



Day 0 Configuration

Cisco Catalyst 8000V supports both Cisco IOS XE and the Cisco IOS XE SD-WAN functionalities. You can access the Cisco IOS XE functionalities by booting the instance in the autonomous mode. Similarly, to access and use the Cisco SD-WAN functionalities, boot your instance in the controller mode.

The autonomous mode is the default mode in which a Cisco Catalyst 8000V instance boots up. If you are a user who wants to proceed with the day 0 configuration in the autonomous mode, refer this chapter.



Note If you wish to deploy the Cisco Catalyst 8000V instance in the controller mode, see [Install and Upgrade for Cisco Catalyst 8000V Controller Mode](#).



Attention If the system is unable to detect any of the following four parameters – OTP, UUID, VBOND, ORG, the device boots in the autonomous mode.

Bootstrap Support Across Hypervisors and Clouds

The following tables provide an overview of the bootstrap support across the hypervisors and the clouds for Cisco Catalyst 8000V in the autonomous mode:

Hypervisor	iosxe_config.txt on CD-ROM	ovf-env.xml on CD-ROM	OVA Installation	Config-drive Format	Custom Data	User Data
VMware	Yes	Yes	Yes	Yes	No	No
KVM	Yes	Yes	No	Yes	No	No
AWS	No	No	No	No	Yes	Yes
Azure	No	No	No	No	Yes	Yes
GCP	No	No	No	Yes	Yes	Yes

Feature Support for Day 0 Configuration

Hypervisor	iosxe_config.txt on CD-ROM	ovf-env.xml on CD-ROM	OVA Installation	Config-drive Format	Custom Data	User Data
Raw configuration copy and paste	Yes	Yes	No	Yes	Yes	Yes
Availability of specific configuration fields	No	Yes	Yes	Yes	Yes	Yes
GUI Availability	No	No	Yes	No	No	No
Guestshell Bootstrapping	Yes; via manual IOS configuration	Yes; via manual IOS configuration	No	Yes; via manual IOS configuration	Yes	Yes; via manual IOS configuration

- Public clouds have one input mechanism through which you can provide the bootstrap information to a VM. However, on the device side, three bootstrap input formats are supported for each cloud – custom-data, user-data, and SDWAN (via the `ciscosdwan_cloud_init.cfg` file downloaded from vManage). For example, in AWS, you can provide the bootstrap information in any of the above-mentioned formats to the instance at launch via the EC2 user data text box or the File Upload option. Cisco Catalyst 8000V then determines and processes the configuration information that you provided.
 - The custom-data and the user-data columns in the table mentioned above refer to the bootstrapping input formats and not the cloud native bootstrap input mechanisms for which they were originally named. All the public clouds support both the formats, but the custom-data format is more mature and is the recommended option for most applications.
 - For private clouds, you can perform the bootstrap configuration by providing a configuration file in the `iosxe_config.txt` format or the `ovf-env.xml` format. You must upload the configuration file to the VM during Cisco Catalyst 8000V installation through an attached CD-ROM.
- [Prerequisites for the Day0 Configuration, on page 3](#)
 - [Restrictions for the Day Zero Configuration, on page 3](#)
 - [Selecting the Bootstrapping Mechanism, on page 3](#)
 - [Day 0 Configuration Using .txt or .xml Files, on page 4](#)
 - [Day 0 Configuration for OVF Templates, on page 8](#)
 - [Day 0 Configuration Using Config-drive, on page 8](#)
 - [Day 0 Configuration Using Custom Data, on page 9](#)
 - [Day 0 Configuration in the Controller Mode, on page 18](#)
 - [Verifying the Router Operation Mode and Day 0 Configuration, on page 18](#)
 - [Frequently Asked Questions, on page 19](#)

Prerequisites for the Day0 Configuration

- If you want to deploy the Cisco Catalyst 8000V instance in the controller mode, generate the bootstrap config file from vManage and rename the generated config file to `ciscosdwan_cloud_init.cfg`. Use the same file for the device to automatically bootup in the Controller mode and register to vManage.

Do not manually edit the automatically generated config file from vManage. This might cause the controller to go out of sync and the device's first power-on and bootup might not be successful.

Restrictions for the Day Zero Configuration

- If you use the PayG licensing model, you cannot perform a mode switch as controller mode does not support the PayG licensing model.
- Only the autonomous mode supports Dual-IOSd.
- Images without payload encryption and NO-LI images are not supported in the controller mode.
- After onboarding and determining the mode of operation, if you switch from the controller mode to the autonomous mode or vice versa, it results in the loss of configuration.
- When you switch from the autonomous mode to the controller mode or vice versa, Cisco Federal Licensing and Smart Licensing registration does not work. You must reregister for the licenses to work.
- When you deploy a Cisco Catalyst 8000V VM by using GUI, the order of network interfaces added to the VM may not match the order in which the interfaces are created. This is because the interface numbering order is based on the name of the driver and the PCI address. Due to this behavior, the Day Zero Configuration might be applied incorrectly for some network interfaces. If you encounter this scenario, you must manually configure the affected network interfaces after you deploy the VM.

Selecting the Bootstrapping Mechanism

Now that you know the supported bootstrap methods across the hypervisors and clouds, the next step is to decide the mechanism that you should choose to perform the day 0 configuration. You can configure the day 0 settings for your device by using:

- **The GUI tool:** If you have installed Cisco Catalyst 8000V on VMware, and you chose an OVA deployment, you can perform the configuration by using the OVA deployment wizard. This wizard supports the bootstrap-specific fields, and you don't have to manually create a bootstrap configuration file.
- **.txt file/.xml file:** If you are in a private cloud and you want to configure the day 0 settings through IOS configuration commands, we recommend choose the `iosxe_config.txt` file. This method allows you to take the CLIs that you wish to apply, paste them into a file, and provide it to the VM as a CD-ROM.
- **Custom data:** When you deploy Cisco Catalyst 8000V on AWS, Microsoft Azure, or GCP, the custom-data formatted bootstrap configuration is the recommended method. This configuration method is more functional and flexible compared to configuration by using user-data. Configuring the day 0 settings using user-data is primarily meant for users with an already established user-data deployment.

