SJSU SAN JOSÉ STATE UNIVERSITY

HW #1 - Ansible

Team: Orion

- Babu Rajendran
- Eric Cheng
- Archana Shokeen

GitHub: https://github.com/sjsu-cmpe272/orion-hw01-ansible

- In the DCloud lab, https://dcloud2-sjc.cisco.com/content/demo/90426?returnPathTitleKey=content-view
- Configure Ansible server on ***ubuntu*** to deploy a webserver on ***centos1*** and ***centos2***, and bring it up on port 8080 with a web page that is accessible on from the Windows 10 Workstation, **wkst1,** that displays the message: “Hello World print from CentosX.” (Where X is 1 or 2 depending on the server)
- Include in the Ansible playbook, plays to **deploy** and **un-deploy** the webserver resources

SJSU SAN JOSÉ STATE UNIVERSITY

Schedule DevNet Data Center

Cisco DevNet Express Data Center v2

Start Date 13-Sep-2020 *Start Time* 22:30 Now

End Date 14-Sep-2020 *End Time* 10:30

Cancel Next

Day Week Month Sep 13 – 19, 2020 Today

	Sun 13	Mon 14	Tue 15	Wed 16	Thu 17	Fri 18	Sat 19
12am	00:00 - 22:26 Unavailable	00:00 - 10:30 Cisco DevNet Express Data Center v2					
1am							
2am							
3am							
4am							
5am							
6am							
7am							
8am							
9am							
10am							
11am							
12pm							
1pm							
2pm							
3pm							
4pm							
5pm							
6pm							
7pm							
8pm							
9pm							
10pm							
11pm	22:30 - 00:00 Cisco DevNet						

Cancel Next

SJSU SAN JOSÉ STATE UNIVERSITY

Schedule Cisco DevNet Express Data Center v2

Please tell us about how you will be using dCloud to help prioritize future enhancements. We recommend you test connectivity before your session using the Connection Test tool available from the Dashboard page.

* Required Field

Primary Use * Test Drive

Revenue Impact * Not Applicable

Customer Name SJSU

Partner Name

Campaign/Promo/Tag

Account Manager ID

Deal ID

Cancel Schedule

Sessions
Favorites
History
Custom content
Routers
Connection tests

Sessions

Sort By Session Status

Edit Sessions Download Details

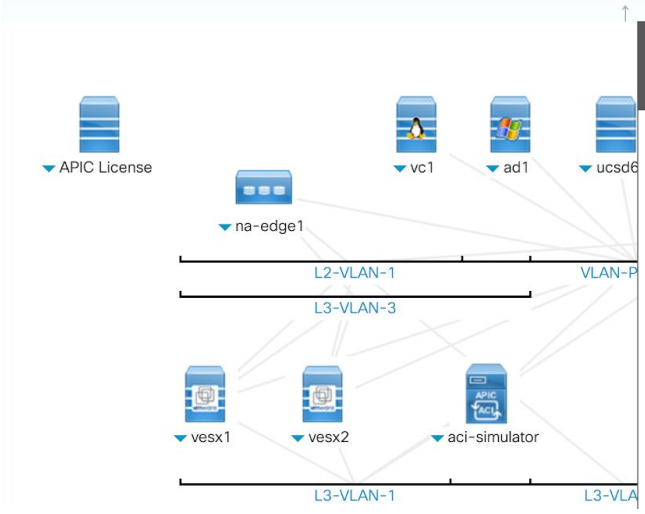
Cisco DevNet Express Data Center v2

Start: 13-Sep-2020 22:31 End: 14-Sep-2020 10:31 Session ID: 253275 Virtual Center: 0

Edit Share Cancel Scheduled

Cisco DevNet Express Data Center v2

Details Servers Resources Devices 11:44:06



Session Details

Session Configuration

Provide the session details when you open a support request.

