root privileges; that is, the port number must be above 1024. Default is 8443.
        - ◊ This option can work with the default certificates generated during the federation of the Appliances.
    3. Tick **Allow Collector UDP port** to enable the communication of the Collectors through UDP.
        1. Type in the port number for the UDP connection that sends device information and user activity to the Engine. Default is 999.
        - ◊ The UDP connection does not require digital certificates.
    4. In **Web API**, type in the port number of the TCP connection used to integrate with the Engine via the NXQL language. This connection also requires a port number higher than 1024. Default is 1671.

5. Click **SAVE** to make your changes permanent. The Engine restarts if you

changed either the UDP port or the Web API port.

After saving your changes, remember to configure the Collectors in accordance with the selected port numbers, either during their deployment or on already deployed Collectors with the Collector Configuration Tool.

In the same way, adapt your integrations to use the new Web API port of the Engine and, to test your NXQL queries, include the new port number in the URL of the NXQL Editor.

Changing the ports in the Portal

The Portal receives assignment requests from the Collectors through their TCP connection when rule-based assignment is enabled.

To change the configured port number of the TCP connection between the Collector and the Portal:

1. Log in to the Web Console of the Appliance that hosts the Portal.
2. Select the **APPLIANCE** tab at the top of the Web Console.
3. Click **Security** on the left-hand side menu.
4. Under **Nexthink Ports**, find the only configurable port:
    1. Tick **Allow Collector TCP port 443** to enable the communication with the Collectors through the default TCP port 443.
        - ◊ Remember that this option requires your own digital certificates.
    2. Tick **Allow Collector TCP port** to enable the communication with the Collectors through a custom TCP port.
        1. Type in the port number for the custom TCP connection of the Collector with the Portal. For this connection, you must use a port that does not require root privileges; that is, the port number must be above 1024. Default is 8443.
        - ◊ This option can work with the default certificates generated during the federation of the Appliances.
5. Click **SAVE** to make your changes permanent.

Remember to configure the Collectors with the same TCP port number as set in the Portal either during their installation or later via the Collector Configuration tool.

Changing the ports in a single Appliance

If the Portal and the Engine are installed in a single Appliance, the **Nexthink Ports** section appears twice on the **Security** page of the Web Console:

- The first holds the three ports to configure the Collector communication with the Engine.
- The second holds the TCP port number for the Collector to connect to the Portal.

Follow the instructions in the two preceding sections to change the port numbers accordingly.

The operations described in this article should only be performed by a Nexthink Engineer or a Nexthink Certified Partner.

If you need help or assistance, please contact your Nexthink Certified Partner.
Related tasks

- Importing and replacing Certificates
- Specifying the Collector Assignment port

Related references

- Connectivity requirements
- Appliance Hardening
- Introducing the Web API V2
- Nxtcfg - Collector Configuration Tool

Centralized Management of Appliances and Engines

Overview

The Centralized Management solution lets central administrators perform selected management actions on connected Appliances from the Portal. The solution also lets you get information from the Engines running on the connected Appliances, as well as perform some maintenance actions on them. This avoids the need to connect individually to each Appliance via the Web Console for performing commonplace management actions.

Centralized Appliance Management

Enable Appliance for centralized management

To enable the centralized management of an Appliance:

1. Log in to the Web Console of the Appliance to be centrally managed.
2. Click the **Appliance** tab at the top of the window.
3. Select the section **Accounts** from the left-hand side menu.
4. Under **Portal remote management account**, tick the box **Enable Portal remote management account**.
5. Type in twice the password to manage the Appliance from the Portal. If no value is entered, the default password is **api**.

Adding an Appliance for centralized management

To add a new Appliance to centralized management in the Portal:

1. Log in to the Portal as a central administrator.
2. In the **ADMINISTRATION** top menu, select the **Appliances** dashboard under the section **SYSTEM CONFIGURATION**.
3. In the **Appliances** section, click on the plus icon.
4. Enter the Appliance **Name**, **Address**, **Port**, the **Remote management password** and **Description**. Note that in the **Address** field, one should enter either the IP address or the DNS name in order to match what was entered in the **Address** field of the Engines dashboard in the Nexthink Portal section (refer to Connecting the Portal to the Engines). Validate your entry to add the Appliance to the centralized management.

Central configuration of Appliances

Select the Appliances to configure and press on the configuration icon. You have several configuration options:

Enable Web Console
: Allow the connection of users to the Web Console in the Appliance.

Disable Web Console
: Prevent users from connecting to the Web Console in the Appliance.

Configure NTP
: Nexthink synchronizes with NTP servers using the Network Time Protocol. Enter the list of NTP server addresses seperated by a space. One option is to use the NTP servers offered by http://www.pool.ntp.org/en/, such as 0.pool.ntp.org, 1.pool.ntp.org, etc.

Reboot
: Reboot the machine that hosts the Appliance.

Add or remove Engines related to an Appliance

1. Click on the information icon corresponding to the Appliance of interest.
2. Click on the chain icon to start or stop the central management of an Engine. If the central management is started, the Engine will show up in the Engines section of the Appliances dashboard.

Edit or remove an Appliance from centralized management

- To edit an Appliance, click on the corresponding pencil icon
- To delete an Appliance, click on the corresponding trashcan icon

Centralized Engine Management

The Engines section shows the Engines associated to the Appliances managed centrally. Select the Engine to configure and press on the configuration icon. You have several configuration options:

Configure LDAP
: Set up your LDAP servers to get Active Directory information for Nexthink objects.

Information
: Displays a table with the different configuration settings and general information about the Engine.

Refresh DNS
: Use this option to refresh DNS information on the Engine. This can be necessary if you wish to reflect changes on a DNS server (configuration changes or updates in the resolution of a particular destination). The Engine resolves new Destinations, but it does not refresh their DNS automatically if it changes.

Refresh LDAP
: This option is generally used in a scenario where the LDAP server integration is performed after Engine installation. In this case, launch this option to trigger the Engine refresh of its LDAP information. The Engine gets information from the configured LDAP servers on every new user detected.

Restart
: Stop and restart the Engine. Note that restarting the Engine results in a temporal loss of data received from Collectors during the time of starting up.

Related tasks

- Managing Appliance accounts
- Connecting the Portal to the Engines

Monitoring the performance of the Appliance

Overview

To keep track of the hardware and network resources consumed during their operation, Nexthink Appliances internally run collectd.

The `collectd` daemon gathers the following performance metrics by default:

- **aggregation**: Sum and average of the usage over all CPUs.
- **cpu**: Amount of time spent by the CPU in various states; namely executing user code, executing system code, waiting for IO-operations and being idle.
- **df**: File system usage information; basically how much space on a mounted partition is used and how much is available. Named after the `df(1)` UNIX command and very similar to it.
- **disk**: Performance statistics of hard-disks and, where supported, partitions.
- **interface**: Information about the traffic (bytes, packets and errors per second) of interfaces.
- **load**: System load. Rough overview over the usage of a machine.
- **memory**: Physical memory usage as reported by the operating system.
- **ntpd**: Drifts of the `ntpd` daemon from the reference NTP server.
- **processes**: Process-related information about the Engine and the Portal.
- **swap**: Swap usage.
- **vmem**: Virtual memory usage.

Getting performance data

To get the RRD files generated by `collectd`:

1. Log in to the Web Console of the Appliance to monitor (Portal or Engine).
2. Select the **APPLIANCE** tab at the top of the Web Console.
3. Click **Monitoring** on the left-hand side menu.
4. Find the **Performance data** section:

5. Click **DOWNLOAD**. Your web browser downloads a compressed tape archive file (`tar.gz` extension) that holds the `collectd` performance data and includes the current date and time in its name.

Related references

- Collectd (external)

Configuring the system log

Overview

Syslog is a de facto standard solution for logging messages in UNIX-derived systems, such as the operating system of the Appliance. Programs use the syslog system call to send arbitrary messages to the syslog service. In addition to the message itself, two parameters are provided to the syslog system call: the facility and the level. The facility refers to the type of program that required the logging of a message. Facilities are named after typical UNIX services such as mail, ftp, or cron, subsystems such as the kernel, the printer, or the clock, and others are reserved for local use. The level indicates the importance or seriousness of the message. Possible values for the level are critical, warning, notice, etc.

The Nexthink components in the Appliance use the system log service to keep a record of significant occurrences, including:

- Audit trail events
- System alerts
- Investigation-based global alerts
- Internal state of the Engine

For writing to the system log, the Appliance relies on the *rsyslog* package, which has become the default logging service in many Linux distributions. Although it adds new advanced features, rsyslog still keeps backwards compatibility with the configuration files of the original syslog daemon. If you are familiar with the

configuration of rsyslog, you may easily customize the output of the logs written by the Nexthink components and adapt them to your needs.

From this point on, we may refer to rsyslog as syslog when we talk about the logging service in general and not about specific features of rsyslog.

Default configuration and log files

The configuration file for rsyslog is found in `/etc/rsyslog.conf`. For the sake of clarity, the specific modifications of Nexthink to the configuration of rsyslog are stored in a separate file, which is found in `/etc/rsyslog.d/nx_rsyslog.conf`. This file is applied to the main configuration file by means of an include directive in `/etc/rsyslog.conf` that reads all additional configuration files in the `/etc/rsyslog.d` folder.

