Integrate Cisco ESC and OpenStack

Introduction

In this activity, you will integrate Cisco NSO with Cisco ESC and OpenStack. You will learn how to onboard new images and flavors on ESC by using NSO.

After completing this activity, you will be able to:

- Configure NSO to work with ESC.
- Onboard images onto ESC.

Job Aids

The following job aid is available to help you complete the lab activities:

- This Lab Guide

The following table contains credentials and addresses that you might need.

Device	Username	Password	Address
Student-VM	student	1234QWer	10.0.0.102
NSO application	admin	admin	10.0.0.102
ESC	admin	admin	10.0.0.104
OpenStack	admin	admin	10.0.0.103
OpenStack SSH	root	1234QWer	10.0.0.103

Required Resources

The following resources and equipment are required for completing the activities in this lab guide:

- PC or laptop with a web browser
- Access to the Internet

Command Syntax Reference

This lab guide uses the following conventions for command syntax:

Formatting	Description and Examples
show running config	Commands in steps use this formatting.
Example	Type **show running config**
Example	Use the **name** command.
`show running config`	Commands in CLI outputs and configurations use this formatting.
highlight	CLI output that is important is highlighted.

Example	`student@student-vm:~$ ncs --version`
	`5.3.2`

Example	Save your current configuration as the default **startup config**. `Router Name# copy running startup`
brackets ([ ])	Indicates optional element. You can choose one of the options.
Example:	`(config-if)# frame-relay lmi-type {ansi|cisco|q933a}`
italics font	Arguments for which you supply values.
Example	Open file **ip tcp window-size** *bytes*
angle brackets (<>)	In contexts that do not allow italics, arguments for which you supply values are enclosed in angle brackets [<>]. Do not type the brackets when entering the command.
Example	If the command syntax is **ping** <*ip_address*>, you enter ping *192.32.10.12*

Formatting	Description and Examples
string	A non-quoted set of characters. Type the characters as-is.
Example	(config)# **hostname MyRouter**
vertical line (|)	Indicates that you enter one of the choices. The vertical line separates choices. Do not type the vertical line when entering the command.
Example	If the command syntax is **show ip route|arp**, you enter either **show ip route** or **show ip arp**, but not both.

Command List

The following are the most common commands that you will need.

Linux Shell:

Command	Comment
source /opt/ncs/ncs-5.3.2/ncsrc	Source NSO environmental variable in Docker container.
ls|ll	Display contents of the current directory.
cd	Move directly to user home directory.
cd ..	Exit out of current directory.
cd test	Move into folder "test" which is a subfolder of the current directory.
cd /home/student/nso300	Move into folder "nso300" by specifying direct path to it starting from the root of directory system.
ncs_cli -C -u admin	Log in to NSO CLI directly from local server.

NSO CLI:

Command	Comment
switch cli	Change CLI style.
show ?	Display all command options for current mode.
configure	Enter configuration mode.
commit	Commit new configuration (configuration mode only command).
show configuration	Display new configuration that has not yet been committed (configuration mode only command).

Makefile commands for Docker environment:

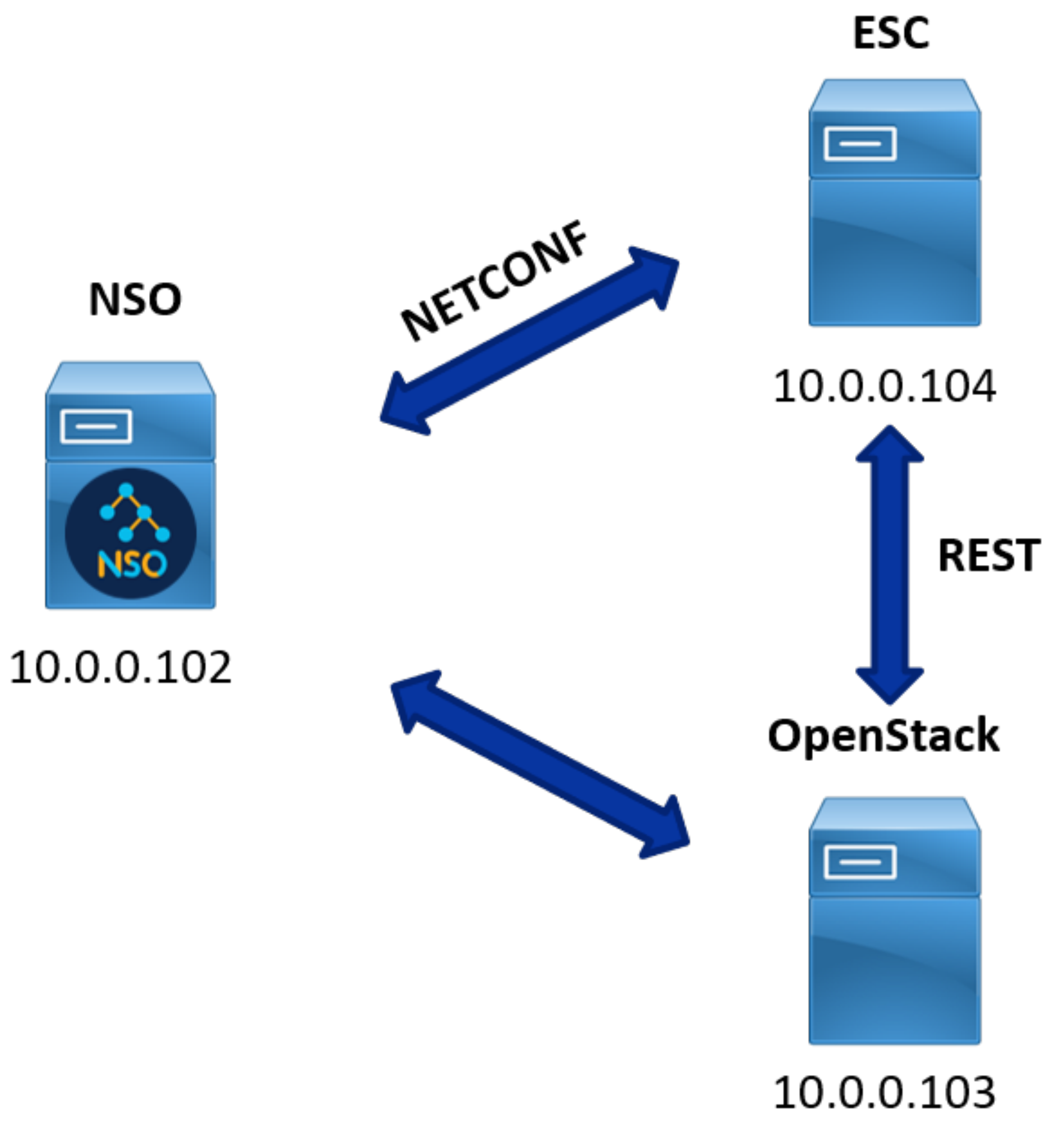
Command	Comment
make build	Builds the main NSO Docker image.
make testenv-start	Starts the NSO Docker environment.

Command	Comment
make testenv-stop	Stops the NSO Docker environment.
make testenv-build	Recompiles and reloads the NSO packages.
make testenv-cli	Enters the NSO CLI of the NSO Docker container.
make testenv-shell	Enters the Linux shell of the NSO Docker container.
make dev-shell	Enters the Linux shell of the NSO Docker development container.

