# Preparing for Installation on Servers Without Internet Access

This section describes the procedures to install Cisco NFVI in a management node without Internet access.

In this scenario, you must:

1. Download the Cisco NFVI installation files to a 64 GB (minimum) USB 2.0 drive on a staging server with Internet access. If the management node is based on M5, you can optionally use USB 3.0 64GB to increase the installation speed significantly.
2. Copy the files to the management node.

• Preparing to Install Cisco NFVI on Management Nodes Without Internet Access, on page 1

# Preparing to Install Cisco NFVI on Management Nodes Without Internet Access

Following procedure describes how to download the Cisco NFVI installation files onto a USB drive of the staging server with Internet access. You can use the USB to load the Cisco NFVI installation files onto the management node without Internet access.

**Note** Cisco recommends you to use Virtual Network Computing (VNC), other terminal multiplexer, or similar screen sessions to complete these steps.

### Before you begin

You must have a CentOS 7 staging server (VM, laptop, or UCS server) with a 64 GB USB 2.0 drive only. You can use USB 3.0 64GB if the management node is of type M5. The staging server must have wired Internet connection to download the Cisco VIM installation files onto the USB drive. Once downloaded, you can copy the installation files onto the management node from USB drive.

**Note** Downloading of the installation files (over 25 GB in size) to the USB drive might take several hours depending on the speed of your Internet connection. Ensure that you disable the CentOS to the sleep mode, for faster installation.

**Step 1** On the staging server, use yum to install the following packages:

- PyYAML (yum install PyYAML)
- python-requests (yum install python-requests)
- Python 3.6

  Check if python 3.6 binary is located at `/opt/rh/rh-python36/root/bin/`, if not copy the python 3.6 binary to `/opt/rh/rh-python36/root/bin/`.

**Step 2** Log into Cisco VIM software download site and download the **getartifacts.py** script from external registry:

```
# download the new getartifacts.py file (see example below)
curl -o getartifacts.py
https://username:password@cvim-registry.com/mercury-releases/cvim24-rhel7-osp10/releases/2.4.4/getartifacts.py

curl -o getartifacts.py-checksum.txt
https://username:password@cvim-registry.com/mercury-releases/cvim24-rhel7-osp10/releases/2.4.4/getartifacts.py-checksum.txt

# calculate the checksum and verify that with one in getartifacts.py-checksum.txt
sha512sum getartifacts.py

# Change the permission of getartificats.py
chmod +x getartifacts.py
```

**Step 3** Run **getartifacts.py**. The script formats the USB 2.0 drive (or USB 3.0 drive for M5/Quanta based management node) and downloads the installation files. You must provide the registry username and password, tag ID, and USB partition on the staging server.

```
# ./getartifacts.py -h
usage: getartifacts.py [-h] -t TAG -u USERNAME -p PASSWORD -d DRIVE
                       [--proxy PROXY] [--retry]
                       [--artifacts [ARTIFACTS [ARTIFACTS ...]]]

Script to pull container images.

optional arguments:
  -h, --help            show this help message and exit
  -t TAG, --tag TAG     installer version to pull
  -u USERNAME, --username USERNAME
                        Registry username
  -p PASSWORD, --password PASSWORD
                        Registry password
  -d DRIVE, --drive DRIVE
                        Provide usb drive path
  --proxy PROXY         https_proxy if needed
  --retry               Try to complete a previous fetch
  --artifacts [ARTIFACTS [ARTIFACTS ...]]

                        Only supported parameter is all and defaults to all if not passed anything
```

```
This script pulls images from remote registry and copies the contents to usb drive
```

To identify the USB drive, execute the **lsblk** command before and after inserting the USB drive. The command displays a list of available block devices. The output data will help you to find the USB drive location. Provide the entire drive path in the `-d` option instead of any partition as shown below. Here, the tag_id refers to the Cisco VIM release version 3.4.x.

For example:

```
sudo ./getartifacts.py -t <tag_id> -u <username> -p <password> -d </dev/sdc> [--artifacts …] [--proxy
 proxy.example.com] –
```

**Note** Ensure that you do not remove the USB drive during synchronization.

**Note** On executing getartifacts.py, the following message: *stderr: mount: wrong fs type, bad option, bad superblock on /dev/sdc1, missing codepage or helper program, or other error.* is displayed to notify bad superblock and mount failure. In this case, reformat the drive and use the **fsck** command to recover the drive as given below:

```
fsck.ext4 -pv /dev/sdc .
```

**Note** As the size of the artifacts is greater than 25G, We recommend that you execute this step over a wired internet connection. It will take few hours to download and populate data on USB drive, depending on the internet connectivity.

