



Deploying a Cisco CSR 1000v VM on Microsoft Azure using a Day 0 Bootstrap File



Note Deploying a Cisco CSR 1000v VM using a Day 0 bootstrap file is supported on Cisco IOS XE Fuji 16.9.1 or later releases.

When you deploy a Cisco CSR 1000v VM instance on Microsoft Azure, you can optionally choose to use a "Day 0" bootstrap file to achieve a variety of automation goals. The Day 0 bootstrap file in Azure 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 CSR with a desired technology package.

To launch a CSR 1000v instance with Day 0 bootstrapping, perform the following steps:

- [Editing the Day 0 Bootstrap File, on page 1](#)
- [Providing the Day 0 Bootstrap File, on page 4](#)
- [Verifying the Configuration after Uploading the Day 0 Bootstrap File, on page 5](#)

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>.

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 CSR1
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`, you can enter a list of Cisco IOS XE configuration commands to be run on the Cisco CSR 1000v router.

When you run this command, the above mentioned IOS configuration is applied to the CSR 1000v 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 CSR 1000v instance. To access the guestshell container, use the **guestshell** EXEC mode command. For more information on guestshell command, see the [Programmability Configuration Guide](#).

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

```
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 CSR1000v instance.

To view more examples of the scripts, see "scripts" at: <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 CSR 1000v 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 CSR 1000v 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](#).

Prior to uploading the bootstrap file and running the bash or python script, we recommend 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 username 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 username (`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 the *Configuring Scripts Property* section.

Configuring the Python package Property

If a Python package is required by a script in the Scripts property and it is not 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 CSR 1000v downloads and installs the required Python package before running the script that you specified in the Scripts property.

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

```
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 “{“ and “}” braces.

If the *version* argument is specified, a specific version number is downloaded.

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

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 the Cisco CSR 1000v.

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

Example 2

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

In this example, the second line of the Python package property specifies that the *package_name* is "csr_azure_guestshell" and the *version* is "1.1.2". When the bootstrap file is uploaded, version 1.1.2 of the csr_azure_guestshell package is installed in the guestshell container of the Cisco CSR 1000v. The following command is executed as a sudo user: `sudo pip install csr_azure_guestshell==1.1.2 --user`.

Configuring the License property

Configure the license property to specify the license technology level for the Cisco CSR 1000v.

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*. (*tech_level* values: 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 CSR 1000v VM, by performing the following Azure CLI command:

```
az vm create --name CSR-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>.

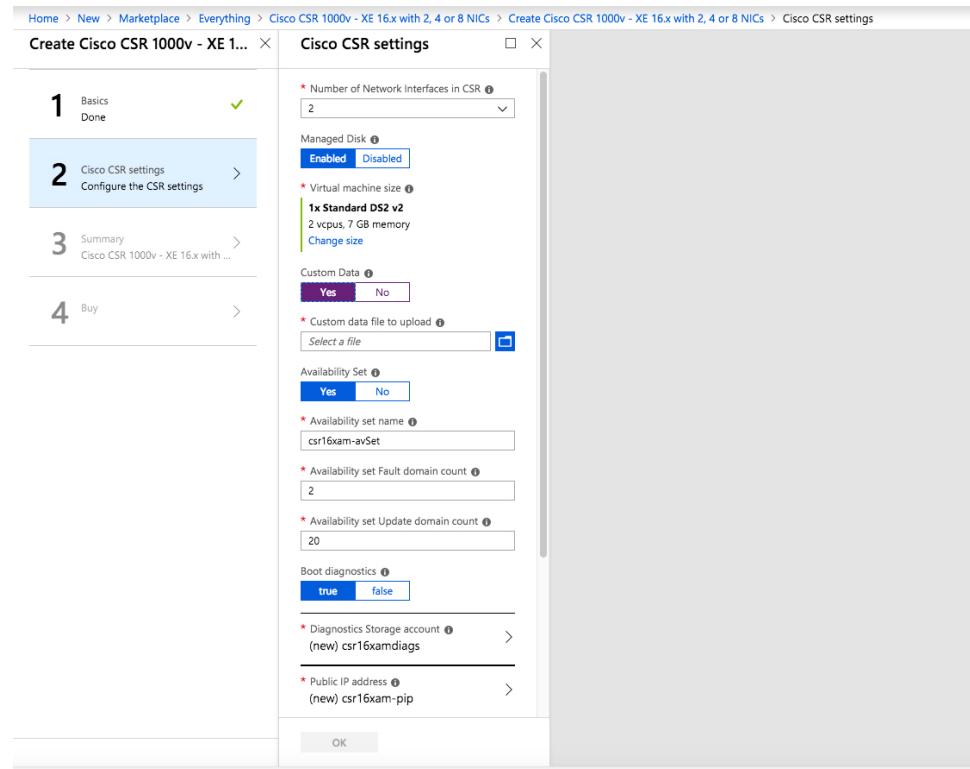
See the following example:

```
az vm create -n CSR-VM-Name -g MyResourceGroup --image
cisco:cisco-csr-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 CSR 1000v VM is created. The router is configured using the commands in the bootstrap file: "bootstrap.txt".

If you are using a Cisco 16.x template to create a CSR 1000v instance, the custom data upload box is provided as shown in the following image:

Figure 1: Uploading Day0 Bootstrap File



Use the **Cisco CSR 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 Configuration after Uploading the Day 0 Bootstrap File

After the Day 0 bootstrap file is uploaded, 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 the CSR 1000v, enter the **show version** command. For example, you should see 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 `/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, searching for the package `package-name` in which you are interested.

To check the Cisco IOS XE commands were successful in the IOS Configuration property, enter the `show running-configuration` command.