HW #1 - Ansible

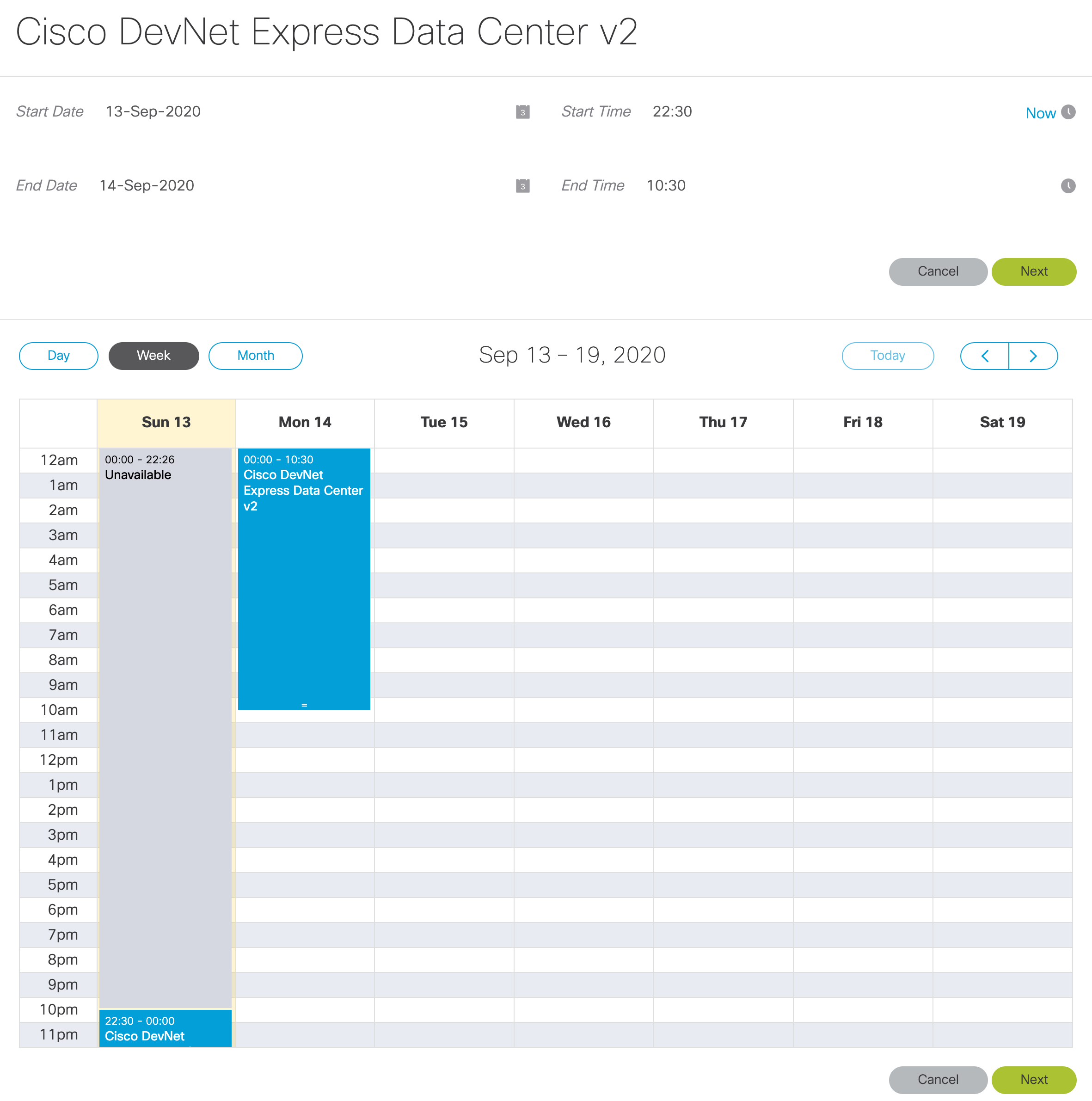
Team: Orion

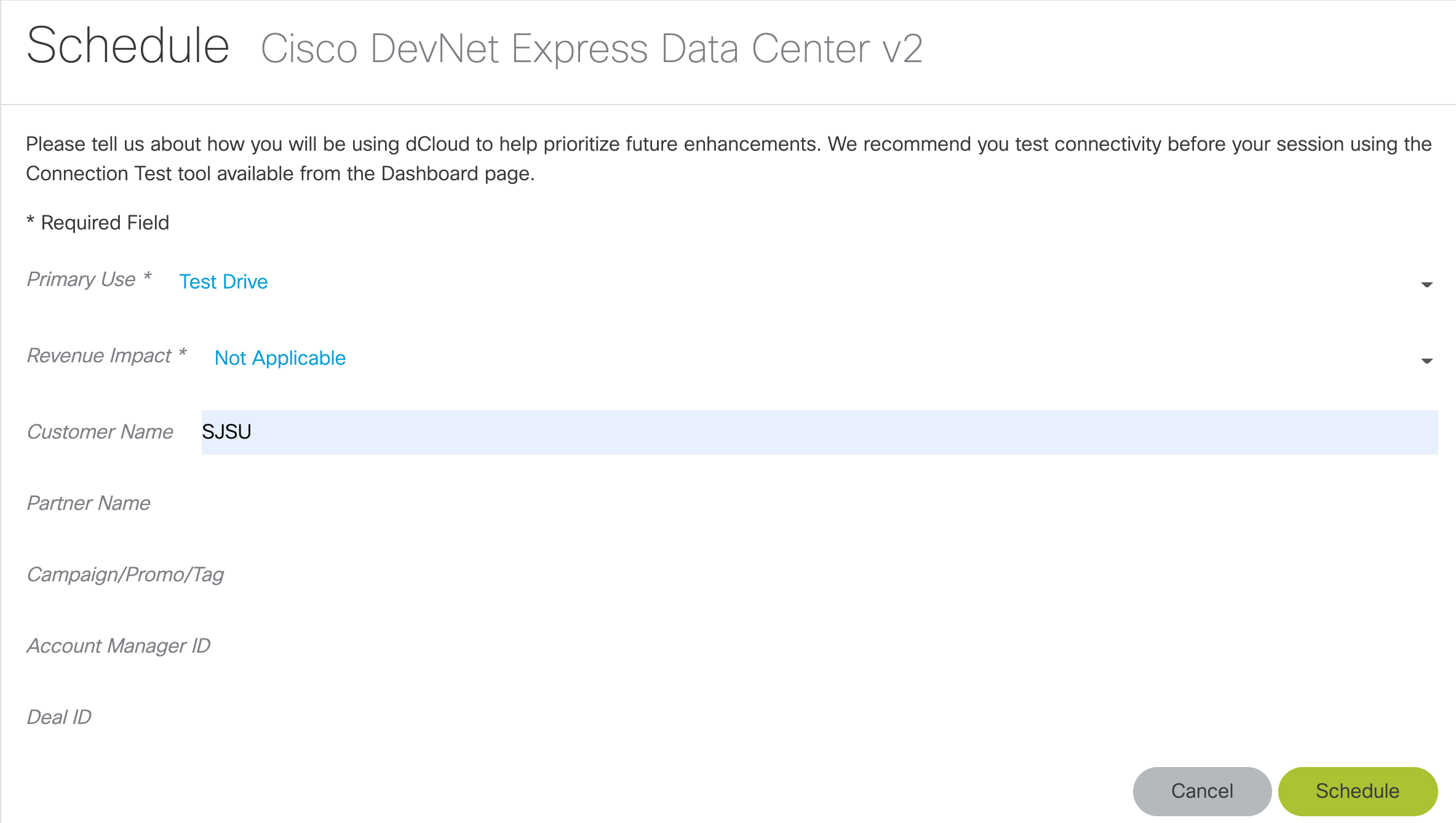
* Babu Rajendran
* Eric Cheng
* Archana Shokeen