Parent Demo:	Cisco DevNet Express Data Center v2
Owner:	baburajendran32028
Session ID:	253275
End Time:	14-Sep-2020 10:31
Last Modified:	13-Sep-2020 22:42
VPN Available:	true
Session Devices:	apic_license_7

SJSU SAN JOSÉ STATE UNIVERSITY

Connect to dCloud

1. Install a VPN client

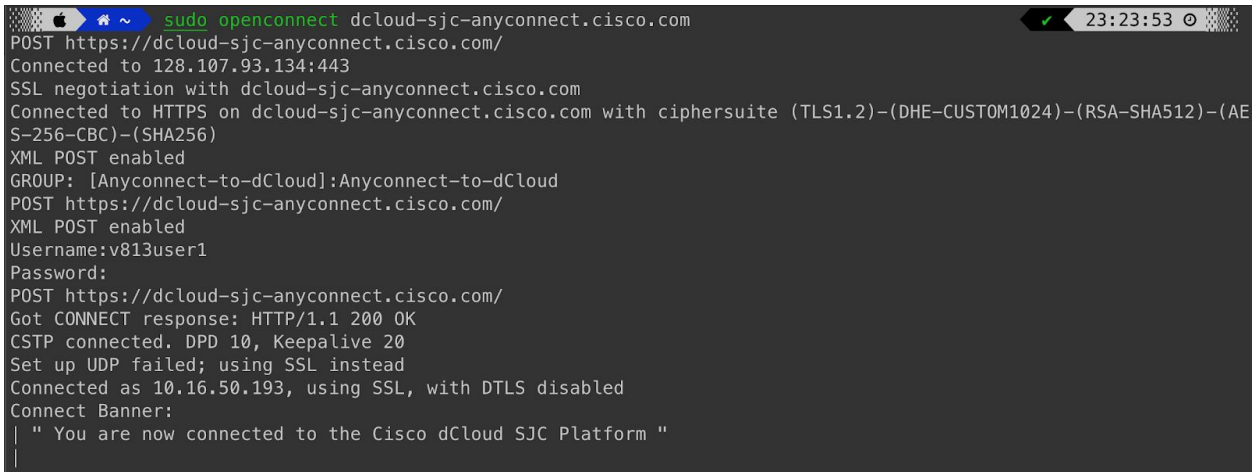
```
brew install openconnect
```

```
~ brew install openconnect                                    ✔ 23:07:49
Updating Homebrew...
==> Auto-updated Homebrew!
Updated 2 taps (homebrew/core and homebrew/cask).
==> Updated Formulae
Updated 52 formulae.
```

```
==> Downloading https://homebrew.bintray.com/bottles/p11-kit-0.23.21.catalina.bottle.
==> Downloading https://homebrew.bintray.com/bottles/gnutls-3.6.15.catalina.bottle.ta
==> Downloading from https://d29vzk4ow07wi7.cloudfront.net/513407ec28ac63623dbc05ac68
==> Downloading https://homebrew.bintray.com/bottles/openconnect-8.10.catalina.bottle
Already downloaded: /Users/babu/Library/Caches/Homebrew/downloads/394df595c4816e1ab4f732b205cfd51055cf3887bdbf46f
8f4c8bed73dc8f056--openconnect-8.10.catalina.bottle.tar.gz
==> Installing dependencies for openconnect: p11-kit and gnutls
==> Installing openconnect dependency: p11-kit
==> Pouring p11-kit-0.23.21.catalina.bottle.tar.gz
🍺  /usr/local/Cellar/p11-kit/0.23.21: 63 files, 3.0MB
==> Installing openconnect dependency: gnutls
==> Pouring gnutls-3.6.15.catalina.bottle.tar.gz
==> Caveats
If you are going to use the Guile bindings you will need to add the following
to your .bashrc or equivalent in order for Guile to find the TLS certificates
database:
  export GUILE_TLS_CERTIFICATE_DIRECTORY=/usr/local/etc/gnutls/
==> Summary
🍺  /usr/local/Cellar/gnutls/3.6.15: 1,250 files, 10.8MB
==> Installing openconnect
==> Pouring openconnect-8.10.catalina.bottle.tar.gz
🍺  /usr/local/Cellar/openconnect/8.10: 53 files, 2.7MB
==> Caveats
==> gnutls
If you are going to use the Guile bindings you will need to add the following
to your .bashrc or equivalent in order for Guile to find the TLS certificates
database:
  export GUILE_TLS_CERTIFICATE_DIRECTORY=/usr/local/etc/gnutls/
~                                                  ✔ 34s ⌛ 23:08:44
```

