

# 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

# 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

---

# Connect to dCloud

## 1. Install a VPN client

```
brew install openconnect
```

```

brew install openconnect
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/394df595c4816e1ab4f732b205cfd51055cf3887bdbf46f8f4c8bed73dc8f056--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/
  
```

## 2. Establish VPN connection

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

```
sudo openconnect dcloud-sjc-anyconnect.cisco.com
POST https://dcloud-sjc-anyconnect.cisco.com/
Connected to 128.107.93.134:443
SSL negotiation with dcloud-sjc-anyconnect.cisco.com
Connected to HTTPS on dcloud-sjc-anyconnect.cisco.com with ciphersuite (TLS1.2)-(DHE-CUSTOM1024)-(RSA-SHA512)-(AES-256-CBC)-(SHA256)
XML POST enabled
GROUP: [Anyconnect-to-dCloud]:Anyconnect-to-dCloud
POST https://dcloud-sjc-anyconnect.cisco.com/
XML POST enabled
Username:v813user1
Password:
POST https://dcloud-sjc-anyconnect.cisco.com/
Got CONNECT response: HTTP/1.1 200 OK
CSTP connected. DPD 10, Keepalive 20
Set up UDP failed; using SSL instead
Connected as 10.16.50.193, using SSL, with DTLS disabled
Connect Banner:
| " You are now connected to the Cisco dCloud SJC Platform "
```

## 3. Connect to the control node

```
ssh cisco@ubuntu
```

```
ssh cisco@ubuntu
cisco@ubuntu's password:
Welcome to Ubuntu 14.04.5 LTS (GNU/Linux 4.4.0-42-generic x86_64)

* Documentation:  https://help.ubuntu.com/

460 packages can be updated.
379 updates are security updates.

New release '16.04.7 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Your Hardware Enablement Stack (HWE) is supported until April 2019.
Last login: Sun Sep 13 16:35:00 2020 from 10.16.50.193
Which Virtual Environment would you like to activate?
2. Python2
3. Python3
>>> 2

Activating Virtual Environment for Python 2

cisco@ubuntu:~/CiscoDevNet/code/dci$
```

## 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 ]-----+
|           ==00       |
|        +. =. 0       |
|       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
```

```
cisco@ubuntu: ~
cisco@ubuntu:~$ ssh-copy-id root@centos1.dcloud.cisco.com
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any th
at are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it
is to install the new keys
root@centos1.dcloud.cisco.com's password:

Number of key(s) added: 1

Now try logging into the machine, with:  "ssh 'root@centos1.dcloud.cisco.com'"
and check to make sure that only the key(s) you wanted were added.

cisco@ubuntu:~$ ssh-copy-id root@centos2.dcloud.cisco.com
The authenticity of host 'centos2.dcloud.cisco.com (198.18.134.50)' can't be established.
ECDSA key fingerprint is 85:f3:cc:8f:38:c7:e5:2d:e7:e8:ad:a7:3a:76:27:82.
Are you sure you want to continue connecting (yes/no)? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any th
at are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it
is to install the new keys
root@centos2.dcloud.cisco.com's password:

Number of key(s) added: 1

Now try logging into the machine, with:  "ssh 'root@centos2.dcloud.cisco.com'"
and check to make sure that only the key(s) you wanted were added.

cisco@ubuntu:~$
```

### 3. Verify that the keys have been copied

```
root@centos1:~ (ssh)
ssh root@centos1
root@centos1's password:
Last login: Sun Sep 13 22:09:35 2020 from 10.16.50.193
[root@centos1 ~]# ls -al .ssh/
total 16
drwx----- 2 root root 4096 Sep 13 22:03 .
dr-xr-x--- 5 root root 4096 Jun 26 2018 ..
-rw----- 1 root root 738 Sep 13 22:03 authorized_keys
-rw-r--r-- 1 root root 611 Aug 5 2016 known_hosts
[root@centos1 ~]#