The default configuration of Nexthink dispatches log messages to different files depending on their content. Find these log files under `/var/log/nexthink`:

File	Purpose
`alert.log`	• System and investigation-based global alerts • Debug info from logger (rsyslog)
`audit.log`	Audit trail events
`engine.log`	Internal state of the Engine

By default, the Portal, the Engine, and the Web Console write their audit events to the `audit.log` file of the Appliance that hosts each one of them. Only the Engine writes to the `alert.log` and the `engine.log` files. In turn, the Portal does not use the syslog service to write information about its internal state, but its own logging tools. The internal logs of the Portal are found under `/var/nexthink/portal/log`.

The audit and alert logs are suitable for automatic processing, since their format is well-defined and stable. However, the format of the internal logs of the Engine is not guaranteed and may be subject to change. Therefore, do not rely on the contents of the Engine log for automating your processes.

Nexthink uses UTF-8 encoding for its log messages. Rsyslog preserves the encoding.

Configuration of alerts

In addition to email, you can use the system log as a notification mechanism for both system and global alerts. For global alerts, you need to enable syslog notification when creating the alert.

The part of the syslog configuration file `/etc/rsyslog.d/nx_rsyslog.conf` which is relevant for alerts is shown below:

```
 $template
RFC5424format,"<%pri%>1 %timestamp:::date-rfc3339% %hostname%
 %programname% %procid%%msg%\n"
 ...
 # alerts
 local5.=notice -/var/log/nexthink/alert.log;
 ...
 # alerts
 local6.=notice -/var/log/nexthink/alert.log; RFC5424format
```

The first line defines an output format for syslog messages by means of a template. The template is named *RFC5424format* because it follows the recommended format for syslog messages which is described in the most recent Internet standard about the syslog protocol: RFC 5424. The template defines the output to be composed of a priority number followed by the timestamp, the host name, the program name, the id of the process which issued the syslog message and the message itself. Once defined in this way, a template can be applied to one or several message filters.

For alerts, you can see that we declare two filters in the syslog configuration file, depending on the facility specified to log the alert. Both filters are instructed to write their output to the same file: `/var/log/nexthink/alert.log`. The minus sign before the file name is there to improve the performance of the syslog daemon. It indicates that syslog output to the file is buffered, so the syslog system will not directly write to the filesystem but to a buffer in memory and then really write to the disk once the buffer is full. The two filters however accept messages from different facilities. If the facility used is local5, rsyslog will use the default syslog output format. On the other hand, if the facility used is local6, rsyslog will use the output format defined by the template *RFC5424format* for every logged alert.

To choose between legacy (local5) or modern (local6) format for the log messages of global alerts, set the following parameter in the main section of the

configuration file of the Engine (`/var/nexthink/engine/01/etc/nxengine.xml`):

```
  <syslog>
   <legacy_alert_format>true</legacy_alert_format>
 </syslog>
```

For details on the formatting of alerts, see the article on integrating alerts.

Customizing the syslog configuration

To preserve the standard log files and formats used by Nexthink, avoid directly editing the Nexthink configuration file for rsyslog `/etc/rsyslog.d/nx_rsyslog.conf`. Instead, to customize the destination and format of the logs, write your own formatting templates and rules (aka selectors) in a custom configuration file located here: `/etc/nexthink/nx_rsyslog_custom.conf`. The Nexthink configuration file for rsyslog includes the custom configuration file before declaring its own rules:

```
$IncludeConfig /etc/nexthink/nx_rsyslog_custom.conf
```

Custom rules are thus processed before standard Nexthink rules. Again, to preserve the default configuration and standard log files of Nexthink, do not write any rules in the custom configuration files that stop the processing of messages by the standard Nexthink rules.

Logging to a remote server

The syslog protocol lets you send log messages through the network to be consumed by syslog servers other than the local Appliance.

To send log messages to a remote syslog server, copy each line in the syslog configuration file of Nexthink that you want to modify to your custom configuration file and substitute the name of the log file for the name or IP address of the receiving server. The name of the server must be preceded by a single or a double at-sign (@ or @@), depending on whether you want to send the log messages via UDP or TCP, respectively. Follow the name or IP address of the remote server by a colon (**:**) and the port number where the server is listening for syslog messages. For example, to send different types of log messages to remote servers.

Remember to use either local5 or local6 entries in slave Appliances, depending on the setting for the Engine `legacy_alert_format` to be true or false, respectively. For master Appliances, recall that the Portal always uses syslog local5 facility and exclusively for audit events:

```
# Send general log to a server listening to UDP port 514
local5.=debug;local5.=info;local5.=error @udp-server.example.com:514;
nxFormat

# Send audit logs to a server listening to UDP port 514
local5.=warning @udp-server.example.com:514; nxAuditFormat

# Send alert logs to a server listening to TCP port 10514
local5.=notice @@tcp-server.example.com:10514;
```

Note that you do not have to choose between saving the logs in a file and send them to a remote server. It is possible to do both by repeating the same line in the syslog configuration changing the destination of the logs. Check the rsyslog documentation for options when sending log messages through the network, specially when using TCP.

Logging accesses to the CLI

Besides user access to the Nexthink components such as the Finder and the Portal, the access to the command line interface of the Appliance is an event of interest in the audit trail.

To filter the syslog messages related to accesses to the CLI of the Appliance and send them to a destination of your choice, specify the programs that control the command line inside conditional statements in your custom configuration file for syslog:

```
# Log access to the CLI
if $programname == 'sshd' then -/var/log/nexthink/audit.log
if $programname == 'sudo' then @udp-server.example.com:514
if $programname == 'login' then @@tcp-server.example.com:10514
```

These programs control remote access (**sshd**) to the Appliance, logging in (**login**) to the Appliance, and execute as the superuser (**sudo**) in the Appliance. In the example above, each program is sending its output to a different destination, but you can send the output of all programs to the same destination.

Restarting the Engine and the syslog service

Restart the Engine if its configuration file required any change:

```
sudo systemctl restart nxengine@1
```

After any modification to the configuration files of syslog, restart the service for the changes to be effective:

```
sudo systemctl restart rsyslog
```

Related tasks

- Creating an investigation-based alert
- Integrating alerts
- Examining the logs in the Portal

Related references

- System alerts
- Audit trail
- Rsyslog (external link)

Examining the logs in the Portal

The log files of the Portal are located in the Appliance that hosts it under:

/var/nexthink/portal/log/

The names of the all the log files of the Portal are prefixed with the word **portal_**.

Log files

The names of the log files reflect the current running mode of the Portal. Note that, for medium and large modes, subsystems of the Portal write to different log files:

- **portal_*<running mode>*.log**: Standard log file.

- **portal_activity_*<running mode>*.log**: Extract higher level information such as Engine connection state, size of the database, memory consumption of the JVM.
- **portal_*<running mode>*.err** : Standard error stream with low-level error messages (for support and unexpected cases).
- **portal_*<running mode>*.out** : Standard output stream with low-level information (for support and unexpected cases).

Running Modes

Depending on the size of the Portal database, there are different running modes.

To know the current running mode, take a look at the file:

/var/nexthink/portal/conf/startup.properties

The name and the number of log files depend on the running mode, as listed below:

Small mode

- **SMALL**: Single node running in single JVM mode.

Medium mode

- **MEDIUM_UI**: Portal UI, Portal compute and HTTP server when running in dual JVM mode.
- **MEDIUM_INFRA**: Content manager, login manager, communication layer, real-time layer when running in dual JVM mode.

Related tasks

- Allocating resources for the Portal

GDPR - Retrieving or anonymizing personal data

Overview

The *General Data Protection Regulation* (GDPR) introduces a single legal data protection framework for both businesses and individuals within the European Union (EU). The GDPR was approved on April 2016 and became directly

applicable on 25 May 2018. As of that date, all companies (including those outside the EU) that control or process personal data relative to EU residents are obliged by the regulation to satisfy certain user rights.

When using Nexthink, companies store data that describes the digital behavior of end-users and allows their personal identification. This kind of personal data usually lies in the context of employment; that is, end users are generally employees of the company that controls and processes their data, although this may not always be the case. Even if the GDPR allows for specific rules to the processing of personal data in the context of employment (see article 88), the protection of this data is still under the GDPR, as long as your employees are EU residents. Consult your legal department in case of doubt.

GDPR in the Web Console

Prerequisites

Both the retrieval of user data from the Portal and the anonymization of user data in the Portal require no special feature or additional module.

On the other hand, the retrieval of user data from the Engine makes use of the Web API and thus requires the purchase of the Nexthink Integrate module. In addition, the following conditions apply:

- The Engines are federated with the Portal.
- The Web Console in the Portal either trusts the server certificates of the Engines or disables certificate validation for GDPR.

Functions

To help you comply with the regulation, the Web Console includes a GDPR-specific section to let you run specialized scripts that perform the following functions:

Retrieve user data
: Article 15 of the GDPR grants data subjects the right to access their personal data. Run the script in this mode to retrieve all the data relative to a particular user or device.

Anonymize user data
: Article 17 of the GDPR grants data subject the right to be forgotten. The script transforms the name of a user or a device to render the personal data unidentifiable.