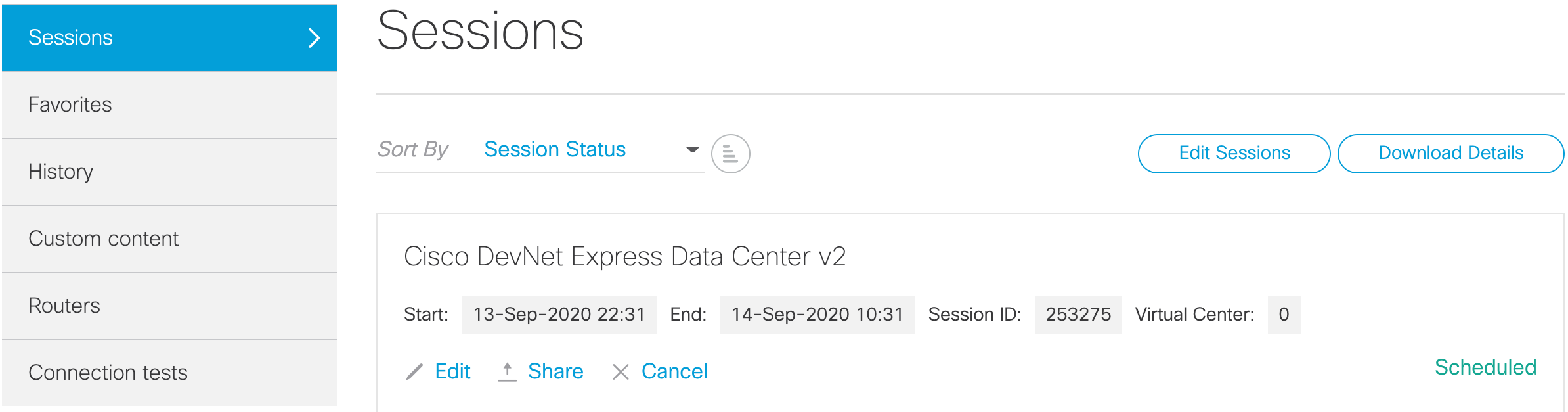
GitHub: <https://github.com/sjsu-cmpe272/orion-hw01-ansible>

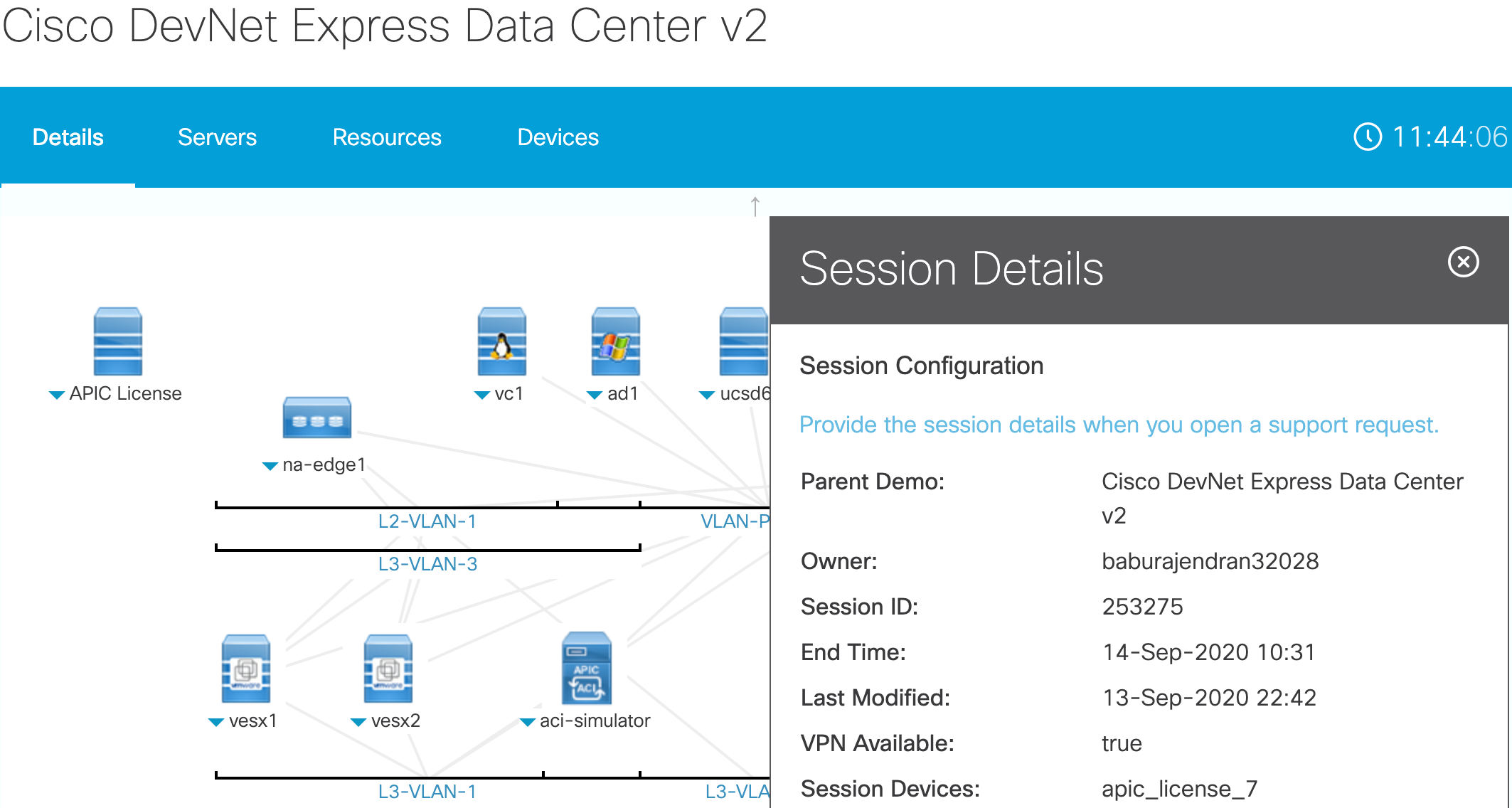
* In the DCloud lab,  
  <https://dcloud2-sjc.cisco.com/content/demo/90426?returnPathTitleKey=content-view>
* Configure Ansible server on ***ubuntu*** to deploy a webserver on ***centos1***and ***centos2***, and bring it up on port 8080 with a web page that is accessible on from the Windows 10 Workstation, **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