root@centos2:~ (ssh)
ssh root@centos2
root@centos2's password:
Last login: Sun Sep 13 22:09:35 2020 from 10.16.50.193
[root@centos2 ~]# ls -al .ssh/
total 16
drwx----- 2 root root 4096 Sep 13 22:05 .
dr-xr-x--- 5 root root 4096 Jun 26 2018 ..
-rw----- 1 root root 738 Sep 13 22:05 authorized_keys
-rw-r--r-- 1 root root 611 Aug 5 2016 known_hosts
[root@centos2 ~]#
```

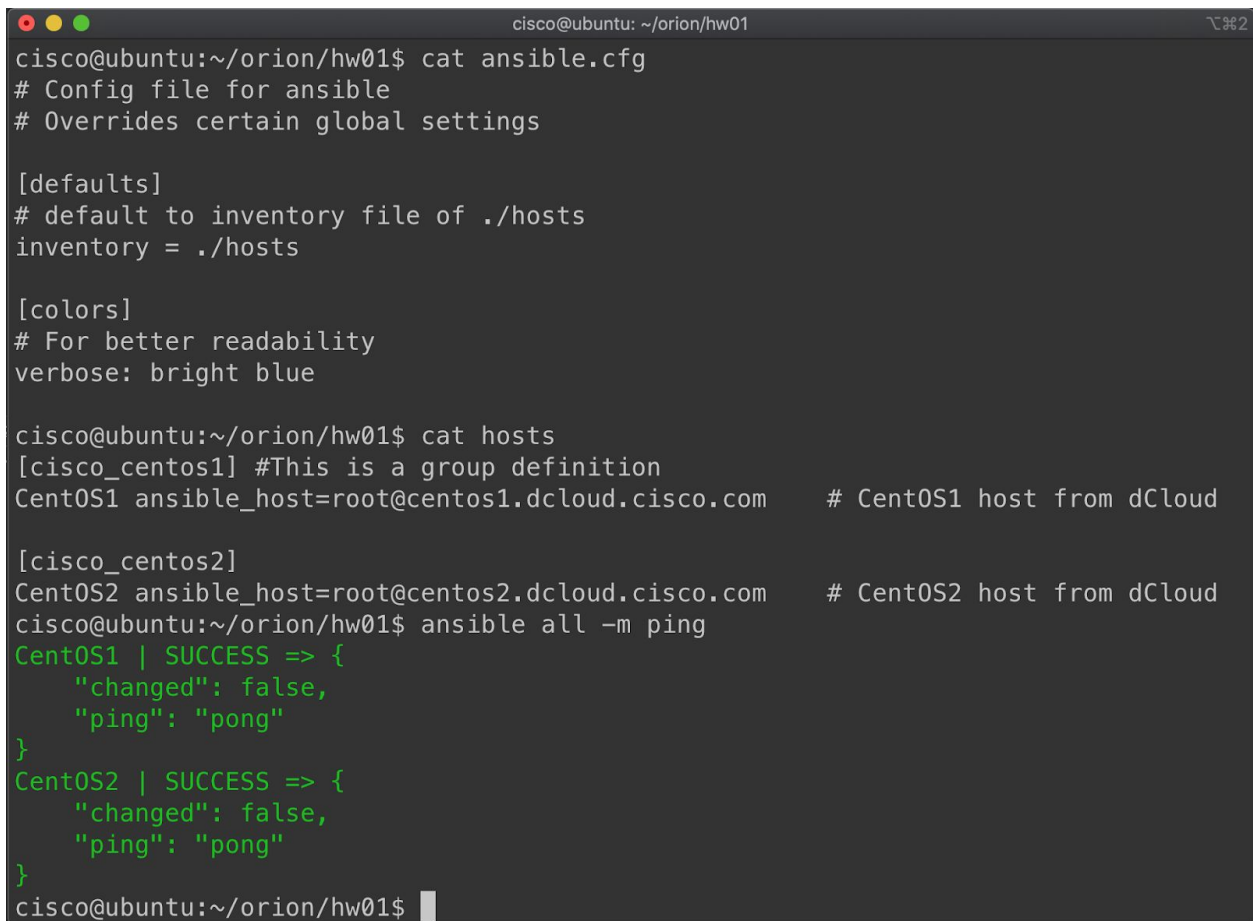


## Create Ansible inventory

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.

A terminal window titled 'cisco@ubuntu: ~/orion/hw01' showing the configuration of an Ansible environment. The user first displays the contents of 'ansible.cfg', which sets the default inventory file to './hosts' and enables verbose output. Then, the user displays the 'hosts' file, which defines two hosts: 'CentOS1' and 'CentOS2', both pointing to root@centos1.dcloud.cisco.com and root@centos2.dcloud.cisco.com respectively. Finally, the user runs 'ansible all -m ping', which results in green output indicating successful connections to both hosts, with 'changed' status as false and 'ping' response as 'pong'.