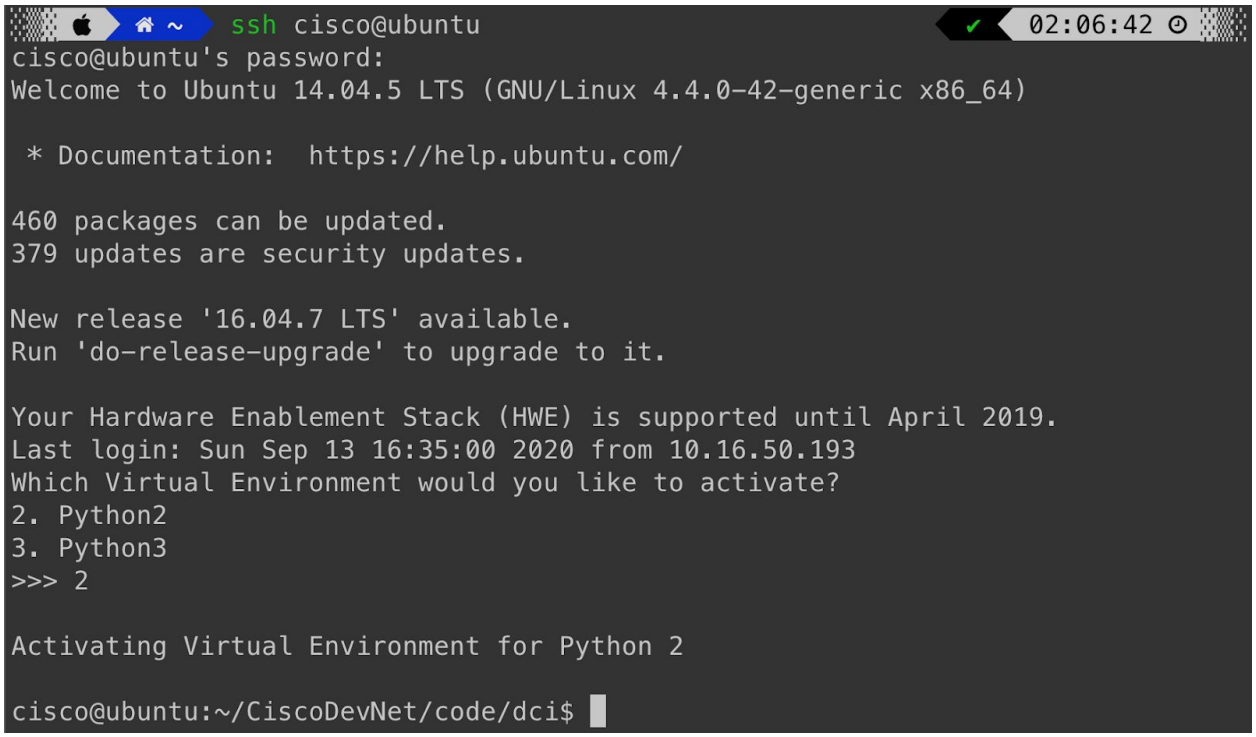
2. Establish VPN connection

```
sudo openconnect dcloud-sjc-anyconnect.cisco.com
```



3. Connect to the control node

```
ssh cisco@ubuntu
```



Setup SSH keys

To communicate with the centos servers from the ubuntu control node, SSH keys need to be generated and transferred to the centos hosts.

