Cisco Ultra Services Platform v1

Last Updated: 9-MAY-2017

About This Demonstration

This guide for the preconfigured **Cisco® Ultra Services Platform** demonstration includes:

Limitations

The current demonstration setup focuses on the Ultra Gateway Platform. The Ultra Services Framework and the Ultra Policy Platform as additional USP components are not yet in the scope of this demonstration.

Customization Options

The demonstration is based on a fully functional OpenStack environment that allows you to deploy virtual machines (VMs) manually instead of using the automation services. Alternatively, the Heat service is included, which can be used to automate the setup by using Heat templates. The setup can also be used to deploy other USP scenarios, such as CUPs or the Ultra Services Framework, unless this causes a resource limitation. All this is for further study and may be implemented as preconfigured and tested scenarios in a later version of the demo.

Requirements

The table below outlines the requirements for this preconfigured demonstration.

Table 1. Requirements

Required	Optional
• Laptop	• Cisco AnyConnect®

Cisco dCloud recommends mapping the controller host name to the IP address 198.18.134.30 on the user's laptop (in /etc/hosts on a Mac or in C:\Windows\System32\drivers\etc\hosts on a Windows system) when using the OpenStack GUI. On the workstation machine shown in the dCloud topology, this and other configurations were already prepared.

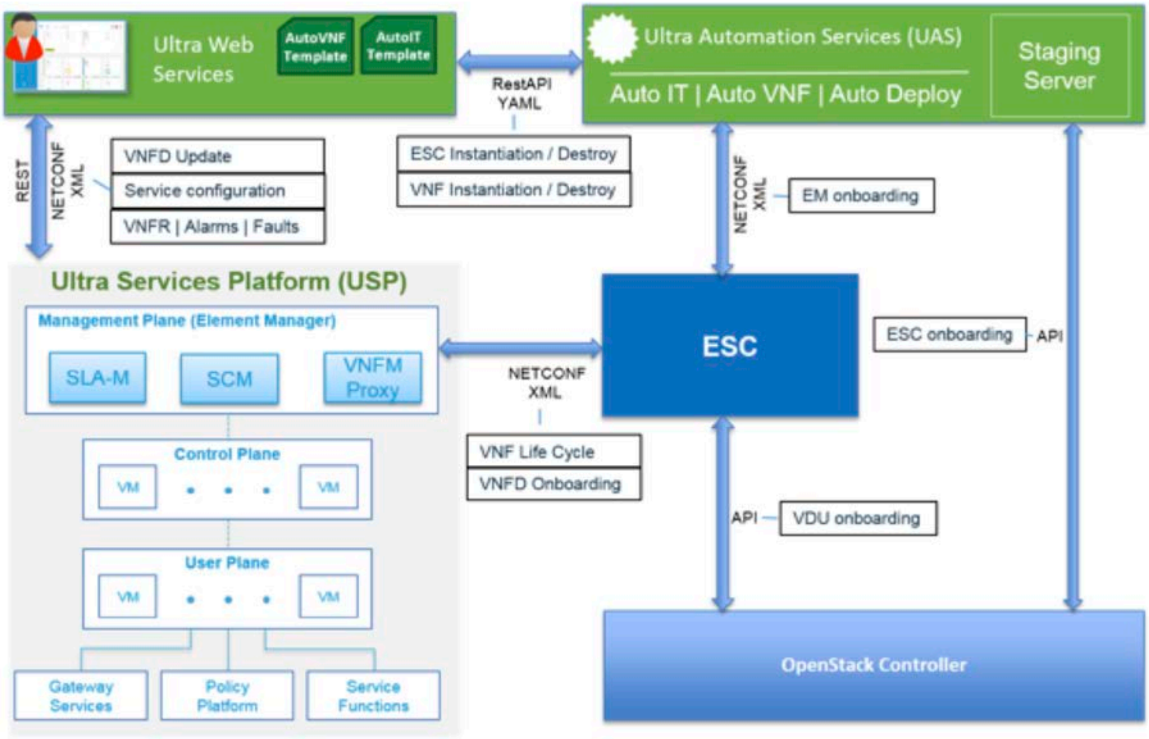
About This Solution

This dCloud demonstration provides access to a fully functional Ultra Services Platform, including the Ultra Gateway Platform (UGP) running as a virtual network function (VNF), the VNF Element Manager (VNF-EM), the Elastic Services Controller (ESC) providing the VNF-M functionality, and the full automation suite to launch all of this automatically.

The demonstration is preconfigured with a VM running the staging server, three VMs for the Ultra Automation Services, and an already deployed VM running ESC as VNF-M. A bootstrap script will also initially start the deployment of the VNF-EM and UGP VNF so that the system is fully operational after it has been launched. The demonstration can then be used to monitor all of the running VMs and related interfaces and UIs.

The demo can also be used to run the same set of end-to-end Virtualized Packet Core (VPC) scenarios that already exist in the Cisco Virtualized Packet Core v1.2 dCloud demo. VPC v1.2 runs all VPC services on a single instance of StarOS in a single VM (called VPC-SI), whereas this demonstration deploys the distributed instance for the SAEGW and uses a separate VPC-SI implementation for the MME/SGSN.

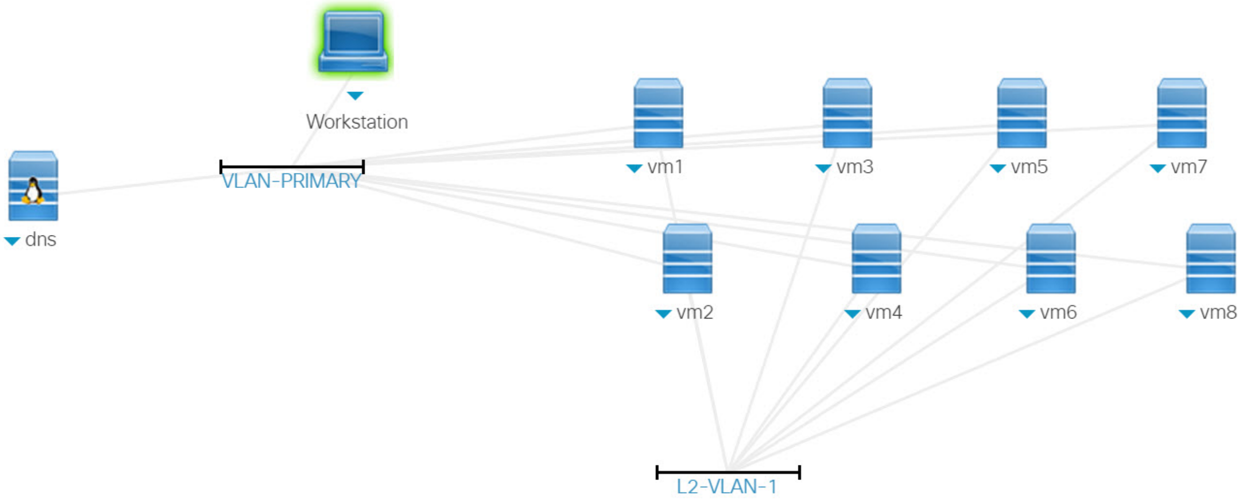
Figure 1. USP Solution Overview



Topology

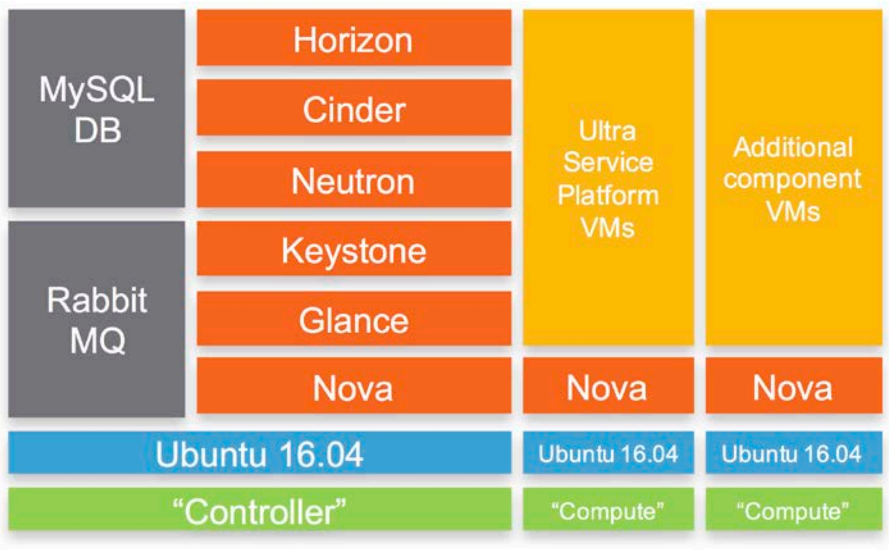
The **Topology** menu of your active session, which is shown in the figure below, lists the eight VMs used in this demonstration, plus the usual Workstation for Remote Desktop Access.

Figure 2. dCloud Topology



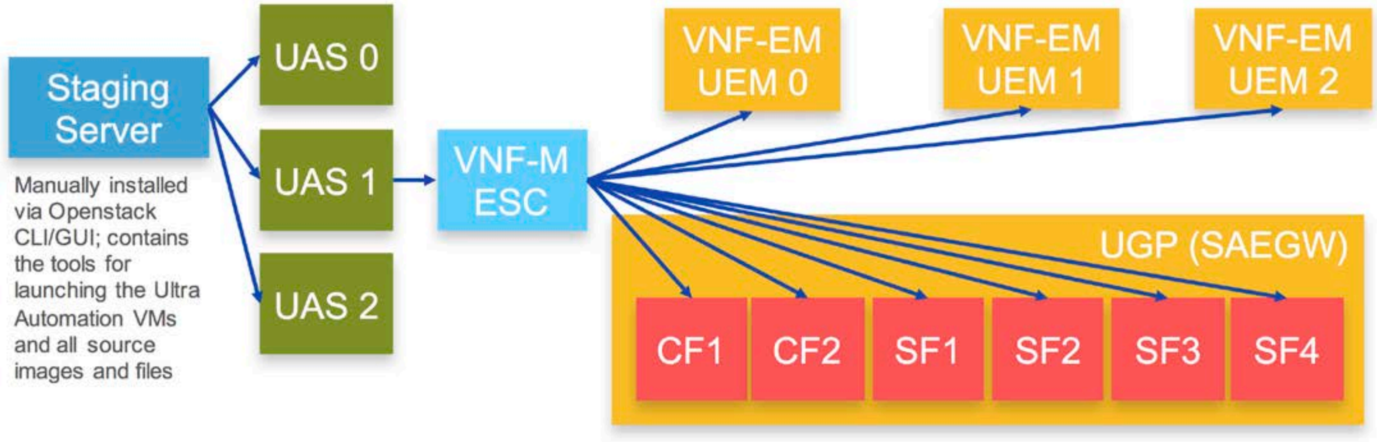
These VMs run a standard Ubuntu 16.04 TLS Linux distribution where all OpenStack modules were manually installed using the instructions at http://docs.openstack.org/mitaka/install-guide-ubuntu/. VM1 hosts the controller with the main important OpenStack services while VM2 through VM8 are used as computes nodes (compute1 to compute7), as shown in Figure 3.

Figure 3. OpenStack Components



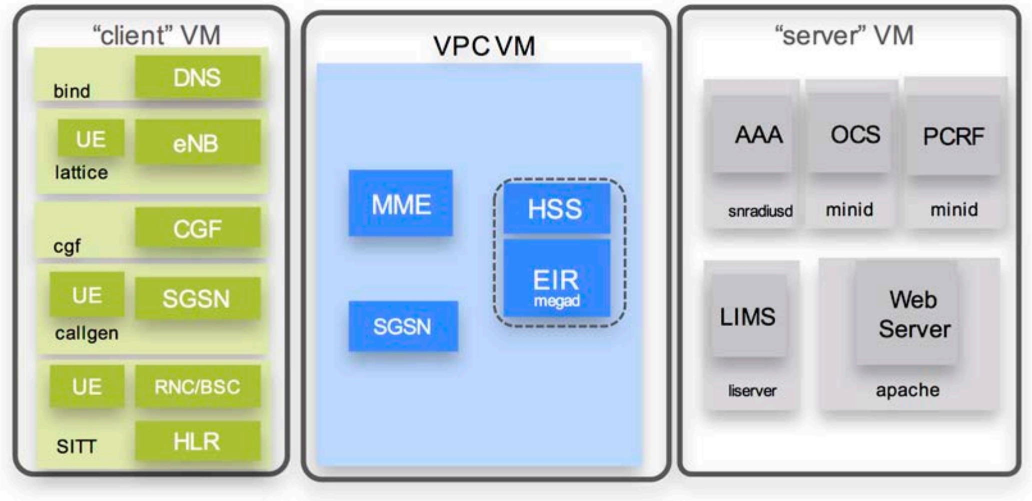
The Ultra Service Platform VMs shown in Figure 3 include not only the actual VMs for the Ultra Gateway Platform, but also the related VMs for the Element Manager, the VNF-M (using the Elastic Services Controller), and the UAS machines for the Ultra Automation Services.

