Nexthink for AWS Installation Guide

It is not possible to install directly the Nexthink ISO when deploying Nexthink in Amazon AWS environment, therefore Nexthink provides a VHD image that can be uploaded to an Amazon S3 storage account and be used to create a virtual machine in Amazon EC2. The VHD is simply a hard disk image of a pre-installed Nexthink Appliance, however it doesn't contain Nexthink Engine or Nexthink Portal and you must still install it afterwards.

Upload the Nexthink for Amazon VHD image

Before provisioning a Nexthink Appliance in Amazon, it is necessary to configure your amazon account so that you will be able to create virtual machines form imported VHD. The first step for that is to install the Amazon client on a regular CentOS machine that you run locally or in an on-premise environment (you can use a Nexthink Appliance for that):

```
http://docs.aws.amazon.com/cli/latest/userguide/awscli-install-linux.html#awscli-install-linux-pip
```

On the machine you installed the Amazon client, you may login using the command:

```
aws configure
```

You will be prompted for entering your AWS Access Key ID, AWS Secret Access Key (in order to create a pair of those, please refer to http://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_access-keys.html), and the region where you are located (a list of regions can be found on http://docs.aws.amazon.com/general/latest/gr/rande.html). (you may leave default output format option to the default value)

Upload the file *release-6.x.x.x-cloud-amazon.vhd* on an Amazon S3 bucket and create an import service role and its associated policy as described on http://docs.aws.amazon.com/vm-import/latest/userguide/vmimport-image-import.html

When the VHD file is stored in the S3 storage, it is seen as a simple file from Amazon perspective and must be imported in EC2 and converted to AMI format that is the Virtual machine image format used in Amazon, *you must create a file called containers.json that will contain:*

```
[
  {
    "Description": "release vhd 6.x.x.x",
    "Format": "vhd",
    "UserBucket": {
        "S3Bucket": "my-import-bucket",
        "S3Key": "release-6.x.x.x-cloud-amazon.vhd"
    }
}]
```

Now run the command:

```
aws ec2 import-image --description "Nexthink Cloud 6.x.x.x" --disk-containers file://containers.json
```

From the returned response, please take a note of the import task id:

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

You can follow the state of the conversion using the command:

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

The conversion process from vhd to ami format will now take place and will take several minutes.

(Optional) Once the operation completes, you should see a newly created AMI image in your Amazon EC2 dashboard under section Images /AMIs. By default this AMI file has a non-descriptive name (import-ami-xxx) and there is no possibility to change, it if you need to identify it in a better way, you have to copy it:

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

Or you may as well edit the name from the Amazon EC2 interface by selecting the tab AMIs.

Network configuration

The following steps depend on if you have an existing AWS environment or not, if you are acquainted with Amazon network configuration, you may skip this section, however keep in mind that the configuration of the security group is mandatory to secure the access to your machine instance. The security group must be configured according to the Nexthink connectivity requirements: http://doc.nexthink.com/Documentation/Nexthink/V6.7/InstallationAndConfiguration/Connectivityrequirements.
The Amazon documentation covers the network configuration of a VPC (virtual private cloud), creation of security group etc... thoroughly http://docs.aws.amazon.com/AmazonVPC/latest/GettingStartedGuide/getting-started-ipv4.html, you have two options from here:

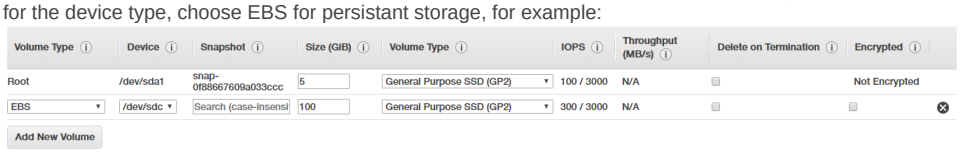
- Manually create a VPC, a subnet, an Internet gateway and a route table
- A default VPC is also available, but it is recommended to configure its attached security group according to the Nexthink Connectivity requirements.

Create Appliance instance

- Go to Instances/Instances in your EC2 Management Console and click on Launch Instance
- Choose "My AMIs" and select the recently imported Nexthink Appliance AMI
- Choose the right Instance type based on the Hardware Requirements of your Nexthink Installation:

  More information about the different AWS Instance types and nomenclatures can be found in the following webpage: https://aws.amazon.com/ec2/instance-types/
  The complete requirements for Nexthink in Amazon are described in the Hardware Requirements.