1. **Generate an SSH key-pair**

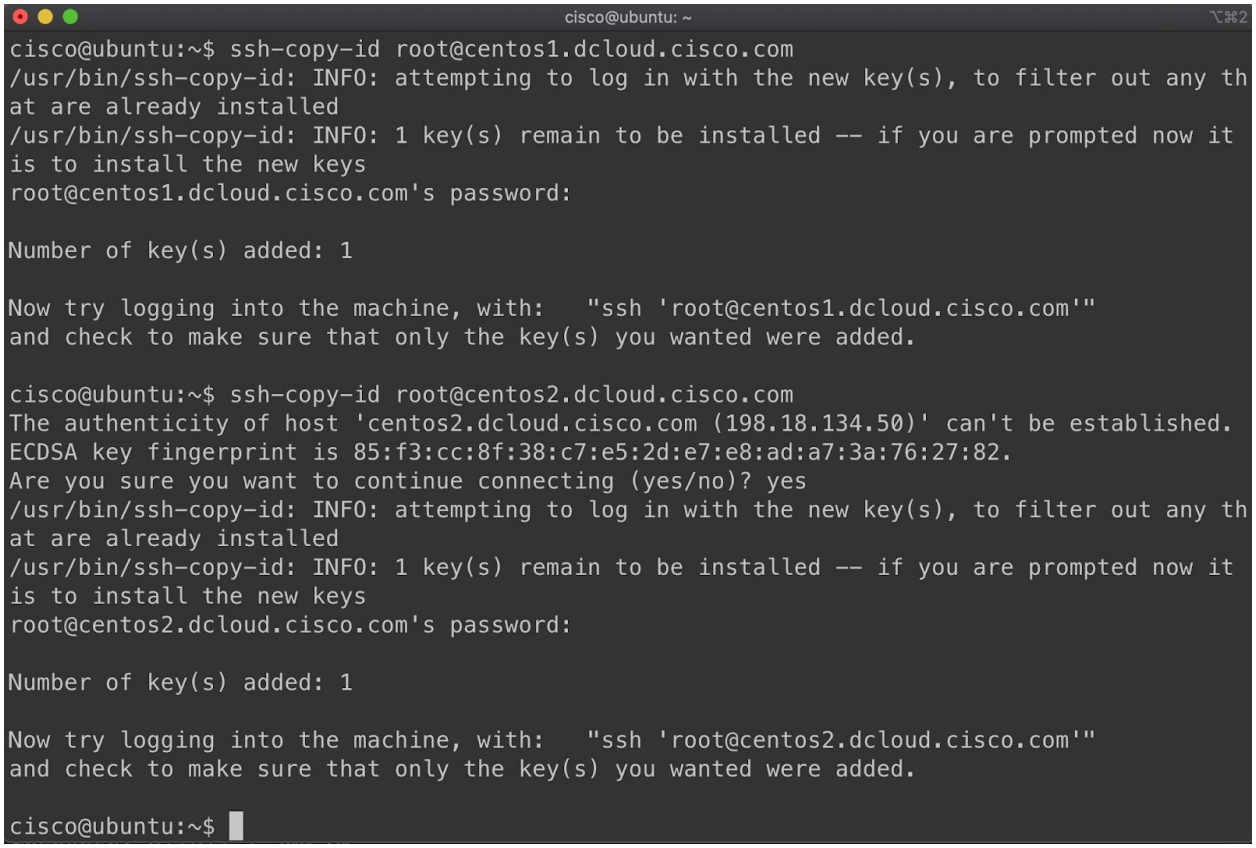
```
ssh-keygen -t rsa -b 4096
```

```
cisco@ubuntu:~/orion/hw01$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/cisco/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/cisco/.ssh/id_rsa.
Your public key has been saved in /home/cisco/.ssh/id_rsa.pub.
The key fingerprint is:
1d:ec:12:53:d5:1e:a0:4b:d1:18:4b:eb:37:28:c8:53 cisco@ubuntu
The key's randomart image is:
+--[ RSA 4096]----+
|          ==oo   |
|         +.=. o  |
|        E B  . . |
|     . o B +  .  |
|      + S * o    |
|       . o . .   |
|                 |
|                 |
|                 |
+-----------------+
cisco@ubuntu:~/orion/hw01$
```

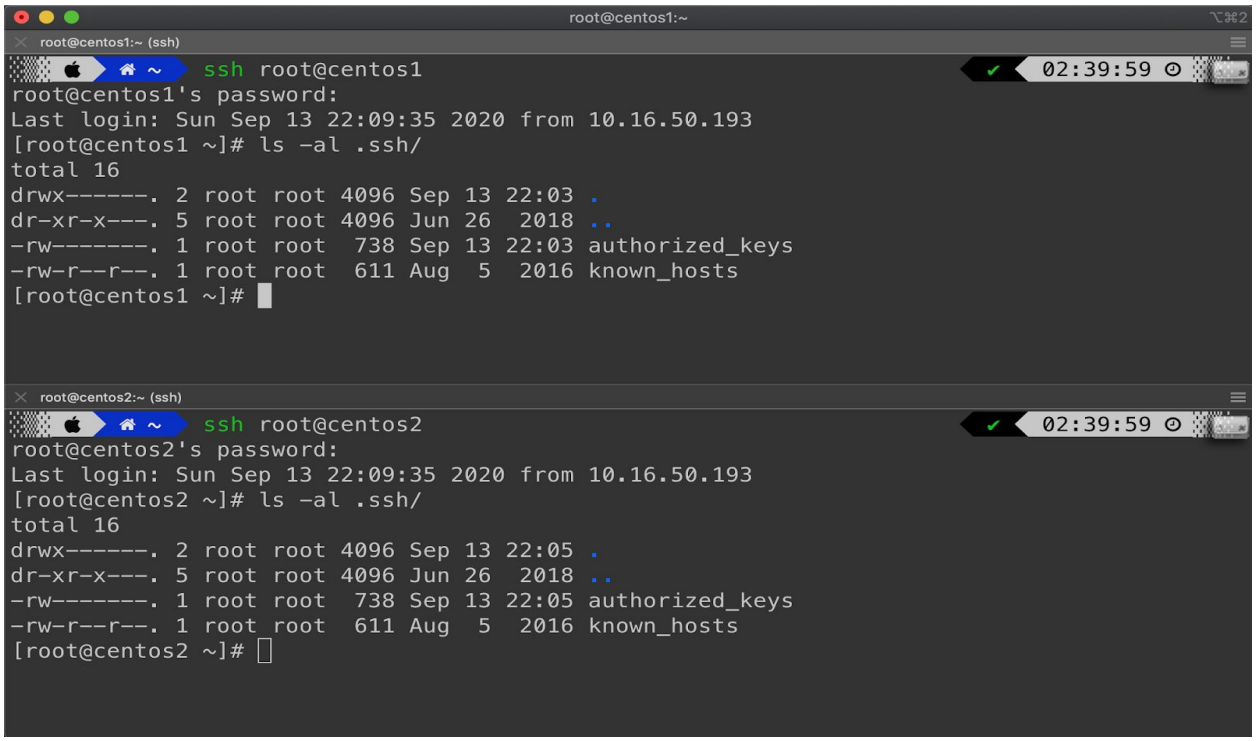
2. **Copy the keys to centos hosts**

```
ssh-copy-id root@centos1.dcloud.cisco.com
```

```
ssh-copy-id root@centos2.dcloud.cisco.com
```