Lab Topology Information

Your lab session is your own personal sandbox. Whatever you do in your session will not be reflected in anyone else's session. There are two topologies. The general one is your lab environment with a Linux server (Student-VM), an OpenStack server instance and an ESC server instance. On the Linux server within that topology is your second topology—a Docker environment, which in this activity contains only an NSO Docker image with your NSO installation.

Topology



Task 1: Start ESC

In this task, you will activate the ESC VM instance via OpenStack.

The final solution for this lab is Cisco NSO being integrated with ESC and ready to deploy NFVs.

Activity

Complete these steps:

Step 1

Connect to the Student-VM server by clicking the icon labelled NSO in the topology.

Step 2

Open the terminal window using the Terminal icon on the bottom bar.

```
student@student-vm:~$
```

Step 3

Establish an SSH session with the OpenStack VM (10.0.0.103) with the credentials **root/1234QWer**.

```
student@student-vm:~$ ssh root@10.0.0.103
The authenticity of host '10.0.0.103 (10.0.0.103)' can't be
established.
ECDSA key fingerprint is SHA256:hio5HmNNpMk/P/qQwb0I/
ajgHv6q3yV9VzlTWj6nLGk.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.0.0.103' (ECDSA) to the list of known
hosts.
root@10.0.0.103's password:
Activate the web console with: systemctl enable --now cockpit.socket

Last login: Thu Aug 27 03:20:53 2020 from 10.6.0.33
[admin@openstack ~]$
```

Step 4

Source file keystonerc_admin.

```
[admin@openstack ~]$ source keystonerc_admin
[admin@openstack ~(keystone_admin)]$
```

Step 5

List all available instances in OpenStack and observe the ESC ID, which you will later use to start the ESC instance.

```
[admin@openstack ~(keystone_admin)]$ nova list
+--------------------------------------+--------+---------+------------
+-------------+---------------------+
| ID                                   | Name   | Status  | Task State
| Power State | Networks            |
+--------------------------------------+--------+---------+------------
+-------------+---------------------+
| 56c70d20-0ca7-4a53-92a8-3ecf2f8eeaf5 | esc-53 | SHUTOFF | -
| Shutdown    | mgmt=10.0.0.104 |
+--------------------------------------+--------+---------+------------
+-------------+---------------------+
```

Step 6

Start the ESC instance. Make sure you use the correct ID.

```
[admin@openstack ~(keystone_admin)]$nova start
56c70d20-0ca7-4a53-92a8-3ecf2f8eeaf5
Request to start server 56c70d20-0ca7-4a53-92a8-3ecf2f8eeaf5 has been
accepted.
[admin@openstack ~(keystone_admin)]$
```

Step 7

Verify that the ESC instance is running.

```
[admin@openstack ~(keystone_admin)]$ nova list
+--------------------------------------+--------+--------+------------
+-------------+------------------+
| ID                                   | Name   | Status | Task State |
Power State | Networks         |
+--------------------------------------+--------+--------+------------
+-------------+------------------+
| 56c70d20-0ca7-4a53-92a8-3ecf2f8eeaf5 | esc-53 | ACTIVE | -          |
Running     | mgmt=10.0.0.104 |
+--------------------------------------+--------+--------+------------
+-------------+------------------+
[admin@openstack ~(keystone_admin)]$
```

Step 8

Exit the SSH session with Openstack.

```
[admin@openstack ~(keystone_admin)]$ exit
logout
Connection to 10.0.0.103 closed.
student@student-vm:~$
```

Activity Verification

You have completed this task when you attain this result:

- ESC instance has reached ACTIVE status.

Task 2: Add ESC to NSO

In this task, you will add the ESC VM instance to NSO. In this way, you can manage the ESC and onboard additional images and deployment flavors via NSO.

Activity

Complete these steps:

Step 1

Copy the **tailf-etsi-rel2-nfvo** package, located in **~/packages** to the nso300 project **packages** folder.

```
student@student-vm:~$ cp -r packages/tailf-etsi-rel2-nfvo nso300/
packages/
student@student-vm:~$
```

Step 2

Navigate to the *nso300* folder and list the contents of the *packages* folder.

There are now two packages present—the **esc** package is the NED package for the Elastic Services Controller, and the **tailf-etsi-rel2-nfvo** is a package meant for orchestration and deployment of virtualized network functions.

```
student@student-vm:~$ cd nso300/
student@student-vm:~/nso300$ ls packages/
esc  tailf-etsi-rel2-nfvo
student@student-vm:~/nso300$
```

Step 3

Recompil the ESC package just to make sure that NED is up-to-date with your NSO version. Use the **make testenv-build** command.

Some warnings might be present, but no errors should occur in the build process. Make sure that the packages are successfully reloaded.

```
student@student-vm:~/nso300$ make testenv-build
for NSO in $(docker ps --format '{{.Names}}' --filter label=testenv-
nso300-5.3.2-student --filter label=nidtype=nso); do \
        echo "-- Rebuilding for NSO: ${NSO}"; \
        docker run -it --rm -v /home/student/nso300:/src --volumes-from
${NSO} --network=container:${NSO} -e NSO=${NSO} -e PACKAGE_RELOAD= -e
SKIP_LINT= -e PKG_FILE=nso300.gitlab.local/nso300/package:5.3.2-student
nso300.gitlab.local/cisco-nso-dev:5.3.2 /src/nid/testenv-build; \
done
-- Rebuilding for NSO: testenv-nso300-5.3.2-student-nso
(^package-meta-data.xml$|\.cli$|\.yang$)
make: Entering directory '/var/opt/ncs/packages/esc/src'
=== Build esc

< … Output Omitted … >

=== Build tailf-etsi-rel2-nfvo

< … Output Omitted … >

-- Reloading packages for NSO testenv-nso300-5.3.2-student-nso
reload-result {
    package esc-nc-1.0
    result true
}
reload-result {
    package tailf-etsi-rel2-nfvo
    result true
}
student@student-vm:~/nso300$
```

Step 4

Enter the NSO CLI with the **make testenv-cli** command and switch the CLI mode.

```
student@student-vm:~/nso300$ make testenv-cli
docker exec -it testenv-nso300-5.3.2-student-nso bash -lc 'ncs_cli -u
admin'

admin connected from 127.0.0.1 using console on ccb73064494d
admin@ncs> switch cli
admin@ncs#
```

Step 5

Enter the configuration mode and create an authentication group **esc**, that contains the **admin/admin** authentication credentials. Commit the authentication group.

```
admin@ncs# config
admin@ncs(config)# devices authgroups group esc default-map remote-name
admin remote-password admin
admin@ncs(config-group-esc)# commit
Commit complete.
admin@ncs(config-group-esc)# top
admin@ncs(config)#
```

Step 6

Add Cisco ESC into Cisco NSO. The NED is already loaded, and Cisco ESC is running.