Figure 4. Ultra Services Platform Components


Staging Server
Manually installed via Openstack CLI/GUI; contains the tools for launching the Ultra Automation VMs and all source images and files
UAS 0
UAS 1
UAS 2
VNF-M ESC
VNF-EM UEM 0
VNF-EM UEM 1
VNF-EM UEM 2
UGP (SAEGW)
CF1
CF2
SF1
SF2
SF3
SF4

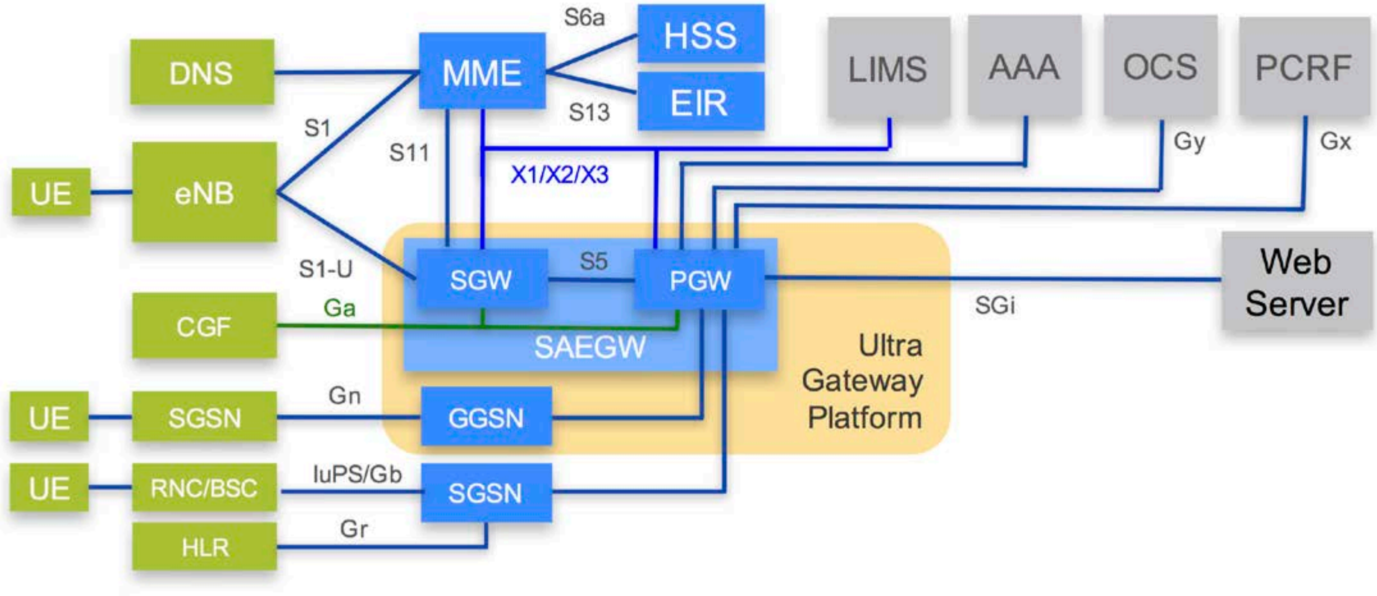
Additional VMs were installed to offer the same end-to-end VPC demonstration capabilities like the Cisco Virtualized Packet Core v1.2 dCloud demonstration, which uses only a VPC-SI machine for the VPC system. The corresponding VPC VM from the VPC demonstration is used here only as an MME/SGSN; the gateway functionality is provided by the UGP components shown in Figure 4.

Figure 5. Additional Components



The end-to-end application layer topology which is provided by this demo is then shown in Figure 6.

Figure 6. End-to-End Application Layer Topology



The tables below summarize all the equipment and access details for the demonstration components.

Table 2. OpenStack hosts and SSH access details

Name	Description	IP Address	Username	Password
Controller	OpenStack controller node	198.18.134.30	root	C1sco12345
Compute1	OpenStack compute node	198.18.134.31	root	C1sco12345
Compute2	OpenStack compute node	198.18.134.32	root	C1sco12345
Compute3	OpenStack compute node	198.18.134.33	root	C1sco12345
Compute4	OpenStack compute node	198.18.134.34	root	C1sco12345
Compute5	OpenStack compute node	198.18.134.35	root	C1sco12345
Compute6	OpenStack compute node	198.18.134.36	root	C1sco12345
Compute7	OpenStack compute node	198.18.134.37	root	C1sco12345

Table 3. Virtual machines and SSH access details

Name	Description	IP Address	Username	Password
Staging Server	Initial staging server to start installation	198.18.135.8	ubuntu	cisco123
UAS	Ultra Automation Services (floating IP for active server)	198.18.135.12	ubuntu	cisco123
ESC	Elastic Services Controller	198.18.135.15	admin	cisco123
Client	Simulators for RAN access (UE, eNodeB etc.)	198.18.135.19	dcloud	cisco
Server	Simulators for PCRF, OCS, AAA, LIMS, Web server	198.18.135.13	dcloud	cisco
VPC-SI	MME/SGSN configured on StarOS single instance	198.18.135.14	admin	cisco
UGP	SAEGW configured on StarOS distributed instance	198.18.135.6	admin	Cisco123

Table 4. URLs and access details for the GUIs

Name	Description	URL	Username	Password
UWS	Ultra Web Services	http://198.18.135.12:8008 -> https://198.18.135.12:8888	admin	cisco123
ESC	Elastic Services Controller	http://198.18.135.15:9000 -> https://198.18.135.15:9001	admin	cisco123
Horizon	OpenStack GUI access	http://controller/horizon (Controller should be linked to 198.18.134.30 in /etc/hosts)	core	core123

Get Started

BEFORE PRESENTING

Cisco dCloud strongly recommends that you perform the tasks in this document with an active session before presenting in front of a live audience. This will allow you to become familiar with the structure of the document and content.

It may be necessary to schedule a new session after following this guide in order to reset the environment to its original configuration.

PREPARATION IS KEY TO A SUCCESSFUL PRESENTATION.

Follow the steps to schedule a session of the content and configure your presentation environment.

1. Initiate your dCloud session. [Show Me How]

NOTE: It may take up to 10 minutes for your session to become active. Also allow an additional 15 minutes once the demo is running to complete the initial deployment scenario which is running automatically

2. For best performance, connect to the workstation with **Cisco AnyConnect VPN** [Show Me How] and the **local RDP client on your laptop** [Show Me How], or connect directly using SSH or browser to the addresses shown in the tables on the previous pages.

   - Workstation 1: **198.18.133.252**, Username: **administrator**, Password: **C1sco12345**

Scenario 1. Explore the Demonstration Environment

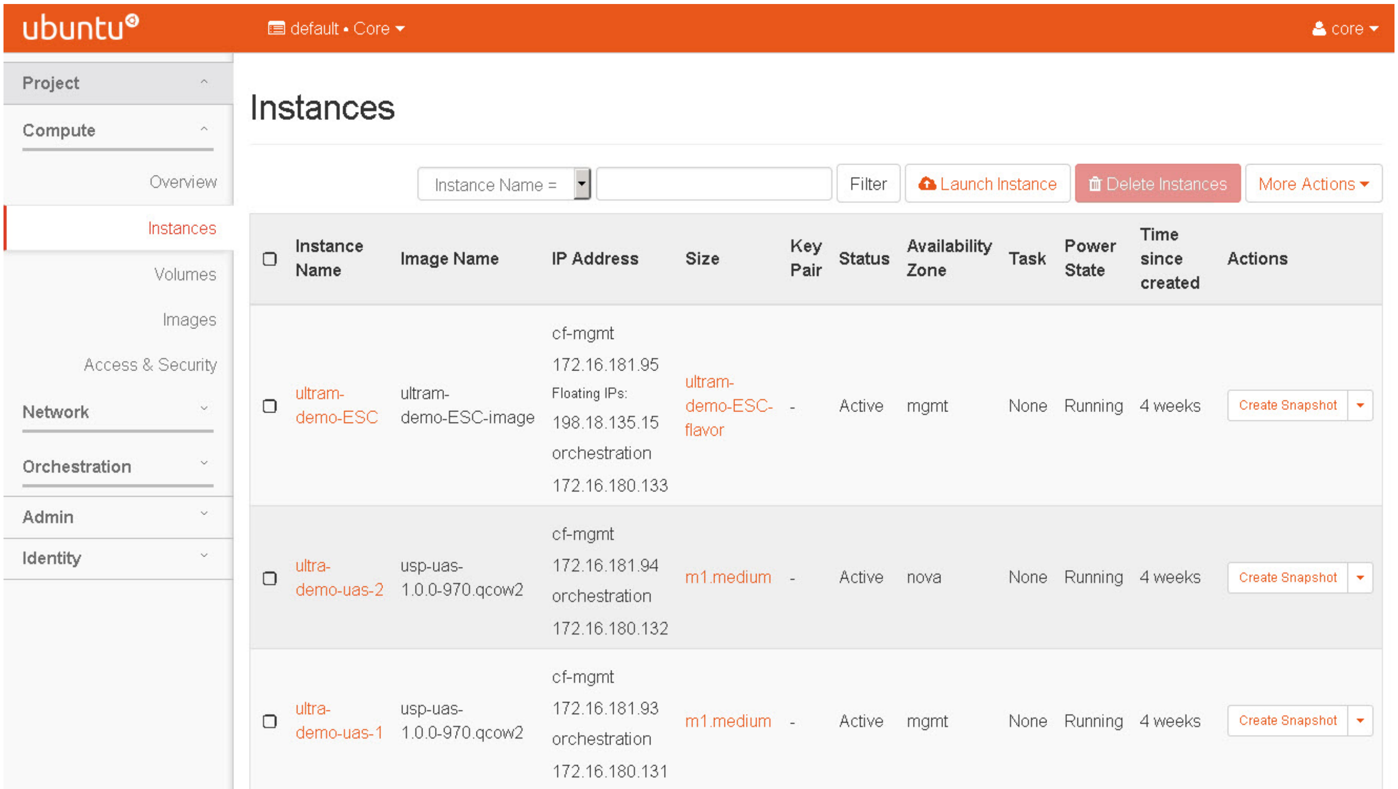
This scenario explores the various UIs and interfaces available in the demonstration environment. No preparation is required to run this scenario after the demonstration has been launched.

Steps

1. Using a web browser, navigate to http://controller/horizon to access the **OpenStack Horizon** UI.
2. Log in with user name: **core**, password: **core123**.