3. Verify that the keys have been copied

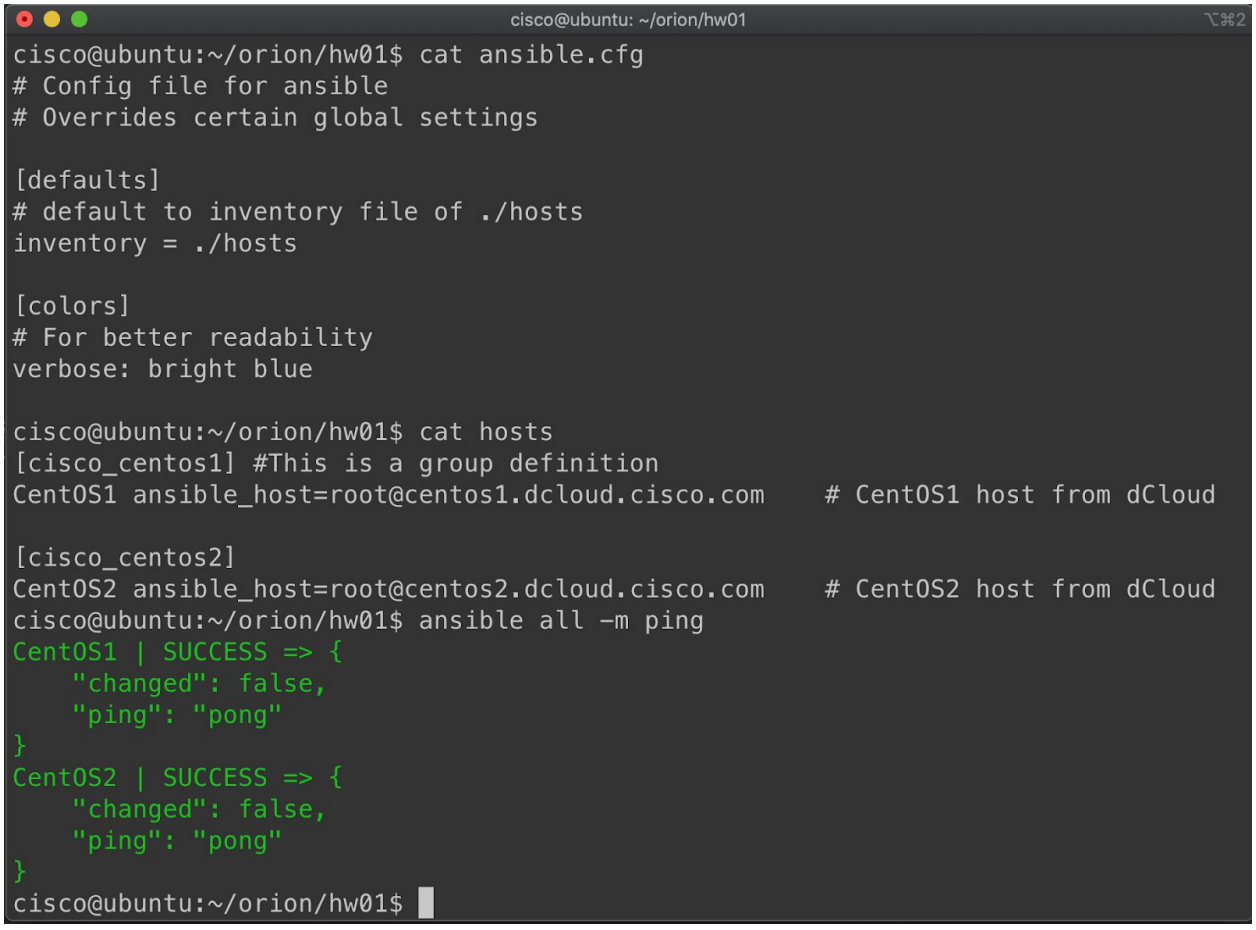


Create Ansible inventory

Setting up the ansible directory required a few files. First, a directory was created in order to keep the files in one place. Then, the default *ansible.cfg* and *hosts* file were copied over. The host file already holds the IP address of the two centOS clients, which we can check the connection by running:

```
ansible all -m ping
```

Correctly setting it up will generate the green success outputs as seen below.



Deploy the web server

After confirming that the ansible pings were successful, the .yml and .html files could be created. The .yml files are ansible playbooks, which will be used to deploy and undeploy the web server from the hosts. Running *deploy.yml* will install the necessary dependencies and deploy the web server to the port 8080, as seen below. The command is:

```
ansible-playbook deploy.yml
```

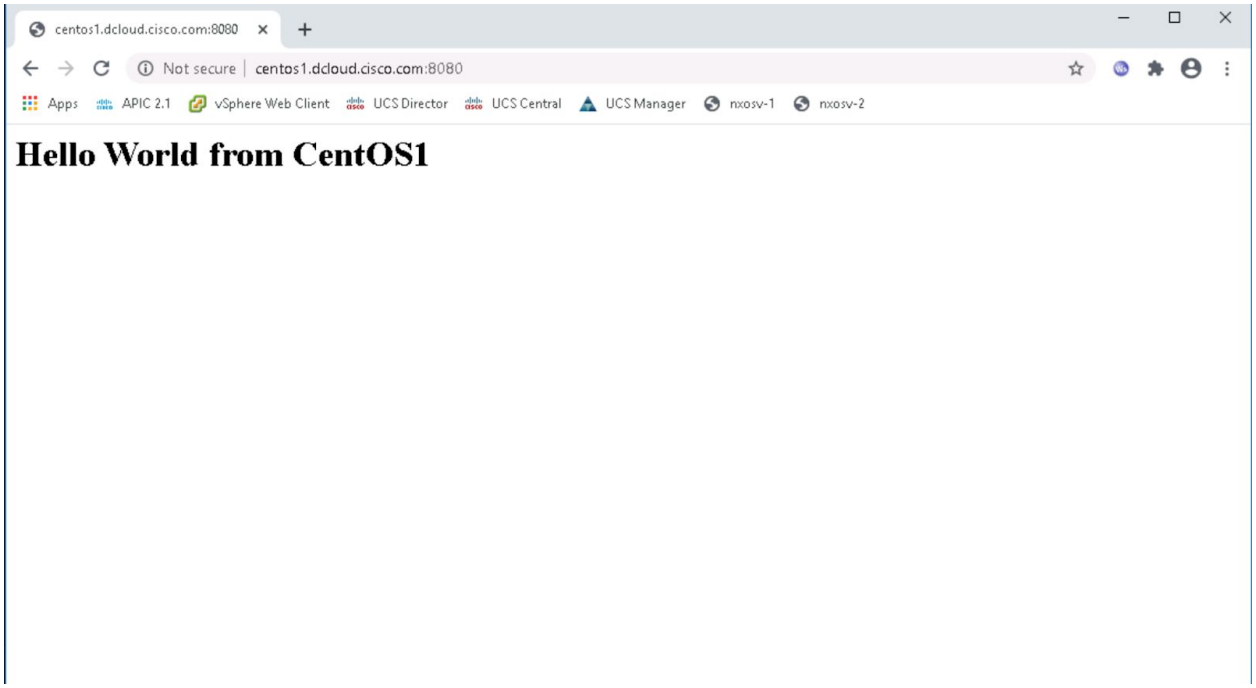
```
cisco@ubuntu:~/orion/hw01$ ansible-playbook deploy.yml

PLAY [all] *********************************************************************

TASK [Gathering Facts] *********************************************************
ok: [CentOS1]
ok: [CentOS2]

TASK [Add epel-repo] ***********************************************************
changed: [CentOS1]
changed: [CentOS2]

TASK [Install Nginx] ***********************************************************
changed: [CentOS1]
changed: [CentOS2]

TASK [add site.html file] ******************************************************
ok: [CentOS1]
ok: [CentOS2]

TASK [Add nginx.conf] **********************************************************
changed: [CentOS1]
changed: [CentOS2]

TASK [allow all access to tcp port 8080] ***************************************
changed: [CentOS1]
changed: [CentOS2]

TASK [Reload Firewall] *********************************************************
changed: [CentOS1]
changed: [CentOS2]
```