Use the following parameters for the new device:

- name: **esc0**
- address: **10.0.0.104**
- port: **830**
- authgroup: **esc**
- device-type: **netconf**
- ned-id: **esc**

```
admin@ncs(config)# devices device esc0 address 10.0.0.104 port 830
authgroup esc device-type netconf ned-id esc-nc-1.0
admin@ncs(config-device-esc0)# state admin-state unlocked
admin@ncs(config-device-esc0)# commit
Commit complete.
admin@ncs(config-device-esc0)# exit
```

```
admin@ncs(config)# exit
admin@ncs#
```

Step 7

Fetch the SSH keys and perform a **sync-from** on the newly added **esc0** device. No errors or warnings should pop up.

```


admin@ncs# devices fetch-ssh-host-keys
fetch-result {
    device esc0
    result updated
    fingerprint {
        algorithm ssh-rsa
        value 75:29:28:ed:a9:e4:5f:cb:32:4e:db:3f:fd:c4:6e:21
    }
}
admin@ncs# devices sync-from
sync-result {
    device esc0
    result true
}
admin@ncs# exit
```

It is important to do a **sync-from** at this point, otherwise, testing any of the services will fail because the Cisco ESC device will be out-of-sync. Make sure that the ESC instance is active before performing a sync.

Activity Verification

You have completed this task when you attain these results:

- **esc** and **tailf-etsi-rel2-nfvo** packages are successfully built and reloaded.
- The ***esc0*** device has been added and synchronized with NSO.

Task 3: Check ESC Status

In this task, you will verify the ESC operational state.

Activity

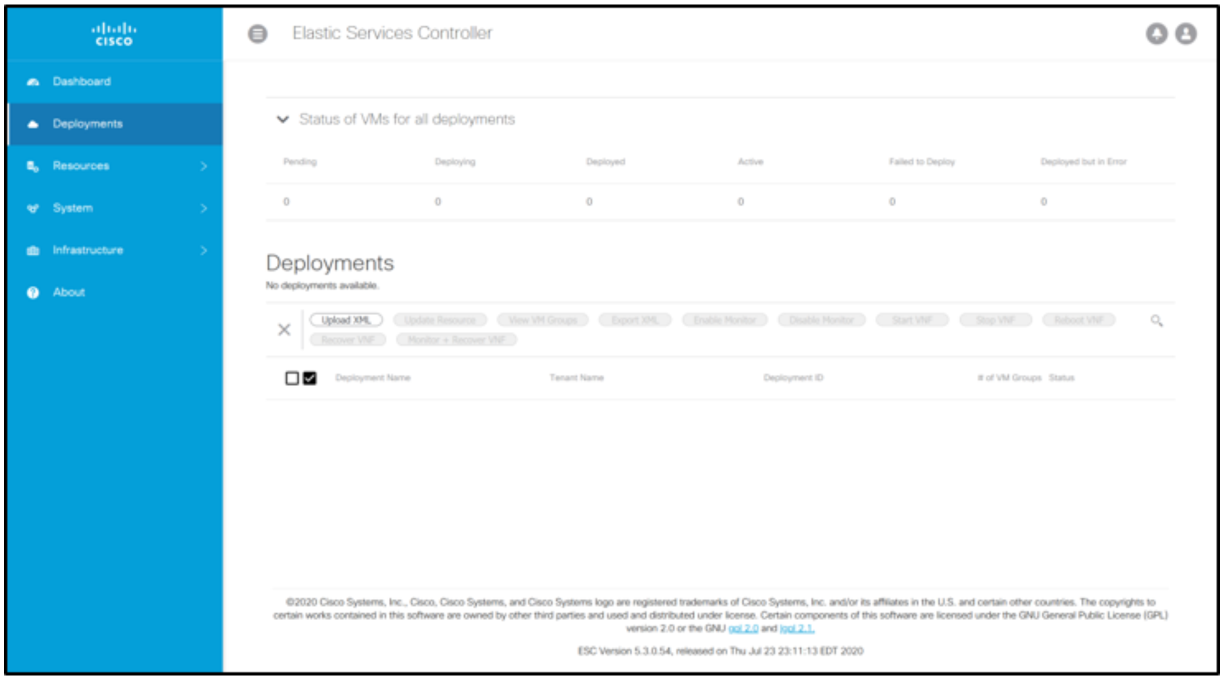
Complete these steps:

Step 1

From the desktop open the Mozilla Firefox browser.

Step 2

Log in to the ESC Web Portal through a web browser, using the URL https://10.0.0.104 and the credentials **admin/admin**. Familiarize yourself with the UI.



Elastic Services Controller

Status of VMs for all deployments

Pending	Deploying	Deployed	Active	Failed to Deploy	Deployed but in Error
0	0	0	0	0	0

Deployments

No deployments available.

Deployment Name	Tenant Name	Deployment ID	# of VM Groups	Status

©2020 Cisco Systems, Inc., Cisco, Cisco Systems, and Cisco Systems logo are registered trademarks of Cisco Systems, Inc. and/or its affiliates in the U.S. and certain other countries. The copyrights to certain works contained in this software are owned by other third parties and used and distributed under license. Certain components of this software are licensed under the GNU General Public License (GPL) version 2.0 or the GNU gpl 2.0 and lgpl 2.1.

ESC Version 5.3.0.54, released on Thu Jul 23 23:11:13 EDT 2020

Step 3

Open the terminal window using the Terminal icon on the bottom bar.

```
student@student-vm:~$
```

Step 4

Establish an SSH session with ESC. Use the same credentials as you used for the ESC Web Portal.

```
student@student-vm:~/nso300$ ssh admin@10.0.0.104
                           WARNING!!!
              READ THIS BEFORE ATTEMPTING TO LOGON

This System is for the use of authorized users only.  Individuals using
this
computer without authority, or in excess of their authority, are
subject to
having all of their activities on this system monitored and recorded by
system personnel.  In the course of monitoring individuals improperly
using
this system, or in the course of system maintenance, the activities of
authorized users may also be monitored.  Anyone using this system
expressly
consents to such monitoring and is advised that if such monitoring
reveals
possible criminal activity, system personnel may provide the evidence
of
such monitoring to law enforcement officials.

Copyright (c) 2020 by Cisco Systems, Inc.
All Rights Reserved
```

```
2018 Cisco Systems, Inc., Cisco, Cisco Systems, and Cisco Systems logo
are
registered trademarks of Cisco Systems, Inc. and/or its affiliates in
the
U.S. and certain other countries.  The copyrights to certain works
contained
in this software are owned by other third parties and used and
distributed
under license.  Certain components of this software are licensed under
the
GNU General Public License (GPL) version 2.0 or the GPL v2.0 and
LGPLv2.1.
admin@10.155.1.20's password:
Permission denied, please try again.
admin@10.155.1.20's password:
Last failed login: Thu Aug 27 08:07:11 UTC 2020 from 10.201.4.95 on
ssh:notty
There was 1 failed login attempt since the last successful login.
Last login: Thu Aug 27 07:24:26 2020 from 10.201.4.95

    ######################################################################
#         ESC on esc-53.novalocal
    ######################################################################

[admin@esc-53 ~]$
```

Step 5

Exit the SSH session.

```
[admin@esc-53 ~]$ exit
logout
Connection to 10.0.0.104 closed.
student@student-vm:~/nso300$
```

Activity Verification

You have completed this task when you attain these results:

- ESC is reachable both via Web Portal and SSH.

Task 4: Configure NSO to Work with ESC and OpenStack

In this task, you will create a Netconf subscription for ESC notifications, create authentication groups for images you will onboard and host these images inside your NSO Docker container.