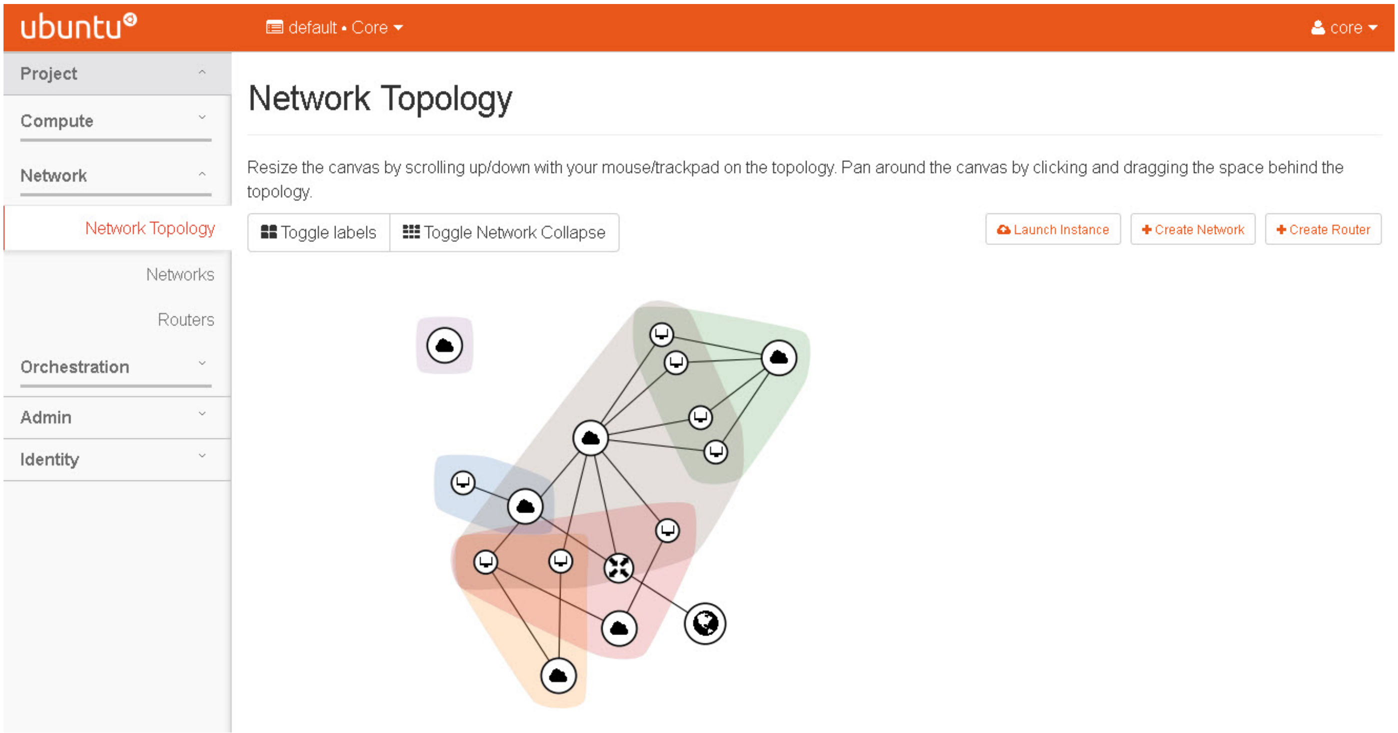
NOTE: Do not change any aspect of the UI; the UI is required to be used as-is in the scenarios that follow.

3. Click **Project > Compute > Instances** to verify the currently running instances of the USP platform that have been deployed automatically.



□	Instance Name	Image Name	IP Address	Size	Key Pair	Status	Availability Zone	Task	Power State	Time since created	Actions
□	ultram-demo-ESC	ultram-demo-ESC-image	cf-mgmt 172.16.181.95 Floating IPs: 198.18.135.15 orchestration 172.16.180.133	ultram-demo-ESC-flavor	-	Active	mgmt	None	Running	4 weeks	Create Snapshot
□	ultra-demo-uas-2	usp-uas-1.0.0-970.qcow2	cf-mgmt 172.16.181.94 orchestration 172.16.180.132	m1.medium	-	Active	nova	None	Running	4 weeks	Create Snapshot
□	ultra-demo-uas-1	usp-uas-1.0.0-970.qcow2	cf-mgmt 172.16.181.93 orchestration 172.16.180.131	m1.medium	-	Active	mgmt	None	Running	4 weeks	Create Snapshot

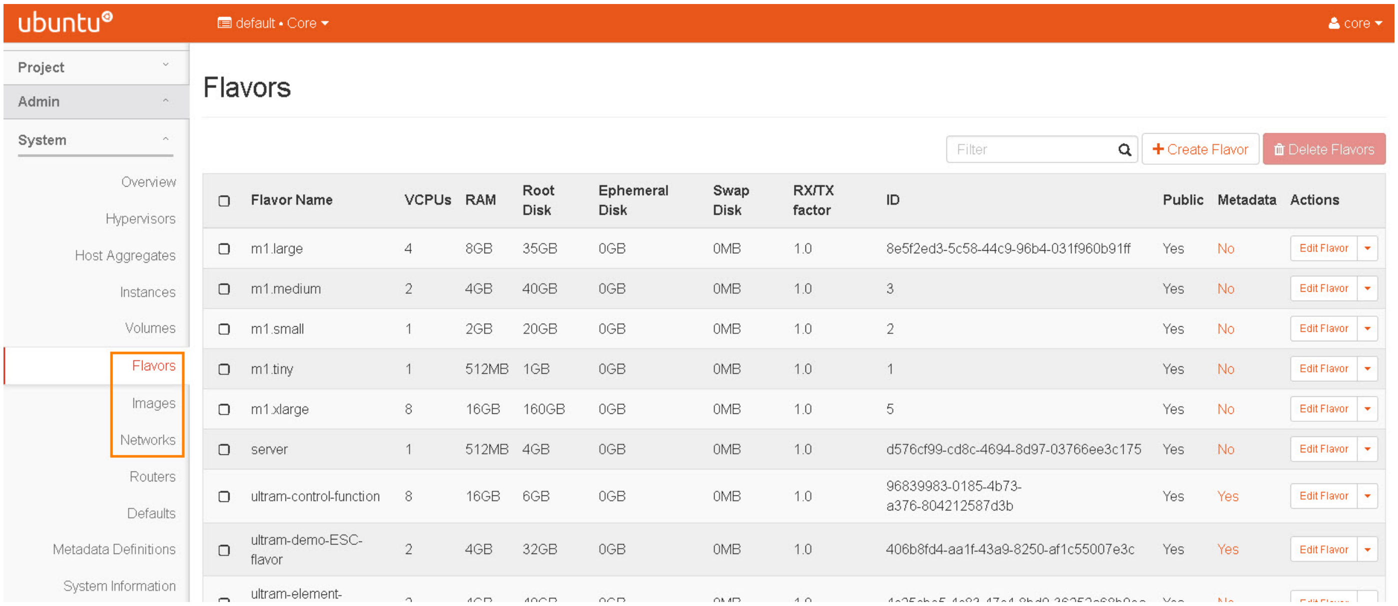
4. Click **Project > Network > Network Topology** to verify the network setup required for orchestration and management and for the end-to-end application functionality.



Network Topology

Resize the canvas by scrolling up/down with your mouse/trackpad on the topology. Pan around the canvas by clicking and dragging the space behind the topology.

5. Click **Admin > System** and explore **Flavors**, **Volumes**, and **Images** that were automatically added to the OpenStack environment through the automation tools.



Flavors

☐	Flavor Name	VCPUs	RAM	Root Disk	Ephemeral Disk	Swap Disk	RX/TX factor	ID	Public	Metadata	Actions
☐	m1.large	4	8GB	35GB	0GB	0MB	1.0	8e5f2ed3-5c58-44c9-96b4-031f960b91ff	Yes	No	Edit Flavor ▾
☐	m1.medium	2	4GB	40GB	0GB	0MB	1.0	3	Yes	No	Edit Flavor ▾
☐	m1.small	1	2GB	20GB	0GB	0MB	1.0	2	Yes	No	Edit Flavor ▾
☐	m1.tiny	1	512MB	1GB	0GB	0MB	1.0	1	Yes	No	Edit Flavor ▾
☐	m1.xlarge	8	16GB	160GB	0GB	0MB	1.0	5	Yes	No	Edit Flavor ▾
☐	server	1	512MB	4GB	0GB	0MB	1.0	d576cf99-cd8c-4694-8d97-03766ee3c175	Yes	No	Edit Flavor ▾
☐	ultram-control-function	8	16GB	6GB	0GB	0MB	1.0	96839983-0185-4b73-a376-804212587d3b	Yes	Yes	Edit Flavor ▾
☐	ultram-demo-ESC-flavor	2	4GB	32GB	0GB	0MB	1.0	406b8fd4-aa1f-43a9-8250-af1c55007e3c	Yes	Yes	Edit Flavor ▾
☐	ultram-element-	2	4GB	40GB	0GB	0MB	1.0	[illegible]	Yes	No	Edit Flavor

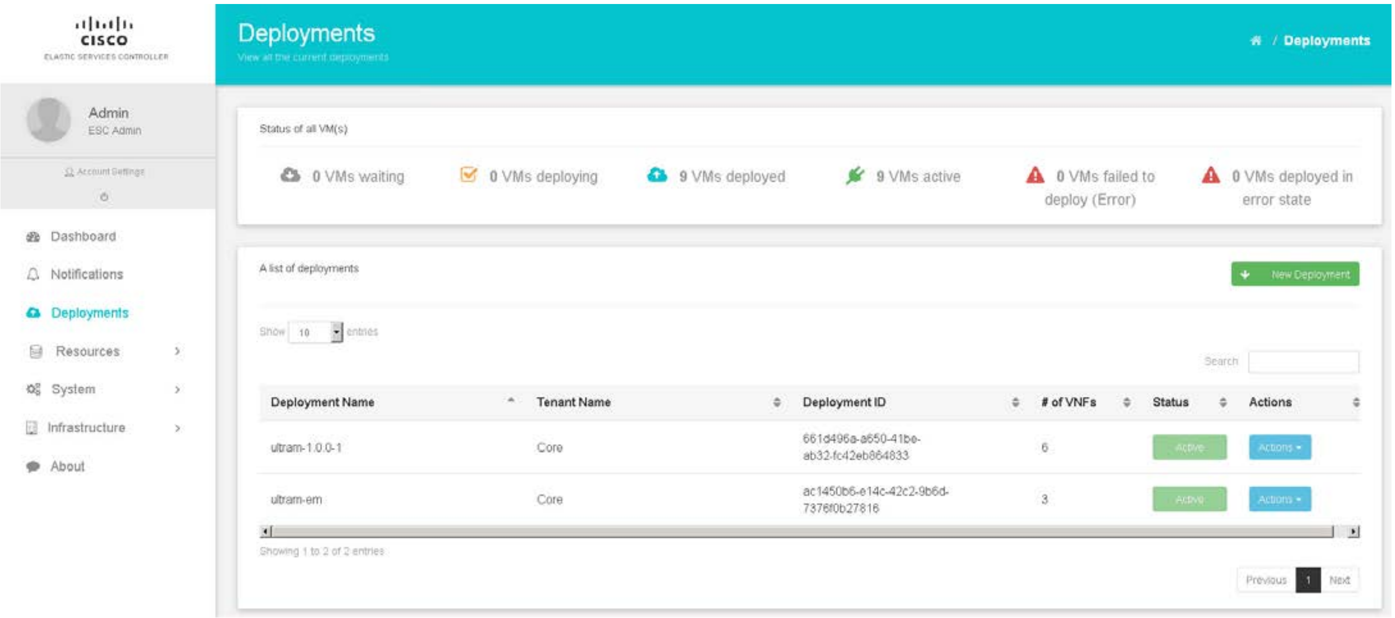
6. Navigate to https://198.18.135.15:9001/ to access the **Elastic Services Controller** UI. If prompted, confirm the security exception.

7. Log in with user name: **admin**, password: **cisco123**. If ESC asks to change the password; simply re-enter **cisco123** for convenience.

NOTE: Do not change any aspect of the UI; the UI is required to be used as-is in the scenarios that follow.

8. Explore the ESC UI. For example, click **Deployments** to verify the deployments sent to ESC.

Figure 7. Deployments shown in ESC GUI



Status of all VM(s)

0 VMs waiting | 0 VMs deploying | 9 VMs deployed | 9 VMs active | 0 VMs failed to deploy (Error) | 0 VMs deployed in error state

A list of deployments

Deployment Name	Tenant Name	Deployment ID	# of VNFs	Status	Actions
ultram-1.0.0-1	Core	661d496a-a650-41be-ab32-fc42eb864833	6	Active	Actions ▾
ultram-em	Core	ac1450b6-e14c-42c2-9b6d-7376f0b27816	3	Active	Actions ▾

Showing 1 to 2 of 2 entries

NOTE: Ignore the error shown when accessing the menu items under Infrastructure. This is a known bug that does not affect the functionality of the demonstration.