- Click on 'Next: Configure instance details':
  - network / subnet: if you wish to use a particular VPC / subnet, select them, otherwise, choose the default one.
    - If you will run multiple Appliances, make sure that they have got the same values in Network and Subnet
  - Auto-assign Public IP: You may choose "Enable" but the assigned IP won't be static, please see later in this document how to assign Elastic IPs (which are static)
  - IAM role: choose None
  - Other options can be left at default.
- Click on 'Next: add storage':
  - The VHD image we uploaded is 5GB and contains only the Operating system, it will be labelled as 'Root' volume, although you can extend it, we recommend to add a new volume having a size that will meet up the Nexthink Hardware Requirements, it should have the following characteristics
    - For the volume type, choose General Purpose SSD or Provisioned IOPS SSD depending on the column Hard disk type in the Hardware Requirements.
    - for the device name, choose /dev/sdc (it may have a different name however after booting the machine)
    - for the device type, choose EBS for persistant storage, for example:



- Click on 'Next: Add tags':
  - (Optional) you may add tags to be able to look for your machine more easily if you have a big infrastructure
- Click on 'Next: Configure Security Groups':
  - Assign the Security Group(s) created previously with the purpose of meeting the Connectivity Requirements (you must have previously selected the VPC that includes this security group)
- Review the configuration
- Press Launch, Amazon GUI will ask you which set of SSH keys you wish to use to connect to the Instance you're about to create, you can either:
  - Create a new keypair as specified in http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-key-pairs.html
  - Proceed without keypair, you will be able to connect using the standard 'nexthink' account of the Appliance
- Now go back to the main dashboard and select Instances, you should see the creation in progress

Once created, you have the possibility to change the name of the instance from the Amazon GUI

- At this stage, the virtual machine will have been created but the assigned public IP is dynamic, we need to allocate an Amazon Elastic IP and assign it to the instance we just created, this page describes how to proceed: http://docs.aws.amazon.com/fr_fr/AWSEC2/latest/UserGuide/elastic-ip-addresses-eip.html#using-instance-addressing-eips-allocating

Format the data disk

When you access your VM using SSH for the first time, only the 5 GB of the initial VHD are usable disk space. We need to format properly the disk we just add, for that, you need to check which it actually has (we specified /dev/sdc, but depending on the kernel version of the Appliance, it may be renamed), run the command:

```
lsblk -pa
```

you should have the following output

Example 1:

```
NAME                    MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
/dev/xvda                 202:0    0    5G  0 disk
/dev/xvda1              202:1    0    1M  0 part
/dev/xvda2              202:2    0 1000M  0 part /boot
/dev/xvda3              202:3    0  3.9G  0 part
  /dev/mapper/nxt-root 253:0    0  3.9G  0 lvm  /
/dev/xvdc                 202:32   0  100G  0 disk
```

In our example, we have two disks /dev/xvda of 5G (the actual VHD image we uploaded) and /dev/xvdc of 100G (the additional disk we provisioned in Amazon console, we can see it has been renamed).

Example 2 (case of NVMe disk on AWS instance with type c5):

```
NAME                   MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
/dev/nvme0n1               259:1    0    8G  0 disk
/dev/nvme0n1p1             259:2    0    1M  0 part
/dev/nvme0n1p2             259:3    0 1000M  0 part /boot
/dev/nvme0n1p3             259:4    0  6.9G  0 part
  /dev/mapper/nxt-root     253:0    0  6.9G  0 lvm  /
/dev/nvme1n1                 259:0    0   50G  0 disk
```

Here /dev/nvme0n1 represents the OS disk of 8GB and /dev/nvme1n1 the data disk of 50GB

Then we will have to run the format script that comes with the VHD image in the /root folder (or /tmp folder if it cannot be found in /root)

```
sudo sh /root/formatDataDisk.sh /dev/xvdc
```

or

```
sudo sh /root/formatDataDisk.sh /dev/nvme1n1
```

Once the script completed, you should now see after running the */sblk -pa* command:

```
NAME                               MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
/dev/xvda                            202:0    0    5G  0 disk
/dev/xvda1                         202:1    0    1M  0 part
/dev/xvda2                         202:2    0 1000M  0 part /boot
/dev/xvda3                         202:3    0  3.9G  0 part
  /dev/mapper/nxt-root             253:0    0  3.9G  0 lvm  /
/dev/xvdc                            202:32   0  100G  0 disk
/dev/xvdc1                         202:33   0  100G  0 part
  /dev/mapper/nxtdatapool-nxtdata 253:1    0  100G  0 lvm  /var/nexthink
```