Activity

Complete these steps:

Step 1

Connect to the NSO CLI and switch the CLI mode.

```
student@student-vm:~/nso300$ make testenv-cli
docker exec -it testenv-nso300-5.3.2-student-nso bash -lc 'ncs_cli -u
admin'

admin connected from 127.0.0.1 using console on ccb73064494d
admin@ncs> switch cli
admin@ncs#
```

Step 2

Configure the netconf subscription for ESC notifications. Subscribe to the esc0 device with the admin user.

```
admin@ncs# config
admin@ncs(config)# nfvo settings-esc netconf-subscription username
admin
admin@ncs(config)# nfvo settings-esc netconf-subscription esc-device
esc0
admin@ncs(config-esc-device-esc0)# top
admin@ncs(config)# commit
Commit complete.
admin@ncs(config)#
```

Step 3

Create and configure authgroups for CSR and ASA virtual devices with **admin/admin** credentials. These will be needed later for management of those types of devices. Exit the NSO CLI after that.

```
admin@ncs(config)# devices authgroups group csr default-map remote-name
admin remote-password admin remote-secondary-password admin
admin@ncs(config-group-csr)# exit
admin@ncs(config)# devices authgroups group asa default-map remote-name
admin remote-password admin remote-secondary-password admin
admin@ncs(config-group-asa)# top
admin@ncs(config)# commit
Commit complete.
admin@ncs(config)# exit
admin@ncs# exit
student@student-vm:~/nso300$
```

Step 4

List the contents of the **images** folder, located in the **extra-files** directory.

This directory contains two device images—ASA and CSR. When the NSO Docker container is built, the contents of the **extra-files** folder are copied to the root directory of the container. This means these images are located inside the NSO container under ***/fileserver/images*** directory.

```
student@student-vm:~/nso300$ ls extra-files/fileserver/images/
asav992.qcow2  csr1000v-universalk9.16.09.05.qcow2
student@student-vm:~/nso300$
```

Step 5

Enter the NSO CLI again. Switch the CLI and enter the configuration mode.

```
student@student-vm:~/nso300$ make testenv-cli
docker exec -it testenv-nso300-5.3.2-student-nso bash -lc 'ncs_cli -u
admin'

admin connected from 127.0.0.1 using console on ccb73064494d
admin@ncs> switch cli
admin@ncs# config
Entering configuration mode terminal
admin@ncs(config)#
```

Step 6

Enable the image server, provided by the *nfvo* package. This functionality allows you to host the virtual images and day0 configuration inside your Docker container. Use it to expose the ***/fileserver*** folder. You can use the port 8080. Exit the NSO CLI after that.

```
admin@ncs(config)# nfvo settings image-server port 8080 document-root /
fileserver
admin@ncs(config)# commit
Commit complete.
admin@ncs(config)# exit
admin@ncs# exit
student@student-vm:~/nso300$
```

Activity Verification

You have completed this task when you attain these results:

- The ESC netconf subscription has been created.
- Authentication groups have been created for ASA and CSR devices.
- The file server hosting the ASA and CSR images has been started.

Task 5: Image and Flavor Onboarding

In this task, you will onboard the Cisco CSR and ASA images on OpenStack that you will use in the following lab to run VNFs (virtual network functions).

Activity

Complete these steps:

Step 1

Open and study the **vnf_catalog** which contains VNFDs that you will be using for your service. The catalog specifies image locations and other requirements for CSR and ASA VNFs. The VNF catalog can be found in **~/nso300/extra-files/cdb_config/vnf_catalog.xml**.

This catalog has been prepared beforehand and is ready to import. Make sure that the image location matches the location that is used by the image server from the previous task.