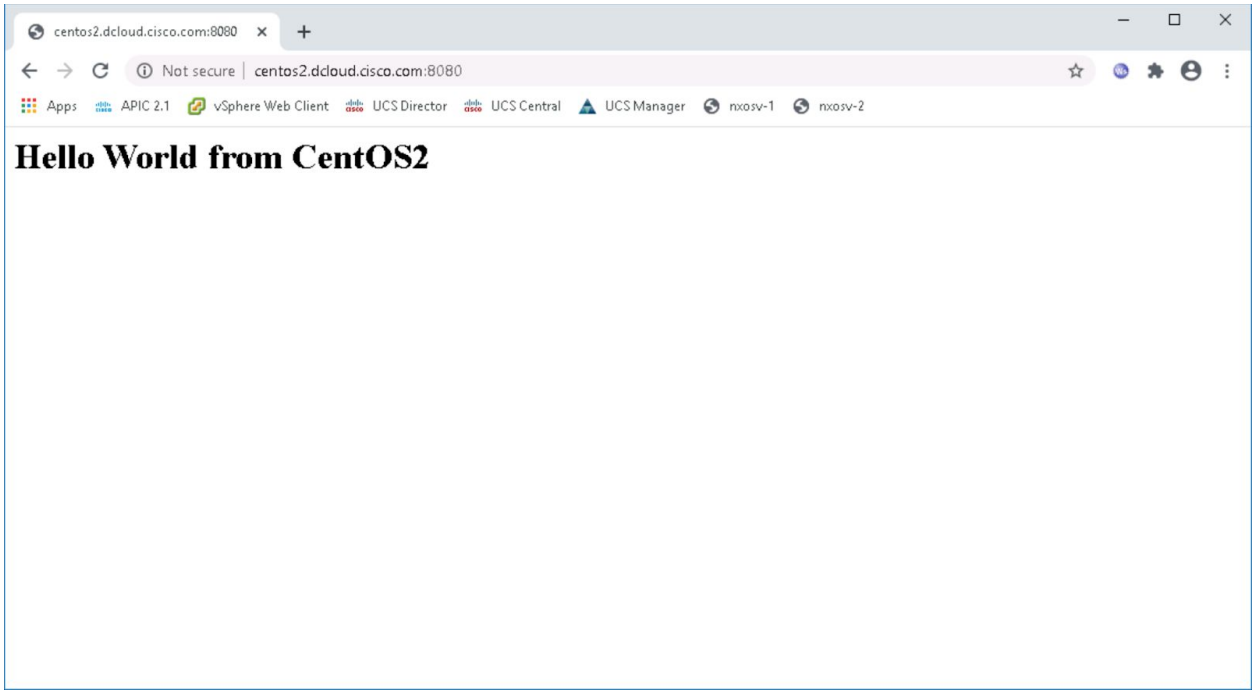
```
TASK [Start Nginx] *************************************************************
changed: [CentOS1]
changed: [CentOS2]

PLAY RECAP *********************************************************************
CentOS1                    : ok=8    changed=6    unreachable=0    failed=0
CentOS2                    : ok=8    changed=6    unreachable=0    failed=0

cisco@ubuntu:~/orion/hw01$
```

Validation from Windows workstation

By using curl <ip address:port>, one can see if the connection was successful. Logging on to the wkst1 remote desktop on dcloud and running the ip address and port 8080 will print out the html file. The centOS print statements change according to the ip address.

```
cisco@ubuntu: ~/orion/hw01
cisco@ubuntu:~/orion/hw01$ curl centos1.dcloud.cisco.com:8080
<h1>Hello World from CentOS1</h1>
cisco@ubuntu:~/orion/hw01$ curl centos2.dcloud.cisco.com:8080
<h1>Hello World from CentOS2</h1>
cisco@ubuntu:~/orion/hw01$
```