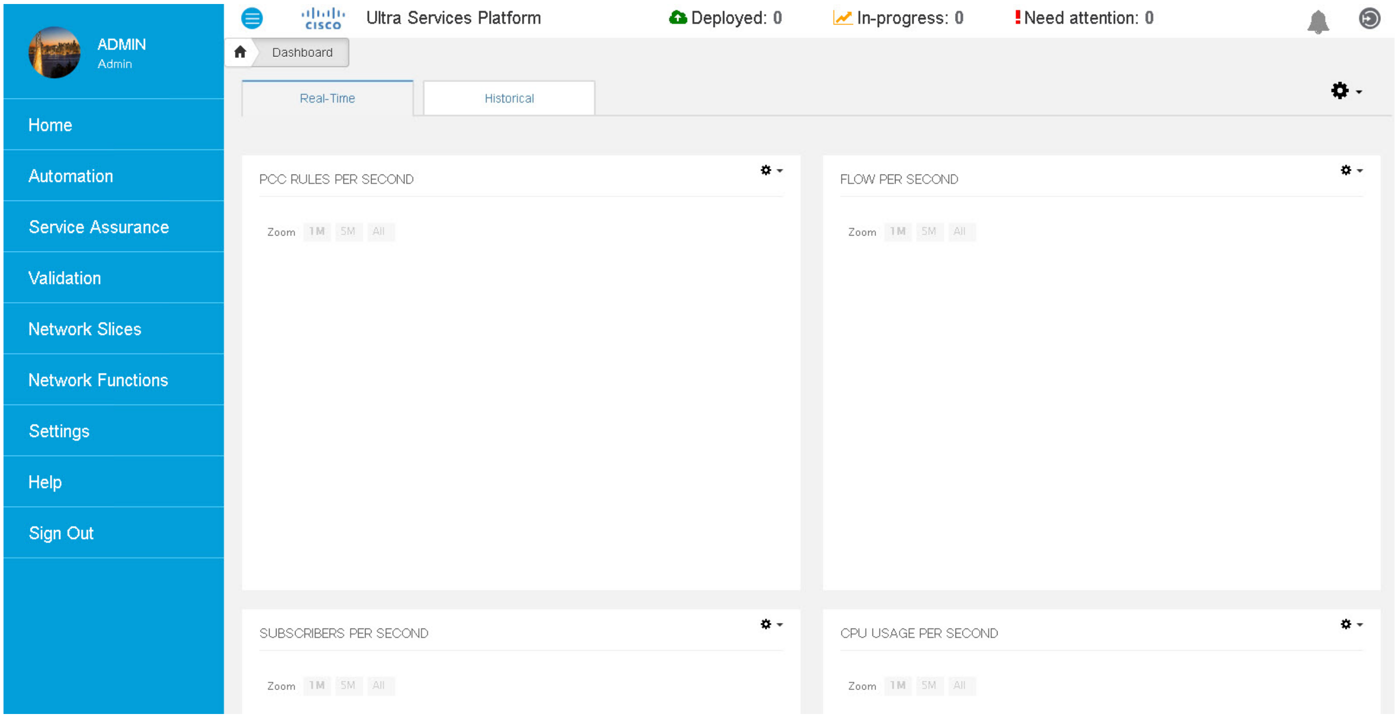
9. Navigate to https://198.18.135.12:8888/ to access the **Ultra Web Services** UI. If prompted, confirm the security exception.

10. Log in with user name: **admin**, password: **cisco123**.

NOTE: Do not change any aspect of the UI; the UI is required to be used as-is in the scenarios that follow.

11. Explore the **Ultra Services Platform** UI.

NOTE: The KPIs shown on the dashboard are not yet supported by the Ultra Gateway Platform and do not show values yet.



Scenario 2. Deploy the VNF

This scenario demonstrates how to deploy the full set of VMs required for the VPC system. The VMs include:

- Two control function VMs
- Four service function VMs
- Three machines for Ultra Element Manager

NOTE: The demonstration is already automatically deployed when launched, therefore, you must first un-deploy the current configuration before using the UI to load XML files for the deployment.

Steps

Un-Deploy VNF

1. From the workstation desktop, double-click the PuTTY icon and SSH to **198.18.135.12** to access Ultra Automation Services (UAS).
2. Click **Yes** in the PuTTY Security Alert.
3. Log in with user name: **ubuntu**, password: **cisco123**.
4. Enter the following commands to open the UAS CLI:

```
ubuntu@ultra-demo-uas-0:~$ source /opt/cisco/usp/uas/confd-6.1/confdrc
ubuntu@ultra-demo-uas-0:~$ confd_cli -u admin -C
```

5. Enter the following commands to deactivate the ultram deployment to un-deploy the VNF:

```
ultra-demo-uas-0#deactivate-deployment deployment-name ultram
transaction-id ad3d2c7c-1492-11e7-95b8-fa163e1bdcaa
```

6. To verify that you successfully deactivated the deployment, open a second terminal window to **198.18.135.12** and enter the following command to view the UAS log:

```
ubuntu@ultra-demo-uas-0:~$ sudo tail -f /var/log/upstart/autovnf.log
2017-03-29 15:16:26,277 - Notify deployment
2017-03-29 15:16:26,293 - Connection to VNFM (esc) at 198.18.135.15
2017-03-29 15:16:26,969 - NETConf Sessions (Transaction/Notifications) estabilished
2017-03-29 15:16:27,007 - NETCONF get-config Request sent, waiting for reply
2017-03-29 15:16:27,463 - NETCONF Transaction success!
2017-03-29 15:16:27,499 - Removing deployment ultram ...
2017-03-29 15:16:27,532 - Destroy VNF for deployment: ultram
2017-03-29 15:16:27,568 - Destroy VNF: ultram-1.0.0-1
2017-03-29 15:16:27,600 - Destroy VNF: ultram-em
2017-03-29 15:16:27,635 - NETCONF edit-config Request sent, waiting for reply
2017-03-29 15:16:28,699 - NETCONF Transaction success!
2017-03-29 15:16:28,727 - Waiting for VNFM to process SERVICE_UNDEPLOYED transaction
2017-03-29 15:17:05,278 - | VM_UNDEPLOYED | ultram-em-1 | SUCCESS | Waiting for: SERVICE_UNDEPLOYED|
2017-03-29 15:17:06,748 - | VM_UNDEPLOYED | s3 | SUCCESS | Waiting for: SERVICE_UNDEPLOYED|
2017-03-29 15:17:07,074 - | VM_UNDEPLOYED | s4 | SUCCESS | Waiting for: SERVICE_UNDEPLOYED|
2017-03-29 15:17:07,228 - | VM_UNDEPLOYED | s5 | SUCCESS | Waiting for: SERVICE_UNDEPLOYED|
2017-03-29 15:17:07,744 - | VM_UNDEPLOYED | s6 | SUCCESS | Waiting for: SERVICE_UNDEPLOYED|
2017-03-29 15:17:08,503 - | VM_UNDEPLOYED | ultram-em-2 | SUCCESS | Waiting for: SERVICE_UNDEPLOYED|
2017-03-29 15:17:09,132 - | VM_UNDEPLOYED | ultram-em-3 | SUCCESS | Waiting for: SERVICE_UNDEPLOYED|
2017-03-29 15:17:09,228 - | SERVICE_UNDEPLOYED | ultram-em | SUCCESS | (1/2)
2017-03-29 15:17:19,615 - | VM_UNDEPLOYED | c2 | SUCCESS | Waiting for: SERVICE_UNDEPLOYED|
2017-03-29 15:19:25,580 - | VM_UNDEPLOYED | c1 | SUCCESS | Waiting for: SERVICE_UNDEPLOYED|
2017-03-29 15:19:53,708 - | SERVICE_UNDEPLOYED | ultram-1.0.0-1 | SUCCESS | (2/2)
2017-03-29 15:19:53,767 - NETCONF transaction completed successfully!
2017-03-29 15:19:53,796 - Deployment ultram removed successfully
2017-03-29 15:19:53,822 - Notify EM Down
2017-03-29 15:19:58,821 - VNF Transaction completed successfully!
2017-03-29 15:19:58,907 - Notify deployment
Transaction Commit
2017-03-29 15:20:18,216 - Transaction Commit
2017-03-29 15:20:18,247 - Notify deployment
2017-03-29 15:20:18,256 - Connection to VNFM (esc) at 198.18.135.15
2017-03-29 15:20:18,989 - NETConf Sessions (Transaction/Notifications) estabilished
2017-03-29 15:20:19,021 - VNF Transaction completed successfully!
2017-03-29 15:20:19,046 - Notify deployment
```

7. From the first PuTTY session, enter **config** mode and enter the following commands to remove the deployment configuration from the UAS database:

```
ultra-demo-uas-0#config
Entering configuration mode terminal
no deployments
no vdu-catalog
no volume-catalog
no network-catalog
no vnfm-instance
commit
Commit complete.
```

NOTE: If you encounter any issues with this configuration sequence, refer to the Troubleshooting section.

8. To verify that you successfully removed the deployment configuration, return to the second terminal window to **198.18.135.12** and enter the following command to view the UAS log:

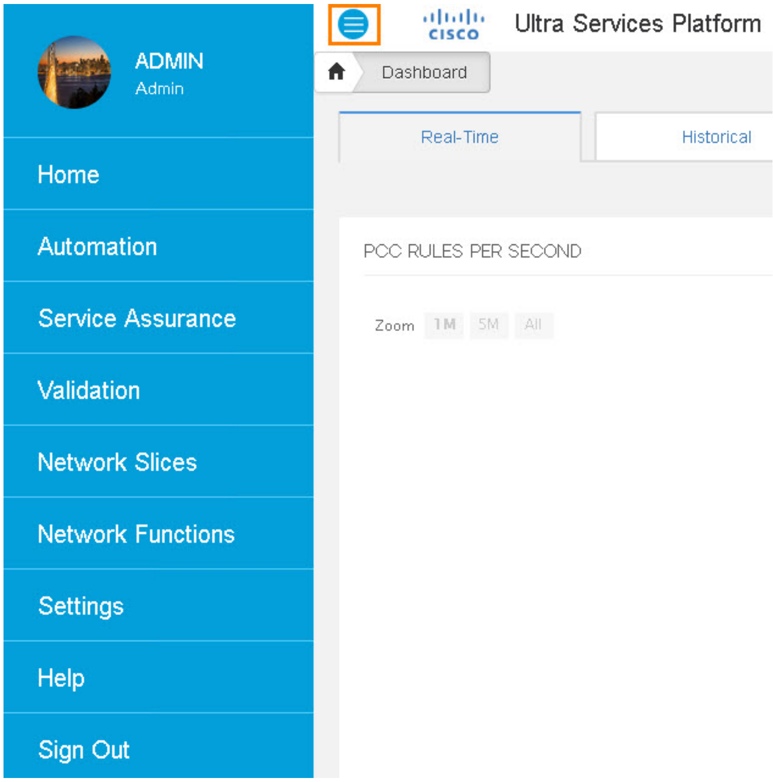
```
ubuntu@ultra-demo-uas-0:~$ sudo tail -f /var/log/upstart/autovnf.log
2017-03-29 15:21:35,306 - Transaction Commit
2017-03-29 15:21:35,324 - Notify deployment
2017-03-29 15:21:35,333 - Connection to VNFM (esc) at 198.18.135.15
2017-03-29 15:21:36,062 - NETConf Sessions (Transaction/Notifications) estabilished
2017-03-29 15:21:36,080 - Get Images
2017-03-29 15:21:36,098 - NETCONF get-config Request sent, waiting for reply
2017-03-29 15:21:36,494 - NETCONF Transaction success!
2017-03-29 15:21:36,512 - Get Flavors List
2017-03-29 15:21:36,533 - Deleting Images ..
2017-03-29 15:21:36,549 - Deleting Images
2017-03-29 15:21:36,578 -   image: ultram-element-manager
2017-03-29 15:21:36,596 - NETCONF edit-config Request sent, waiting for reply
2017-03-29 15:21:38,033 - NETCONF Transaction success!
2017-03-29 15:21:38,051 - Waiting for VNFM to process DELETE_IMAGE transaction
2017-03-29 15:21:55,613 - | DELETE_IMAGE | ultram-element-manager | SUCCESS | (1/1)
2017-03-29 15:21:55,637 - NETCONF transaction completed successfully!
2017-03-29 15:21:55,657 - Deleting Images
2017-03-29 15:21:55,689 -   image: ultram-control-function
2017-03-29 15:21:55,712 - NETCONF edit-config Request sent, waiting for reply
```

```
2017-03-29 15:21:55,802 - NETCONF Transaction success!
2017-03-29 15:21:55,824 - Waiting for VNFM to process DELETE_IMAGE transaction
2017-03-29 15:21:56,773 - | DELETE_IMAGE | ultram-control-function | SUCCESS | (1/1)
2017-03-29 15:21:56,802 - NETCONF transaction completed successfully!
2017-03-29 15:21:56,824 - Deleting Images
2017-03-29 15:21:56,848 -   image: ultram-session-function
2017-03-29 15:21:56,869 - NETCONF edit-config Request sent, waiting for reply
2017-03-29 15:21:57,008 - NETCONF Transaction success!
2017-03-29 15:21:57,037 - Waiting for VNFM to process DELETE_IMAGE transaction
2017-03-29 15:21:57,351 - | DELETE_IMAGE | ultram-session-function | SUCCESS | (1/1)
2017-03-29 15:21:57,381 - NETCONF transaction completed successfully!
2017-03-29 15:21:57,423 - Images deleted successfully.
2017-03-29 15:21:57,449 - Deleting Flavors ..
2017-03-29 15:21:57,476 - Deleting flavors
2017-03-29 15:21:57,492 -   flavor: ultram-element-manager
2017-03-29 15:21:57,515 - NETCONF edit-config Request sent, waiting for reply
2017-03-29 15:21:57,700 - NETCONF Transaction success!
2017-03-29 15:21:57,735 - Waiting for VNFM to process DELETE_FLAVOR transaction
2017-03-29 15:21:57,906 - | DELETE_FLAVOR | ultram-element-manager | SUCCESS | (1/1)
2017-03-29 15:21:57,940 - NETCONF transaction completed successfully!
2017-03-29 15:21:57,965 - Deleting flavors
2017-03-29 15:21:57,983 -   flavor: ultram-control-function
2017-03-29 15:21:58,002 - NETCONF edit-config Request sent, waiting for reply
2017-03-29 15:21:58,244 - NETCONF Transaction success!
2017-03-29 15:21:58,264 - Waiting for VNFM to process DELETE_FLAVOR transaction
2017-03-29 15:21:58,415 - | DELETE_FLAVOR | ultram-control-function | SUCCESS | (1/1)
2017-03-29 15:21:58,435 - NETCONF transaction completed successfully!
2017-03-29 15:21:58,467 - Deleting flavors
2017-03-29 15:21:58,492 -   flavor: ultram-session-function
2017-03-29 15:21:58,512 - NETCONF edit-config Request sent, waiting for reply
2017-03-29 15:21:58,701 - NETCONF Transaction success!
2017-03-29 15:21:58,718 - Waiting for VNFM to process DELETE_FLAVOR transaction
2017-03-29 15:21:58,861 - | DELETE_FLAVOR | ultram-session-function | SUCCESS | (1/1)
2017-03-29 15:21:58,895 - NETCONF transaction completed successfully!
2017-03-29 15:21:58,914 - Flavors deleted successfully.
2017-03-29 15:21:58,930 - Purging configuration data ...
2017-03-29 15:21:58,968 - Configuration purged successfully.
2017-03-29 15:21:58,997 - Purging configuration data ...
2017-03-29 15:21:59,035 - Configuration purged successfully.
2017-03-29 15:21:59,063 - Purging configuration data ...
2017-03-29 15:21:59,116 - Configuration purged successfully.
2017-03-29 15:21:59,157 - Notify VDU Delete Catalog for : element-manager, status: SUCCESS, txid:
65762dfc-1493-11e7-95b8-fa163e1bdcaa
2017-03-29 15:21:59,163 - Notify VDU Delete Catalog for : control-function, status: SUCCESS, txid:
65762dfc-1493-11e7-95b8-fa163e1bdcaa
2017-03-29 15:21:59,169 - Notify VDU Delete Catalog for : session-function, status: SUCCESS, txid:
65762dfc-1493-11e7-95b8-fa163e1bdcaa
2017-03-29 15:21:59,176 - VNF Transaction completed successfully!
2017-03-29 15:21:59,199 - Notify deployment
```

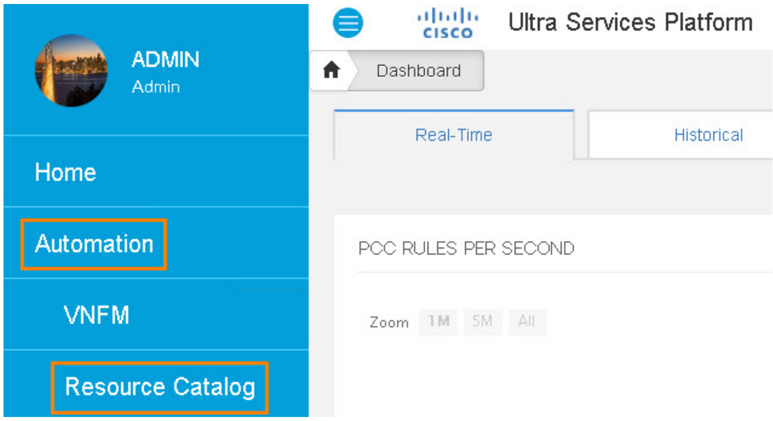
Deploy the VNF

The deployment described in this section uses a set of XML files which have been prepared for this demo. They are stored on the Windows desktop in the directory C:\dcloud and are listed in the Appendix for reference. When using the VPN via AnyConnect and a browser running on your laptop then these files need to be stored on your laptop; either copy/paste the content from this document or download the files via SFTF from the Windows machine.

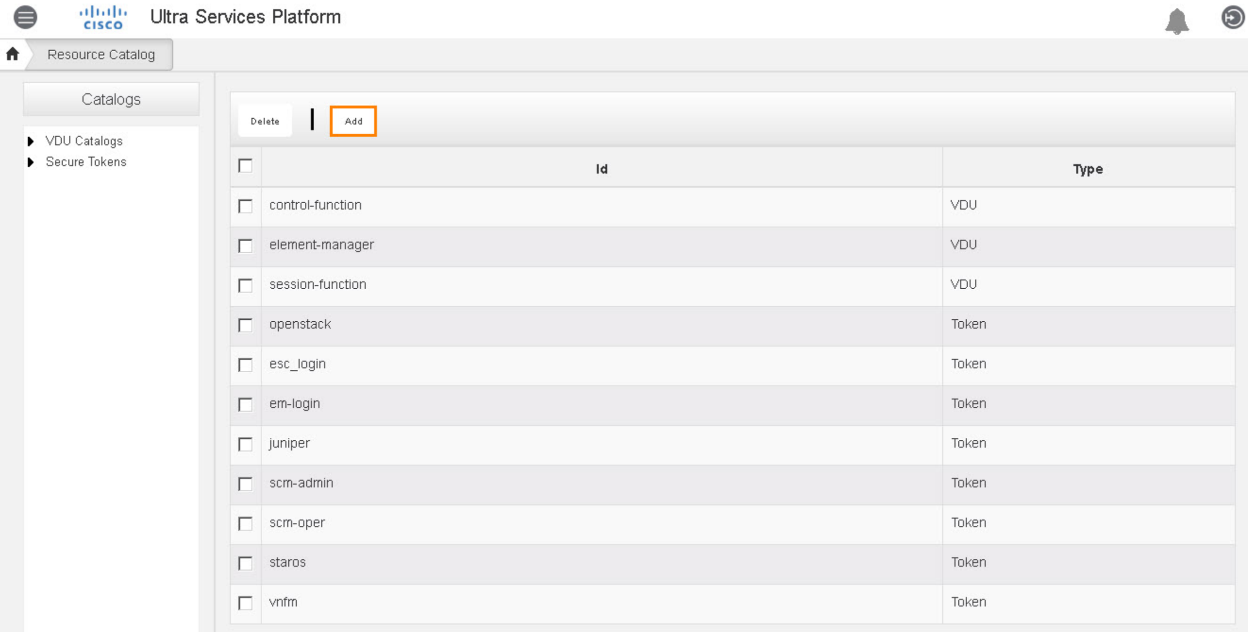
1. Navigate to https://198.18.135.12:8888/ to access the **Ultra Services Platform** UI
2. Log in with user name: **admin**, password: **cisco123**.
3. Click **menu** icon to display the navigation pane.



4. Click **Automation > Resource Catalog**.

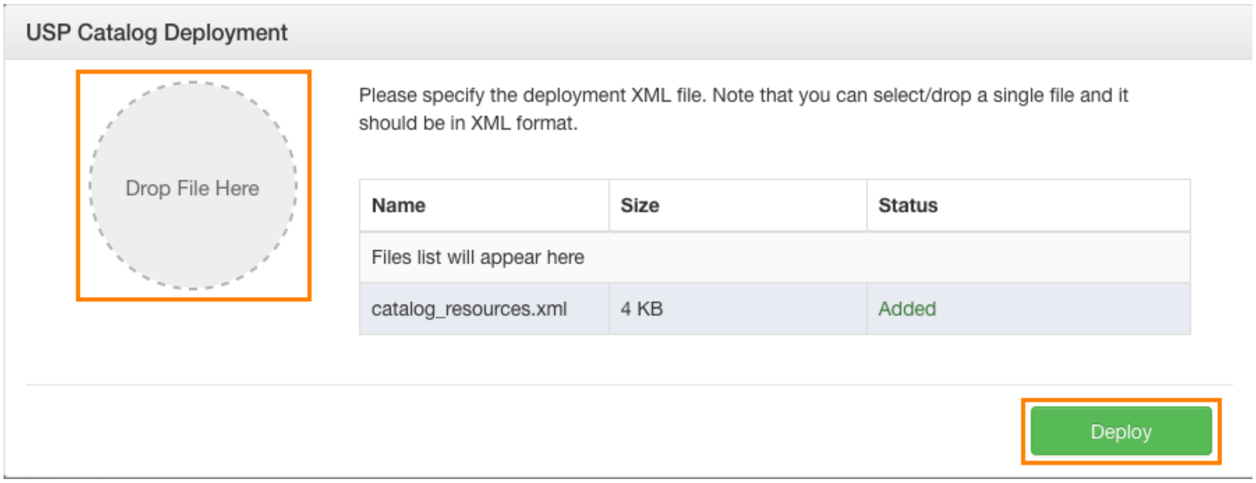


5. Click **Add**.



□	Id	Type
□	control-function	VDU
□	element-manager	VDU
□	session-function	VDU
□	openstack	Token
□	esc_login	Token
□	em-login	Token
□	juniper	Token
□	scm-admin	Token
□	scm-oper	Token
□	staros	Token
□	vnfm	Token

6. Drag and drop the **catalog_resources.xml** file from your local machine (or from **C:\dcloud** on the workstation) to the **Drop File Here** area. Click **Deploy**. The automation tools add the VNF resources (VDUs, Volumes, and VNFM-Instance).