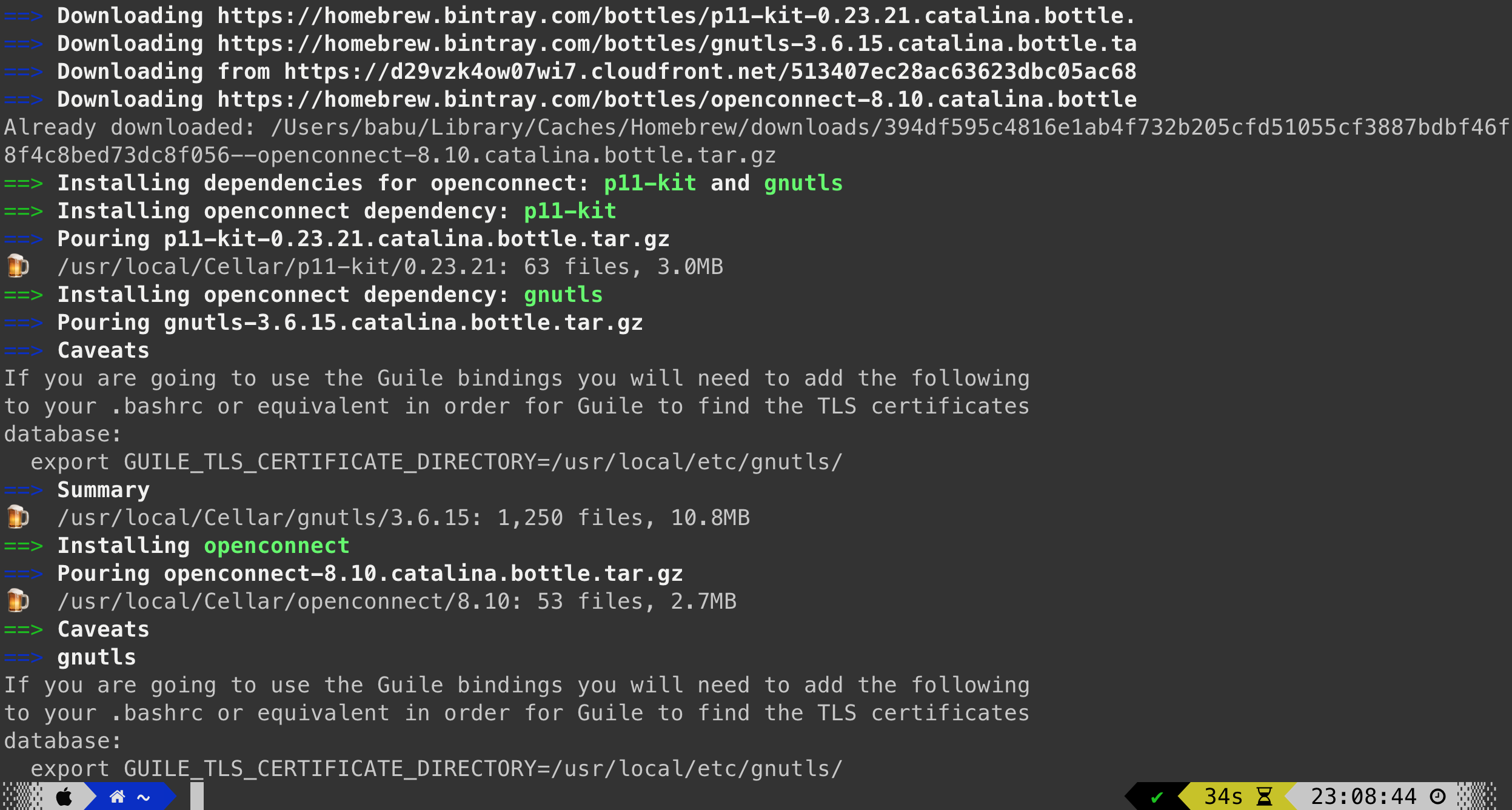
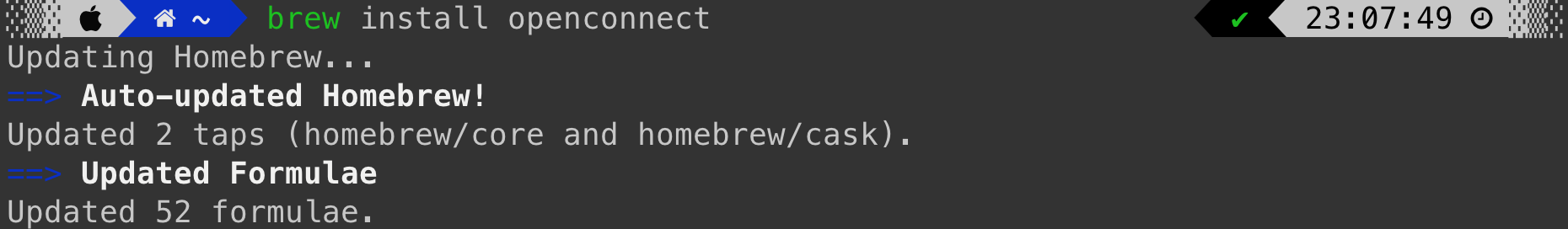