```
cisco@ubuntu: ~/orion/hw01
cisco@ubuntu:~/orion/hw01$ cat ansible.cfg
# Config file for ansible
# Overrides certain global settings

[defaults]
# default to inventory file of ./hosts
inventory = ./hosts

[colors]
# For better readability
verbose: bright blue

cisco@ubuntu:~/orion/hw01$ cat hosts
[cisco_centos1] #This is a group definition
CentOS1 ansible_host=root@centos1.dcloud.cisco.com    # CentOS1 host from dCloud

[cisco_centos2]
CentOS2 ansible_host=root@centos2.dcloud.cisco.com    # CentOS2 host from dCloud
cisco@ubuntu:~/orion/hw01$ ansible all -m ping
CentOS1 | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
CentOS2 | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
cisco@ubuntu:~/orion/hw01$
```



## 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]

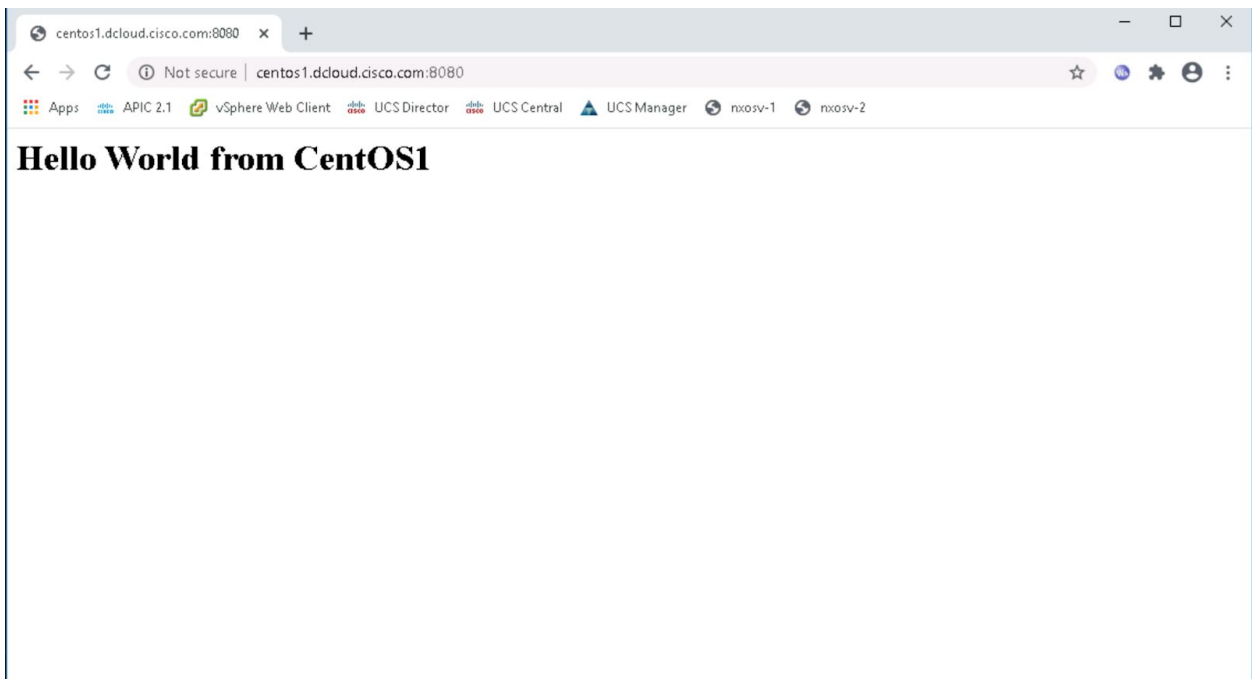
PLAY RECAP *****
CentOS1      : ok=8    changed=6    unreachable=0    failed=0
CentOS2      : ok=8    changed=6    unreachable=0    failed=0

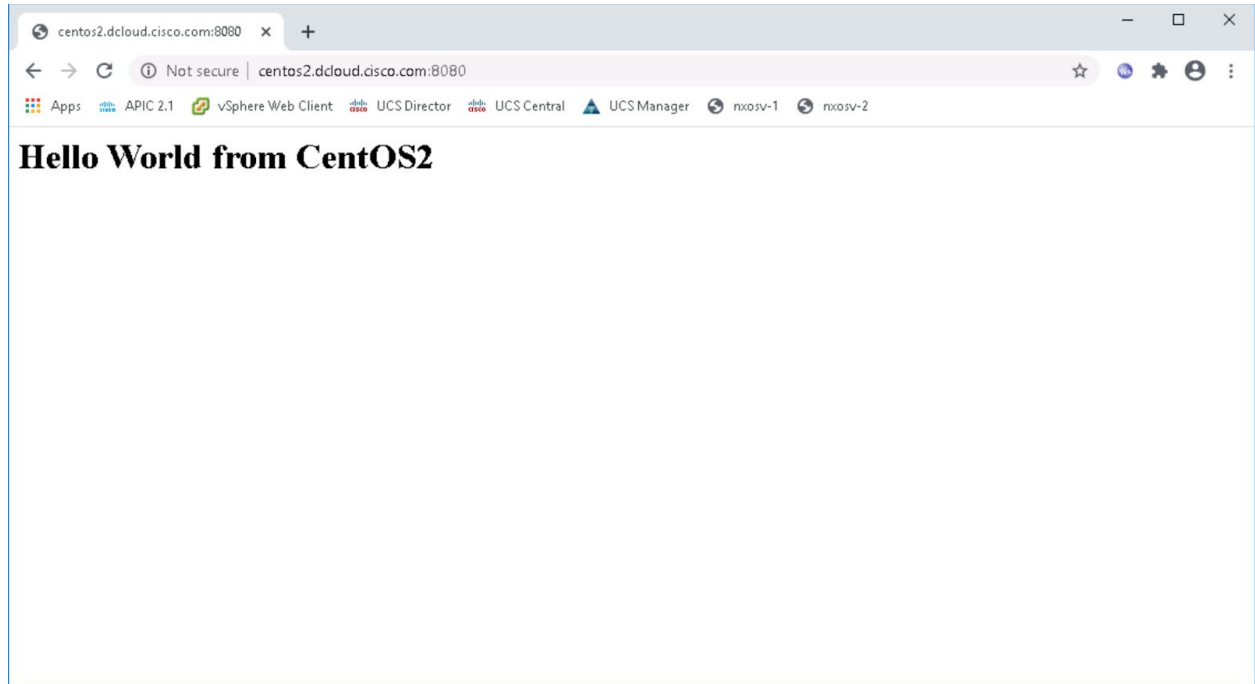
cisco@ubuntu:~/orion/hw01$
```

## 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
```

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

PLAY [Undeploy website on port 8080] *****

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

TASK [stop nginx] *****
changed: [CentOS1]
changed: [CentOS2]

TASK [Remove index.html] *****
changed: [CentOS2]
changed: [CentOS1]

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

TASK [uninstall nginx] *****
changed: [CentOS1]
changed: [CentOS2]

TASK [uninstall epel-release] *****
changed: [CentOS1]
changed: [CentOS2]

TASK [disable port 8080] *****
changed: [CentOS1]
changed: [CentOS2]

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

PLAY RECAP *****
CentOS1      : ok=8    changed=7    unreachable=0    failed=0
CentOS2      : ok=8    changed=7    unreachable=0    failed=0

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

## Validate after undeployment

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

```
cisco@ubuntu: ~/orion/hw01
cisco@ubuntu:~/orion/hw01$ curl centos1.dcloud.cisco.com:8080
curl: (7) Failed to connect to centos1.dcloud.cisco.com port 8080: No route to host
cisco@ubuntu:~/orion/hw01$ curl centos2.dcloud.cisco.com:8080
curl: (7) Failed to connect to centos2.dcloud.cisco.com port 8080: No route to host
cisco@ubuntu:~/orion/hw01$
```