These functions are available in either the Portal or the Engine databases according to the following table:

Function	Portal	Engine
Retrieve user data	Yes	Yes
Anonymize user data	Yes	No

Retrieving user data

To retrieve user data:

1. Log in to the Web Console of the master appliance.
2. Select the **APPLIANCE** tab at the top of the Web Console.
3. Click the **GDPR** section on the left-hand side menu.
4. Under **Retrieve user data**, check either:
    - **Username**, to retrieve all the data related to a particular user.
      Type in the name of the user as it appears in either the Finder or the Portal.
    - **Device name**, to retrieve all the data related to a particular device.
      Type in the name of the device as it appears in either the Finder or the Portal.
5. Tick either one or both:

- **Engine data** to retrieve user or device data from the federated Engines (requires Nexthink Integrate).
- **Portal data** to retrieve user or device data from the Portal.

6. Click **DOWNLOAD DATA**.
    1. In the **GDPR Retrieve data** dialog that shows up, read first the confirmation message about the operation that you are about to make.
    2. Type in the credentials of a Nexthink user with the **Data privacy** property set to **none (full access)** so that the user has access to the Web API:
        1. As **Username**, type in the name of the existing Nexthink user.
        2. As **Password**, type in the password of the existing Nexthink user.
    3. If you want to retrieve Engine data, ensure that the Engines that hold the user data show up in the **Engines List**. Remember that you can only retrieve data from federated Engines.
    4. Click **CONTINUE**.
7. Wait for the data retrieval to finish.
8. Optional: Click **CLOSE** to cancel the data retrieval process.
9. Once the data retrieval is finished, the download of the file `gdpr-data.tar.gz` starts automatically.
    - If the download does not start automatically, click the link **CLICK HERE**.
10. Click **CLOSE**.

The downloaded file is the result of compressing a set of CSV files that hold all the recorded activity of the user (if Engine data was retrieved) and the results of metrics that have any information related to the user (if Portal data was retrieved).

Anonymizing user data

To anonymize user data in the Portal:

1. Log in to the Web Console of the master appliance.
2. Select the **APPLIANCE** tab at the top of the Web Console.
3. Click the **GDPR** section on the left-hand side menu.
4. Under **Anonymize user data**, check either:
    - **Username**, to anonymize the data related to a particular user.
      Type in the name of the user as it appears in either the Finder or the Portal.
    - **Device name**, to anonymize the data related to a particular device.

Type in the name of the device as it appears in either the Finder or the Portal.

5. Tick **Irreversibly anonymize Portal data** to confirm your choice.
6. Click **ANONYMIZE DATA**.
    1. In the **GDPR Anonymize Portal data** dialog that shows up, read first the confirmation message about the operation that you are about to make.
    2. Type in the credentials of a Nexthink user with the **Data privacy** property set to **none (full access)** so that the user has access to the Web API:
        1. As **Username**, type in the name of the existing Nexthink user.
        2. As **Password**, type in the password of the existing Nexthink user.
    3. Click **ANONYMIZE**.
7. Wait for the data anonymization to finish.
8. Optional: Click **CLOSE** to cancel the data anonymization process.
9. Once the data anonymization is finished, the user is anonymized in the Portal.
10. Click **CLOSE**.

Other mechanisms to comply with GDPR

In addition to the GDPR menu of the Web Console, remember that Nexthink provides you with other mechanisms that can help you comply with the GDPR:

Removal of devices
: Helps you comply with the right to erasure by completely removing all the stored information about a particular device from the Engine.

Anonymization in traffic redirection
: Helps you comply with the GDPR by removing all the information that can potentially identify a person from the Collector traffic received by the Engine.

Related tasks

- Removing devices

Related references

- Access rights and permissions

- Redirecting Collector traffic

Finding out unlicensed devices

Overview

When ordering a license for Nexthink, you must specify the number of end-user devices on which you want to deploy the Collector. From that total, you allocate the maximum number of devices to each Engine. If the number of devices reporting to an Engine is actually higher than the maximum number of allocated devices for that Engine, the Engine issues a system alert and discards the devices in excess. No data are stored for these *unlicensed* devices and, therefore, they are not visible in the Finder or the Portal.

Learn here how to find out unlicensed devices from the Engine log, so you can adapt your license or your allocation strategy accordingly.

The operations described in this article should only be performed by a Nexthink Engineer or a Nexthink Certified Partner.

If you need help or assistance, please contact your Nexthink Certified Partner.

Reaching the maximum number of devices

When the Collector is installed in a device, it soon starts sending information relative to the device to its configured Engine. At that point, the Engine realizes that the device exists. When the Engine identifies a new device in the network, it increases its count of devices and checks its maximum number of allocated devices, according to the distribution of the license. If the count is within the limit, the Engine keeps the device and accepts the data coming from the Collector. On the other hand, if the count is above the limit, the Engine discards the device and any other data coming from the Collector.

When the Engine discards a device because it exceeds the maximum number of allocated devices, it logs the following message:

```
machine <Name>|<MAC>[|<MAC>]* out of license
```

The message states the name of the device followed by its detected MAC addresses. While the Engine is running, it logs the message only once per device, even if the Collector in the device may keep sending information. If the Engine is restarted, it logs the message again as soon as it receives data from

the offending Collector.

Note that if you choose to ignore a particular device by setting its storage policy to **none** or **remove** , the device will never appear in the log as unlicensed.

Looking for unlicensed devices in the log

To extract the list of unlicensed devices from the Engine log, look for messages matching the format shown above.

If the Engine has been running for a long time, the information about unlicensed devices in the log may be outdated (you may have removed some Collectors and installed some others in the meantime or the log file may have rotated). To make sure that you get up-to-date information regarding unlicensed devices from the Engine log, restart the Engine previous to examining the log file:

1. Log in to the CLI of the appliance hosting the Engine.
2. Stop the Engine:
   ```
   sudo systemctl stop nxengine@1
   ```
3. Make a backup of the log file:
   ```
   sudo cp /var/log/nexthink/engine.log
   /var/log/nexthink/engine.log.bk
   ```
4. Reset the log file:
   ```
   sudo truncate -s 0 /var/log/nexthink/engine.log
   ```
5. Start the Engine:
   ```
   sudo systemctl start nxengine@1
   ```
6. Wait for the Collectors to send information.

After waiting for a reasonable amount of time (one full day, for instance), examine the Engine log as described below:

1. Type in the following command to retrieve the list of *out of license* messages:
   ```
   sudo grep -E ".+machine.+out of license"
   /var/log/nexthink/engine.log
   ```
2. Optional: Count the lines of the previous result to get the total number of devices in excess:
   ```
   sudo grep -E ".+machine.+out of license"
   /var/log/nexthink/engine.log | wc -l
   ```

Related tasks

- Setting up a software license
- Establishing a privacy policy

- Removing devices

Related references

- System alerts

Removing devices

Manually removing devices

To manually remove a device from the Finder:

1. Log in to Finder with administrative rights.
2. Type the name of the device in the Search field.
3. Right-click the device in the results of the search and select **Drill-down**.
4. Right-click the device listed and select **Edit...** (or type **Ctrl+Alt+E**). The **Edit device** dialog shows up.
5. Select the option **remove** from the list **Storage** at the bottom of the dialog.
6. Click **Apply**. The device is marked for removal.

The Finder still displays the device until the Engine removes it from the database. During the nightly cleanup, the Engine removes from the database the devices that were not active for the last 24 hours and whose storage policy is set to **remove**.

Uninstall the Collector from the devices being removed to stop them from sending new activity data to the Engine. Failing to do so results in the Engine not removing the device from the database or, if the device was inactive for more than 24 hours and actually removed from the database, recreating the device in the database as soon as the Engine receives new data from it.

Once the device is tagged as **remove** the Portal almost immediately frees one license from the pool.

Applies to platforms:

Automatic removal of inactive devices

In case that:

- A device is inactive for more than 90 days (configurable).

- There are no events associated to the device left in the database of the Engine.

The Engine purges all the data related to the device and automatically frees one license from the pool.

Changing the maximum inactivity period of devices

Modify the maximum inactivity period of devices for the Engine to identify a device as inactive either more quickly or more slowly and, accordingly, remove it from its database. Note that modifying the maximum inactivity period of devices is local to each Engine.

To modify the maximum inactivity period of devices:

1. Log in to the CLI of the Appliance that hosts the Engine.
2. Edit the configuration file of the Engine:
   ```
   sudo vi /var/nexthink/engine/01/etc/nxengine.xml
   ```
3. Inside the limit section, set the new inactivity period in seconds (default value is 7776000 seconds, that is, 90 days):
   ```
   <limit>
   <max_inactivity_period>7776000</max_inactivity_period>
   </limit>
   ```
4. Save your changes and exit by typing in:
   ```
   :wq
   ```
5. Restart the Engine:
   ```
   sudo systemctl restart nxengine@1
   ```

Beware that setting the maximum inactivity period too low may result in an inefficient removal and recreation of devices with regular inactivity intervals, when these intervals are longer than the specified maximum period.

The operations described in this article should only be performed by a Nexthink Engineer or a Nexthink Certified Partner.

If you need help or assistance, please contact your Nexthink Certified Partner.
Related references

- Nightly tasks schedules timetable

Installing third-party software in the Appliance

The Appliance consists of a Linux-based operating system on which you can install the Portal or the Engine. The software packages included in the Appliance have been carefully selected and fine tuned to work together with both Nexthink products in order to deliver the best performance possible. Both the Portal and the Engine are very demanding in terms of computing resources and they usually require the full dedication of the hardware specified to run them.