# Connect to dCloud

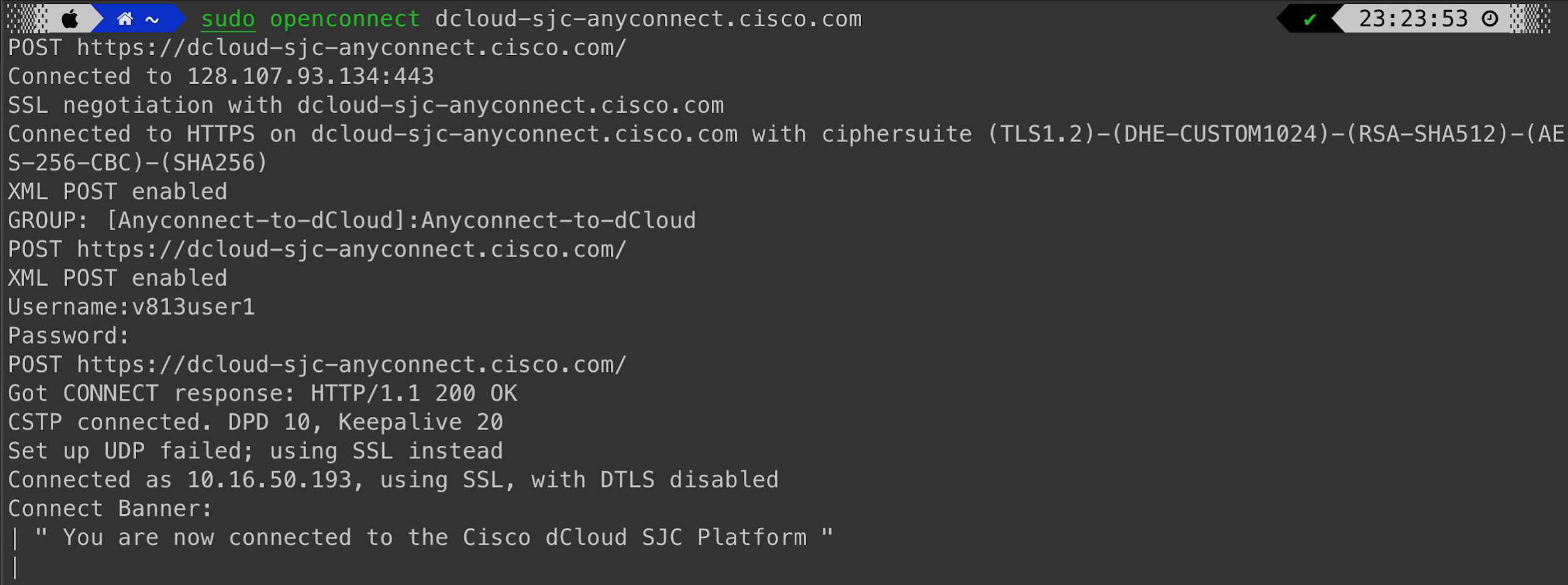
###### Install a VPN client

|  |
| --- |
| brew install openconnect |



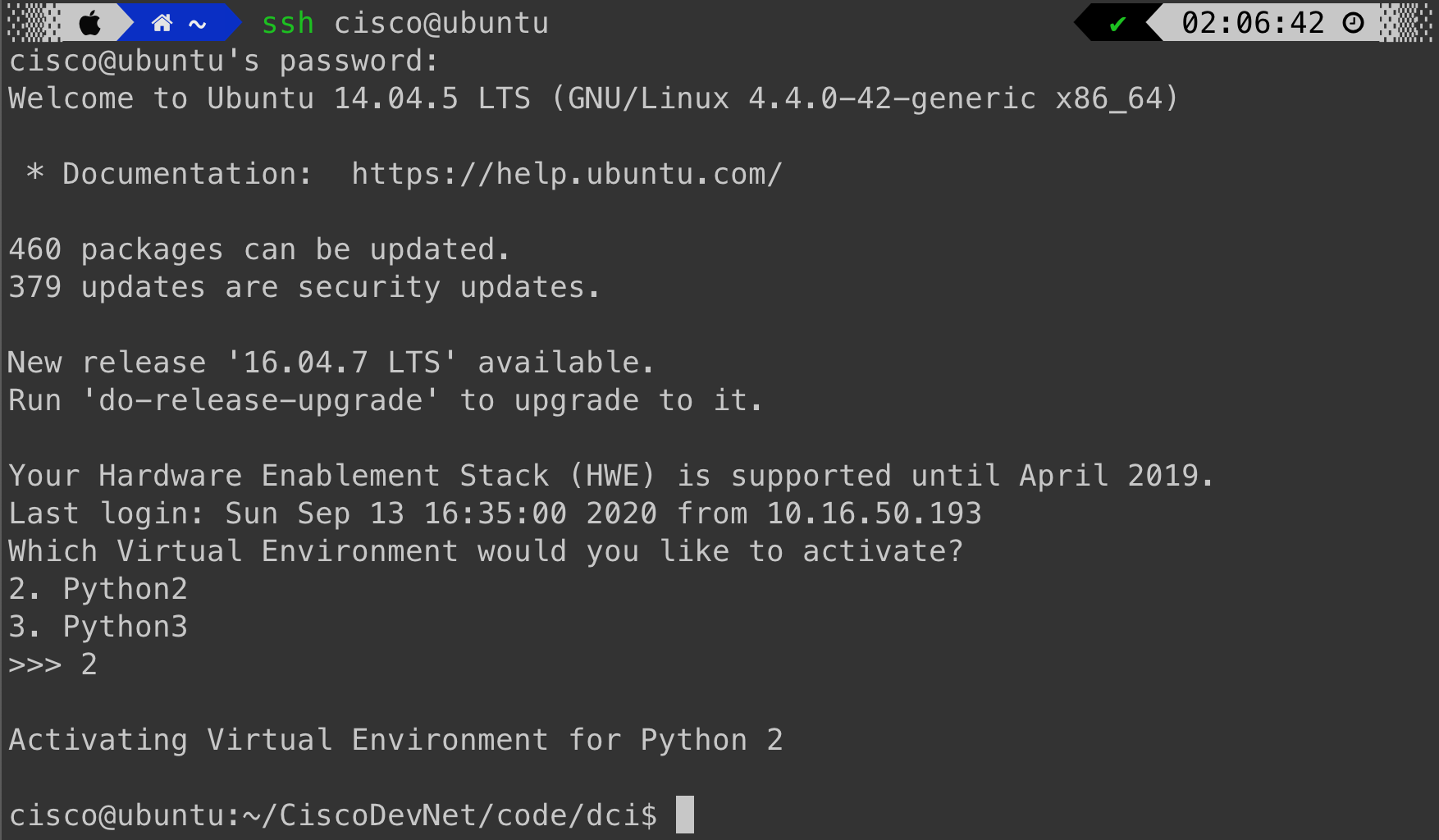
###### Establish VPN connection

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



###### 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 |

# 

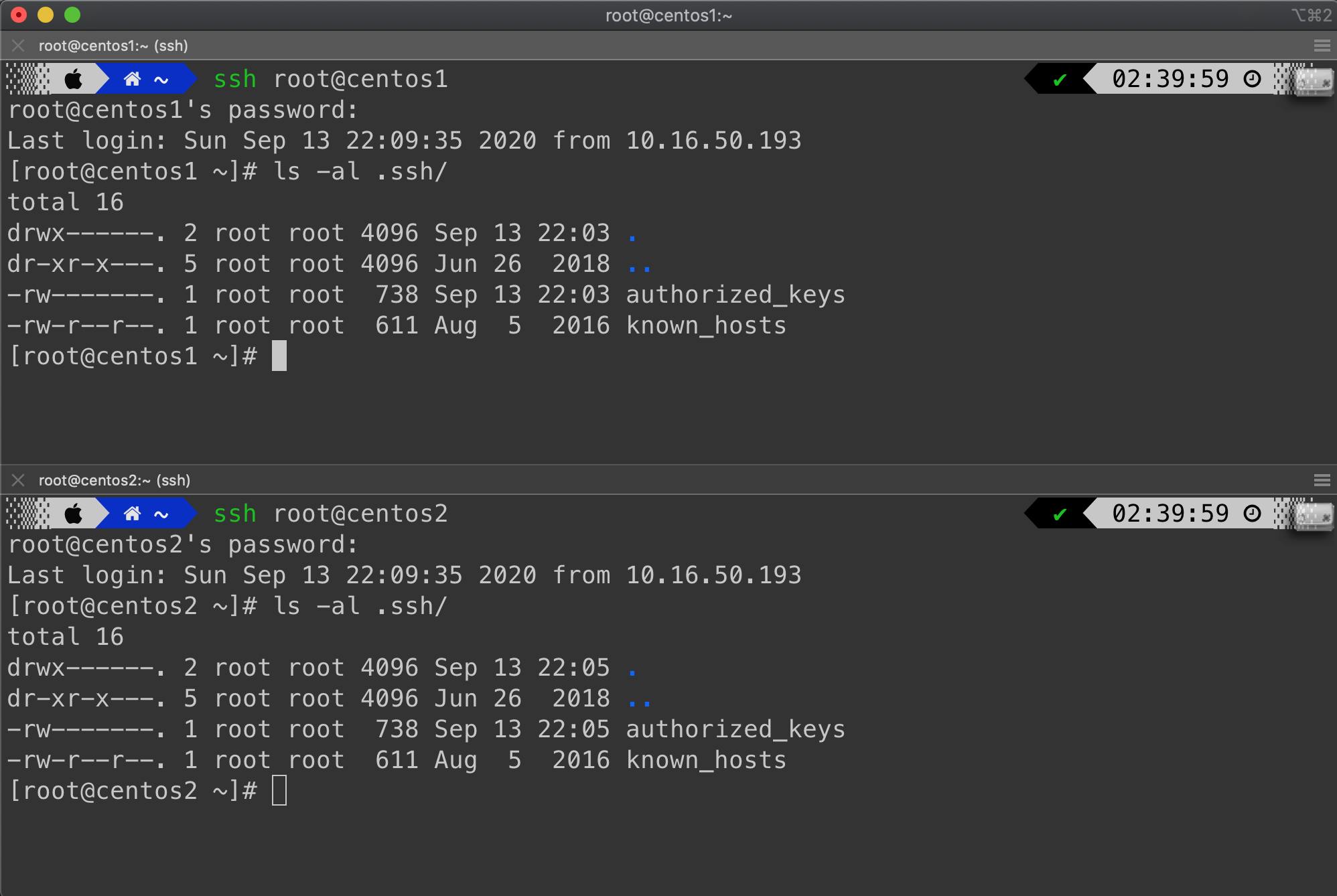