Read on to know more about each of these mechanisms in detail.

Day 0 Configuration Using .txt or .xml Files

On a new, out-of-box device, during the installation, if you want to boot up the device in the autonomous mode, you can provide the bootstrap related configuration.

In a private cloud such as KVM environment, you can perform the bootstrap configuration by providing a `iosxe_config.txt` file or an `ovf-env.xml` file. This method allows you to gather the configurations that you wish to apply via the CLI, paste them into a file, and provide this content to the VM as a CD-ROM. Depending on the hypervisor environment, the data is then used for the bootstrap configuration.

The following sections explain this bootstrap configuration method in detail:

Creating the Bootstrap File

This procedure provides the steps that you need to perform to create a bootstrap configuration file. This file, which is either in the .txt or .xml format, allows you to provide the day0 configuration for your device in a simple and flexible manner.

You can perform this procedure when you create the virtual machine in hypervisors such as KVM.

Step 1 Create the `iosxe_config.txt` or the `ovf-env.xml` file.

- a) To create the `iosxe_config.txt` file, create a file with this name that contains the IOS config commands line by line.
- b) To create the `ovf-env.xml` file, select the properties that you wish to configure from Bootstrap Properties, and place them in a file with the specified name.

Note To know more about the individual properties in the .xml file, see [Bootstrap Properties, on page 4](#).

Step 2 To convert the .xml or the .txt file to a consumable form for the virtual machine, create a disk image from the file using the following command:

Example:

```
mkisofs -l -o /my/path/c8000v_config.iso <configuration_filename>
```

Step 3 Mount the `c8000v_config.iso` as an additional disk during creation of the Cisco Catalyst 8000V virtual machine.

Bootstrap Properties

See the following table to know about the individual bootstrap properties using which you can create the `ovf-env.xml` file.

Table 1: Bootstrap Properties

Property	Description
console	Configures the console mode. Possible values include auto, virtual, serial.

Property	Description
domain-name	Domain name of the router.
enable-scp-server	Enables the IOS SCP feature.
enable-ssh-server	Enables remote login using SSH and disables remote login via Telnet. Requires that the login user name and password are set.
hostname	The host name of the router.
ios-config	<p>Enables execution of a Cisco IOS command.</p> <p>To execute multiple commands, use multiple instances of ios-config, with a number appended to each instance. For example, ios-config-1, ios-config-2. The commands are executed in numerical order according to the appended number.</p> <p>Example</p> <pre>ios-config-1="username cisco priv 15 pass ciscoxyz" ios-config-2="ip scp server enable" ios-config-3="ip domain lookup" ios-config-4="ip domain name cisco.com"</pre>
license	Configures the license technology level that is available when the Cisco Catalyst 8000V instance boots.
login-password	The login password for the router.
login-username	The user name for the router.
mgmt-interface	Designates the management interface for the Cisco Catalyst 8000V instance. The format must be GigabitEthernetx or GigabitEthernetx.xxx.
mgmt-ipv4-addr	The management gateway address/mask in the IPv4 format for the GigabitEthernet0 management interface.
mgmt-ipv4-gateway	The IPv4 management default gateway address. If you're using DHCP, enter dhcp in the field.
mgmt-ipv4-network	Configures the IPv4 Network (such as "192.168.2.0/24" or "192.168.2.0 255.255.255.0") that the management gateway should route to. If this value is not specified, the default route (0.0.0.0/0) is used.
mgmt-vlan	Configures the dot1Q VLAN interface. Requires the management interface to be configured using the GigabitEthernetx.xxx format.
pnscc-agent-local-port	<p>(Optional) Configures the Cisco Prime Network Services Controller service agent SSL port on the local Cisco Catalyst 8000V to receive policies from the service manager.</p> <p>This setting is used if you plan to remotely manage the Cisco Catalyst 8000V using the Cisco Prime Network Services Controller.</p>

Property	Description
pns-c-ipv4-addr	Configures the IP address of the Cisco Prime Network Services Controller. This setting is used if you plan to remotely manage the Cisco Catalyst 8000V instance using the Cisco Prime Network Services Controller.
pns-c-shared-secret-key	Configures the Cisco Prime Network Services Controller shared secret key for the Cisco Prime Network Services Controller agent to set the SSL certificate from the controller. This setting is used if you plan to remotely manage the Cisco Catalyst 8000V instance using the Cisco Prime Network Services Controller.
privilege-password	Configures the password for privileged (enable) access.
resource-template	Configures the Resource Template. Possible values include default, service_plane_medium, and service_plane_heavy.



Note For a sample `ovf-env.xml` file, see [Sample ovf-env.xml File, on page 7](#).

Sample iosxe_config.txt File

```
hostname ultra-ios_cfg
license smart enable
username lab privilege 15 password lab
ip domain-name cisco.com
crypto key generate rsa modulus 1024
interface GigabitEthernet1
ip address 10.0.0.5 255.255.255.0
no shut
exit
ip route 0.0.0.0 0.0.0.0 10.0.0.1
line vty 0 4
login local
exit
```

Sample iosxe_config.txt File for OpenStack Environment

```
hostname c8kv-ios_cfg
license smart enable
username lab priv 15 secret lab
ip domain-name cisco.com
interface GigabitEthernet1
ip address 10.0.0.5 255.255.255.0
no shut
exit
ip route 0.0.0.0 0.0.0.0 10.0.0.1
line vty 0 4
login local
exit
```