Therefore, the installation of third-party software that competes for computing resources with the Nexthink products in the Appliance can degrade the overall performance of the Appliance or hinder the proper functioning of the Portal or the Engine.

As an exception, Nexthink recommends the installation of VMware Tools in those virtualized Appliances that run on VMware products.

Third-party software

The installation and the update of third-party software must be supervised by Nexthink Customer Success services.

The update of a Nexthink appliance on which a third-party software is installed must also be supervised by Nexthink Customer Success services.

In case of a performance issue or of any issue that could be related to third-party software, Nexthink Support may ask the customer to uninstall the software.

Installing typical third-party tools

Usually, you may want to install third-party software in the Appliance to perform any of the following tasks:

- Backup the Appliance
- Monitor the Appliance
- Protect the Appliance against computer viruses

The tools that typically perform these tasks may have a major impact in the performance of the system; therefore, Nexthink recommends not to install any additional tool. Should you choose to go ahead and install third-party software (because it is mandated by the security policy of your company, for example), we strongly recommend that you first test your setup in a pre-production

environment.

Backup the Appliance

Starting from Nexthink V4.1, the Appliance includes an automatic backup mechanism that lets you push all the database and configuration files to a shared directory. Configure the automatic backup of the Appliance from the Web Console to recover from a full or partial data loss.

If you are compelled to install a third-party backup tool, schedule it to perform the backup when the Appliance is less active and always test it first in a pre-production environment. Depending on the product that you installed in the Appliance, follow the corresponding piece of advice:

Engine
: The Engine is less active during the night, when it receives less data from Collectors and it has finished the cleanup of its database. Schedule the backup at around 04h30.

Portal
: The Portal is less active when fewer users are connected to it and it is not collecting data from the Engine. Since data collection starts at 01h00 and it can last for several hours, schedule the backup of the Portal between the end of the working hours and 01h00.

Monitor the Appliance

Currently, Nexthink Appliances run collectd to monitor their own performance and resource consumption.

If you really need to install another third-party monitoring tool in the Appliance, be specially careful if it is the Appliance that hosts the Engine. A monitoring tool can greatly interfere with the Engine during periods of high activity.

Protect the Appliance against computer viruses

The Appliance is always delivered with the latest security updates of the underlying Linux-based operating system. The risk of vulnerabilities is thus reduced to a minimum.

If an antivirus software is installed, please make sure that the following items are not scanned, as it would drastically impact the performance:

- Engine DB - /var/nexthink/engine/01/data/nxengine.db

- Portal directory - /var/nexthink/portal
- RAM of the Engine Appliance
- RAM of the Portal Appliance

Related references

- Nexthink Appliance (hardware requirements)
- Planning for disaster recovery (backup)
- Monitor the performance of the Appliance

- Collectd (external)

Installing VMware Tools in the Appliance

Nexthink recommends installing VMware Tools in any Appliance that runs on top of VMware virtualization products such as vSphere. VMware Tools significantly improves the performance and manageability of virtualized Appliances.

Starting from Nexthink V6, the Appliance is distributed with the **open-vm-tools** package already pre-installed. Therefore, no action is required on your part. When you deploy the Appliance in a VMware environment, it directly benefits from the features provided by the package. In addition, the package is automatically updated via the Appliance updates whenever a new version is available.

If for some reason you need to install the commercial version of VMware Tools, uninstall the open-vm-tools package first and then proceed as follows. Note however that VMware recommends the use of open-vm-tools on those platforms where the package is available, so **do not install the commercial version of VMware Tools** unless you really know what you are doing.

To install the commercial version of VMware Tools in the Appliance:

1. Open the vSphere Web Client and log in to connect to your vCenter Server.
2. On the left-side pane, click **vCenter** and select **Virtual Machines** from the **Inventory Lists** section.
3. Click the name of the virtual machine that runs the Appliance.
4. In the **Summary** tab, a yellow warning box displays the message **VMware Tools is not installed on this virtual machine**.

5. Click the link to the right of the warning message that reads **Install VMware Tools**.
6. Click **Mount** in the pop up dialog. A virtual CD with VMware Tools is now attached to your VM.
7. Open a terminal connection to the Nexthink Appliance (e.g. click **Launch Console** or connect to it via ssh) and log in to its CLI.
8. Type the following commands to mount the virtual CD:
   ```
   sudo mkdir /mnt/cdrom
   sudo mount -t iso9660 /dev/cdrom /mnt/cdrom
   ```
9. Check whether the mount was successful by listing the contents of the cdrom folder. The file **VMwareTools-<version>.tar.gz** must appear in the list:
   ```
   ls /mnt/cdrom/
   ```
   Copy the VMware Tools file to the tmp folder and extract its contents:
   ```
   cp /mnt/cdrom/VMwareTools-*.tar.gz /tmp/
   cd /tmp
   tar -xvzf VMwareTools-*.tar.gz
   cd vmware-tools-distrib
   ```
10. Install the VMware tools executing the following script:
    ```
    sudo ./vmware-install.pl
    ```
11. Press **Enter** to accept the default option whenever asked during the installation process.
12. Reboot the Appliance after install:
    ```
    sudo reboot
    ```

After installing VMware Tools, you should be able to see the IP addresses of the VM hosting the Appliance in the **Summary** tab. The warning message about the installation of VMware Tools disappears.

Related references

- Open-VM-Tools project on GitHub

Operational data sent to Nexthink

Overview

Nexthink SA automatically gathers non-personal data about product usage, product performance, asset inventory, and user activity from every installation of the Appliance to provide value-added services to customers:

Support Telemetry

To give customers the best support experience, Nexthink SA collects non-personal product and feature usage data as well as configuration and performance data from the Nexthink Appliances.

Cloud Intelligence

To provide insights and quarterly reports to customers, the Cloud Intelligence program collects non-personal benchmark and intelligence data, including the Digital Experience Score. Data is anonymized at the customer level and only higher level information, such as the customer's industry, is kept for benchmarking purposes.

Nexthink Enhance

By anonymously submitting to Nexthink SA the hash values of the binary files that the end users run on their devices and the names of the web sites that the end users visit, Nexthink Enhance provides compliance and risk information about applications and web domains to customers.

Customers can decide whether to participate in any of these programs independently. Declining the participation on a particular program implies losing the benefits associated to its value-added services.

Support Telemetry

The Support Telemetry program aims at providing customers a better support experience by gathering information about the actual configuration and use of the product.

Support Telemetry collects the following data points:

- Product configuration
- Product performance
- Product and feature usage
- Product feedback

With the help of these data, Nexthink Support is able to respond customer requests more efficiently, by potentially saving time in communications with the customer to ask for these data.

Enabling and disabling Support Telemetry

Starting from Nexthink V6.21, the collection of Support Telemetry data is enabled by default in the product license. To disable Support Telemetry:

- New customers must purchase premium support or managed services.

- Existing customers must contact Nexthink Support and ask for an update on their license to disable Support Telemetry. Customers that disable Support Telemetry assume that they might need longer communication efforts to get effective support from Nexthink.

Cloud Intelligence

The purpose of the Cloud Intelligence program is to provide customers with insights and benchmark information about the digital experience of their employees. Thanks to quarterly reports from Nexthink Customer Success Services, customers can compare themselves with other players in the same or in other industries to get a glimpse of where they are doing good in terms of digital employee experience and where they have room for improvement.

Cloud Intelligence collects information about the following objects from the deployed Engines, including the Digital Experience Score:

- Application - Executable - Binary
- Domain
- Package
- Device
- Digital Experience Score

Enabling and disabling Cloud Intelligence

Starting from Nexthink V6.21, Cloud Intelligence is enabled by default on every installation of the product. To disable Cloud Intelligence:

1. Log in to the Web Console as administrator.
2. Click the **Appliance** tab at the top of the window.
3. Select the section **Cloud services** from the left-hand side menu.
4. Untick the option **Enable Nexthink Cloud Intelligence. Please read the OFFICIAL DOCUMENTATION for more details.**

5. Click **Save Changes**.

Customers that disable Cloud Intelligence cannot benefit from the insights and benchmarking data associated to the program.

The following content applies exclusively to the Nexthink Cloud offering.

As this is a built-in feature for the Cloud offering, customers of Nexthink Cloud cannot disable Cloud Intelligence.

Nexthink Enhance module

The Nexthink Enhance module complements the data that is directly retrieved by the Collector from the devices of the end users with additional compliance and risk information about binary and domain objects.

To that end, Nexthink Enhance collects information about the following objects from the deployed Engines:

- Binary
- Domain

In return, Nexthink Enhance adds compliance and risk info to binaries and domains objects in the Engine database by updating the following fields:

- **Binary**
  - **Threat level**: Dangerousness of executing the binary. There are four threat levels defined: none, low, intermediate or high. Binaries unknown to Enhance receive a **-** as threat level.
  - **Application Category**: The type of application to which the binary belongs. Possible values are Email, Browser, Antivirus, Multimedia, etc.
- **Domain**
  - **Reputation**: The confidence with which employees can navigate the web domain without receiving malicious content. The following levels of reputation are defined: trustworthy, low risk, moderate risk and high risk. Domains unknown to Enhance display a dash sign (**-**) as reputation.
  - **Domain category**: The main type of online content offered by the servers behind the domain. Possible values are Social, VoIP, News and information, Search engines and portals, etc.
  - **Hosting Country**: The country that hosts the servers behind the domain.