###### Copy the keys to centos hosts

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

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

# 

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

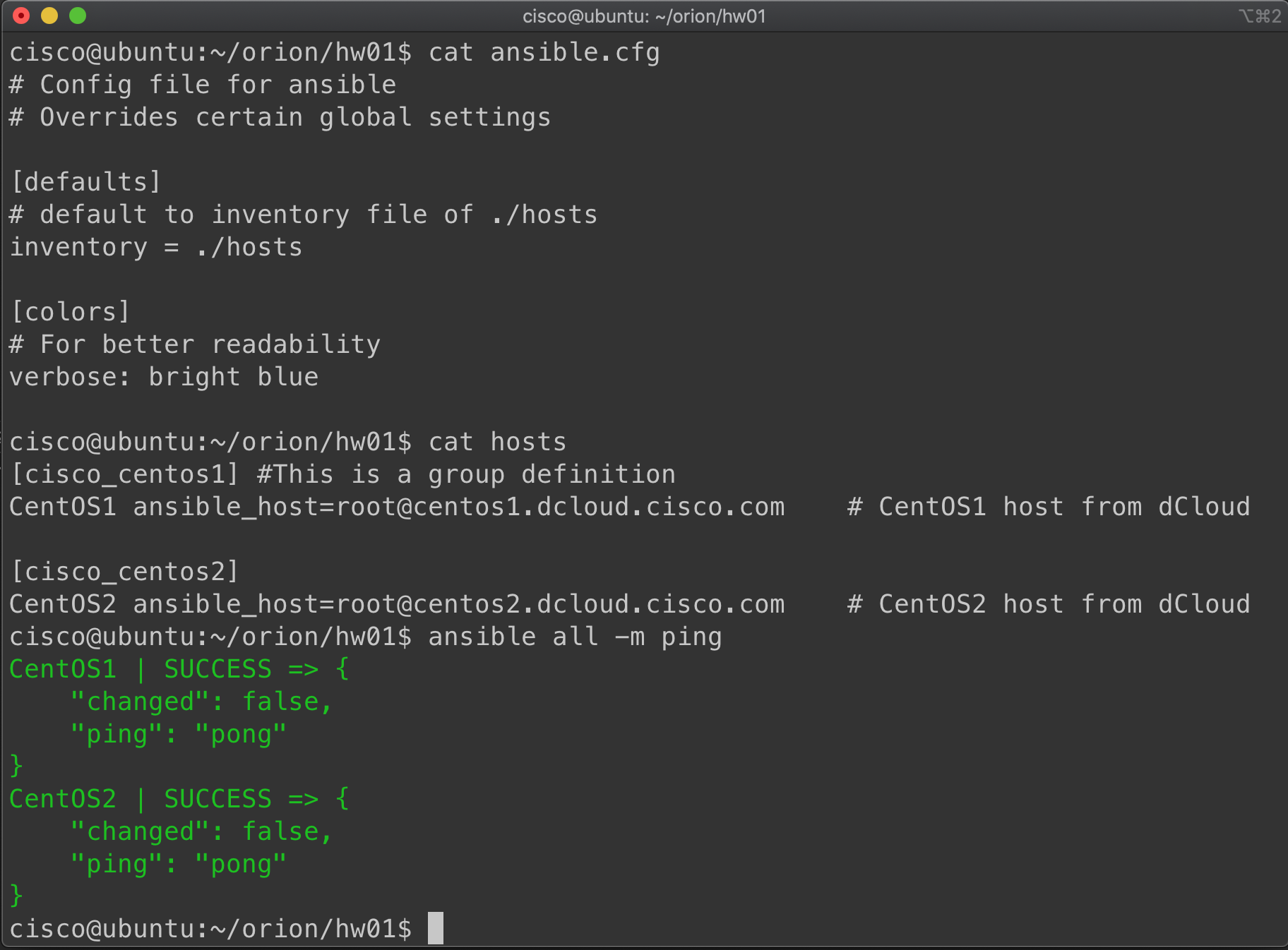
****

# 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 |

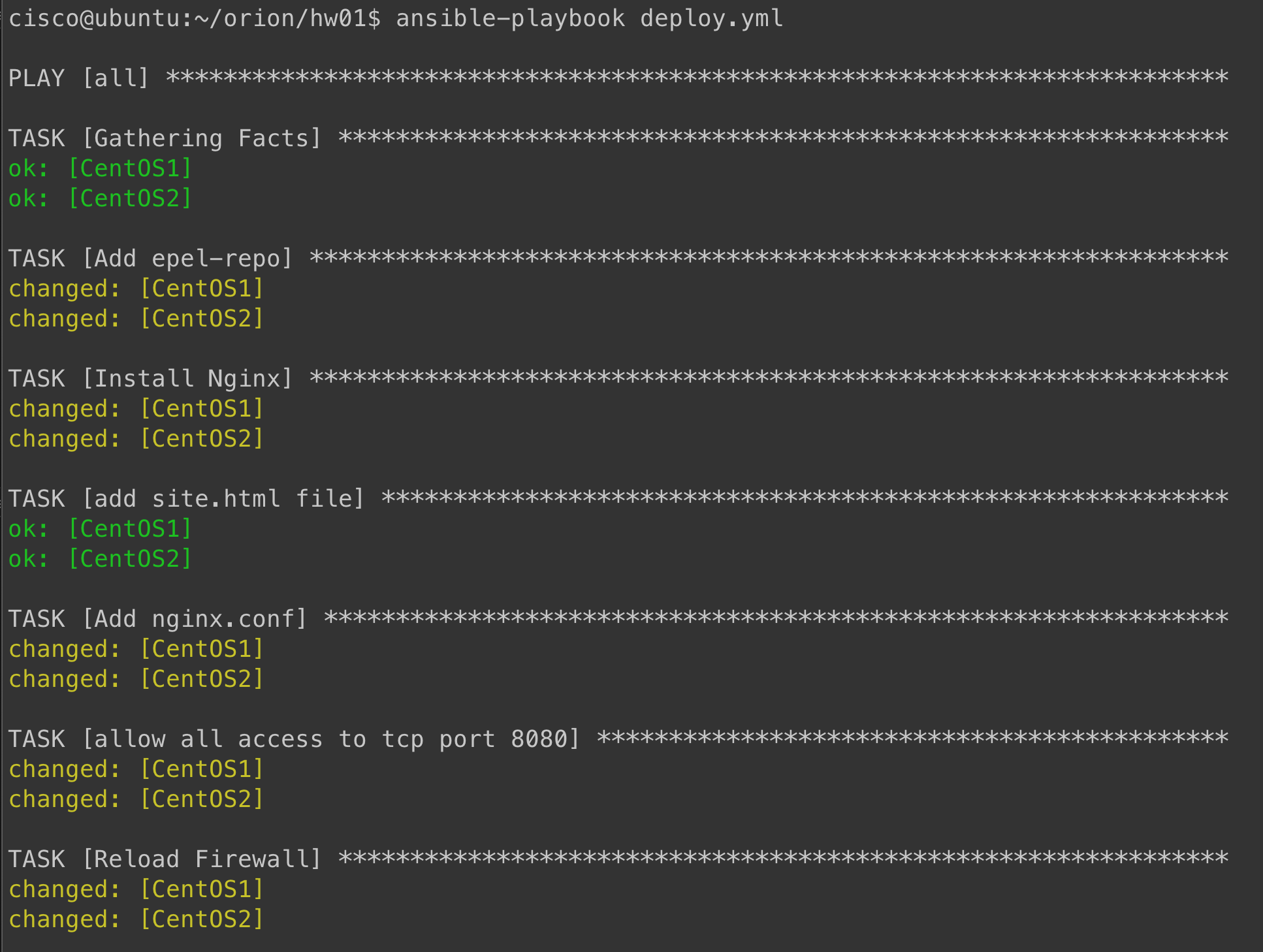