When working with ESC in real life, you will have to create this catalog yourself via NSO CLI, while referring to the ESC and Cisco NFVO pack documentation and examples.

```
student@student-vm:~/nso300$ cat extra-files/cdb_config/vnf_catalog.xml
<config xmlns="http://tail-f.com/ns/config/1.0">
  <nfvo xmlns="http://tail-f.com/pkg/tailf-etsi-rel2-nfvo">
  <vnfd>
    <id>ASAv</id>
    <product-name>virtual ASA</product-name>
    <provider>Cisco</provider>
    <version>9.9.2</version>
    <product-info-description>Virtual security appliance</product-info-
description>
    <vdu>
      <id>ASA</id>
      <internal-connection-point-descriptor>
        <id>inside</id>
        <external-connection-point-descriptor>cp-inside</external-
connection-point-descriptor>
        <layer-protocol>IPv4</layer-protocol>
        <interface-id xmlns="http://tail-f.com/pkg/tailf-etsi-rel2-
nfvo-esc">1</interface-id>
      </internal-connection-point-descriptor>
      <internal-connection-point-descriptor>
        <id>mgmt</id>
        <external-connection-point-descriptor>cp-mgmt</external-
connection-point-descriptor>
        <layer-protocol>IPv4</layer-protocol>
        <interface-id xmlns="http://tail-f.com/pkg/tailf-etsi-rel2-
nfvo-esc">0</interface-id>
      </internal-connection-point-descriptor>
      <internal-connection-point-descriptor>
        <id>outside</id>
        <external-connection-point-descriptor>cp-outside</external-
connection-point-descriptor>
        <layer-protocol>IPv4</layer-protocol>
        <interface-id xmlns="http://tail-f.com/pkg/tailf-etsi-rel2-
nfvo-esc">2</interface-id>
      </internal-connection-point-descriptor>
      <virtual-compute-descriptor>vcd</virtual-compute-descriptor>
      <virtual-storage-descriptor>root</virtual-storage-descriptor>
      <software-image-descriptor>
        <container-format>bare</container-format>
        <disk-format>qcow2</disk-format>
        <image>http://10.0.0.102:8080/images/asav992.qcow2</image>
        <additional-setting xmlns="http://tail-f.com/pkg/tailf-etsi-
rel2-nfvo-esc">
          <id>disk_bus</id>
          <value>ide</value>
        </additional-setting>
        <additional-setting xmlns="http://tail-f.com/pkg/tailf-etsi-
rel2-nfvo-esc">
          <id>e1000_net</id>
          <value>true</value>
        </additional-setting>
      </software-image-descriptor>
      <device-type xmlns="http://tail-f.com/pkg/tailf-etsi-rel2-nfvo-
esc">
        <cli>
          <ned-id>cisco-asa-cli-6.10</ned-id>
        </cli>
      </device-type>
      <day0 xmlns="http://tail-f.com/pkg/tailf-etsi-rel2-nfvo-esc">
        <destination>asa_config.txt</destination>
        <mandatory/>
      </day0>
    </vdu>
    <virtual-compute-descriptor>
      <id>vcd</id>
      <virtual-memory>
        <virtual-memory-size>4.0</virtual-memory-size>
      </virtual-memory>
      <virtual-cpu>
        <number-of-virtual-cpus>1</number-of-virtual-cpus>
      </virtual-cpu>
    </virtual-compute-descriptor>
    <virtual-storage-descriptor>
      <id>root</id>
      <type-of-storage>root</type-of-storage>
      <size-of-storage>10</size-of-storage>
    </virtual-storage-descriptor>
    <external-connection-point-descriptor>
      <id>cp-inside</id>
      <layer-protocol>IPv4</layer-protocol>
    </external-connection-point-descriptor>
    <external-connection-point-descriptor>
      <id>cp-mgmt</id>
      <layer-protocol>IPv4</layer-protocol>
      <management xmlns="http://tail-f.com/pkg/tailf-etsi-rel2-nfvo-
esc"/>
    </external-connection-point-descriptor>
    <external-connection-point-descriptor>
      <id>cp-outside</id>
      <layer-protocol>IPv4</layer-protocol>
    </external-connection-point-descriptor>
    <deployment-flavor>
      <id>asav.basic</id>
      <vdu-profile>
        <vdu>ASA</vdu>
        <min-number-of-instances>1</min-number-of-instances>
        <max-number-of-instances>1</max-number-of-instances>
      </vdu-profile>
      <instantiation-level>
```

```
        <id>small</id>
        <vdu-level>
          <vdu>ASA</vdu>
          <number-of-instances>1</number-of-instances>
        </vdu-level>
      </instantiation-level>
    </deployment-flavor>
  </vnfd>
  <vnfd>
    <id>CSR1kv</id>
    <product-name>CSR 1000v</product-name>
    <provider>Cisco</provider>
    <version>9.16.9</version>
    <product-info-description>Cloud router</product-info-description>
    <vdu>
      <id>CSR</id>
      <internal-connection-point-descriptor>
        <id>left</id>
        <external-connection-point-descriptor>left</external-
connection-point-descriptor>
        <layer-protocol>IPv4</layer-protocol>
        <interface-id xmlns="http://tail-f.com/pkg/tailf-etsi-rel2-
nfvo-esc">1</interface-id>
      </internal-connection-point-descriptor>
      <internal-connection-point-descriptor>
        <id>mgmt</id>
        <external-connection-point-descriptor>mgmt</external-
connection-point-descriptor>
        <layer-protocol>IPv4</layer-protocol>
        <interface-id xmlns="http://tail-f.com/pkg/tailf-etsi-rel2-
nfvo-esc">0</interface-id>
      </internal-connection-point-descriptor>
      <internal-connection-point-descriptor>
        <id>right</id>
        <external-connection-point-descriptor>right</external-
connection-point-descriptor>
        <layer-protocol>IPv4</layer-protocol>
        <interface-id xmlns="http://tail-f.com/pkg/tailf-etsi-rel2-
nfvo-esc">2</interface-id>
      </internal-connection-point-descriptor>
      <virtual-compute-descriptor>vcd</virtual-compute-descriptor>
      <virtual-storage-descriptor>root</virtual-storage-descriptor>
      <software-image-descriptor>
        <container-format>bare</container-format>
        <disk-format>qcow2</disk-format>
        <image>http://10.0.0.102:8080/images/csr1000v-
universalk9.16.09.05.qcow2</image>
        <additional-setting xmlns="http://tail-f.com/pkg/tailf-etsi-
rel2-nfvo-esc">
          <id>disk_bus</id>
          <value>virtio</value>
        </additional-setting>
        <additional-setting xmlns="http://tail-f.com/pkg/tailf-etsi-
rel2-nfvo-esc">
          <id>e1000_net</id>
          <value>false</value>
        </additional-setting>
        <additional-setting xmlns="http://tail-f.com/pkg/tailf-etsi-
```

```
rel2-nfvo-esc">
          <id>serial_console</id>
          <value>true</value>
        </additional-setting>
        <additional-setting xmlns="http://tail-f.com/pkg/tailf-etsi-
rel2-nfvo-esc">
          <id>virtio_net</id>
          <value>false</value>
        </additional-setting>
      </software-image-descriptor>
      <device-type xmlns="http://tail-f.com/pkg/tailf-etsi-rel2-nfvo-
esc">
        <cli>
          <ned-id>cisco-ios-cli-6.54</ned-id>
        </cli>
      </device-type>
      <day0 xmlns="http://tail-f.com/pkg/tailf-etsi-rel2-nfvo-esc">
        <destination>csr_config.txt</destination>
        <mandatory/>
      </day0>
    </vdu>
    <virtual-compute-descriptor>
      <id>vcd</id>
      <virtual-memory>
        <virtual-memory-size>4.0</virtual-memory-size>
      </virtual-memory>
      <virtual-cpu>
        <number-of-virtual-cpus>1</number-of-virtual-cpus>
      </virtual-cpu>
    </virtual-compute-descriptor>
    <virtual-storage-descriptor>
      <id>root</id>
      <type-of-storage>root</type-of-storage>
      <size-of-storage>8</size-of-storage>
    </virtual-storage-descriptor>
    <external-connection-point-descriptor>
      <id>left</id>
      <layer-protocol>IPv4</layer-protocol>
    </external-connection-point-descriptor>
    <external-connection-point-descriptor>
      <id>mgmt</id>
      <layer-protocol>IPv4</layer-protocol>
      <management xmlns="http://tail-f.com/pkg/tailf-etsi-rel2-nfvo-
esc"/>
    </external-connection-point-descriptor>
    <external-connection-point-descriptor>
      <id>right</id>
      <layer-protocol>IPv4</layer-protocol>
    </external-connection-point-descriptor>
    <deployment-flavor>
      <id>csrv.basic</id>
      <vdu-profile>
        <vdu>CSR</vdu>
        <min-number-of-instances>1</min-number-of-instances>
        <max-number-of-instances>10</max-number-of-instances>
      </vdu-profile>
      <instantiation-level>
        <id>basic</id>
        <vdu-level>
          <vdu>CSR</vdu>
          <number-of-instances>1</number-of-instances>
        </vdu-level>
      </instantiation-level>
      <instantiation-level>
        <id>small</id>
        <vdu-level>
          <vdu>CSR</vdu>
          <number-of-instances>1</number-of-instances>
        </vdu-level>
      </instantiation-level>
      <default-instantiation-level>basic</default-instantiation-level>
    </deployment-flavor>
  </vnfd>
  </nfvo>
</config>
student@student-vm:~/nso300$
```