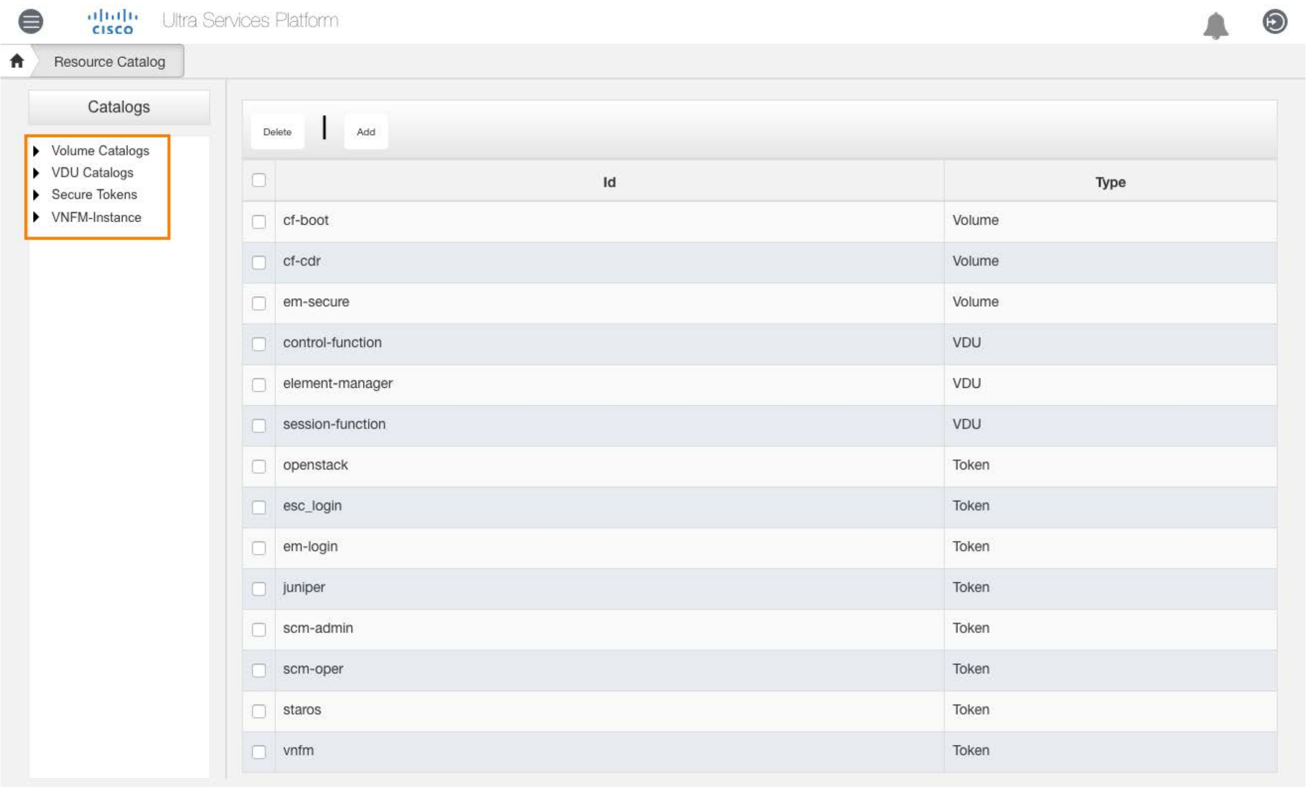
USP Catalog Deployment

Please specify the deployment XML file. Note that you can select/drop a single file and it should be in XML format.

Name	Size	Status
Files list will appear here		
catalog_resources.xml	4 KB	Added

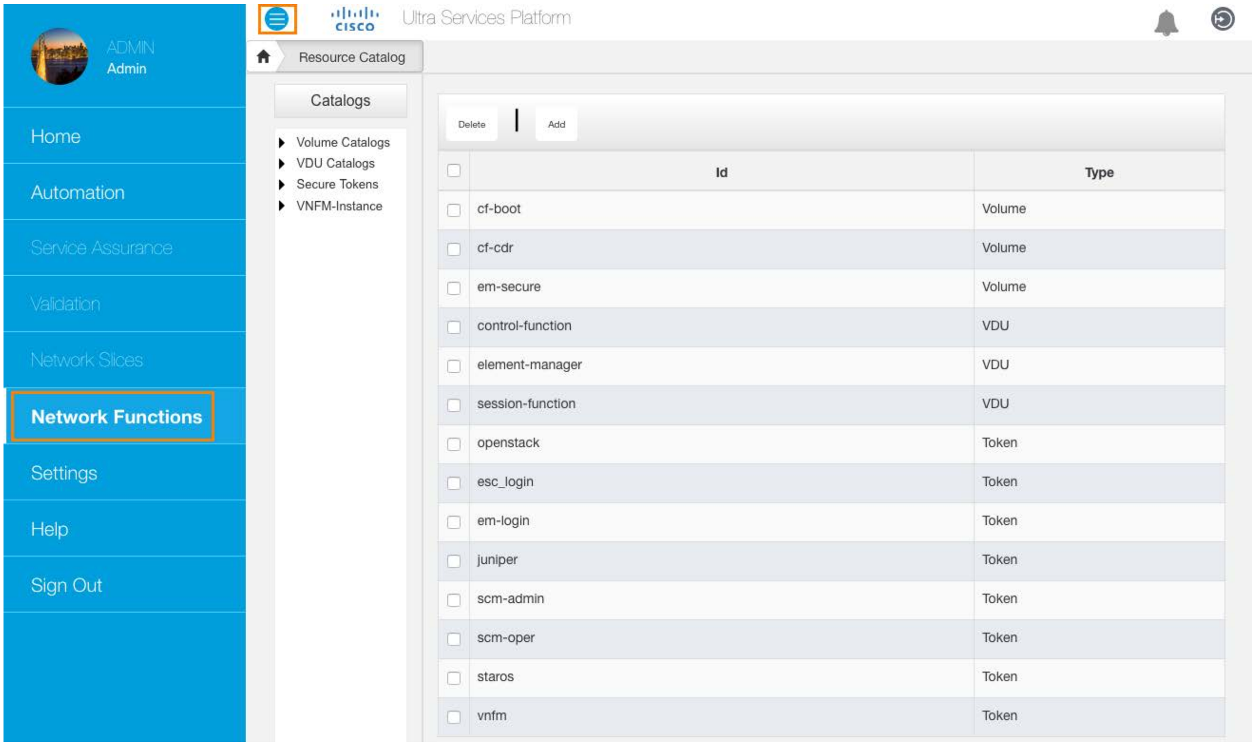
7. Wait for the screen to automatically refresh. Notice the newly added catalogs.

NOTE: Optionally, to review the UAS logs to monitor the progress, access UAS through SSH and enter the **sudo tail -f /var/log/upstart/autovnf.log** command.



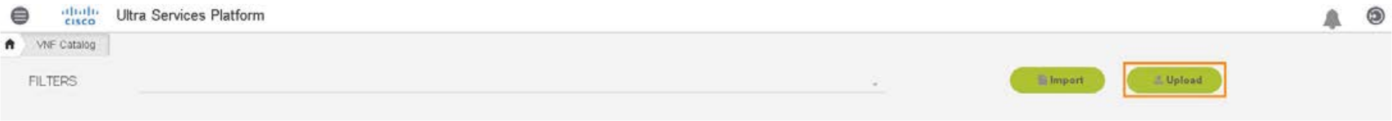
Id	Type
cf-boot	Volume
cf-cdr	Volume
em-secure	Volume
control-function	VDU
element-manager	VDU
session-function	VDU
openstack	Token
esc_login	Token
em-login	Token
juniper	Token
scm-admin	Token
scm-oper	Token
staros	Token
vnfm	Token

8. Click the **menu** icon to display the navigation pane and click **Network Functions**.



Id	Type
cf-boot	Volume
cf-cdr	Volume
em-secure	Volume
control-function	VDU
element-manager	VDU
session-function	VDU
openstack	Token
esc_login	Token
em-login	Token
juniper	Token
scm-admin	Token
scm-oper	Token
staros	Token
vnfm	Token

9. Click **Upload**.



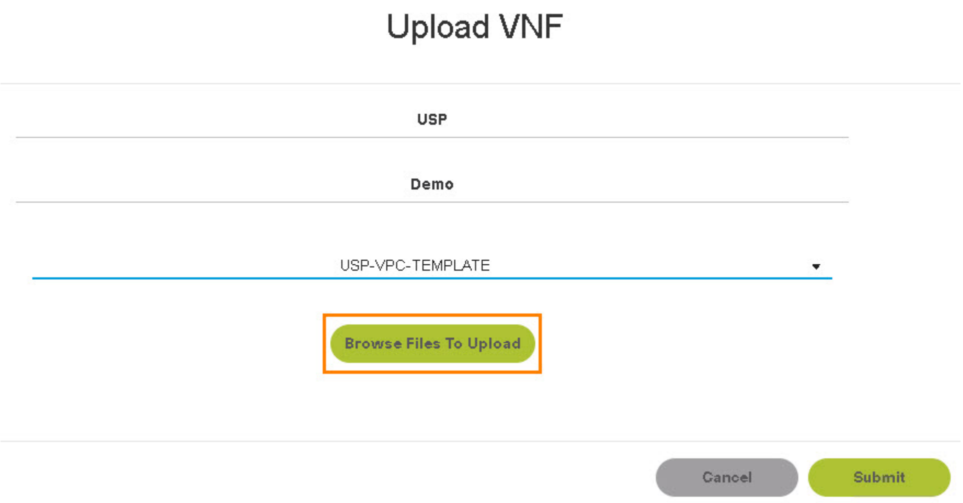
10. From the **Upload VNF** screen, do the following:

    a. Enter **USP** in the **Display Name** field.

    b. Enter **Demo** in the **Description** field.

    c. Choose **USP-VPC-TEMPLATE** from the drop-down list.

    d. Click **Browse Files To Upload**.



Upload VNF

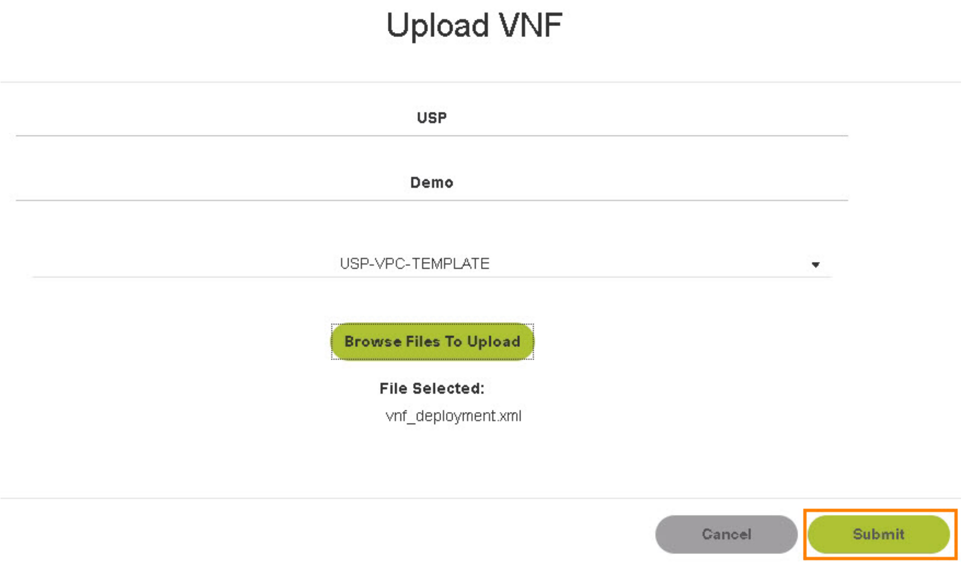
USP

Demo

USP-VPC-TEMPLATE

11. Browse your local machine (or **C:\dcloud** on the workstation) for **vnf_deployment.xml**, choose it, and click **Open**.