# Correctly setting it up will generate the green success outputs as seen below.

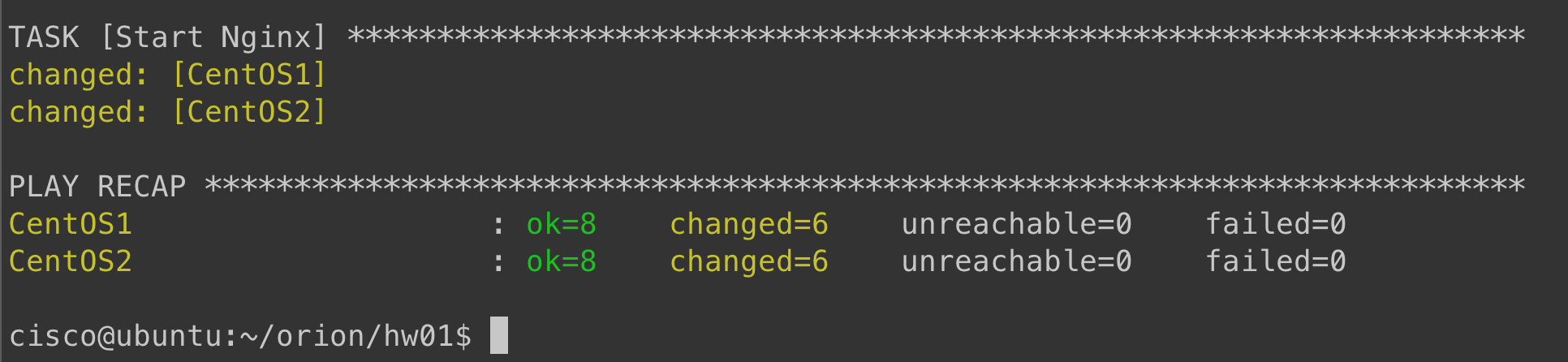


# Deploy the web server

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:

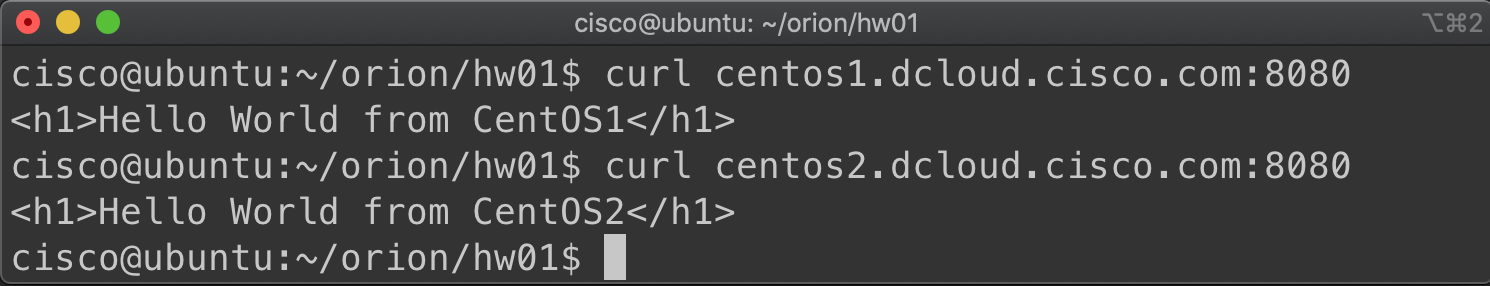
|  |
| --- |
| ansible-playbook deploy.yml |