Nexthink Enhance data is periodically sent to and received from Nexthink:

- The Engine sends Enhance data 5 minutes after start up.

- Following this first connection, binaries that have been active at least once in the last seven days are reevaluated every 24 hours.
- For every new binary or domain that appears in the system, the Engine takes around 5 minutes to fetch its compliance and risk info.

Enabling and disabling Nexthink Enhance

Nexthink Enhance is enabled by purchasing the Enhance module as a separate subscription, which is added to the product license. Data to identify binaries and domains is sent to Nexthink and compliance and risk information about binaries and domains is returned to the Engines deployed by the customer for as long as the subscription is valid.

Connecting to cloud services

All three value-added services: Support Telemetry, Cloud Intelligence, and Nexthink Enhance use the same connection to Nexthink Cloud Services. Therefore, the same connectivity requirements are applicable to all these value-added services.

Storage of collected data

The following cloud vendors provide secure storage for the non-personal customer data collected by Nexthink in the indicated countries:

Provider	Data center location
Azure	Netherlands
OVH	France
GCS	European Union

Related references

- Connectivity requirements
- Digital Experience Score

Sending additional data to Support

Overview

To help Nexthink Support correctly diagnose issues within your setup, send data about the performance and health of your Appliances from the Web Console.

Running the Support script

After contacting Nexthink Support, you may be requested to run the Support script on one or more of your Nexthink Appliances and send the results back to the Support team for analysis.

To run the Support script from the Web Console:

1. Log in to the Web Console of the concerned Appliance.
2. Select the **APPLIANCE** tab at the top of the Web Console.
3. Click **Support** from the left-hand side menu.
    - When retrieving data from the Portal Appliance, tick any of the choices under **Extra options**, if requested by Support:
        1. Tick **SMTP logs** to include the logs related to the sending of emails from the Appliance (notifications and digests).
        2. Tick **SSO logs** to include the logs about the authentication of users through single-sing on.
4. Click the button **RUN SCRIPT** under **Run support script**.

The execution time of the Support script can take up to approximately 20 minutes. Avoid restarting the Appliance while the script is running.

Sending the results to support

To get the results from the Web Console, still from the **Support** page:

1. Click the button **DOWNLOAD RESULTS** under **Script results**.
    - The date and time of the last created file appear to the right of the button.
    - If no results are yet available, the message **No Support script results found** is displayed instead.
2. Attach the downloaded file to your answer in the support ticket.

Related tasks

- Sending email notifications from the Appliance

Disaster recovery

Planning for disaster recovery

The Nexthink Appliance provides you with different backup techniques that allow you to recover from either a partial or a full disaster:

- A partial disaster is a failure that affects one or several of the server components of Nexthink (Web Console, Engine or Portal), while the Appliance is still accessible.
- A full disaster is a complete system failure that prevents any further access to the Appliance.

The mechanisms for partial disaster recovery are automatically put in place after the installation of the Appliance. Each one of the server components in the Appliance generates a daily backup of its data for its own recovery. In this way, if any of the components crashes, you can at least get the component back to the state it had the day before the crash.

Full disaster recovery, on the other hand, requires you to save the backups to an external storage device outside the Appliance before total breakdown. You can automate this process by activating the provided mechanism to save backup files externally. If you want to install your own backup tool, first read and follow the recommendations of the article on installing third-party software in the Appliance. Beware that a serious hardware issue in your Appliance can make your data unrecoverable if you do not save it elsewhere.

Partial disaster recovery

In case of a server component malfunction, use its daily backup files for recovery. In addition to the daily backups, the server components make an automatic backup of their data before migration as well. That is useful in the case that the software upgrade process goes wrong.

To learn about the information that is saved during the backup process and how to recover from a partial disaster, read the corresponding documentation for each component:

- Web Console automatic backup and Web Console restore
- Engine automatic backup and Engine restore
- Portal automatic backup and Portal restore

Full disaster recovery

In case of a total failure of the Appliance, you need to be ready to start from anew. As a prerequisite, you must have previously saved the backups of all the server components in the Appliance to an external storage device. Remember that you can automate this process by activating external backups from the Web Console.

In addition to the server components, take a backup of the following two items to recover from a full disaster of the master Appliance:

- The product license. Since it is not included in the automatic backups, take a backup of the license file each time that you renew your subscription.
- The PKI that secures the TCP communication of the Collectors with the Engines. Take a backup of the certificates and keys in the master Appliance to avoid having to recreate them and redistribute them to the deployed Collectors.

To perform full recovery:

1. Download an Appliance ISO with the same version of the Appliance that failed.
2. Install the Appliance following the steps described in Installing the Appliance.
3. Choose to install either the Portal or the Engine as described in Engine & Portal Installation, depending on the main server component that your Appliance was running.
4. Copy the backups to the new Appliance using any SCP client.
5. Restore the Web Console first as described in Restoring the Web Console to set the general parameters of the Appliance.
6. Restore the installed server component: Engine or Portal, as documented in Restoring the Engine or Restoring the Portal.
7. In the case of a complete failure of the appliance that hosts the Portal, restore the license file.

Activating external backups

The Appliance provides a mechanism to automate the saving of backup files to an external SMB share. This mechanism makes a copy of the daily backup of every server component (Web Console, Engine or Portal, including rule-based assignment data, if enabled) to the SMB share right after the backup file is created.

Before activating external backups, you must set up the SMB share:

1. Configure the user account
2. Set the permissions on the destination folder
3. Share the folder

To activate external backups in the Appliance:

1. Log in to the Web Console as admin from a web browser:
   https://<IP_address_of_Appliance>:99
2. Click the **Appliance** tab at the top of the window.
3. Select the section **External backup** from the left-hand side menu. This item only appears in slave Appliances if the mechanism of external backup has not been centralized
4. Tick the option **Enable daily backups to a SMB share** and fill out the form:
   - **SMB share path**: The path of the shared folder in Windows format, that is **server-name****shared-folder****path**.
   - **Username**: The name of the user account with the permissions to write to the shared folder.
   - **Domain**: The name of the domain to which the user account belongs. Leave empty if the user does not have any domain.
   - **Password**: The password of the user account.
   - Optional: Tick the box **Send notificiation by email** to send an email to the recipients specified in the **Accounts** section under **Notifications**, each time that the system makes an external backup.
   - Optional: In **Copy test file to SMB share**, click the **COPY** button to test the access to the given shared folder.

Note that you can centralize the external backup of slave Appliances when you federate them. In that way, the slave Appliance uses the same SMB share as the master Appliance for external backups.

The files saved in the SMB share for the different components have the following format:

- Web Console:
  console-<hostname>-<timestamp>.tgz
- Engine:
  nxengine-<instance>-<hostname>-<timestamp>.tgz
- Portal (main backup and history details of count metrics):
  portal-<hostname>-<timestamp>.tgz

portal-<hostname>-history_YYYYMMDD-<timestamp>.backup

- Rule-based assignment data:
  nxassignment-<hostname>-<timestamp>.tgz

For advanced users, it is possible to customize the mount options of the SMB share for external backups. These are the options found after the **-o** flag of the **mount** command. By default, the Appliance mounts the SMB share using the options **guest** and **credentials**. After activating external backups via the Web Console, set additional mount options for the SMB share by editing the backup config file:

1. Log into the CLI of the Appliance.
2. Edit the backup configuration file of the Appliance:
   ```
   sudo vi /var/nexthink/common/conf/backup-config.xml
   ```
3. Inside the section **BackupDirectory** add a new entry to specify one or more additional options, separated by commas:
   ```
   <ExtraParameters>options</ExtraParameters>
   ```
4. Save your changes and exit:
   ```
   :wq
   ```

The resulting configuration file should look like this:

```
<?xml version="1.0"?>
<Configuration origin=... >
  <BackupDirectory enabled="true">
    <Server>...
     ...
    <ExtraParameters>options</ExtraParameters>
  </BackupDirectory>
</Configuration>
```

Related tasks

- Web Console backup and restore
- Engine backup and restore
- Portal backup and restore
- License backup and restore
- PKI backup and restore
- Rule-based assignment backup and restore
- Installing third-party software in the Appliance

Web Console backup and restore

Manual Backup

To manually back up the Web Console:

1. Log in to Command Line Interface of the Appliance.
2. Execute the following command to create a new backup. Optionally specify a different name for the backup file without the extension (tgz is automatically added):

```
sudo /var/nexthink/console/helpers/backup-console.sh
[backup-file]
```

The backup file contains the full database of the Web Console (`console-db.backup`) and the content of the following files:

- `/var/nexthink/common/*` (all files in the directory)
- `/etc/yum/pluginconf.d/proxy.conf`
- `/var/nexthink/console/etc/certificate.pem`

Find the backup file in the directory:

```
/var/nexthink/console/backup
```

Automatic Backup

Every day at 01:10 an automatic backup is triggered using a crontab entry. Up to 10 backups files are used to keep history, all located in the directory:

```
/var/nexthink/console/backup
```

A link file named console-backup.tgz is also created in that directory and points to the last backup.