Step 2

Now open and study the **ns_catalog**, which defines the VNFs that you need for your service. It can be found here: **~/nso300/extra-files/cdb_config/ns_catalog.xml**.

```
student@student-vm:~/nso300$ cat extra-files/cdb_config/ns_catalog.xml
<config xmlns="http://tail-f.com/ns/config/1.0">
  <nfvo xmlns="http://tail-f.com/pkg/tailf-etsi-rel2-nfvo">
  <nsd>
    <id>dmz</id>
    <version>1.0</version>
    <vnfd>
      <vnfd>ASAv</vnfd>
    </vnfd>
    <vnfd>
      <vnfd>CSR1kv</vnfd>
    </vnfd>
    <service-access-point-descriptor>
      <id>inside</id>
    </service-access-point-descriptor>
    <service-access-point-descriptor>
      <id>mgmt</id>
    </service-access-point-descriptor>
    <service-access-point-descriptor>
      <id>outside</id>
    </service-access-point-descriptor>
    <deployment-flavor>
      <id>dmz-flavor</id>
      <vnf-profile>
        <id>ASAv</id>
        <vnfd>ASAv</vnfd>
        <flavor>asav.basic</flavor>
        <instantiation-level>small</instantiation-level>
        <min-number-of-instances>1</min-number-of-instances>
        <max-number-of-instances>2</max-number-of-instances>
        <local-affinity-or-anti-affinity-rule>
          <affinity-type>affinity</affinity-type>
          <affinity-scope>nfvi-pop</affinity-scope>
        </local-affinity-or-anti-affinity-rule>
        <affinity-or-anti-affinity-group>
          <id>aff-group-1</id>
        </affinity-or-anti-affinity-group>
        <affinity-or-anti-affinity-group>
          <id>aff-group-2</id>
        </affinity-or-anti-affinity-group>
        <sapd-connectivity>
          <sapd>inside</sapd>
          <cp>cp-inside</cp>
        </sapd-connectivity>
        <sapd-connectivity>
          <sapd>mgmt</sapd>
          <cp>cp-mgmt</cp>
        </sapd-connectivity>
        <sapd-connectivity>
          <sapd>outside</sapd>
          <cp>cp-outside</cp>
        </sapd-connectivity>
      </vnf-profile>
      <vnf-profile>
        <id>CSR1kv</id>
        <vnfd>CSR1kv</vnfd>
        <flavor>csrv.basic</flavor>
        <instantiation-level>basic</instantiation-level>
        <min-number-of-instances>1</min-number-of-instances>
        <max-number-of-instances>1</max-number-of-instances>
        <local-affinity-or-anti-affinity-rule>
          <affinity-type>affinity</affinity-type>
          <affinity-scope>nfvi-pop</affinity-scope>
        </local-affinity-or-anti-affinity-rule>
        <local-affinity-or-anti-affinity-rule>
          <affinity-type>anti-affinity</affinity-type>
          <affinity-scope>nfvi-node</affinity-scope>
        </local-affinity-or-anti-affinity-rule>
        <affinity-or-anti-affinity-group>
          <id>aff-group-1</id>
        </affinity-or-anti-affinity-group>
        <sapd-connectivity>
          <sapd>outside</sapd>
          <cp>left</cp>
        </sapd-connectivity>
        <sapd-connectivity>
          <sapd>inside</sapd>
          <cp>right</cp>
        </sapd-connectivity>
        <sapd-connectivity>
          <sapd>mgmt</sapd>
          <cp>mgmt</cp>
        </sapd-connectivity>
      </vnf-profile>
      <scaling-aspect>
        <aspect>sa1</aspect>
        <scaling-level>small</scaling-level>
      </scaling-aspect>
      <affinity-or-anti-affinity-group>
        <id>aff-group-1</id>
```

```
        <affinity-type>affinity</affinity-type>
        <affinity-scope>nfvi-pop</affinity-scope>
      </affinity-or-anti-affinity-group>
      <affinity-or-anti-affinity-group>
        <id>aff-group-2</id>
        <affinity-type>anti-affinity</affinity-type>
        <affinity-scope>nfvi-node</affinity-scope>
      </affinity-or-anti-affinity-group>
      <instantiation-level>
        <id>small</id>
        <description>Small instance</description>
        <vnf-to-level-mapping>
          <vnf-profile>ASAv</vnf-profile>
          <number-of-instances>1</number-of-instances>
        </vnf-to-level-mapping>
        <vnf-to-level-mapping>
          <vnf-profile>CSR1kv</vnf-profile>
          <number-of-instances>1</number-of-instances>
        </vnf-to-level-mapping>
      </instantiation-level>
      <dependency>
        <id>dep1</id>
        <primary-vnf-profile>CSR1kv</primary-vnf-profile>
        <secondary-vnf-profile>ASAv</secondary-vnf-profile>
      </dependency>
    </deployment-flavor>
  </nsd>
  </nfvo>
</config>
student@student-vm:~/nso300$
```

Step 3

Display the running Docker containers with the **docker ps -a** command. The NSO container exposes the 8080 port (on which the file server is running), on a random host's port. Note down that port number and the ID of the container.

```
student@student-vm:~$ docker ps -a
CONTAINER ID        IMAGE
COMMAND             CREATED             STATUS
PORTS
NAMES
ccb73064494d        nso300.gitlab.local/nso300/nso:5.3.2-student   "/
run-nso.sh"       About a minute ago   Up About a minute (healthy)
0.0.0.0:32774->22/tcp, 0.0.0.0:32773->80/tcp, 0.0.0.0:32772->443/tcp,
0.0.0.0:32771->830/tcp, 0.0.0.0:32770->4334/tcp, 0.0.0.0:32769->5678/
tcp, 0.0.0.0:32768->8080/tcp   testenv-nso300-5.3.2-student-nso
student@student-vm:~$
```

Step 4

Open and edit the **~/nso300/extra-files/cdb_config/vnf_catalog.xml** file. Change the port within the URL for the software image location to the location that you observed in the previous step.

```
student@student-vm:~/nso300$ vim extra-files/cdb_config/vnf_catalog.xml

< ... Output Omitted ... >

     <software-image-descriptor>
        <container-format>bare</container-format>
        <disk-format>qcow2</disk-format>
        <image>http://10.0.0.102:32768/images/asav992.qcow2</image>
        <additional-setting xmlns="http://tail-f.com/pkg/tailf-etsi-
rel2-nfvo-esc">
          <id>disk_bus</id>
          <value>ide</value>
        </additional-setting>
        <additional-setting xmlns="http://tail-f.com/pkg/tailf-etsi-
rel2-nfvo-esc">
          <id>e1000_net</id>
          <value>true</value>
        </additional-setting>
      </software-image-descriptor>

< ... Output Omitted ... >

      <software-image-descriptor>
        <container-format>bare</container-format>
        <disk-format>qcow2</disk-format>
        <image>http://10.0.0.102:32768/images/csr1000v-
universalk9.16.09.05.qcow2</image>
        <additional-setting xmlns="http://tail-f.com/pkg/tailf-etsi-
rel2-nfvo-esc">
          <id>disk_bus</id>
          <value>virtio</value>
        </additional-setting>
        <additional-setting xmlns="http://tail-f.com/pkg/tailf-etsi-
rel2-nfvo-esc">
          <id>e1000_net</id>
          <value>false</value>
        </additional-setting>
        <additional-setting xmlns="http://tail-f.com/pkg/tailf-etsi-
rel2-nfvo-esc">
          <id>serial_console</id>
          <value>true</value>
        </additional-setting>
        <additional-setting xmlns="http://tail-f.com/pkg/tailf-etsi-
rel2-nfvo-esc">
          <id>virtio_net</id>
          <value>false</value>
        </additional-setting>
      </software-image-descriptor>

< ... Output Omitted ... >
```

Step 5

Save the file and exit the file editor.