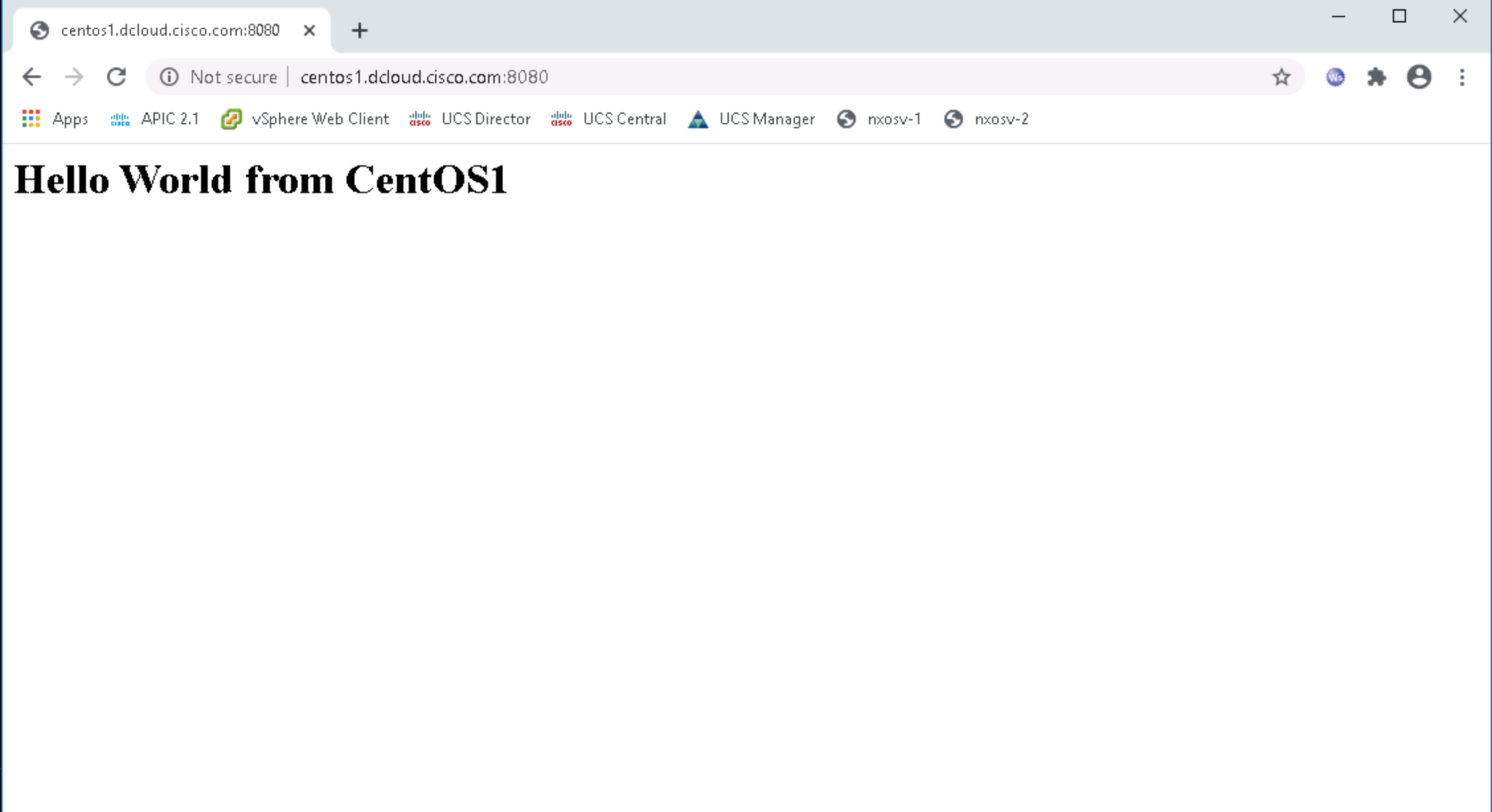


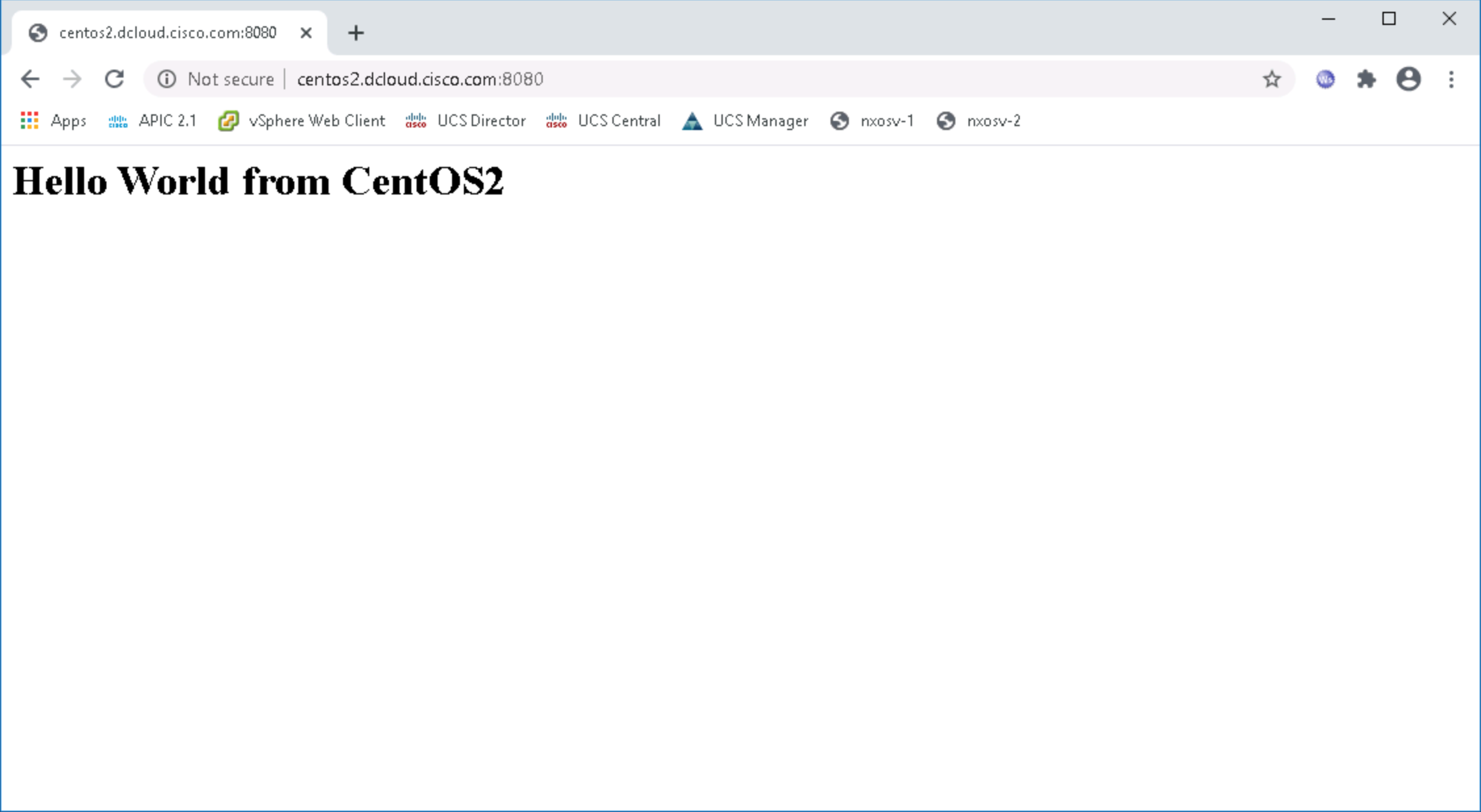


# Validation from Windows workstation

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.



