Nexthink for AWS Installation Guide

This documentation references external sources. Nexthink does not have control over the accuracy of third-party documentation, nor any external updates or changes that might create inconsistencies with the information presented on this page.

This file outlines the process of installing Nexthink for Amazon Web Services (AWS). Due to the technical limitations of deploying Nexthink onto an AWS environment, Nexthink provides a Virtual Hard Disk (VHD) image that can be uploaded onto an Amazon Simple Storage Service (S3) storage account. This image can then be used to create a virtual machine in Amazon Elastic Compute Cloud (EC2).

> *The VHD is an image of a pre-installed Nexthink Appliance, which does not contain the Nexthink Engine or Portal. These must be installed afterwards.*
>
> *Nexthink recommends having basic knowledge of Linux to ensure proper Appliance setup. The Nexthink Appliance is public-facing and requires the Security Hardening protocol.*

Configuring the local machine for AWS interaction

Requirements:

- An Amazon account configured for virtual machine (VM) creation from imported VHDs
- A local or on-premise CentOS machine with the Amazon client installed. To acquire the Amazon client, refer to the Get started with the AWS CLI documentation from Amazon
  - This can be achieved via the Nexthink Appliance

Log into the Amazon client using one of the supported AWS methods. Refer to the Configure the AWS CLI documentation from Amazon.

If you log in as an IAM user with an access key or assume an IAM role, be sure to include the following permissions:

```
"ec2:ModifySnapshotAttribute",
"ec2:CopySnapshot",
"ec2:RegisterImage",
"ec2:Describe*",
"s3:*"
```

Uploading the VHD

Requirements:

- An S3 bucket within your AWS account

1. After completing the authentication process, upload the 'release-6.x.x.x-cloud-amazon-ol8.vhd' file onto the bucket.
2. Create an import service role and the associated policy. Refer to the Importing a VM as an image using VM Import/Export documentation from Amazon.

Converting the VHD to AMI

Amazon Machine Image (AMI) is the VM format compatible with AWS EC2. Refer to the Importing a VM as an image using VM Import/Export documentation from Amazon.

1. Access the **containers.json** file and change the **Format** to **vhd**.
2. Run the **import** command.
3. After getting a response, take note of the **ImportTaskId** value.

   ```
   ...
   "StatusMessage": "pending",
       "ImportTaskId": "import-ami-xxx"
   }
   ...
   ```

4. Follow the state of the conversion with the following command:

```
aws ec2 describe-import-image-tasks --import-task-ids import-ami-xxx
```

5. Wait for the **StatusMessage** to change to **completed**.

Once the conversion process is finished, the newly created AMI image will appear on the EC2 dashboard under **Images > AMIs**. By default, this image has a non-descriptive name, e.g. **import-ami-xxx**, which cannot be changed. For ease of identification, Nexthink recommends copying the image with the following command:

```
aws ec2 copy-image --source-image-id <initial_ami_id> --source-region <your_region_code> --region
<your_region_code> --name <new-name> --description <description>
```

Alternatively, edit the name directly through the **AMIs** tab.

Configuring the network

Requirements:

- An existing AWS environment or familiarity with the Amazon network configuration
- A security group configured according to the Nexthink connectivity requirements
    - This is required for secure access to the machine instance

> *To configure a Virtual Private Cloud (VPC) network and create a security group, refer to the What is Amazon VPC? documentation from Amazon. To configure the security group, refer to the Nexthink connectivity requirements documentation.*

There are two ways of configuring the network:

- Manually create a VPC, a subnet, an internet gateway and a route table
- Use the default VPC. Nexthink recommends configuring its security group according to the Nexthink Connectivity requirements

Creating an instance of the Nexthink Appliance

1. On the EC2 Management Console, select **Instances**.
2. Insert a name for the instance.
3. Select **Application and OS Images > My AMIs**.
4. Select the newly imported Nexthink Appliance AMI.
5. Select the **Instance Type** according to the hardware requirements of your Nexthink installation. Refer to the instance types documentation from Amazon and the Hardware Requirements PDF file from the Installing the Appliance on AWS documentation.
6. Create a new key pair to access the instance. Alternatively, use the default.
    - Nexthink recommends using a strong key pair as the instance will be public.

Network configuration:

1. Configure the network and subnet as needed. Alternatively, use the default network.
    a. If multiple Appliances exist, apply the same values as the network and subnet.
2. Auto-assign a public IP. The **enable** option does not ensure the IP is static.
    a. For Elastic IP assignment, refer to the Elastic IP addresses documentation from Amazon.
3. Create a new security group or assign an existing one.
    a. Both inbound and outbound rules must be set according to the Nexthink connectivity requirements

Storage configuration:

1. Create a **root volume** of **32GB.**
2. Set the **disk type** to **gp3**.
3. Create a new volume for containing Nexthink databases. This volume must be configured according to the AWS hardware requirements.