The `getartifacts.py` script downloads the following:

- Packages
  - buildnode-K9.iso
  - mercury-installer.tar.gz
  - ironic-images-K9.tar.gz
  - registry-2.6.2.tar.gz
  - insight-K9.tar.gz
  - mariadb-app-K9.tar.gz
  - docker images and manifests

  **Note** The python-docker-py-1.10.6-4.el7.noarch.rpm, python-docker-pycreds-1.10.6-4.el7.noarch.rpm, python-websocket-client-0.32.0-116.el7.noarch.rpm, and vim_upgrade_orchestrator.py packages are required for upgrade

- Respective checksums

**Step 4** Use the following command to verify the downloaded artifacts and container images:

```
# create a directory
sudo mkdir -p /mnt/Cisco

# /dev/sdc is the USB drive, same as supplied in getartifacts.py python script

#You need to mount the partition with the steps given below:
sudo mount /dev/sdc1 /mnt/Cisco
cd /mnt/Cisco

# execute the test-usb help to look at the options
./test-usb -h
```

```
usage: ./test-usb [-h] -- Show this program to check integrity of artifacts in this USB drive
                  [-a] -- Check integrity of all (core and insight) artifacts in this USB drive
                  [-l] -- Location of artifacts
                  [-U] -- test upgrade artifacts from 2.4.x to 3.4.y
# execute the verification script
./test-usb

# failures will be explicitly displayed on screen, sample success output below# sample output of
./test-usb with 3.4.1 release
#./test-usb
           INFO: Checking the integrity of artifacts on this USB drive
           INFO: Checking artifact python-docker-py-1.10.6-4.el7.noarch.rpm
           INFO: Checking artifact python-docker-pycreds-1.10.6-4.el7.noarch.rpm
           INFO: Checking artifact python-websocket-client-0.32.0-116.el7.noarch.rpm
           INFO: Checking artifact vim_upgrade_orchestrator.py
           INFO: Checking artifact mercury-installer.tar.gz
           INFO: Checking artifact ironic-images-K9.tar.gz
INFO: Checking artifact buildnode-K9.iso
INFO: Checking artifact registry-2.6.2.tar.gz
INFO: Checking required layers:
           INFO: 764 layer files passed checksum.

# ./test-usb -a
           INFO: Checking the integrity of artifacts on this USB drive
           INFO: Checking artifact python-docker-py-1.10.6-4.el7.noarch.rpm
           INFO: Checking artifact python-docker-pycreds-1.10.6-4.el7.noarch.rpm
           INFO: Checking artifact python-websocket-client-0.32.0-116.el7.noarch.rpm
           INFO: Checking artifact vim_upgrade_orchestrator.py
           INFO: Checking artifact mercury-installer.tar.gz
           INFO: Checking artifact ironic-images-K9.tar.gz
INFO: Checking artifact  buildnode-K9.iso
INFO: Checking artifact registry-2.6.2.tar.gz
INFO: Checking artifact mariadb-app-K9.tar.gz
INFO: Checking artifact insight-K9.tar.gz
INFO: Checking required layers:
INFO: 764 layer files passed checksum.
```

If the download fails, an error message is displayed.

For example:

```
# ./test-usb
INFO: Checking the integrity of this USB stick
INFO: Checking artifact buildnode-K9.iso
ERROR: Checksum for artifact buildnode-K9.iso does not match ('SHA512 (buildnode-K9.iso) =
96ec62a0932a0d69daf60acc6b8af2dc4e5eca132cd3781fc17a494592feb52a7f171eda25e59c0d326fbb09194eeda66036cbdc3870dafe74f59cf1f2dce225'
 != 'SHA512 (buildnode-K9.iso) =
a6a9e79fa08254e720a80868555679baeea2dd8f26a0360ad47540eda831617bea0514a117b12ee5f36415b7540afa112a1c904cd69e40d704a8f25d78867acf')
INFO: Checking artifact registry-2.3.1.tar.gz
ERROR: Artifact registry-2.3.1.tar.gz is not present
INFO: Checking required layers:
ERROR: Layer file sha256:002aa1f0fbdaea7ea25da1d906e732fe9a9b7458d45f8ef7216d1b4314e05207 has a bad
 checksum
ERROR: Layer file sha256:5be3293a81773938cdb18f7174bf595fe7323fdc018c715914ad41434d995799 has a bad
 checksum
ERROR: Layer file sha256:8009d9e798d9acea2d5a3005be39bcbfe77b9a928e8d6c84374768ed19c97059 has a bad
 checksum
ERROR: Layer file sha256:ea55b2fc29b95d835d16d7eeac42fa82f17e985161ca94a0f61846defff1a9c8 has a bad
 checksum
INFO: 544 layer files passed checksum.
```

**Step 5** To resolve download artifact failures, unmount the USB and run the getartifacts command again with the --retry option.

```
sudo ./getartifacts.py -t <tag_id> -u <username> -p <password> -d </dev/sdc> --retry
```

**Step 6** Mount the USB and then run the test-usb command to validate if all the files are downloaded:

```
# /dev/sdc is the USB drive, same as supplied in get artifacts.py python script
sudo mount /dev/sda1 /mnt/Cisco
cd /mnt/Cisco

# execute the verification script
./test-usb

# In case of failures the out of the above command will explicitly display the same on the screen
```

**Step 7** When the USB integrity test is done, unmount the USB drive by running the following command:

```
sudo umount /mnt/Cisco
```