Sample ovf-env.xml File

```
<?xml version="1.0" encoding="UTF-8"?>
<Environment
  xmlns:oe="http://schemas.dmtf.org/ovf/environment/1">
  <PropertySection>
    <Property oe:key="com.cisco.c8000v.license.1" oe:value="security"/>
    <Property oe:key="com.cisco.c8000v.console.1" oe:value="serial"/>

  <Property oe:key="com.cisco.c8000v.config-version.1" oe:value="1.0"/>
    <Property oe:key="com.cisco.c8000v.domain-name.1" oe:value=""/>
    <Property oe:key="com.cisco.c8000v.enable-scp-server.1" oe:value="False"/>
    <Property oe:key="com.cisco.c8000v.enable-ssh-server.1" oe:value="False"/>
    <Property oe:key="com.cisco.c8000v.hostname.1" oe:value="lab"/>
    <Property oe:key="com.cisco.c8000v.license.1" oe:value="ax"/>
    <Property oe:key="com.cisco.c8000v.login-password.1" oe:value=""/>
    <Property oe:key="com.cisco.c8000v.login-username.1" oe:value="lab"/>
    <Property oe:key="com.cisco.c8000v.mgmt-interface.1" oe:value="GigabitEthernet1"/>
    <Property oe:key="com.cisco.c8000v.mgmt-ipv4-addr.1" oe:value="172.25.223.251/25"/>

    <Property oe:key="com.cisco.c8000v.mgmt-ipv4-gateway.1" oe:value="172.25.223.129"/>

    <Property oe:key="com.cisco.c8000v.mgmt-ipv4-network.1" oe:value=""/>
    <Property oe:key="com.cisco.c8000v.mgmt-vlan.1" oe:value=""/>
    <Property oe:key="com.cisco.c8000v.pnsc-agent-local-port.1" oe:value=""/>
    <Property oe:key="com.cisco.c8000v.pnsc-ipv4-addr.1" oe:value=""/>
    <Property oe:key="com.cisco.c8000v.pnsc-shared-secret-key.1" oe:value=""/>
    <Property oe:key="com.cisco.c8000v.privilege-password.1" oe:value=""/>
    <Property oe:key="com.cisco.c8000v.remote-mgmt-ipv4-addr.1" oe:value=""/>
    <Property oe:key="com.cisco.c8000v.resource-template.1"
oe:value="service_plane_medium"/>
    <Property oe:key="com.cisco.c8000v.ios-config-0001" oe:value="logging buffered
10000"/>
    <Property oe:key="com.cisco.c8000v.ios-config-0002" oe:value="hostname uut-ovf"/>
    <Property oe:key="com.cisco.c8000v.ios-config-0003" oe:value="ip domain-name
cisco.com"/>
    <Property oe:key="com.cisco.c8000v.ios-config-0004" oe:value="crypto key generate
rsa modulus 1024"/>
    <Property oe:key="com.cisco.c8000v.ios-config-0005" oe:value="interface
GigabitEthernet2"/>
    <Property oe:key="com.cisco.c8000v.ios-config-0006" oe:value="ip address 10.0.0.5
255.255.255.0"/>
    <Property oe:key="com.cisco.c8000v.ios-config-0007" oe:value="no shut"/>
    <Property oe:key="com.cisco.c8000v.ios-config-0008" oe:value="exit"/>
    <Property oe:key="com.cisco.c8000v.ios-config-0009" oe:value="ip route 0.0.0.0
0.0.0.0 10.0.0.1"/>
  </PropertySection>
</Environment>
```

Sample ovf-env.xml File for OpenStack

```
<?xml version="1.0" encoding="UTF-8"?>
<Environment
  xmlns:oe="http://schemas.dmtf.org/ovf/environment/1">
  <PropertySection>
    <Property oe:key="com.cisco.c8000v.license.1" oe:value="network-premier addon
dna-premier"/>
    <Property oe:key="com.cisco.c8000v.console.1" oe:value="virtual"/>

  <Property oe:key="com.cisco.c8000v.config-version.1" oe:value="1.0"/>
  <Property oe:key="com.cisco.c8000v.domain-name.1" oe:value=""/>
  <Property oe:key="com.cisco.c8000v.enable-scp-server.1" oe:value="False"/>
```

```

<Property oe:key="com.cisco.c8000v.enable-ssh-server.1" oe:value="False"/>
<Property oe:key="com.cisco.c8000v.hostname.1" oe:value="lab"/>
<Property oe:key="com.cisco.c8000v.login-password.1" oe:value="lab#123"/>
<Property oe:key="com.cisco.c8000v.login-username.1" oe:value="lab"/>
<Property oe:key="com.cisco.c8000v.mgmt-interface.1" oe:value="GigabitEthernet1"/>
<Property oe:key="com.cisco.c8000v.mgmt-ipv4-addr.1" oe:value=""/>
<Property oe:key="com.cisco.c8000v.mgmt-ipv4-gateway.1" oe:value="192.168.8.1"/>
<Property oe:key="com.cisco.c8000v.mgmt-ipv4-network.1" oe:value=""/>
<Property oe:key="com.cisco.c8000v.mgmt-vlan.1" oe:value=""/>
<Property oe:key="com.cisco.c8000v.pnsc-agent-local-port.1" oe:value=""/>
<Property oe:key="com.cisco.c8000v.pnsc-ipv4-addr.1" oe:value=""/>
<Property oe:key="com.cisco.c8000v.pnsc-shared-secret-key.1" oe:value=""/>
<Property oe:key="com.cisco.c8000v.privilege-password.1" oe:value="lab#123"/>
<Property oe:key="com.cisco.c8000v.remote-mgmt-ipv4-addr.1" oe:value=""/>
<Property oe:key="com.cisco.c8000v.resource-template.1" oe:value="service-plane-medium"/>
<Property oe:key="com.cisco.c8000v.ios-config-0001" oe:value="logging buffered 10000"/>
<Property oe:key="com.cisco.c8000v.ios-config-0002" oe:value="hostname uut-ovf"/>
<Property oe:key="com.cisco.c8000v.ios-config-0003" oe:value="ip domain name cisco.com"/>
<Property oe:key="com.cisco.c8000v.ios-config-0005" oe:value="interface GigabitEthernet2"/>
<Property oe:key="com.cisco.c8000v.ios-config-0006" oe:value="ip address dhcp"/>
<Property oe:key="com.cisco.c8000v.ios-config-0007" oe:value="no shut"/>
<Property oe:key="com.cisco.c8000v.ios-config-0008" oe:value="exit"/>
<Property oe:key="com.cisco.c8000v.ios-config-0009" oe:value="ip route 0.0.0.0 0.0.0.0
192.168.8.1"/>
<Property oe:key="com.cisco.c8000v.ios-config-0010" oe:value="interface GigabitEthernet1"/>
<Property oe:key="com.cisco.c8000v.ios-config-0011" oe:value="ip address dhcp"/>
<Property oe:key="com.cisco.c8000v.ios-config-0012" oe:value="no shut"/>
</PropertySection>
</Environment>