12. Click **Submit** to add the VNF deployment configuration to Network Functions.



Upload VNF

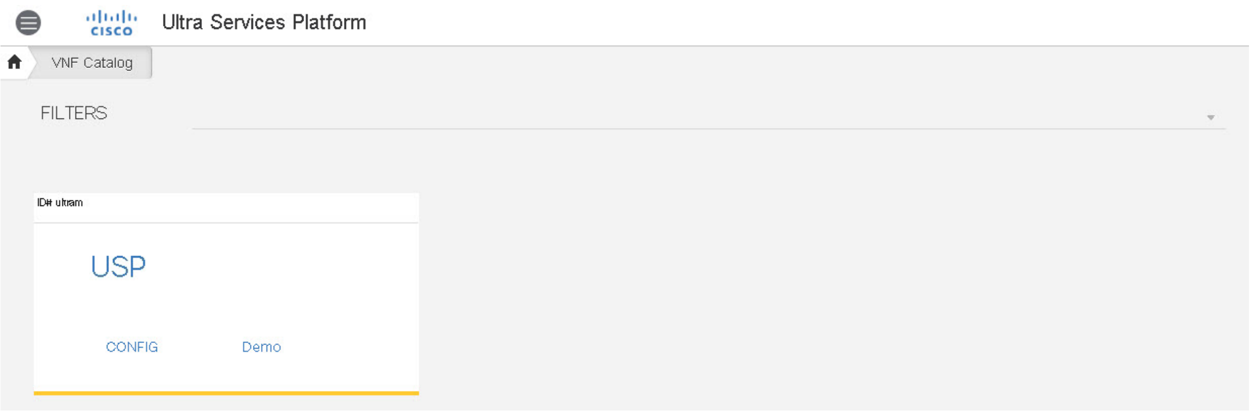
USP

Demo

USP-VPC-TEMPLATE

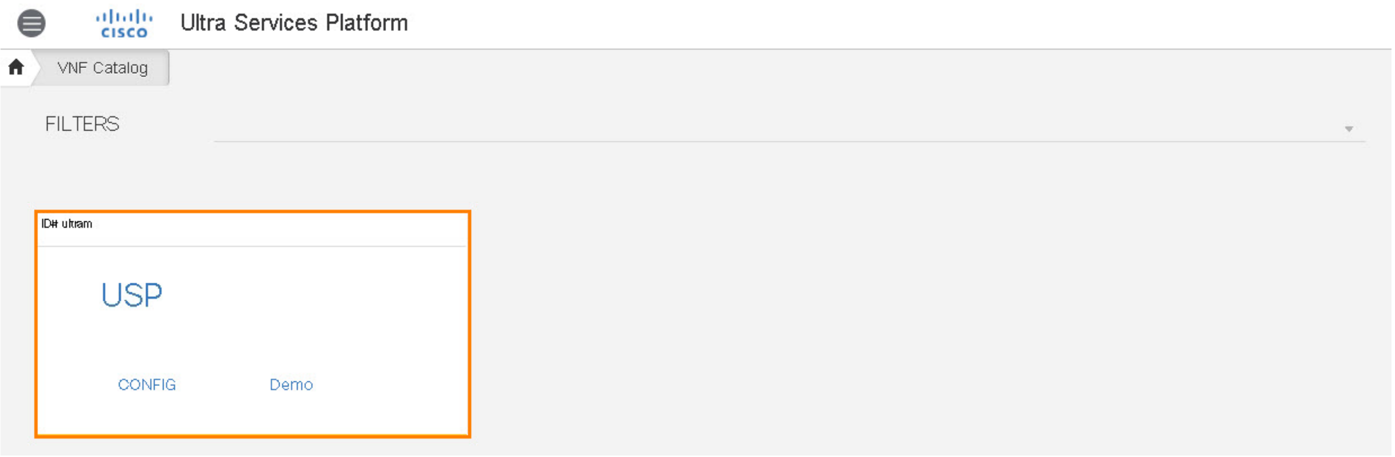
File Selected:
vnf_deployment.xml

13. Notice the USP network function added to the VNF Catalog.

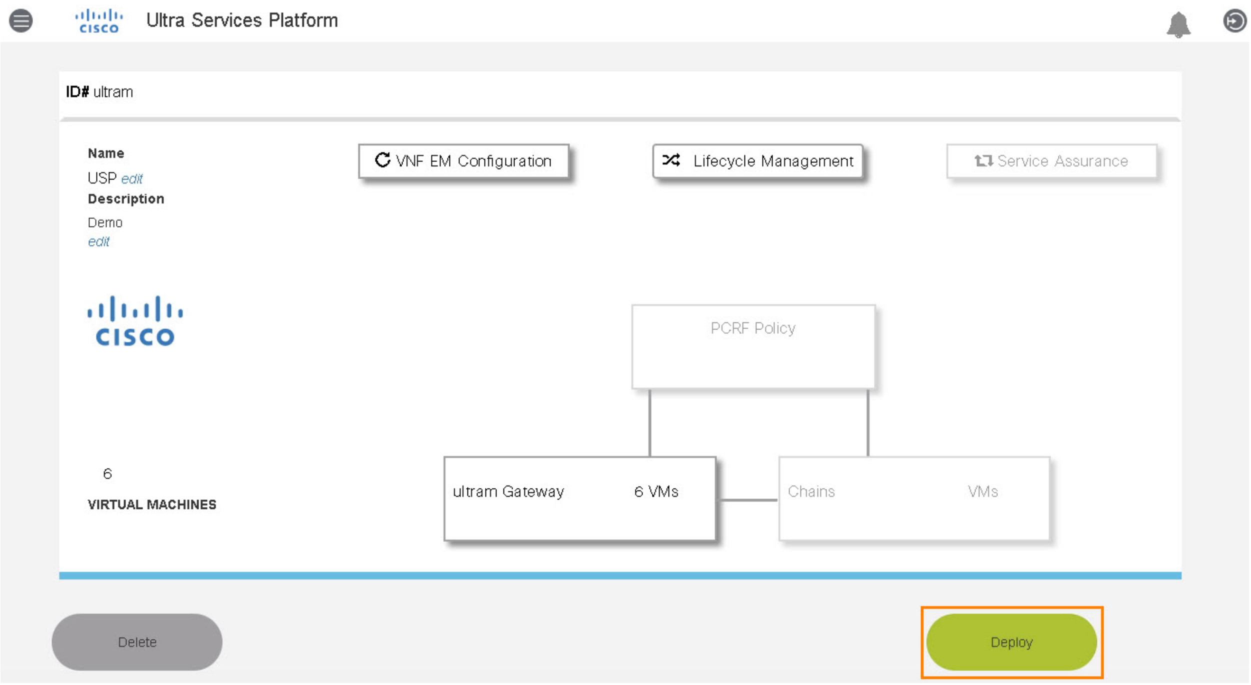


NOTE: Optionally, to review the UAS logs to monitor the progress, access UAS via SSH and enter the **sudo tail -f /var/log/upstart/autovnf.log** command.

14. Click the **USP** network function.

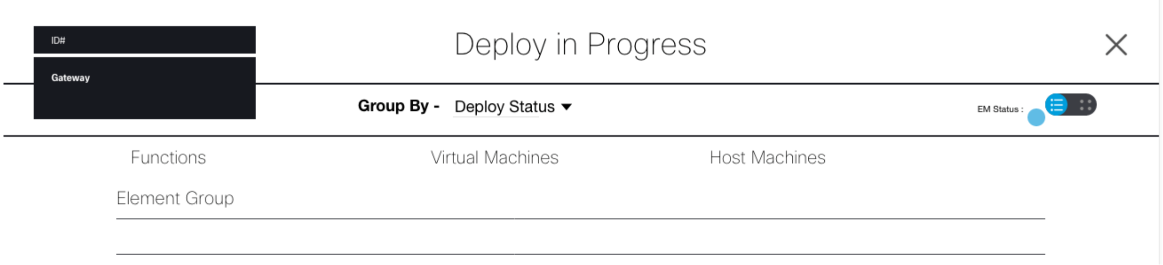


15. Click **Deploy** to deploy the USP VNF.



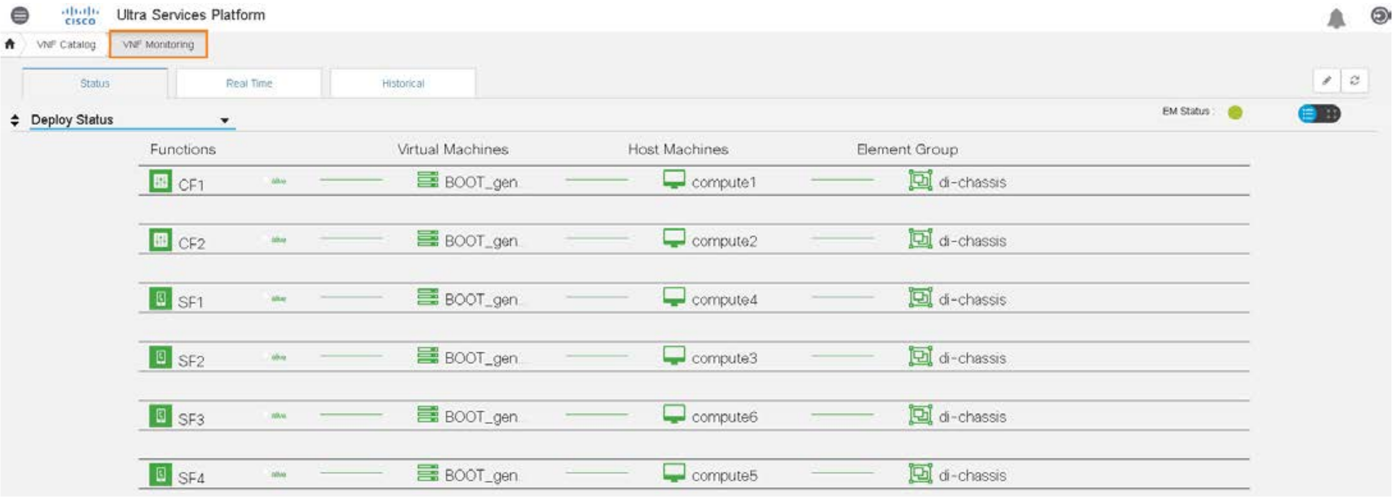
16. The automation tools deploy the USP VNF. Wait for the USP deployment to complete.

NOTE: It may take up to 30 minutes for the deployment to complete. Periodically refresh the screen until you see the **100% Deployment** banner at the top of the screen.



NOTE: Optionally, to review the UAS logs to monitor the progress, access UAS via SSH and enter the **sudo tail -f /var/log/upstart/autovnf.log** command.

17. Click **VNF Monitoring** to view the status of the VNF deployment.



Functions	Virtual Machines	Host Machines	Element Group
CF1	BOOT_gen	compute1	di-chassis
CF2	BOOT_gen	compute2	di-chassis
SF1	BOOT_gen	compute4	di-chassis
SF2	BOOT_gen	compute3	di-chassis
SF3	BOOT_gen	compute6	di-chassis
SF4	BOOT_gen	compute5	di-chassis

18. From the workstation desktop, double-click the PuTTY icon and SSH to **198.18.134.30** to access the controller node.

19. Log in with user name: **root**, password: **C1sco12345**.

20. Enter the following command to allow loopback address traffic from the service functions (SFs) to flow freely in and out of the neutron ports:

```
root@controller:~# ./open_ultra_sf_ports.sh
Updated port: 12ffdbdf-a5cf-4ef3-8c67-4dd258825529
Updated port: 172c8ca5-a931-4ea9-9376-190bc5096240
Updated port: 32fb1840-8008-4be1-b521-2697c7fa0529
Updated port: 3f8bdb5d-9a80-4a05-8370-02f796cb9d6e
Updated port: 54798dc1-97a6-49fd-8257-7805bce5c808
Updated port: 57450c79-5d3b-4265-92c1-5485925cbeff
Updated port: 57e35026-d96d-498b-bbb9-bc689ed65f3a
Updated port: 6c68f609-4d68-4e90-8e10-c286119b0f61
Updated port: 8606f8a4-817d-4845-ac8b-7609f9fd602b
Updated port: 96900aff-6e97-4a87-a339-5ebd7e842e1c
Updated port: a8a3c66a-7719-4b42-8a82-c2b52c669b12
Updated port: bccef99a-0415-48b4-bdf6-35f2b9975c44
Updated port: dc108a83-8230-4ca6-9763-d2854db00040
Updated port: e9d3d982-2863-43fc-99bf-18d078cc349f
Updated port: f5aee3a8-8b7b-4f62-866e-2c320a030bbe
Updated port: f7cc218e-45c7-41d8-8e29-0660a603a168
```

Scenario 3. Establish End-to-End Data Session with Ultra Gateway Platform

This scenario demonstrates how to establish an end-to-end data session using the Ultra Gateway Platform. The additional components added to this demo (such as the client, server, and VPC-SI VM for the MME) offer the same capabilities as the Cisco Virtualized Packet Core v1.2 demonstration in dCloud. For information about the possible use cases and the steps to collect additional information, such as traces and CDRs, refer to the Cisco Virtualized Packet Core v1.2 guide.

Steps

1. Start the UE/eNodeB simulator on the client VM. If you accessed this demonstration through RDP instead of using AnyConnect, start the **client - lattice** profile in PuTTY and keep the screen open.

NOTE: Lattice is preconfigured to log in to the client VM and start the simulator. When logging in through SSH to the client VM directly with AnyConnect and using your own laptop and an SSH client, change to the lattice directory and enter the **./start_lattice.sh** command there.

2. Start the CLI session for the simulator, which loads the configuration for the eNodeB and the APNs and call models to be used.