Restoring the Web Console

To completely restore the Web Console settings and account configuration, log in to the shell of the Appliance, get your backup file, and follow the next steps:

1. Stop the Web Console:

```
sudo systemctl stop nxconsole
```

2. Untar your backup file (suppose that it is named console-backup.tgz) in a directory in your home:

```
mkdir console-bk
tar xvzf console-backup.tgz -C console-bk
```

3. Copy the configuration files in the backup to their intended location:
```
cd console-bk
sudo cp -R var/nexthink/common/* /var/nexthink/common
sudo cp etc/yum/pluginconf.d/proxy.conf
/etc/yum/pluginconf.d
sudo cp var/nexthink/console/etc/certificate.pem
/var/nexthink/console/etc
```
4. Drop the database of the Web Console:
```
dropdb -U postgres console
```
5. Drop the *console* user of the database:
```
dropuser -U postgres console
```
6. Create an empty database:
```
/var/nexthink/console/helpers/create-db.sh
```
7. Restore the database of the Web Console (console-db.backup file from the backup):
```
pg_restore -U postgres -d console console-db.backup
```
8. Restart the Web Console:
```
sudo systemctl start nxconsole
```

The Web Console is now restored with all its users and settings in place.

The operations described in this article should only be performed by a Nexthink Engineer or a Nexthink Certified Partner.

If you need help or assistance, please contact your Nexthink Certified Partner.
Related tasks

- Logging in to the CLI

Related references

- Nightly task schedules timetable

Engine backup and restore

Manual backup

To make a complete backup of the Engine, execute manually the same script that is executed automatically during the daily backup of the Appliance:

1. Log in to the CLI of the Appliance running the Engine.

2. Execute the command:
   ```
   sudo /var/nexthink/engine/common/bin/nightly_backup.sh
   ```
3. Optionally: If you want to keep the logs, copy the log files stored under:
   ```
   /var/log/nexthink/
   ```
   - `engine.log`
   - `audit.log`
   - `alert.log`
   - All the compressed older logs stored in gz files.

For the Engine in the Appliance, the script creates a tgz file (GZIP compressed Tar archive format) with the contents of the Engine database and its configuration. Find the backup files under:

```
/var/nexthink/engine/01/backups/nxengine-backup-<id>.tgz
```

Alternatively, you can make a backup of the Engine database only. Copying the database file while the Engine is running is not a good idea, because the Engine is continuosly modifying the file, and the result could be a corrupted file. Instead, make a safe backup while your Engine is running by following these steps:

1. Log in to Command Line Interface of the Appliance.
2. Execute the following command to generate a compressed gz file with the database of the Engine:
   ```
   sudo nxinfo backup --name <name_of_backup_file>
   ```

The file is copied to your current directory.

Automatic backup

The Appliance automatically makes a backup of all the Engines running on it via a cron job. The job is executed every day at 04h15 by default. Find the cron job specification under:

```
/etc/cron.d/nxengine-crontab
```

And the script executed in here:

```
/var/nexthink/engine/common/bin/nightly_backup.sh
```

The script makes a copy of the database and the configuration files of the Engine that is present in the Appliance and compresses them in separate tgz files. Backup files are stored in the Engine backup directory:

```
/var/nexthink/engine/01/backups/nxengine-backup-<id>.tgz
```

Specify the maximum number of local backups that are kept in the directory by editing the first variable in the script `nightly_backup.sh`. By default, up to ten backups are kept simultaneously:

```
NUM_OF_BACKUPS=10
```

Beware that if the nightly backup script changes on an Engine release, the upgrade of the Engine resets the number of backups to its default value. In such a case, recover your own modified number of backups from a copy of the script, named `nightly_backup.sh.rpmsave`, that the upgrade process saves in the same directory with the contents of the file before the upgrade.

In a test environment, you may want to disable automatic backups to save disk space in the Appliance. To that end, comment out the line that executes the nightly backup in the crontab file by prepending a hash sign (**#**) to it.

Each local backup file gets assigned a number from one to the maximum number of simultaneous backups in the directory. When the maximum number is reached, the count begins again and backup files are progressively overwritten. In order to get the most recent backup file, there is a symbolic link to the latest backup (note the absence of identifier):

```
/var/nexthink/engine/01/backups/nxengine-backup.tgz
```

If external backups have been activated, the automatic script copies the daily backup to external storage right after generating it.

On upgrade backup

In addition to the automatic nightly backup of the Engine, the appliance automatically makes a new backup of the Portal before each upgrade. The file is placed in the same directory as the nightly backups and its name has the following format (where **X.X.X.X** indicates the version to which the Portal is upgrading):

```
/var/nexthink/engine/01/backups/nxengine-backup_before-X.X.X.X.tgz
```

Older upgrade backups are erased in the process.

Restoring the Engine

Restore the Engine either in the same Appliance from which you made the backups or in a different Appliance. In the case that you are restoring your

backups in another Appliance, make sure that its network configuration is the same as the configuration of the original Appliance. Otherwise, you may no longer receive data from the Collectors and have the wrong internal networks configured. In addition, the new Appliance requires you to reallocate the devices assigned to the original Appliance from the Portal. In case that you are using a license with online activation, this process should be transparent. If you are using a license with offline activation, you must repeat the procedure to get your license signed.

To restore a complete backup of the Engine:

1. Log in to CLI of the Appliance where you want to restore the Engine.
2. Stop the running Engine:
   ```
   sudo systemctl stop nxengine@1
   ```
3. Copy the backup file into the Engine directory:
   ```
   sudo cp nxengine-backup-<id>.tgz /var/nexthink/engine/01
   ```
4. Extract the database and configuration files from the backup file:
   ```
   cd /var/nexthink/engine/01
   sudo tar -xvzf nxengine-backup-<id>.tgz
   ```
   - If you are restoring a database from an Engine previous to V6.5, change the owner and mode of the restored configuration files:
     ```
     sudo chown nxengine:nxengine
     /var/nexthink/engine/01/etc/ -R
     sudo chmod 0660 /var/nexthink/engine/01/etc/*
     ```
5. Remove the database of the Engine in place:
   ```
   sudo nxinfo remove -r
   ```
6. Restore the database of the Engine backup:
   ```
   sudo nxinfo restore -n
   /var/nexthink/engine/01/data/nxengine-db.gz
   ```
7. Restart the Engine:
   ```
   sudo systemctl start nxengine@1
   ```
8. Validate that the Engine is running properly:
   ```
   nxinfo info
   ```

While the Engine is starting, the last command displays the message:

```
nxengine is booting...
```

After a few minutes, once the Engine has finished loading the database, the execution of this command displays the basic configuration and some statistics of the Engine. This means that the restore process was successful.

If you made a backup of the database only and you want to restore it in the current Appliance, you just need to restore the database of the Engine, and not any of the configuration files, which are already in place:

1. Log in to CLI of the Appliance where you want to restore the Engine.
2. Stop the running Engine:
   ```
   sudo systemctl stop nxengine@1
   ```
3. Remove the database of the Engine in place:
   ```
   sudo nxinfo remove -r
   ```
4. Restore the database of the Engine backup:
   ```
   sudo nxinfo restore -n nxengine-db.gz
   ```
5. Restart the Engine:
   ```
   sudo systemctl start nxengine@1
   ```
6. Validate that the Engine is running properly:
   ```
   nxinfo info
   ```

The operations described in this article should only be performed by a Nexthink Engineer or a Nexthink Certified Partner.

If you need help or assistance, please contact your Nexthink Certified Partner.
Related tasks

- Planning for disaster recovery
- Setting up a software license
- Logging in to the CLI

Related references

- Nightly task schedules timetable

Portal backup and restore

Manual Backup

To manually back up the Portal:

1. Log in to Command Line Interface of the Appliance that hosts the Portal.
2. Execute the following script, noting that you must not add any extension to the name of the target file. The script automatically appends the **.tgz** extension to the name of the backup file:
   ```
   sudo /var/nexthink/portal/backup/backup-portal.sh
   target-filename
   ```
   - The Portal backup file is stored under:
     `/var/nexthink/portal/backup/`
3. Execute the following script to backup the configuration of Nginx, the reverse proxy component in the Portal that handles connections. As in the case of the Portal, the **.tgz** extension is added automatically to the name

of the backup file:

```
sudo /var/nexthink/nxnginx/bin/backup-nxnginx.sh
target2-filename
```

- The proxy backup file is stored under:
  `/var/nexthink/nxnginx/backup/`

In addition, if you want to take a backup of the history details of count metrics, you must have configured the Portal to automatically keep these history details day by day. See in the next section the directory where the Portal stores the backup files of history details. If the Portal has not been configured to store the history details, it is not possible to recompute them afterwards manually.

Copy the contents of the history directory to another location (e.g. to a USB key) to make a manual backup of the history details of count metrics:

```
cp -r /var/nexthink/portal/backup/history/ target-folder
```

Automatic Backup

Nightly backup

Every day at 22h15, a cron job triggers an automatic backup of the Portal. Find the cron job specification under:

```
/etc/cron.d/portal-crontab
```

The backup files are located in:

```
/var/nexthink/portal/backup
```

Find the script that creates the automatic backups in the same directory:

```
/var/nexthink/portal/backup/backup-portal.sh
```

The file named **portal-backup.tgz** is a symbolic link that points to the last backup file in the history. The backup file holds the main database of the Portal and the content of the configuration folder:

```
/var/nexthink/portal/conf
```

Specify the maximum number of local backups that are kept in the directory by editing the first variable in the script `backup-portal.sh`. By default, up to ten backups are kept simultaneously:

```
NUM_OF_BACKUPS=10
```

Beware that if the nightly backup script changes on a Portal release, the upgrade of the Portal resets the number of backups to its default value. In such a case, recover your own modified number of backups from a copy of the script, named `backup-portal.sh.rpmsave`, that the upgrade process saves in the same directory with the content of the file before the upgrade.