```

Day 0 Configuration for OVF Templates

OVF deployments with full support for Day 0 bootstrapping are only supported in VMware via the vCenter UI or the COT tool. The Day 0 configuration for Cisco Catalyst 8000V running on the ESXi hypervisor is available in [Deploying the OVA to the VM](#).

To know how to perform the day 0 configuration for deployments using the COT tool, see [Editing the Basic Properties of Cisco Catalyst 8000V using COT](#).

Day 0 Configuration Using Config-drive

Use the **--config-drive** option to specify that the configuration is loaded when Cisco Catalyst 8000V is booting. CD-ROMs and the second hard drive can also contain configuration information in the config-drive format. In either of these cases, this information is a file with contents that match the format of either the `iosxe_config.txt` file or the `ovf-env.xml` file.

To use the config drive option for your day zero configuration, set the **--config-drive** option to **true**, and specify the name of the configuration file in which you enter the router configuration to be booted. You can provide the configuration information in the following ways:

As an XML/TXT File

In this option, you must provide the configuration file in one of the two possible formats:

- As an xml file in the `ovf-env.xml` file format (for OVF deployments)

- As a text file in the `iosxe_config.txt` file format

We strongly recommend that you use only one configuration file type, either the `.txt` file or the `.xml` file, and not both.

See the following sample configuration. Use one of these configurations to provide your configuration file in the filesystem:

```
nova boot c8000v-vm-174 --image c8000v-174 --flavor c8000v.2vcpu.4gb --nic
port-id=6773be11-7b95-48cd-b372-fb8a3cae2b50 --config-drive=true --file
ovf-env.xml=/home/stack/conf_files/ut/ovf-env.xml
```

OR

```
nova boot c8000v-vm-174 --image c8000v-174 --flavor c8000v.2vcpu.4gb --nic
port-id=6773be11-7b95-48cd-b372-fb8a3cae2b50 --config-drive=true --file
iosxe_config.txt=/home/stack/conf_files/ut/iosxe_config.txt
```



Note These file names are hard-coded and are required for the `config-drive` settings to boot.

Using User Data

In certain environments such as OpenStack, use the `user_data` option to provide the file into the filesystem with the `config-drive` format. See the following sample user data for the OpenStack environment:

```
openstack server create "admin-VK-C8KISOSerial-20210917"
--config-drive true
--image c8kv-image-176
--flavor m1.large
--network mgmt-nt
--network prod-nt
--block-device-mapping id=admin-VK-EmptyVolume-SerialTest:type=volume
--user-data userdata.txt
```

Day 0 Configuration Using Custom Data

After you download the Cisco Catalyst 8000V installation files and deploy the image in your environment, the Cisco Catalyst 8000V instance requires manual configuration before the device is fully functional. To automate the configuration steps or to connect to on-premise sites, you can upload the Cisco Catalyst 8000V custom data or user data in all the supported public and private clouds.

By uploading the custom data for your cloud service provider or your private cloud, you can automate the day 0 and/or the bootstrap configuration. Upload or attach a bootstrap configuration file, (`iosxe_config.txt` file) or provide the user data to automate these processes to bring up the device into a functional state with minimal to no touch.

The Day 0 bootstrap file allows you to run Cisco IOS XE configuration commands, install Python packages in guestshell on Day0, run scripts in guestshell on Day0, and provide licensing information to boot the Cisco Catalyst 8000V instance with a desired technology package.

To launch a Cisco Catalyst 8000V instance by using custom data, perform the following steps:

Editing the Day 0 Bootstrap File

To edit the bootstrap file, configure these properties: IOS Configuration, Scripts, Script credentials, Python package, and Licensing. The properties can be placed in the bootstrap file in any order. Dependencies between the properties are noted in each of the following property descriptions. See the example bootstrap files at: <https://github.com/csr1000v/customdata-examples>.

After you have defined the properties of the bootstrap file, upload the file .

Configuring the IOS Configuration Property

If you want to bootstrap certain IOS configuration on Day0, configure the IOS Configuration property. See the following example:

```
Section: IOS configuration
hostname C8000V1
interface GigabitEthernet1
description "static IP address config"
ip address 10.0.0.1 255.255.255.0
interface GigabitEthernet2
description "DHCP based IP address config"
ip address dhcp
```

After the first line that reads `Section: IOS configuration`, enter a list of Cisco IOS XE configuration commands to be run on the Cisco Catalyst 8000V router.

When you run this command, the above mentioned IOS configuration is applied to the Cisco Catalyst 8000V router on Day0.

Configuring the Scripts Property

Scripts property helps you to automate your deployment and achieve other automation goals. If you want to run a python or a bash script on Day0 under guestshell context, you can achieve the same by providing the public URL and arguments of the python or the bash script in Scripts property.

A script must include a piece of code that includes the shebang (!) character in the first line of the script. This line tells Cisco IOS-XE which script interpreter (Python or Bash) must be used to parse the script code. For example, the first line of a python script can contain `#!/usr/bin/env python`, while the first line of a bash script can contain `#!/bin/bash`. This line allows the Python or Bash script to run as executable code in a Linux environment.