Step 6

Copy the modified catalog to the **/cdb_config** folder within the NSO container with the **docker cp** command. Use the appropriate container ID, since they are randomly generated and will be different in your case.

```
student@student-vm:~/nso300$ docker cp extra-files/cdb_config/
vnf_catalog.xml ccb73064494d:/cdb_config/
student@student-vm:~/nso300$
```

Step 7

Enter the NSO containers Linux shell with the **make testenv-shell** command.

```
student@student-vm:~/nso300$ make testenv-shell
docker exec -it testenv-nso300-5.3.2-student-nso bash -l
root@ccb73064494d:/#
```

Step 8

First, load the **vnf-catalog.xml** and then the the ns-catalog.xml into NSO with the **load_merge -l -m** command from NSO CLI. The XML files are located in the **/ cdb_config** folder. No errors should be present when executing the commands.

```
root@ccb73064494d:/# ncs_load -l -m cdb_config/vnf_catalog.xml
root@ccb73064494d:/# ncs_load -l -m cdb_config/ns_catalog.xml
root@ccb73064494d:/# exit
logout
student@student-vm:~/nso300$
```

Step 9

Enter the NSO CLI, switch the CLI and enter the configuration mode.

```
student@student-vm:~/nso300$ make testenv-cli
docker exec -it testenv-nso300-5.3.2-student-nso bash -lc 'ncs_cli -u
admin'

admin connected from 127.0.0.1 using console on ccb73064494d
admin@ncs> switch cli
admin@ncs# config
Entering configuration mode terminal
admin@ncs(config)#
```

Step 10

Create flavors **csrv.basic** and **asav.basic** for CSR and ASA for VDUs that are part of loaded VNFD on OpenStack through the NSO CLI.

```
admin@ncs(config)# nfvo onboarding esc flavor csrv.basic vnfd CSR1kv
```

```
vdu CSR esc-device esc0
admin@ncs(config-esc-device-esc0)# top
admin@ncs(config)# nfvo onboarding esc flavor asav.basic vnfd ASAv vdu
ASA esc-device esc0
admin@ncs(config-esc-device-esc0)# commit
Commit complete.
admin@ncs(config-esc-device-esc0)# top
admin@ncs(config)#
```

Step 11

Onboard images for ASA and CSR to OpenStack.

```

admin@ncs(config)# nfvo onboarding esc image asav-9_9_2 vnfd ASAv vdu
ASA esc-device esc0
admin@ncs(config-esc-device-esc0)# top
admin@ncs(config)# nfvo onboarding esc image csrv-9_16_9 vnfd CSR1kv
vdu CSR esc-device esc0
admin@ncs(config-esc-device-esc0)#
admin@ncs(config-esc-device-esc0)# top
admin@ncs(config)# commit
Commit complete.
admin@ncs(config)# exit
admin@ncs# exit
student@student-vm:~/nso300$
```

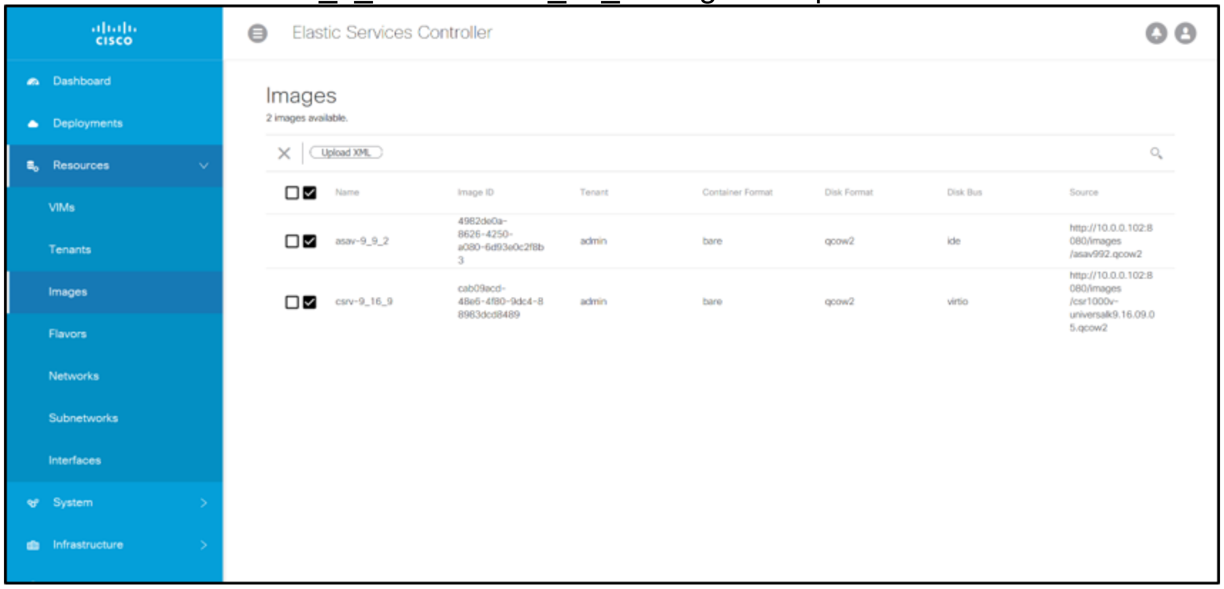
Step 12

Open a web browser. Log in to the ESC Web Portal, using the URL https://10.0.0.104 and the credentials **admin/admin**. Do not close the terminal since you will continue using it soon.

Step 13

Navigate to the *Images* page, using the navigation pane on the left side. Images are located under *Resources* on the navigation pane.

Make sure that *asav-9_9_2* and *csrv-9_16_9* images are present on ESC.



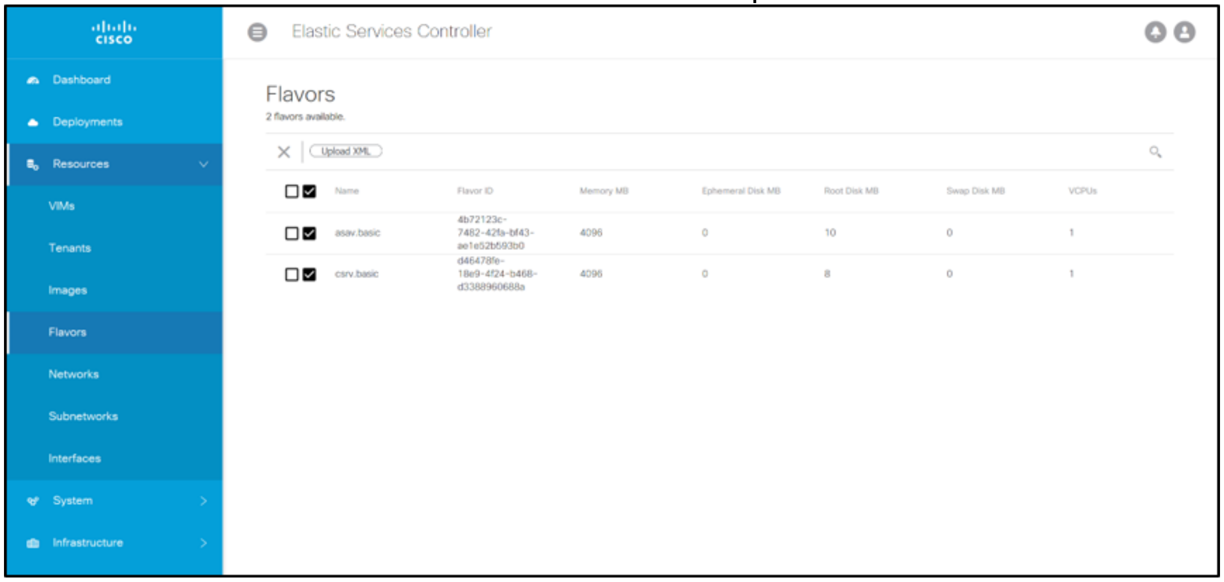
Images
2 images available.