History backup

In addition, if you have configured your Portal to store the history details of count metrics, that is, the lists of objects that contributed to the count metric on a particular day, these are stored under:

```
/var/nexthink/portal/backup/history
```

The name of history detail files has the format *history_YYYYMMDD.backup*. The number of files kept for the history details depend on the disk space reserved for this purpose.

Nginx backup

Later, at 22h30, another cron job triggers the backup of the configuration of Nginx, a reverse proxy component used to enhance the security of the Portal. The automatic backup system keeps a history of up to ten backup files, which are located in:

```
/var/nexthink/nxnginx/backup
```

Find the script that creates the automatic backups of the reverse proxy here:

```
/var/nexthink/nxnginx/bin/backup-nxnginx.sh
```

Specify the maximum number of local backups that are kept in the directory by editing the first variable in the script `backup-nxnginx.sh`. By default, up to ten backups are kept simultaneously:

```
NUM_OF_BACKUPS=10
```

As for the nightly script of the Portal, the backup script of Nginx may be overwritten by an upgrade of the Portal appliance and reset the number of backups to its default value, if the script changed on a release for any reason. To avoid losing your changes in such a case, recover your value for the maximum number of backups from the file `backup-nxnginx.sh.rpmsave` in the same

directory.

Each backup file saves the configuration of Nginx that is located in the following directory:

```
/var/nexthink/nxnginx/conf.d
```

On upgrade backup

In addition to the nightly backups, the appliance automatically makes a new backup of the Portal before each upgrade. The file is placed in the same directory as the nightly backups and its name has the following format (where **X.X.X.X** indicates the version to which the Portal is upgrading):

```
/var/nexthink/portal/backup/portal-backup_before-X.X.X.X.tgz
```

Older upgrade backups are erased in the process.

Because backing up the Portal may be a lengthy process, deactivate the automatic backup on Portal upgrade if you consider that the nightly backup is enough for you to not lose important data in the case of a failed upgrade. To deactivate the automatic backup of the Portal on upgrade, create an empty file in the directory of the Portal with the following command:

```
sudo touch /var/nexthink/portal/conf/skip-update-backup.conf
```

Restoring the Portal

To restore the Portal state from a backup file:

1. Log in to Command Line Interface of the Appliance that hosts the Portal.
2. Execute the restore script:
   ```
   sudo /var/nexthink/portal/backup/restore-portal.sh \
   [-d history_details_directory] <backup-filename>
   ```

If you saved the history details of count metrics, use the **-d** option to specify the directory that holds these files. The history files are expected to have the same name format specified above (*history_YYYYMMDD.backup*). In the case that you configured the Portal to save the backups and history details to an external share, this name format is changed to *portal-<hostname>-history_YYYYMMDD-<timestamp>.backup*. To restore the history files with the script, they must have their original name format. To rename all the history detail files stored in an external share, copy them to a directory in the Appliance and then type in the following command:

```
reg="(history_[0-9]+)"; \
for file in *.backup; do if [[ ${file} =~ ${reg} ]];\
then mv $file ${BASH_REMATCH[1]}.backup;\
fi; done
```

In the external share, you may have stored a set of details files whose total size exceeds the reserved disk size for history details configured in the Portal. Remember to manually select only the more recent files whose total size is within the configured limit. Use the command `du -h` in the folder containing the files with history details to get their total size, compare it to the value that you have configured in the Portal data retention, and remove the oldest files in the set until the total size of the files matches or is below the configured value. Failing to do so results in the Portal taking more time to restore history details that must be removed afterwards anyway, because there is no disk space left reserved for them.

The script only restores the database of the Portal, that is, the state of your dashboards. It does not restore the configuration files though, because you may want to keep your current configuration. If you need to restore the configuration of the Portal and that of Nginx:

1. Stop the Portal:
   ```
   sudo systemctl stop nxportal
   ```
2. Stop Nginx:
   ```
   sudo systemctl stop nginx
   ```
3. Untar the backup file of the Portal:
   ```
   tar -xvzf portal-backup.tgz
   ```
4. Copy the contents of the **conf** directory to the Portal configuration directory:
   ```
   sudo cp -r conf/ /var/nexthink/portal/
   ```
5. Untar the backup file of Nginx:
   ```
   tar -xvzf nxnginx-backup.tgz
   ```
6. Copy the contents of the **conf.d** directory to corresponding Nginx directory:
   ```
   sudo cp -r conf.d/ /var/nexthink/nxnginx
   ```
7. If you made changes to the default PKI, restore it now.
8. Restart Nginx:
   ```
   sudo systemctl start nginx
   ```
9. Restart the Portal:
   ```
   sudo systemctl start nxportal
   ```

The operations described in this article should only be performed by a Nexthink Engineer or a Nexthink Certified Partner.

If you need help or assistance, please contact your Nexthink Certified Partner.

Related tasks

- Logging in to the CLI
- Planning for disaster recovery

Related references

- Data retention
- Nightly task schedules timetable

Rule-based assignment backup and restore

If you activated rule-based Collector assignment on your setup, learn here how to make a backup of the assignment data stored in the Portal appliance.

Manual backup

To manually back up the rule-based assignment data:

1. Log in to the CLI of the Appliance that hosts the Portal.
2. Run the backup script. The script automatically adds the **.tgz** extension to the name of the file:

   ```
   sudo /var/nexthink/nxassignment/bin/backup-nxassignment.sh
   target-filename
   ```

   - The backup file of rule-based assignment is stored under:
     `/var/nexthink/nxassignment/backup/`

Automatic backup

Every day at 22h00, a cron job triggers the automatic backup of the rule-based assignment data. Find the cron job specification under:

```
/etc/cron.d/nxassignment-crontab
```

And find the script executed by the cron job here:

```
/var/nexthink/nxassignment/bin/backup-nxassignment.sh
```

The script makes a copy of the current assignment data and stores them in the following backup folder with the specified format:

```
/var/nexthink/nxassignment/backup/nxassignment-backup-<id>.tgz
```

Specify the maximum number of local backups that are kept in the directory by editing the first variable in the script `backup-nxassignment.sh`. By default, up to ten backups are kept simultaneously:

```
NUM_OF_BACKUPS=10
```

If external backups have been activated, the automatic script copies the daily backup to external storage right after generating it.

Restoring the assignment rules

Restore the rule-based assignment data either to the same or to a different Appliance from which you made the backup.

To restore the assignment data from the backup:

1. Log in to the CLI of the Portal Appliance where you want to restore the assignment data.
2. Copy the backup file into the Portal Appliance using your favorite SCP tool.
3. Run the restoring script:
   ```
   sudo /var/nexthink/nxassignment/bin/restore-nxassignment.sh
   \
   nxassignment-backup-<id>.tgz
   ```

Although rule-based assignment data can be restored independently of Portal data, if both need to be restored, we recommend to restore the Portal first.

Related tasks

- Planning for disaster recovery
- Assigning Collectors to Engines
- Nightly taks schedules timetable

License backup and restore

In the case of a complete failure of the appliance that hosts the Portal, the locally cached license may be lost as well. To avoid this, you can manually save a copy of your license file into the same shared folder that you use for your external Portal backups, for example.

Manual backup

To save the cached license file:

1. Log in to the CLI of the appliance that hosts the Portal.
2. Copy the cached license files to an external storage medium, for instance, the external share for the Portal configured in the Web Console:

```
mkdir -p /<external_share>/LicenseRestore
sudo cp /var/nexthink/llm/data\
/{license.file,llm_private_key.txt,llm_public_key.txt} \
/<external_share>/LicenseRestore
```

Restoring the license

Before restoring the license, restore first the Portal in the new appliance and configure the access to the storage medium (typically the external share) where you stored a copy of the license file.

To restore the cached license file:

1. Log in to the new appliance that hosts the Portal.
2. Copy the the license file backups to the correct folder in the appliance:

```
sudo cp /<external_share>/LicenseRestore\
/{license.file,llm_private_key.txt,llm_public_key.txt} \
/var/nexthink/llm/data/
sudo chown nxtlicense:nxtlicense /var/nexthink/llm/data/*
```

3. Restart the local license manager service:

```
sudo systemctl restart nxllm
```

4. Check that the LLM works correctly:

```
sudo /var/nexthink/llm/bin/check.sh
```

Recreating the license

In the case of a full disaster where you do not have an external backup of the cached license file, deactivate the product from the Portal:

1. Log in to the Portal as admin.
2. In the **ADMINISTRATION** menu, click **Licenses** under the **SYSTEM CONFIGURATION** section.
3. Click the button **Deactivate product** at the top right corner of the **Licenses** panel.
4. Ask Nexthink for a new license and reactivate the product.

Related tasks

- Planning for disaster recovery
- Portal backup and restore
- Logging in to the CLI
- Setting up a software license

PKI backup and restore

Overview

The PKI generated by the master Appliance during federation lets Collectors securely communicate with the Engines through a TCP connection.

Failing to take a backup of the PKI items in the master Appliance (root certificate, private key, and customer key) before a full disaster occurrence, results in the need to re-create the PKI and re-distribute a new root certificate and a new customer key to all the deployed Collectors.