When you execute the script, the script runs in the guestshell container of the Cisco Catalyst 8000V instance. To access the guestshell container, use the **guestshell** EXEC mode command. For more information on guestshell commands, see the [Programmability Configuration Guide](#).

To configure the Scripts property, follow the format given here:

```
Section: scripts
public_url <arg1> <arg2>
```

In this script, the first line of the property should read `Section: Scripts`.

In the second line of the property, enter the URL of the script and the script's arguments. The script can be either a python or a bash script. The script is run in guestshell in the first boot when the bootstrap file is uploaded when you create the Cisco Catalyst 8000V instance.

To view more examples of the scripts, see the *Scripts* section in <https://github.com/csr1000v/customdata-examples>. Also refer to the following two examples:

Example 1

```
Section: Script
https://raw.githubusercontent.com/csr1000v/customdata-examples/master/scripts/smartLicensingConfigurator.py --idtoken "<token_string>" --throughput <throughput_value>
```

The two lines in the scripts property retrieve the `smartLicensingConfigurator.py` script from the `customdata-examples` repository at the specified URL. The script runs in the guestshell container of the Cisco Catalyst 8000V with the arguments `idtoken` and `throughput`.

Example 2

```
Section: Scripts
ftp://10.11.0.4/dir1/dir2/script.py -a arg1 -s arg2
```

These two lines in the Scripts property retrieve the `script.py` script from the ftp server with the IP address 10.11.0.4, and runs the script with the `./script.py -a arg1 -s arg2` bash command in the guestshell container of the Cisco Catalyst 8000V using arguments `arg1` and `arg2`.



Note If a script in the Scripts property requires a Python package that is not included in the standard CentOS Linux release (the CentOS Linux release that is used by the guestshell, which is currently CentOS Linux release 7.1.1503), you must include information about the Python package in the Python package property. For more information, see [Configuring the Python package Property, on page 12](#).

Prior to uploading the bootstrap file and running the bash or python script, Cisco recommends that you test the URL that you intend to use in the Scripts property. You can test the `ftp://10.11.0.4/dir1/dir2/script.py -a arg1 -s arg2` URL by first running the curl software tool to download the script file. In the guestshell, enter the curl command, as shown in the following example:

```
curl -m 30 --retry 5 --user username:password
ftp://10.11.0.4/dir1/dir2/script_needs_credentials.py.
```

If the curl command is successful, a copy of the python script is downloaded which verifies whether the URL is correct.

Configuring the Script credentials Property

If you have specified an FTP server in the Script property, and the server requires a user name and password credentials, specify the credentials using the Script credentials property. If the FTP server can be accessed anonymously, you need not use the Script credentials property.

Configure the Scripts property with a URL and parameters that match those in the Script credentials property. To configure the Script credentials property, follow the format given below:

```
Section: Script credentials
public_url <username> <password>
```

Example 1

```
Section: Script credentials
ftp://10.11.0.4/dir1/dir2/script1.py userfoo foospass
```

The second line in the Script credentials property specifies the values of the user name (`userfoo`) and password (`foospass`) credentials for the python script `script1.py`.

Include the name of the FTP server that is also in the Scripts property. An example line in the Scripts property is: `ftp://10.11.0.4/dir1/dir2/script1.py -a arg1 -s arg2`. See example 2 in [Configuring the Scripts Property, on page 10](#).

Configuring the Python package Property

If a Python package is required by a script in the Scripts property and is not a part of the standard CentOS Linux release 7.1.1503, you must include information about the package in the Python package property. By including the Python package property in the bootstrap file, you ensure that the Cisco Catalyst 8000V downloads and installs the required Python package before running the script that you specified in the Scripts property.



Note Cisco Catalyst 8000V supports only Python3 in guestshell.

To configure the Python package property, follow the format as specified here:

```
Section: Python package
package_name [ version ] [ sudo ] { [ pip_arg1 [ ..[ pip_arg9] ] ] }
```

The arguments: *version*, **sudo**, and *pip_arg1* to *pip_arg9* are optional. You must put the arguments to the pip command between the “{” and “}” braces.

If you specify the *version* argument, the specific version number is downloaded.

If you specify the *sudo* argument, the package is downloaded as a sudo user.

Sample Configuration (Microsoft Azure)

Example 1

In this example, the second line of the Python package property specifies that the *package_name* is `ncclient` and the *version* is "0.5.2". When the bootstrap file is uploaded, version 0.5.2 of the `ncclient` package is installed in the guestshell container of Cisco Catalyst 8000V.

```
Section: Python package
ncclient 0.5.2
```

Example 2

```
Section: Python package
c8000v_azure_guestshell 1.1.2 sudo {--user}
```

In this example, the second line of the Python package property specifies that the *package_name* is "c8000v_azure_guestshell" and the *version* is "1.1.2". When the bootstrap file is uploaded, version 1.1.2 of the `c8000v_azure_guestshell` package is installed in the guestshell container of Cisco Catalyst 8000V. The following command is executed as a sudo user: `sudo pip install c8000v_azure_guestshell==1.1.2 --user`.



Note If you do not specify an argument, `--user` is used as the default argument.

Sample Configuration (Google Cloud Platform)

Example 1

Section: Python package

```
ncclient 0.5.2
```

In this example, the second line of the Python package property specifies that the *package_name* is "ncclient", and the *version* is "0.5.2". When the bootstrap file is uploaded, version 0.5.2 of the ncclient package is installed in the guestshell container of the Cisco Catalyst 8000V instance.

Example 2

Section: Python package

```
c8000v_gcp_ha 3.0.0 sudo {--user}
```

In this example, the second line of the Python package property specifies that the *package_name* is "c8000v_gcp_ha", and the *version* is "3.0.0". When the bootstrap file is uploaded, version 3.0.0 of the c8000v_gcp_ha package is installed in the guestshell container of the Cisco Catalyst 8000V instance. The following command is executed as a sudo user: `pip3 install c8000v_gcp_ha=3.0.0 --user`.