Name	Image ID	Tenant	Container Format	Disk Format	Disk Bus	Source
asav-9_9_2	4982de0a-8626-4250-a080-6d93e0c2f8b3	admin	bare	qcow2	ide	http://10.0.0.102:8080/images/asav992.qcow2
csrv-9_16_9	cab09acd-48e6-4f80-9dc4-88983dcd8489	admin	bare	qcow2	virtio	http://10.0.0.102:8080/images/csr1000v-universalk9.16.09.05.qcow2

Step 14

Navigate to the *Flavors* page, which is also located under *Resources* on the navigation pane.

Make sure that *asav.basic* and *csrv.basic* flavors are present on ESC.



Flavors

2 flavors available.

Name	Flavor ID	Memory MB	Ephemeral Disk MB	Root Disk MB	Swap Disk MB	VCPUs
asav.basic	4b72123c-7482-42fa-bf43-ae1e52b593b0	4096	0	10	0	1
csrv.basic	d46478fe-18e9-4f24-b468-d3388960688a	4096	0	8	0	1

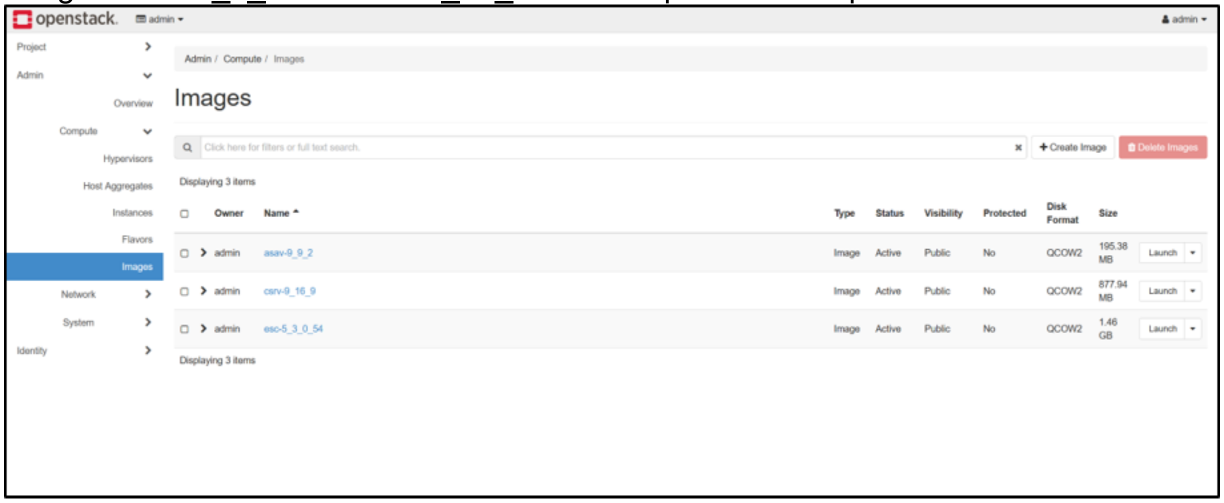
Step 15

Open a new browser tab. Log in to the OpenStack Web Portal, using the URL https://10.0.0.103 and the credentials **admin/admin**.

Step 16

Navigate to the *Images* page, using the navigation pane on the left side. Images are located under *Admin* > *Compute* on the navigation pane. Verify that the image onboarding process has begun (or that it is active already—look at the Status column).

Images *asav-9_9_2* and *csrv-9_16_9* are also present on OpenStack.



Images

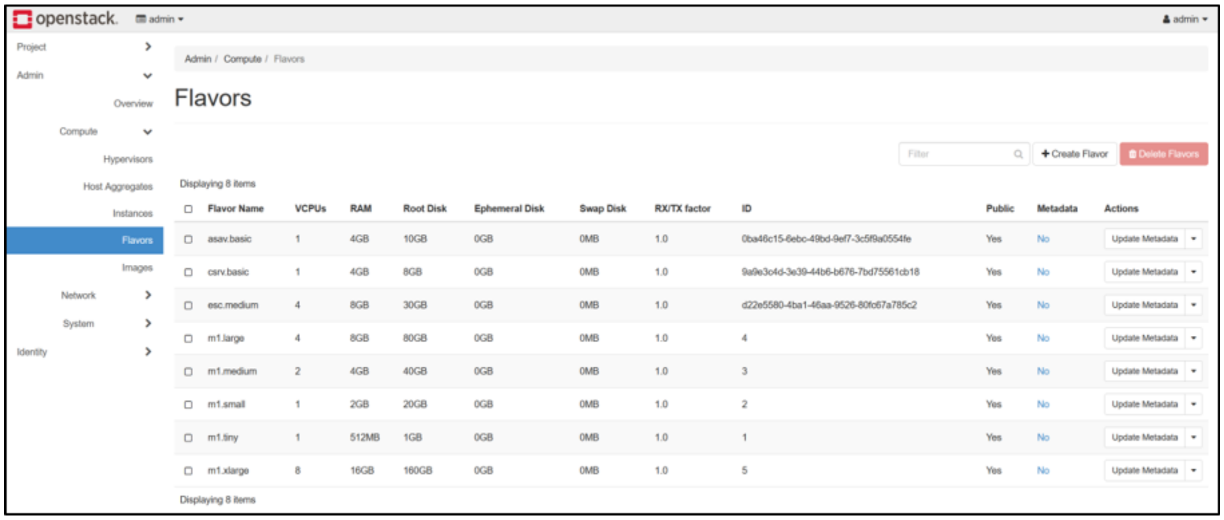
Displaying 3 items

Owner	Name	Type	Status	Visibility	Protected	Disk Format	Size
admin	asav-9_9_2	Image	Active	Public	No	QCOW2	195.38 MB
admin	csrv-9_16_9	Image	Active	Public	No	QCOW2	877.94 MB
admin	esc-5_3_0_54	Image	Active	Public	No	QCOW2	1.46 GB

Displaying 3 items

Step 17

Now navigate to the *Flavors* tab, also located under *Admin* > *Compute* on the navigation pane. The two flavors are present here too.



Flavors

Displaying 8 items

Flavor Name	VCPUs	RAM	Root Disk	Ephemeral Disk	Swap Disk	RX/TX factor	ID	Public	Metadata	Actions
asav.basic	1	4GB	10GB	0GB	0MB	1.0	0ba46c15-6ebc-49bd-9ef7-3c5f9a0554fe	Yes	No	Update Metadata
csrv.basic	1	4GB	8GB	0GB	0MB	1.0	9a9e3c4d-3e39-44b6-b676-7bd75561cb18	Yes	No	Update Metadata
esc.medium	4	8GB	30GB	0GB	0MB	1.0	d22e5580-4ba1-46aa-9526-80fc67a785c2	Yes	No	Update Metadata
m1.large	4	8GB	80GB	0GB	0MB	1.0	4	Yes	No	Update Metadata
m1.medium	2	4GB	40GB	0GB	0MB	1.0	3	Yes	No	Update Metadata
m1.small	1	2GB	20GB	0GB	0MB	1.0	2	Yes	No	Update Metadata
m1.tiny	1	512MB	1GB	0GB	0MB	1.0	1	Yes	No	Update Metadata
m1.xlarge	8	16GB	160GB	0GB	0MB	1.0	5	Yes	No	Update Metadata

Displaying 8 items

Activity Verification

You have completed this task when you attain these results:

- vCSR and vASA images and flavors are successfully onboarded to ESC.