Testing the connection

Test the SSH authentication to the newly created instance with the following command:

```
ssh -i mykeypair.pem cloud-user@ec2-XXX.eu-central-1.compute.amazonaws.com
# mykeypair.pem: private key of the keypair that was used for the instance creation
# ec2-XXX.eu-central-1.compute.amazonaws.com: Route53 Public DNS record automatically assigned by AWS, it can
be found in Instance Details > Public IPv4 DNS
```

Formatting the data disk

The VHD only has 5GB of available disk space on first access.

1. Check the disk type with the following command:

```
lsblk -pa
```

In the following example, **/dev/nvme0n1** is an OS disk with a size of 32GB and **/dev/nvme1n1** is a data disk with a size of 64GB. The root partition **/dev/mapper/nxt-root** requires extension to use the full capacity of the OS disk.

```
NAME                     MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
/dev/nvme1n1             259:0    0   64G  0 disk
/dev/nvme0n1             259:1    0   32G  0 disk
/dev/nvme0n1p1         259:2    0    1M  0 part
/dev/nvme0n1p2         259:3    0 1000M  0 part /boot
/dev/nvme0n1p3         259:4    0  200M  0 part /boot/efi
/dev/nvme0n1p4         259:5    0    1K  0 part
/dev/nvme0n1p5         259:6    0  6.7G  0 part
  /dev/mapper/nxt-root 253:0    0  6.6G  0 lvm  /
```

1. Extend the root partition with the following commands. The script uses LVM for added flexibility:

   ```
   # (optional) if growpart is not installed, install it:
   # sudo yum install --disablerepo=* --enablerepo=ol8_baseos_latest --enablerepo=ol8_appstream  cloud-
   utils-growpart
   sudo growpart /dev/nvme0n1 4
   sudo growpart /dev/nvme0n1 5
   sudo pvresize /dev/nvme0n1p5
   sudo lvextend -r -l +100%FREE /dev/mapper/nxt-root
   ```

2. Format the disk and create a new partition with the following commands:

   ```
   sudo sh /root/formatDataDisk.sh /dev/nvme1n1
   sudo systemctl daemon-reload
   ```

3. Check disk status with the **lsblk -pa** command.

```
NAME                                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
/dev/nvme1n1                        259:0    0   64G  0 disk
/dev/nvme1n1p1                    259:8    0   64G  0 part
  /dev/mapper/nxtdatapool-nxtdata 253:1    0   64G  0 lvm  /var/nexthink
/dev/nvme0n1                        259:1    0   32G  0 disk
/dev/nvme0n1p1                    259:2    0    1M  0 part
/dev/nvme0n1p2                    259:3    0 1000M  0 part /boot
/dev/nvme0n1p3                    259:4    0  200M  0 part /boot/efi
/dev/nvme0n1p5                    259:6    0 30.8G  0 part
 /dev/mapper/nxt-root            253:0    0 30.8G  0 lvm  /
/dev/nvme0n1p4                    259:7    0    1K  0 part
```

A usable 64 GB partition is now available and mounted on **/var/nexthink**.

Installing Nexthink on the AWS instance

1. Use any SCP client to download the 'Nexthink-offline-install-6.X.tgz' installation package onto the VM. Alternatively, visit the V6 release notes to download the package.
2. Unpack the package with the following command:

```
tar -xzvf Nexthink-offline-install-6.X.tgz
```

3. Install the package contents with the following command. Use the -p parameter to install the Portal or the -e parameter to install the Engine.

```
sudo sh installNexthinkInCloud.sh -p
# or alternatively
sudo sh installNexthinkInCloud.sh -e
```

4. Verify the components are running.
   - For the Engine, use the following command:

   ```
   nxinfo info
   ```

   - For the Portal, connect to the public IP. Monitor the logs in */var/nexthink/portal/logs*.

Configuring the directory for Portal cleanup

In an AWS installation, the scheduled cleanup of the Portal may fail if there is not enough disk space in the root partition. In V6.20 or later, avoid this by overriding the default directory where the Portal performs the cleanup with the path of the data disk.

1. Find the Portal configuration file (*/var/nexthink/portal/conf/portal.conf*).
   - If the Portal configuration file does not exist, copy the sample file located in */var/nexthink/portal/conf/portal.conf.sample.*
2. Add a new line with the following command:

```
globalconfig.portal.computation.compute-tables-cleanup-postgres-location = "/var/nexthink/database"
# the path must correspond to the mount point of the data disk
```

3. Restart the Portal with the following command:

```
sudo systemctl restart nxportal
```

Important notes and considerations

Compared to an on-premise installation of Nexthink, Nexthink Appliance faces a public connection and an internal network. Regarding the Portal to Engine Configuration, both the public and private IP/DNS of the machines must be configured in:

- Internal and External DNS on the Webconsole parameters
- Portal IP/Hostname on the Engine's Webconsole
- Engine DNS name when performing Appliance federation, the same name
  - must be resolved as the Engine's internal IP address by the Portal machine
  - must be resolved as the Engine's external IP address using the Finder
    - This is so the Finder can have access to the Engine

> *An implementation example is having a DNS server within an AWS environment that allows the Portal to resolve the Engine DNS to its private IP, and a public DNS server resolving the same Engine DNS to the Engine's public IP.*
>
> *This example may not fit the needs of a specific Nexthink deployment. For more information, contact Nexthing professional services.*