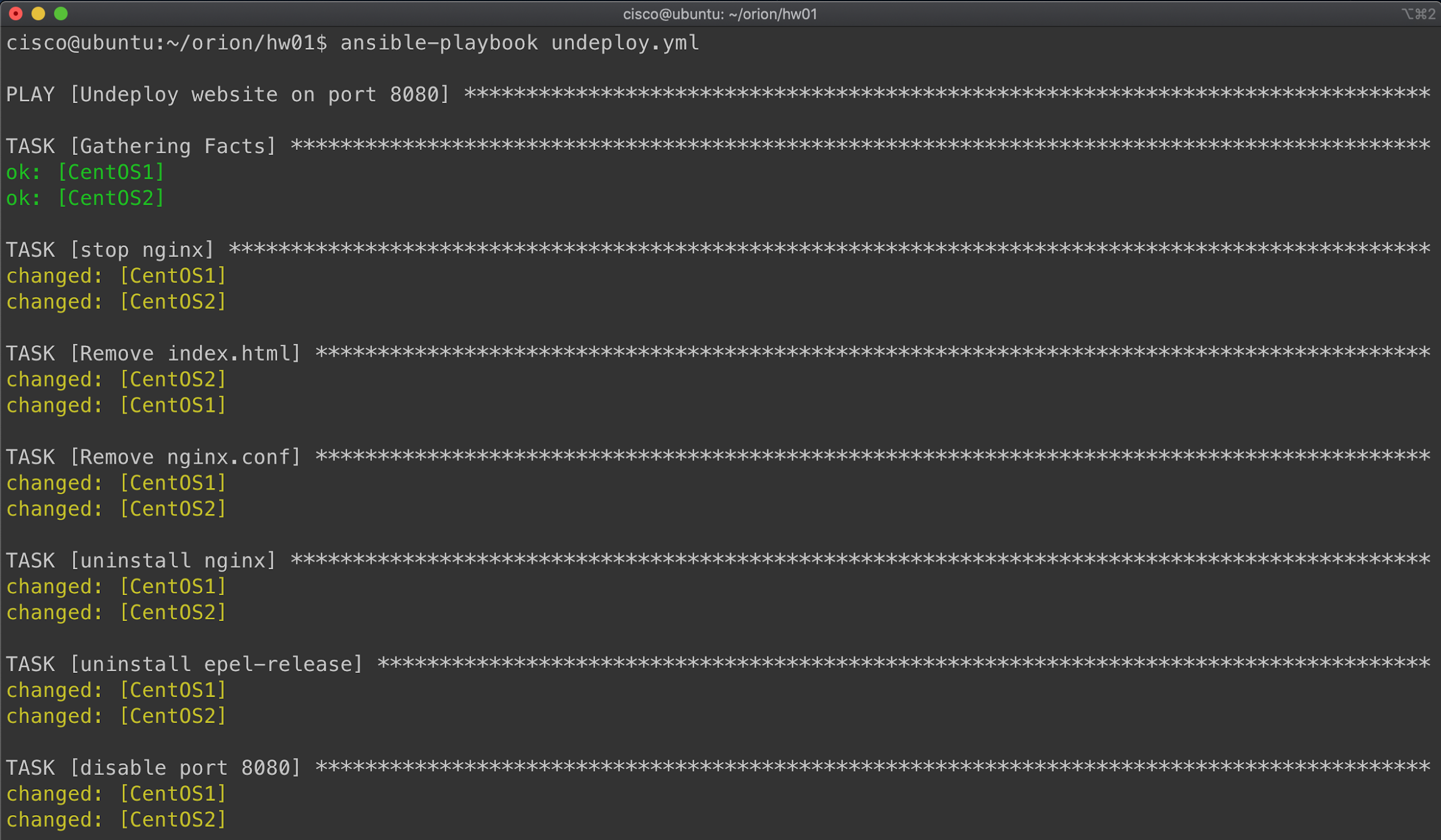


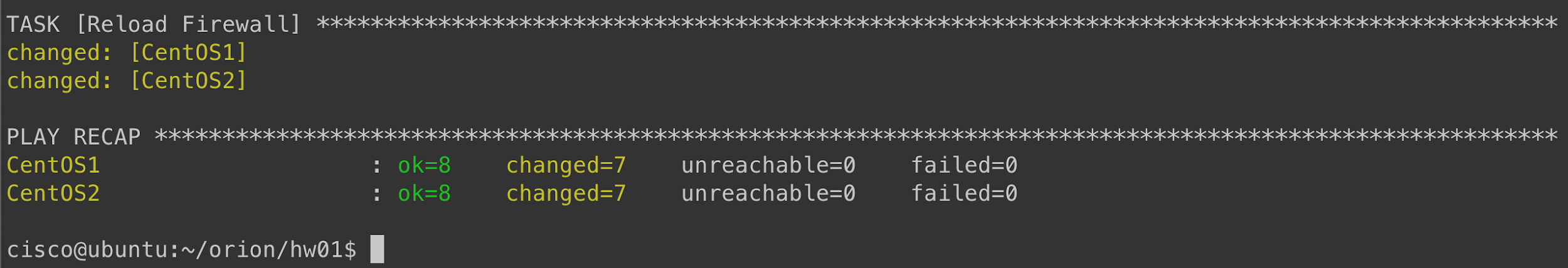


# Undeploy the web server

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.