Note If you do not specify an argument, --user is used as the default argument.

Configuring the License property

Configure the license property to specify the license technology level for Cisco Catalyst 8000V.

Enter the first line of the property: `Section: License`. Enter the second line of the property which specifies the tech level of the license, using the following format: **TechPackage:***tech_level* .



Note There must be no spaces between TechPackage: and the *tech_level*. The possible *tech_level* values include ax, security, appx, or ipbase)

tech_level must be in lowercase.

Example 1

Section: License

```
TechPackage:security
```

Providing the Day 0 Bootstrap File

Provide the Day 0 bootstrap file which creates a Cisco Catalyst 8000V VM by executing the following Azure CLI command:

```
az vm create --name C8000V-name --resource-group resource-group { [ arg1 [ ..[ arg9 ] ] ] }
--custom-data bootstrap-file
```

For further information on the **az vm create** command, see: <https://docs.microsoft.com/en-us/cli/azure/vm?view=azure-cli-latest#az-vm-create>.

See the following example:

```
az vm create -n c8000V-VM-Name -g MyResourceGroup --image
cisco:cisco-c8000V-1000v:16_6:16.6.120170804 --data-disk-sizes-gb 8 --availability-set
myAvlSet --nics nic1 nic2 nic3 nic4 --admin-username azureuser --admin-password "+Cisco123456"
--authentication-type password -l westus --size Standard_DS4_v2 --custom-data bootstrap.txt..
```

When you execute this command, a Cisco Catalyst 8000V VM is created. The router is configured using the commands in the bootstrap file: "bootstrap.txt".

Use the **Cisco C8000V Settings** option to provide the custom data bootstrap config file.

For further information on managing Linux VMs, see: [Tutorial: Create and Manage Linux VMs with the Azure CLI 2.0](#).

Verifying the Custom Data Configuration (Microsoft Azure)

After you upload the Day 0 bootstrap file, the VM is created and configuration commands are executed. Perform the following commands to verify the configuration commands of each property.

To help determine if the license property worked, in Cisco IOS XE CLI on Cisco Catalyst 8000V, enter the **show version** command. For example, you should see a reference to the security license.

To see if errors occurred after running the commands in the scripts property, look at the customdata.log file in the /home/guestshell/customdata directory. The *scriptname*.log file stores any output sent to STDOUT by the script.

To check if the Python property worked, enter the **pip freeze | grep package-name** command to view the currently installed python packages. Search for the package *package-name* in which you are interested.

To check if the Cisco IOS XE commands were successful in the IOS Configuration property, enter the **show running-configuration** command. The following is a sample output for this command:

```
Router#show version
Cisco IOS XE Software, Version
Copyright (c) 1986-2020 by Cisco Systems, Inc.
```

```
Cisco IOS-XE software, Copyright (c) 2005-2020 by cisco Systems, Inc.
All rights reserved. Certain components of Cisco IOS-XE software are
licensed under the GNU General Public License ("GPL") Version 2.0. The
software code licensed under GPL Version 2.0 is free software that comes
with ABSOLUTELY NO WARRANTY. You can redistribute and/or modify such
GPL code under the terms of GPL Version 2.0. For more details, see the
documentation or "License Notice" file accompanying the IOS-XE software,
or the applicable URL provided on the flyer accompanying the IOS-XE
software.
```

```
ROM: IOS-XE ROMMON
```

```
Router uptime is 1 minute
Uptime for this control processor is 7 minutes
System returned to ROM by reload
System image file is "bootflash:packages.conf"
Last reload reason: Unknown reason
```

This product contains cryptographic features and is subject to United States and local country laws governing import, export, transfer and use. Delivery of Cisco cryptographic products does not imply

third-party authority to import, export, distribute or use encryption. Importers, exporters, distributors and users are responsible for compliance with U.S. and local country laws. By using this product you agree to comply with applicable laws and regulations. If you are unable to comply with U.S. and local laws, return this product immediately.

A summary of U.S. laws governing Cisco cryptographic products may be found at: <http://www.cisco.com/wl/export/crypto/tool/stqrg.html>

If you require further assistance please contact us by sending email to export@cisco.com.

License Level: ipbase
License Type: N/A(Smart License Enabled)
Next reload license Level: ipbase

The current throughput level is 250000 kbps

Smart Licensing Status: Registration Not Applicable/Not Applicable

cisco C8000V (VXE) processor (revision VXE) with 2271486K/3075K bytes of memory.
Processor board ID 9MUG8CATY8R
Router operating mode: Controller-Managed
1 Gigabit Ethernet interface
32768K bytes of non-volatile configuration memory.
8106756K bytes of physical memory.
11530240K bytes of virtual hard disk at bootflash:.