   - With RDP, start the **client - cli (basic)** profile in PuTTY.
   - Alternatively, if you connected through VPN, change to the lattice directory and enter the **./start_cli.sh** command.

3. Enter the following command to establish a single IPv4 data session:

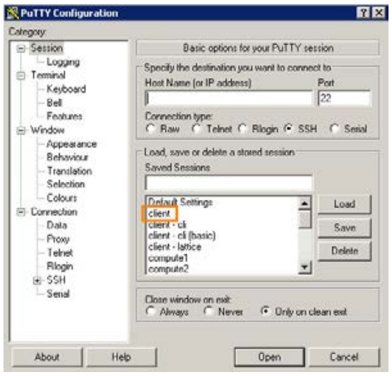
```
[lattice]> start call-model m4
```

NOTE: The **start call-model m4** command is a predefined call model loaded with the Lattice configuration file. The same file also includes the m6 model, which starts a single LTE session using IPv6, and the m46 model, which enables demonstration of a dual-stack IPv4v6 data session.

4. From PuTTY, start the **saegw** profile to start a CLI session to the UGP VM (using your own terminal via AnyConnect or from the saved session in PuTTY on the Remote Desktop) and enter the following command to verify the subscriber status:

```
[local]saegw# show subscribers all
```

5. From PuTTY, start the **client** profile and run the following commands to send actual data through the SAEGW (SGW and PGW) service of the UGP:



Port: 22

Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

Saved Sessions:
Default Settings
client
client - cli
client - cli (basic)
client - lattice
compute1
compute2

Close window on exit: ○ Always ○ Never ● Only on clean exit

NOTE: The ltc-up.sh command below is a wrapper to configure the tunnel interface established via the command above with the IP addresses assigned by the PGW, and to setup the routing for the traffic.

```
dcloud@client:~# sudo ltc-up.sh
dcloud@client:~# ping -c 3 server
dcloud@client:~# wget server/10M.bin
```

6. In the **UGP** VM, enter the following command to display the results of the traffic detection:

   ```
   [local]saegw# show active-charging sessions full all
   ```

7. From the simulator CLI session, enter **stop call-model m4** to delete the data session.

   ```
   [lattice]> stop call-model m4
   ```

Appendix A. Additional Guidelines

Save a Demo

To save the demo for your own purposes, such as to run your own customized version, complete the following steps to prepare for the automated deployment after a restart.

1. To un-deploy the ultram deployment, connect to the UAS and enter the following commands:

```
ubuntu@ultra-demo-uas-0:~$ source /opt/cisco/usp/uas/confd-6.1/confdrc
ubuntu@ultra-demo-uas-0:~$ confd_cli -u admin -C
ultra-demo-uas-0#deactivate-deployment deployment-name ultra
```

2. Enter the following command to monitor the UAS log and verify that the deployment was deactivated:

```
sudo tail -f /var/log/upstart/autovnf.log
```

3. Enter the following commands to remove the configuration for the deployment:

```
config
no deployments
no vdu-catalog
no volume-catalog
no network-catalog
no vnfm-instance
commit
```

4. Load the configuration for the deployment in config mode and monitor the result in the UAS log as described in step 2.

```
load merge autovnf.cfg
commit
```

5. Connect to the controller and enter **./dcloud_cleanup.sh**.

Transfer Files to Workstation Machine Used as Windows Remote Desktop

NOTE: An SSH daemon runs on this machine as well, which allows you to connect through SFTP.

1. Connect to **198.18.133.252** via SFTP.

2. Log in with user name: **administrator**, password: **C1sco12345**.

3. Upload the files, which then appear on the Windows machine in **C:\dcloud**.

XML Deployment Files

catalog_resources.xml

```
<config xmlns="http://tail-f.com/ns/config/1.0">
  <vdu-catalog xmlns="http://www.cisco.com/usp/nfv/usp-autovnf">
    <vdu-type>element-manager</vdu-type>
    <image>
      <location>http://198.18.135.8/bundles/em-bundle/em-1_0_0_877.qcow2</location>
    </image>
    <flavor>
      <vcpus>2</vcpus>
      <ram>4096</ram>
      <root-disk>40</root-disk>
    </flavor>
    <volumes>
      <volume>em-secure</volume>
    </volumes>
    <login-credential>em-login</login-credential>
  </vdu-catalog>
  <vdu-catalog xmlns="http://www.cisco.com/usp/nfv/usp-autovnf">
    <vdu-type>control-function</vdu-type>
    <image>
      <location>http://198.18.135.8/bundles/ugp-bundle/qvpc-di-cf.qcow2</location>
    </image>
    <flavor>
      <vcpus>8</vcpus>
      <ram>16384</ram>
      <root-disk>6</root-disk>
      <host-aggregate>mgmt</host-aggregate>
      <anti-affinity-placement>true</anti-affinity-placement>
    </flavor>
    <configurations>
      <destination-path>staros_config.txt</destination-path>
      <source-url>http://198.18.135.8:5001/uploads/system.cfg</source-url>
    </configurations>
    <neds>
      <ned-type>netconf</ned-type>
      <ned-id>cisco-staros-nc</ned-id>
      <port-number>830</port-number>
      <authentication>staros</authentication>
    </neds>
    <volumes>
      <volume>cf-boot</volume>
    </volumes>
    <volumes>
      <volume>cf-cdr</volume>
    </volumes>
  </vdu-catalog>
  <vdu-catalog xmlns="http://www.cisco.com/usp/nfv/usp-autovnf">
    <vdu-type>session-function</vdu-type>
    <image>
      <location>http://198.18.135.8/bundles/ugp-bundle/qvpc-di-sf.qcow2</location>
    </image>
    <flavor>
      <vcpus>8</vcpus>
      <ram>16384</ram>
      <root-disk>6</root-disk>
      <host-aggregate>service</host-aggregate>
    </flavor>
    <upp>
      <cores>30</cores>
    </upp>
  </vdu-catalog>
  <volume-catalog xmlns="http://www.cisco.com/usp/nfv/usp-autovnf">
    <volume-id>cf-boot</volume-id>
    <volume>
      <type>LUKS</type>
      <size>4</size>
      <bus>ide</bus>
      <bootable>true</bootable>
    </volume>
  </volume-catalog>
  <volume-catalog xmlns="http://www.cisco.com/usp/nfv/usp-autovnf">
    <volume-id>cf-cdr</volume-id>
    <volume>
      <type>LUKS</type>
      <size>16</size>
      <bus>ide</bus>
    </volume>
  </volume-catalog>
  <volume-catalog xmlns="http://www.cisco.com/usp/nfv/usp-autovnf">
    <volume-id>em-secure</volume-id>
    <volume>
      <type>LUKS</type>
      <size>1</size>
    </volume>
  </volume-catalog>
  <network-catalog xmlns="http://www.cisco.com/usp/nfv/usp-autovnf">
    <network-id>di-internal1</network-id>
    <pre-created>di_internal1</pre-created>
    <ip-prefix>192.168.1.0/24</ip-prefix>
  </network-catalog>
  <network-catalog xmlns="http://www.cisco.com/usp/nfv/usp-autovnf">
    <network-id>management</network-id>
    <pre-created>cf-mgmt</pre-created>
    <ip-prefix>172.16.181.0/24</ip-prefix>
    <dhcp>true</dhcp>
    <gateway>172.16.181.1</gateway>
    <reserved-ips>
      <start>172.16.181.1</start>
      <end>172.16.181.20</end>
    </reserved-ips>
  </network-catalog>
  <network-catalog xmlns="http://www.cisco.com/usp/nfv/usp-autovnf">
    <network-id>orch</network-id>
    <pre-created>orchestration</pre-created>
    <ip-prefix>172.16.180.0/24</ip-prefix>
    <reserved-ips>
      <start>172.16.180.1</start>
      <end>172.16.180.30</end>
    </reserved-ips>
  </network-catalog>
  <network-catalog xmlns="http://www.cisco.com/usp/nfv/usp-autovnf">
    <network-id>service-network1</network-id>
    <pre-created>service1</pre-created>
    <ip-prefix>10.10.10.0/24</ip-prefix>
  </network-catalog>
  <network-catalog xmlns="http://www.cisco.com/usp/nfv/usp-autovnf">
    <network-id>service-network2</network-id>
    <pre-created>service2</pre-created>
    <ip-prefix>20.20.20.0/24</ip-prefix>
  </network-catalog>
  <vnfm-instance xmlns="http://www.cisco.com/usp/nfv/usp-autovnf">
    <vnf-em-peer>172.16.181.95</vnf-em-peer>
    <autovnf-peer>198.18.135.15</autovnf-peer>
    <tenant>Core</tenant>
    <netconf-credential>vnfm</netconf-credential>
    <dep-prefix>ultram</dep-prefix>
  </vnfm-instance>
</config>
```

vnf_deployment.xml

```
<config xmlns="http://tail-f.com/ns/config/1.0">
  <deployments xmlns="http://www.cisco.com/usp/nfv/usp-autovnf">
    <id>ultram</id>
    <di-internal-networks>
      <di-internal-network>di-internal1</di-internal-network>
    </di-internal-networks>
    <domain-name>openstacklocal</domain-name>
    <dns-servers>
      <dns-ip>198.18.133.1</dns-ip>
    </dns-servers>
    <vnf-em>
      <vdu-id>element-manager</vdu-id>
      <ha-vip>172.16.181.10</ha-vip>
      <high-availability>true</high-availability>
      <orchestration-network>orch</orchestration-network>
      <management-network>management</management-network>
      <scm-admin>scm-admin</scm-admin>
      <scm-oper>scm-oper</scm-oper>
    </vnf-em>
    <infra-element-groups>
      <name>di-chassis</name>
      <high-availability>true</high-availability>
      <vdus>
        <vdu-id>CF</vdu-id>
        <vdu-ref>control-function</vdu-ref>
        <instances>2</instances>
        <interfaces>
          <ifname>di_intf1</ifname>
          <network>di-internal1</network>
        </interfaces>
        <interfaces>
```

```
          <ifname>mgmt</ifname>
          <network>management</network>
          <uplink-actions>
            <action>mount-ned-nc</action>
          </uplink-actions>
        </interfaces>
        <interfaces>
          <ifname>orch1</ifname>
          <network>orch</network>
        </interfaces>
      </vdus>
      <vdus>
        <vdu-id>SF</vdu-id>
        <vdu-ref>session-function</vdu-ref>
        <instances>4</instances>
        <interfaces>
          <ifname>di_intf1</ifname>
          <network>di-internal1</network>
        </interfaces>
        <interfaces>
          <ifname>orch1</ifname>
          <network>orch</network>
        </interfaces>
        <interfaces>
          <ifname>svc_intf1</ifname>
          <network>service-network1</network>
        </interfaces>
        <interfaces>
          <ifname>svc_intf2</ifname>
          <network>service-network2</network>
        </interfaces>
      </vdus>
    </infra-element-groups>
  </deployments>
</config>
```

Troubleshooting

This dCloud demonstration is thoroughly tested. The system is expected to come up in the same working configuration whenever a new instance is loaded. It is possible, however, particularly when running the un-deployment and deployment scenarios, for the system to behave unexpectedly or for steps and procedures to not work. If this happens, you have to option of exiting your session and launching a new session. Alternatively, explore the following troubleshooting steps.

UAS Issues

UAS-0 must be active to use it for configuration. Occasionally, the AutoVNF process may fail, triggering a restart of the node so that UAS-1 comes up. In that case, a known recovery scenario is to reboot UAS-1 to force UAS-0 to come back up when connecting to the UAS address. After launching confd, run the cleanup commands to deactivate the deployment and clear the configuration, which may bring the system back to a usable state.

In case there is no reaction to the **deactivate-deployment** command, verify whether there are possibly previous transactions in the In-Progress state by entering the **show transactions** command. Running the **clear-transactions transaction-id all** command and rebooting the UAS node may assist in recovering.

VPC Issues

If VPC does not behave as expected, try running a reload.

Americas Headquarters
Cisco Systems, Inc.
San Jose, CA

Asia Pacific Headquarters
Cisco Systems (USA) Pte. Ltd.
Singapore

Europe Headquarters
Cisco Systems International BV Amsterdam,
The Netherlands

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco Website at **www.cisco.com/go/offices.**