Manual backup

Once you have federated at least one slave Appliance, take a backup of the generated PKI:

1. Open a web browser and log in to the Web Console of the master Appliance as admin.
2. In the **Appliance** tab, select the **Collector management** section on the left-hand side menu.
3. Under **Collector default certificates** at the bottom of the page, click the button **BACKUP CERTIFICATE AND KEY** to get a backup of the generated Root CA certificate and Customer Key. The backup file has the name **root-ca-backup.tgz**.

Restoring the PKI

To restore the backup of the PKI, we assume that you have a new master Appliance in place with the same network configuration as the original Appliance and a restored license.

Follow this procedure before federating any Engine back.

1. Copy the backup file **root-ca-backup.tgz** to the master Appliance using any SCP tool.

2. Download the following script for deploying the Customer Key and Root CA: deploy_rck.sh.
3. Copy the script to the master Appliance using any SCP tool.
4. Log in to the CLI of the master Appliance.
5. Execute the script as root, passing the backup file as argument.
   ```
   sudo sh deploy_rck.sh root-ca-backup.tgz
   ```
6. Open a web browser and log in to the Web Console of the master Appliance as admin.
7. If the new Appliance has a different DNS name from the original:
   1. In the **Appliance** tab, select the **Network Parameters** section on the left-hand side menu.
   2. Type in the **External DNS name** and the **Internal DNS name** of the new master Appliance.
8. Select the **Collector management** section on the left-hand side menu.
9. If you are running the Portal and the Engine in the same Appliance, click the button **GENERATE CERTIFICATE** that is displayed in red.
10. If your Engines reside in separate slave Appliances, federate them now:
    1. Select the **Federated appliances** section on the left-hand side menu.
    2. Click **ADD APPLIANCE** to add a new slave and provide the necessary information.

Related tasks

- Planning for disaster recovery
- License backup and restore
- Logging in to the CLI

Branding

Branding the Portal

Overview

To customize the visual appearance of the Portal and adapt it to your corporate image, brand the following elements:

- Login page backgroung
- Login page logo
- Navigation bar logo
- Email digests logo

Login page background

To replace the image that is displayed as background in the login page:

1. Upload your background image (in JPEG format) to the home directory of the *nexthink* account (`/home/nexthink`) in the Appliance using your favorite SCP client.
2. Log in to the CLI of the Appliance that hosts the Portal.
3. Move the uploaded background image to the folder where the Portal expects to find it and rename it to `portal-signin-bg.jpg`:

```
sudo mv <your_background_image>.jpg \
/var/nexthink/portal/custom/portal-signin-bg.jpg
```

4. Make Portal the owner of the file:

```
sudo chown nxportal:nexthink \
/var/nexthink/portal/custom/portal-signin-bg.jpg
```

The background image is displayed centered with respect to the login page, preserving its original aspect ratio. To cover the full display area of the web browser, the image is automatically stretched and cropped. For an optimal result, use an image with a minimum resolution of 1280x850 pixels.

Login page logo

To replace the Nexthink logo that appears in the login page:

1. Upload your logo image (in GIF format) to the home directory of the *nexthink* account (`/home/nexthink`) in the Appliance using your favorite SCP client.
2. Log in to the CLI of the Appliance that hosts the Portal.
3. Move the uploaded logo to the folder where the Portal expects to find it and rename it to `logo.gif`:

```
sudo mv <your_logo>.gif
/var/nexthink/portal/custom/logo.gif
```

4. Make the Portal the owner of the file:

```
sudo chown nxportal:nexthink \
/var/nexthink/portal/custom/logo.gif
```

5. Check the login page to verify that your logo is displayed correctly.

Provide an image with a height of 48 pixels and a width up to 200 pixels (200x48 pixels):

- If the image has a different height, it is scaled such that the resulting image is 48 pixels tall.
- If the resulting image is more than 200 pixels wide, it is horizontally scaled down to fit 200 pixels. Otherwise, if the width of the resulting image is 200

pixels or less, the aspect ratio is preserved.

Navigation bar logo

To replace the logo that appears at the top of the navigation bar:

1. Upload your logo image (in GIF format) to the home directory of the *nexthink* account (`/home/nexthink`) in the Appliance using your favorite SCP client.
2. Log in to the CLI of the Appliance that hosts the Portal.
3. Move the uploaded logo to the folder where the Portal expects to find it and rename it to `nav-logo.gif`:
   ```
   sudo mv <your_logo>.gif
   /var/nexthink/portal/custom/nav-logo.gif
   ```
4. Make the Portal the owner of the file:
   ```
   sudo chown nxportal:nexthink \
   /var/nexthink/portal/custom/nav-logo.gif
   ```
5. Check the login page to verify that your logo is displayed correctly.

Provide a square format image of 40 pixels (40x40 pixels).

Email digest logo

Mail clients retrieve the logo displayed in email digests from a public online location. Therefore, to replace the logo in email digests:

1. Upload your logo image to a publicly accessible location. Use a common web image format (PNG, GIF, or JPEG) so that most mail clients can display it.
2. Log in to the CLI of the Appliance that hosts the Portal.
3. Optional: If the Portal has no configuration file yet, that is, if `portal.conf` does not exist in folder `/var/nexthink/portal/conf`, create it by copying the defaults from the sample configuration file:
   ```
   sudo -u nxportal cp
   /var/nexthink/portal/conf/portal.conf.sample \
   /var/nexthink/portal/conf/portal.conf
   ```
4. Edit the configuration file of the Portal:
   ```
   sudo vi /var/nexthink/portal/conf/portal.conf
   ```
5. Add the following line to the configuration:
   ```
   globalconfig.portal.digest.logo-url =
   "http://<URL_of_your_logo>"
   ```
6. Save your changes and quit the editor by typing:
   ```
   :wq
   ```
7. Restart the Portal:
   ```
   sudo systemctl restart nxportal
   ```

The logo image is displayed in the email digest with a resolution of 128x34 pixels. For an optimal result, use a logo image of this exact size or with the same

aspect ratio.

Related tasks

- Receiving email digests
- Logging in to the Portal

Branding of campaigns

Overview

To catch the eye of your employees and give them confidence to willingly answering the questions of a campaign, brand campaign notifications with your own corporate logo and colors.

The configuration settings for branding your campaigns is centralized in the Portal. The logo, selected theme, and colors apply to the notifications of the campaign only and not to the questions, which keep their usual appearance.

Applies to platforms:

Supplying the corporate logo

Provide the logo of your company in PNG format. Use preferably a version of the logo with transparent background. The logo is displayed to the left of the notification text, inside an area of 84 x 84 pixels. If the provided logo has a different size, it is scaled up or down so that its biggest dimension fits the display area, while keeping the original aspect ratio of the image.

To supply the corporate logo to the Portal:

1. Copy the logo file (e.g. `mylogo.png`) to the home directory of the nexthink account in the Portal by using your favorite SCP tool.
2. Log in to the CLI of the Portal.
3. Move the logo file to the configuration directory of the Portal with the following name:
   ```
   sudo mv mylogo.png
   /var/nexthink/portal/conf/end_user_feedback_logo.png
   ```
4. Make Portal the owner of the file:
   ```
   sudo chown nxportal:nexthink
   /var/nexthink/portal/conf/end_user_feedback_logo.png
   ```

From that point on, campaigns display the custom logo.

Choosing the theme and color

Choose between two themes to determine the general appearance of notifications:

- **Dark** (default), to display a black background with white colored text.
- **Light**, to display a white background with black colored text.

Additionally, specify the color of the buttons inside the notification in hexadecimal RGB format (e.g. `#4C4C4C` for the default dark grey color).

To set up the theme and the colors of the buttons:

1. Log in the CLI of the Portal.
2. Optional: If the Portal has no configuration file yet, that is, if `portal.conf` does not exist in folder `/var/nexthink/portal/conf`, create it by copying the defaults from the sample configuration file:
   ```
   sudo -u nxportal cp
   /var/nexthink/portal/conf/portal.conf.sample \
   /var/nexthink/portal/conf/portal.conf
   ```
3. Edit the main configuration file of the Portal:
   ```
   sudo vi /var/nexthink/portal/conf/portal.conf
   ```
4. Add the following two lines to the configuration file (or modify them if they already exist):
   ```
   globalconfig.euf-service.customization.theme="light"
   globalconfig.euf-service.customization.button-background-colour="#4C4C4C"
   ```
5. Save your changes and quit the vi editor by typing in:
   ```
   :wq
   ```
6. Restart the Portal:
   ```
   sudo systemctl restart nxportal
   ```

Depending on the specified color for the background of the buttons, the system automatically chooses the color of the text inside the buttons to be either black or white for ensuring good readability, as well as the color of the border of the buttons when they get focus.

Examples

Test your customizations by previewing them in the Finder at the final stage of creating a campaign.

Settings	Notification example
Standard Button color: #4C4C4C	
Light theme Button color: #88a2CC	
Dark theme Button color: #365D91	

Collector awareness

To see the branding in campaign notifications, employees require Collector V6.10 or later installed on their devices.

Once you have saved your branding customizations and restarted the Portal, Collectors acknowledge the changes as soon as they are restarted or after a maximum of 12 hours.

Related tasks

- Creating a campaign

Related references

- Campaign display compatibility