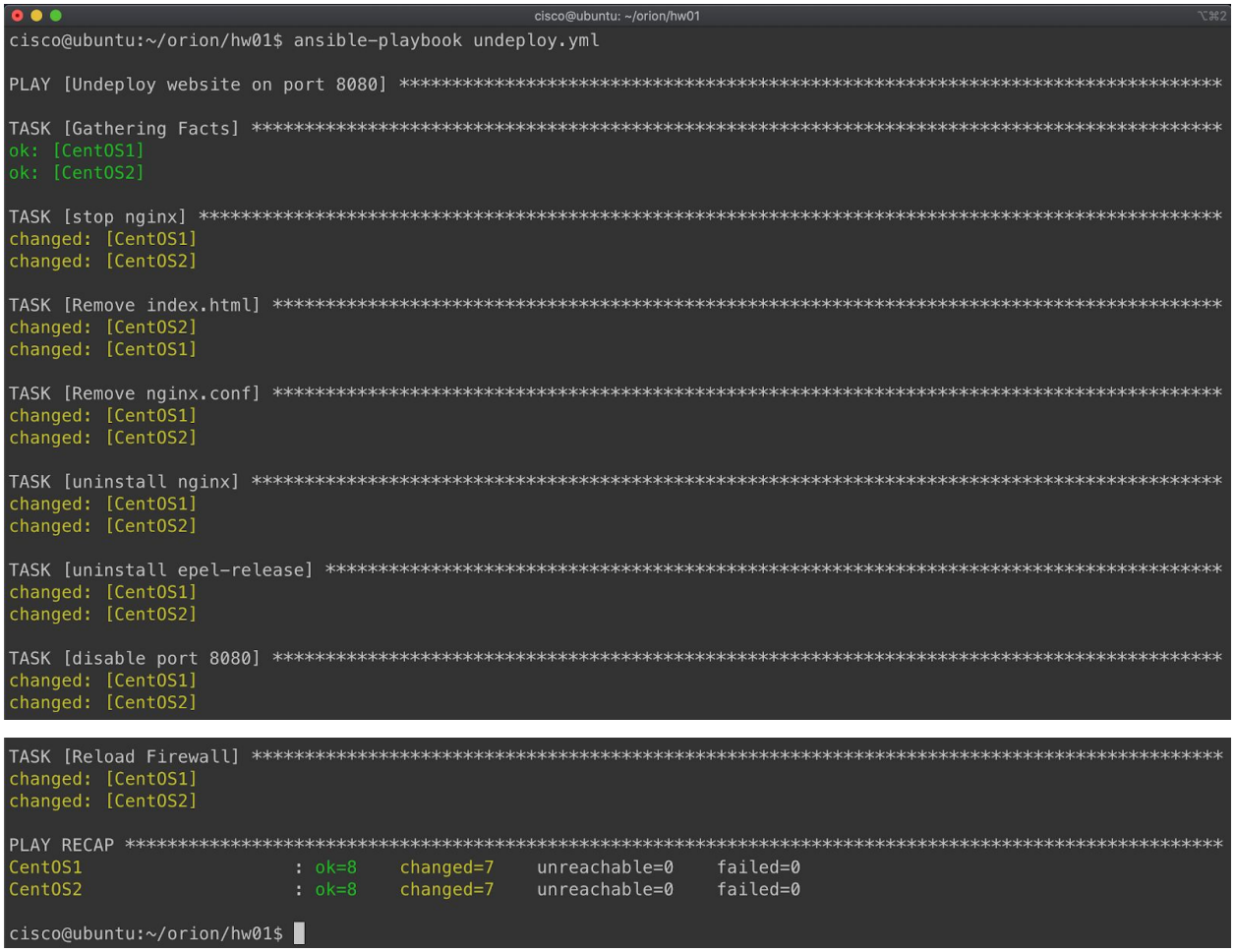




Undeploy the web server

Running the *undeploy.yml* file will undeploy the web server from port 8080, reinstate the firewall, and uninstall nginx and its dependencies. The command is

```
ansible-playbook undeploy.yml
```



Validate after undeployment

After undeployment validate that the target hosts are not reachable on port 8080.