Configuration register is 0x2102

```
[guestshell@guestshell ~]$ pip3 freeze | grep gpg==1.10.0  
gpg==1.10.0  
[guestshell@guestshell ~]$
```

```
Router#show running-config  
Building configuration...
```

```
Current configuration : 6982 bytes  
!  
! Last configuration change at 14:34:36 UTC Fri Nov 6 2020 by NETCONF  
!  
version 17.3  
service timestamps debug datetime msec  
service timestamps log datetime msec  
service password-encryption  
! Call-home is enabled by Smart-Licensing.  
service call-home  
platform qfp utilization monitor load 80  
platform punt-keepalive disable-kernel-core  
platform console serial  
!  
hostname Router  
!  
boot-start-marker  
boot-end-marker  
!  
vrf definition 65528  
!  
  address-family ipv4  
  exit-address-family  
!  
no logging buffered  
no logging rate-limit  
!
```

```

aaa new-model
!
aaa authentication login default local
aaa authentication enable default enable
aaa authorization console
aaa authorization exec default local
!
aaa session-id common
fhrp version vrrp v3
!
no ip dhcp use class
!
no ip igmp ssm-map query dns
login on-success log
ipv6 unicast-routing
!
subscriber templating
!
multilink bundle-name authenticated
!
crypto pki trustpoint TP-self-signed-2465303444
  enrollment selfsigned
  subject-name cn=IOS-Self-Signed-Certificate-2465303444
  revocation-check none
  rsakeypair TP-self-signed-2465303444
!
crypto pki trustpoint SLA-TrustPoint
  enrollment pkcs12
  revocation-check crl
!
!
crypto pki certificate chain TP-self-signed-2465303444
certificate self-signed 01
  30820330 30820218 A0030201 02020101 300D0609 2A864886 F70D0101 05050030
  31312F30 2D060355 04031326 494F532D 53656C66 2D536967 6E65642D 43657274
  69666963 6174652D 32343635 33303334 3434301E 170D3230 31313036 31343333
  35345A17 0D333031 31303631 34333335 345A3031 312F302D 06035504 03132649
  4F532D53 656C662D 5369676E 65642D43 65727469 66696361 74652D32 34363533
  30333434 34308201 22300D06 092A8648 86F70D01 01010500 0382010F 00308201
  0A028201 0100B02F AD33A0FF 0C50D3F2 D06CFDC6 F3CB73BB 4070D649 E07D16CE
  E6271C90 34E86882 822C8D71 E4BAC29D 85285258 51E748E1 8C9FB2C5 12242A22
  7FB71551 02CB4DBC 64089D2F 8DBB6C4A D3E2F112 8E16E71F FE70D102 F59862A3
  E920E77E 52E62E02 1979F800 3D13601F 27C42F81 483BFB34 697F1C20 3952626A
  CA1F5805 26D50A39 33F264D6 1AD485A0 8EB45882 FC97DCA2 106C8FAD 8CDBC0E6
  FF609188 B4677AB0 FBBE77F2 359EA002 E1A5D37D EA895FF3 92732A2B 63465DFD
  4A2A277C 17E7F720 2007A6B6 A7C7296F D0CD2707 8C7C9690 F86B0642 1BA9F28C
  F729157B 8C472E40 78A4E6BE 70471018 4B62EE36 48193FCA 062DB09F 38BC420B
  687E5866 DFA10203 010001A3 53305130 0F060355 1D130101 FF040530 030101FF
  301F0603 551D2304 18301680 14ABBD00 3D02C6E1 7706FA96 29B037A8 583E7B2E
  69301D06 03551D0E 04160414 ABBD003D 02C6E177 06FA9629 B037A858 3E7B2E69
  300D0609 2A864886 F70D0101 05050003 82010100 40C60BF0 2184CF86 08CACB66
  73E74D63 E87A6661 DC839037 D0DB08D0 33C4993C EC326432 E3573D1B EC3B42AF
  F410BF72 2AAB6D8F 1406B352 FE6B5365 CCA7E094 96980FC7 A4B77A02 49CB8C01
  3EC87F01 58BFEE33 0DA222DB 0A1BA130 0AC01F1F FDBF2085 D41EFA45 7A4C7F5E
  2D004D04 D11433BF 69337D90 117A86ED 2CF57A49 AD7DA227 129E53DF 55E12E03
  4D8E0097 A29DC365 11E8B386 891C310E F19EDF6D D9B3EA1E E26ABDBD EF82D8E9
  B0484E26 C0FC1D71 91B19B70 221E1A1A 090F8EA1 3A5FC4FD A4EF36CD EFD2F1F4
  6056C87D 8A76ED1A 68FB76F5 956C6B50 7EFA9D8C 90EA910F 187EBD13 0BF76E5A
  0B9CE20E AA5927C4 7AD13C28 58C6E920 76E36475
      quit
crypto pki certificate chain SLA-TrustPoint
certificate ca 01
  30820321 30820209 A0030201 02020101 300D0609 2A864886 F70D0101 0B050030
  32310E30 0C060355 040A1305 43697363 6F312030 1E060355 04031317 43697363

```



```

6F204C69 63656E73 696E6720 526F6F74 20434130 1E170D31 33303533 30313934
3834375A 170D3338 30353330 31393438 34375A30 32310E30 0C060355 040A1305
43697363 6F312030 1E060355 04031317 43697363 6F204C69 63656E73 696E6720
526F6F74 20434130 82012230 0D06092A 864886F7 0D010101 05000382 010F0030
82010A02 82010100 A6BCBD96 131E05F7 145EA72C 2CD686E6 17222EA1 F1EFF64D
CBB4C798 212AA147 C655D8D7 9471380D 8711441E 1AAF071A 9CAE6388 8A38E520
1C394D78 462EF239 C659F715 B98C0A59 5BBB5CBD 0CFEBEA3 700A8BF7 D8F256EE
4AA4E80D DB6FD1C9 60B1FD18 FFC69C96 6FA68957 A2617DE7 104FDC5F EA2956AC
7390A3EB 2B5436AD C847A2C5 DAB553EB 69A9A535 58E9F3E3 C0BD23CF 58BD7188
68E69491 20F320E7 948E71D7 AE3BCC84 F10684C7 4BC8E00F 539BA42B 42C68BB7
C7479096 B4CB2D62 EA2F505D C7B062A4 6811D95B E8250FC4 5D5D5FB8 8F27D191
C55F0D76 61F9A4CD 3D992327 A8BB03BD 4E6D7069 7CBADF8B DF5F4368 95135E44
DFC7C6CF 04DD7FD1 02030100 01A34230 40300E06 03551D0F 0101FF04 04030201
06300F06 03551D13 0101FF04 05300301 01FF301D 0603551D 0E041604 1449DC85
4B3D31E5 1B3E6A17 606AF333 3D3B4C73 E8300D06 092A8648 86F70D01 010B0500
03820101 00507F24 D3932A66 86025D9F E838AE5C 6D4DF6B0 49631C78 240DA905
604EDCDE FF4FED2B 77FC460E CD636FDB DD44681E 3A5673AB 9093D3B1 6C9E3D8B
D98987BF E40CBD9E 1AECA0C2 2189BB5C 8FA85686 CD98B646 5575B146 8DFC66A8
467A3DF4 4D565700 6ADF0F0D CF835015 3C04FF7C 21E878AC 11BA9CD2 55A9232C
7CA7B7E6 C1AF74F6 152E99B7 B1FCF9BB E973DE7F 5BDDEB86 C71E3B49 1765308B
5FB0DA06 B92AFE7F 494E8A9E 07B85737 F3A58BE1 1A48A229 C37C1E69 39F08678
80DDCC16 D6BACECA EEBC7CF9 8428787B 35202CDC 60E4616A B623CDBD 230E3AFB
418616A9 4093E049 4D10AB75 27E86F73 932E35B5 8862FDAE 0275156F 719BB2F0
D697DF7F 28
quit
!
license udi pid C8000V sn 9MUG8CATY8R
diagnostic bootup level minimal
memory free low-watermark processor 69848
!
!
username admin privilege 15 secret 9
$14$vKLj$yfnFjRidlKJg9.$4obKgKyy4TsoUs0sJ2t3HXpna3XjYWRBnnYKBwVeJrw
!
redundancy
!
interface Loopback65528
 vrf forwarding 65528
 ip address 192.168.1.1 255.255.255.255
!