As you can see a usable partition of 100 GB was created and is mounted on /var/nexthink

In case of a NVMe disk, we would have:

```
NAME                    MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
/dev/nvme0n1                259:1    0    8G  0 disk
/dev/nvme0n1p1              259:2    0    1M  0 part
/dev/nvme0n1p2              259:3    0 1000M  0 part /boot
/dev/nvme0n1p3              259:4    0  6.9G  0 part
  /dev/mapper/nxt-root      253:0    0  6.9G  0 lvm  /
/dev/nvme1n1                  202:32   0  100G  0 disk
/dev/nvme1n1p1              202:33   0  100G  0 part
  /dev/mapper/nxtdatapool-nxtdata 253:1    0  100G  0 lvm  /var/nexthink
```

the script uses LVM partitioning for better flexibility

Install Nexthink on the AWS Instance

- Transfer the installation package '*Nexthink-offline-install-6.X.tgz*' to the VM using your favorite SCP client. Again this file can be obtained from the Support Team.
- Now, unpack the installation package:

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

- The script '*installNexthinkInCloud.sh*' takes two optional parameters: -p to install the Portal, -e to install the engine, depending on what you want on this Appliance:

```
sudo sh installNexthinkInCloud.sh -p -e
```

- Nexthink is now installed, you may wish to check the engine is up and running:

```
nxinfo info
```

- Or the Portal by connecting to https://<your public ip> (please note that Portal can take some that to start, you can monitor the logs in */var/nexthink/portal/logs*).
- As written in the security hardening guide, you can now change the default password for the Webconsole and the Portal and make sure that all the steps of the guide are strictly followed.

Important configuration notes:

Compared to a standard installation of Nexthink, the fact that the Appliance(s) may be facing Internet on one side and facing an internal network on another side must be taken into account. Notably regarding the Portal Engine Configuration. The Internal IP/DNS Name of the machines must be used when configuring:

- Internal and External DNS in the Web Console parameters section.
- Portal IP/hostname in the Web Console of the Engine.
- Engine hostname in the Portal's Engines configuration tab.
  (here it must be a hostname such as this host will be resolved as the Engine's internal IP address by the Portal machine and it will be resolved as engine's external IP address on the machine using the Finder, this is important so that the Finder can have access to the engine).

An implementation example is to have a DNS server within AWS environment that allow the Portal appliance to resolve Engine DNS name to its private IP and in the meantime another Internet-facing DNS server resolving the same Engine DNS name to the public IP of the engine. Please note that this example may not fit the needs of every Nexthink deployment. If your are unsure about the best way to deploy and configure Nexthink in AWS environment, please contact your Nexthink Professional services representative.

Security Hardening

The hardening of the appliance is automated and will be deployed as soon as you install Nexthink. It is strongly advised to change the default product password.

- After the first login on the Web Console, it will prompt you to change the Web Console admin password and also the password of the Nexthink support account.
- When you login to the Portal, navigate to *account name* > My account to change the Portal's default admin password.

Directory configuration for Portal cleanup

In an AWS installation, the nightly cleanup of the Portal may fail if there is not enough space in the root partition. To avoid such situation in V6.20 and later, override the default directory where the Portal computes the cleanup and indicate the path of the data disk. Add the following line to the configuration file of the portal */var/nexthink/portal/conf/portal.conf* (if the file does not exist, create it first by copying the sample file */var/nexthink/portal/conf/portal.conf.sample*):

```
globalconfig.portal.computation.compute-tables-cleanup-postgres-location = "/var/nexthink/database"
```

The path indicated must correspond to the mount point of your data disk.

Restart the Portal:

```
sudo systemctl restart nxportal
```

Appendix:

- Available regions and their code (for the API for instance): http://docs.aws.amazon.com/general/latest/gr/rande.html
- EC2 Instance types: https://aws.amazon.com/ec2/instance-types/