```

Verifying the Custom Data Configuration (Google Cloud Platform)

After you run the custom data script, the VM is created and the configuration commands are executed. To verify the same, use the following commands and scripts:

- **show version:** To help determine if the license property worked in Cisco IOS XE CLI on the Cisco Catalyst 8000V instance, enter the **show version** command. For example, the output displays a reference to the security license.
- To see if errors occurred after running commands in the scripts property, look at the `customdata.log` file in the `/bootflash/<cloud>/` directory. The `scriptname.log` file stores any output that is sent to STDOUT by the script.
- To verify whether the Python property worked, enter the `pip freeze | grep <package-name>` command from the Guestshell to view the currently installed Python packages. Here, *package-name* refers to the package that you are specifically searching for.
- To verify the Cisco IOS XE commands in the IOS Configuration property, run the **show running-configuration** command.

Day 0 Configuration in the Controller Mode

If you want to perform the day 0 configuration for a Cisco Catalyst 8000V in the controller (SD-WAN) mode, you must provide the contents of the `ciscosdwan_cloud_init.cfg` file downloaded from vManage.

If you want to switch to the Controller mode, or if you are looking to bootstrap Cisco Catalyst 8000V with the Cisco SD-WAN functionalities, see [Install and Upgrade for Cisco Catalyst 8000V Controller Mode](#).



Note For a Cisco Catalyst 8000V instance running on Cisco CSP-5000 hypervisor, when you enter the settings in the **Day Zero Config** screen, ensure that you maintain the format mentioned here:

- **Source File Name:** Enter the value for this field in the format: `day0_ciscosdwan_cloud_init.cfg`.
- **Destination File Name:** Enter the value for this field in the format: `day0-dest-filename/openstack/content/ciscosdwan_cloud_init.cfg`.



Note With the SD-WAN format configurations, if the `confd` cannot apply the config successfully at the first boot, the box might not have a working config at Day0. This is particularly critical in public cloud environments where SSH is necessary to login. Review the configuration carefully if you encounter issues upon provisioning.

Verifying the Router Operation Mode and Day 0 Configuration

To verify whether you've deployed or upgraded to the IOS XE 17.4 or later releases successfully, run the **show version** command. This command displays the version of your instance, and the **operating device-mode** parameter displays the mode in which your Cisco Catalyst 8000V instance is running.

Sample configuration output for a Cisco Catalyst 8000V instance in autonomous mode

```
Device# show version | inc operating
Router operating mode: Autonomous
Device# show platform software device-mode
Operating device-mode: Autonomous
Device-mode bootup status:
-----
Device# show platform software chasfs r0 brief | inc device_managed_mode
/tmp/chassis/local/rp/chasfs/etc/device_managed_mode : [autonomous]
/tmp/fp/chasfs/etc/device_managed_mode : [autonomous]
Device# show version | inc Last reload
Last reload reason: Enabling autonomous-mode
```

Frequently Asked Questions

- Q.** I have been using Cisco IOS XE image until now. Which mode should I now choose?
- A.** If you have been using the Cisco IOS XE universalk9 image so far, deploy the IOS XE 17.4 image and enter the autonomous mode.
- Q.** If I am upgrading to the Cisco Catalyst 8000V 17.4 release, do I need to provide the bootstrap configuration?
- A.** If you are an existing non-SD WAN user and are upgrading to the IOS XE 17.4 release (autonomous mode), you can directly perform the upgrade. You need not perform the Day 0 or custom data configuration again.

For a Cisco Catalyst 8000V instance running on Microsoft Azure or Google Cloud Platform, the device uses the custom data that you provided the first time you configured your Cisco Catalyst 8000V instance.

For Cisco Catalyst 8000V instances running on AWS, the device fetches the custom data from the cloud service provider.

- Q.** What happens to my custom data configuration after switching modes?
- A.** The existing configuration data is deleted. You must perform the bootstrap or custom data configuration just as you do for a fresh installation.
- Q.** What happens to my custom data after a factory reset?
- A.** When you perform a factory reset, the configuration and the files present on the disk are erased. The router boots up like a fresh install and looks for configuration files at the appropriate location. This action determines the mode and the associated configuration.
- Q.** Can I deploy my Cisco Catalyst 8000V instance in any mode with PayG license?
- A.** If you use the PayG licensing model, you cannot deploy the Cisco Catalyst 8000V instance in the controller mode or switch to the controller mode. This mode does not support the PayG licensing model.

