Network Automation using Ansible

Version 1.0

July 2018

Contents

Environment setup.. 3

Network Automation using Ansible raw module.. 8

Automation Cisco Nexus devices using Ansible.. 11

Perform Clean up activity on nexus device.. 15

Automating Cisco CSR1000v using Ansible.. 17

Automation Arista EOS using Ansible 21

Perform Cleanup activity on the EOS Device using Ansible.. 26

Appendix A.. 30

Running Ansible environment setup script.. 30

Note: We will be using Ansible version 2.4.2.0 for this LAB due to some existing bugs on network_cli module. We will be leveraging one of the Managed Nodes VM to install the Ansible 2.4

We are not going to install Ansible as package instead we will be cloning the Ansible github repository and switching out to the desired version.

Environment setup

1. Login as root use on ssh root@podx-node1.origin.com
2. Execute the following command to Install Python-pip

```
# subscription-manager repos --enable rhel-server-rhscl-7-rpms

# yum install python27-python-pip -y

# scl enable python27 bash
```

Note: Since our LAB RHEL subscription is expired, we cannot use the Yum to install the python-pip, we can use a workaround to install the python pip

Please follow the steps from this link – https://linuxconfig.org/installation-of-pip-the-python-packaging-tool-on-rhel-7-linux

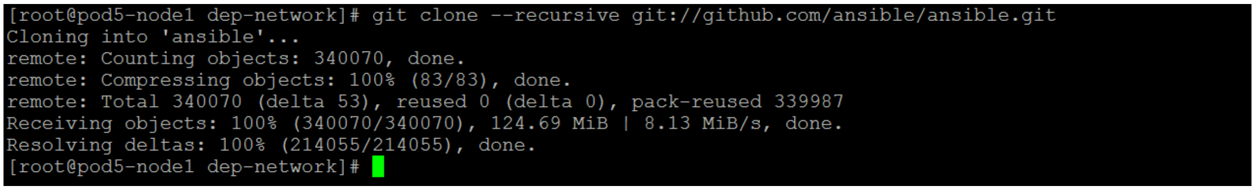
For the actual Training we will use the above Yum install method to install the packages

3. Create a directory for setting up Ansible

```
mkdir ~/lab11-dep-network && cd ~/lab11-dep-network
```

4. Clone the latest ansible repo from GitHub

```
git clone --recursive git://github.com/ansible/ansible.git
```



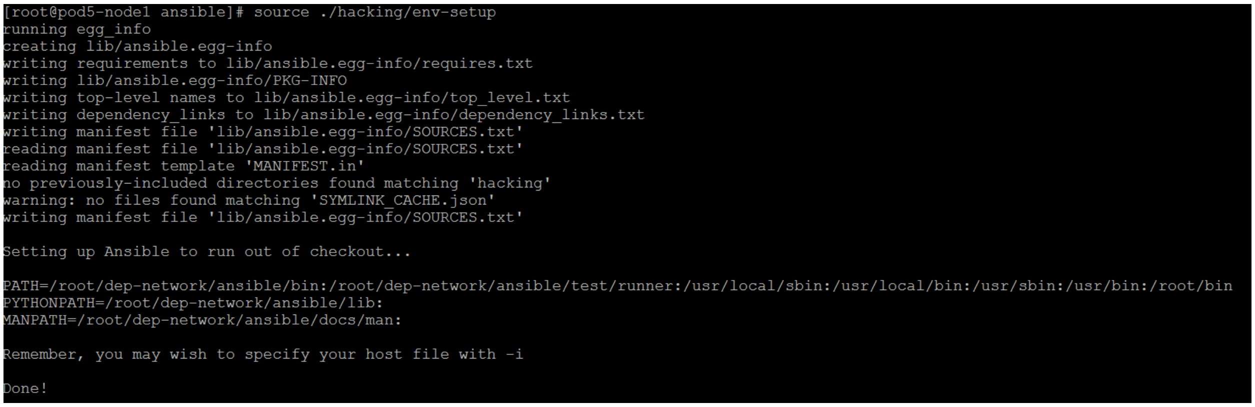
```
[root@pod5-node1 dep-network]# git clone --recursive git://github.com/ansible/ansible.git
Cloning into 'ansible'...
remote: Counting objects: 340070, done.
remote: Compressing objects: 100% (83/83), done.
remote: Total 340070 (delta 53), reused 0 (delta 0), pack-reused 339987
Receiving objects: 100% (340070/340070), 124.69 MiB | 8.13 MiB/s, done.
Resolving deltas: 100% (214055/214055), done.
[root@pod5-node1 dep-network]# 
```

5. Cd to ansible directory

```
cd ./ansible
```

6. Execute the env-setup script

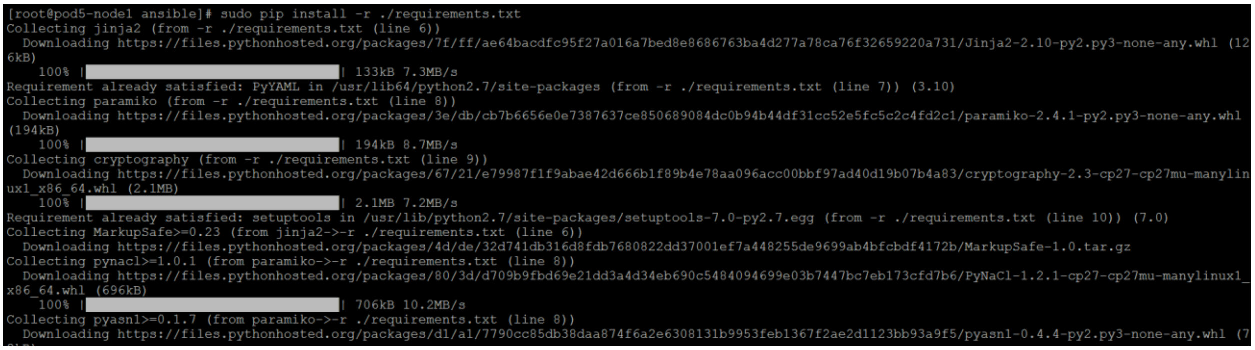
```
source ./hacking/env-setup
```



```
[root@pod5-node1 ansible]# source ./hacking/env-setup
running egg_info
creating lib/ansible.egg-info
writing requirements to lib/ansible.egg-info/requires.txt
writing lib/ansible.egg-info/PKG-INFO
writing top-level names to lib/ansible.egg-info/top_level.txt
writing dependency_links to lib/ansible.egg-info/dependency_links.txt
writing manifest file 'lib/ansible.egg-info/SOURCES.txt'
reading manifest file 'lib/ansible.egg-info/SOURCES.txt'
reading manifest template 'MANIFEST.in'
no previously-included directories found matching 'hacking'
warning: no files found matching 'SYMLINK_CACHE.json'
writing manifest file 'lib/ansible.egg-info/SOURCES.txt'

Setting up Ansible to run out of checkout...

PATH=/root/dep-network/ansible/bin:/root/dep-network/ansible/test/runner:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/root/bin
PYTHONPATH=/root/dep-network/ansible/lib:
MANPATH=/root/dep-network/ansible/docs/man:

Remember, you may wish to specify your host file with -i

Done!
```

7. Install the Requirements

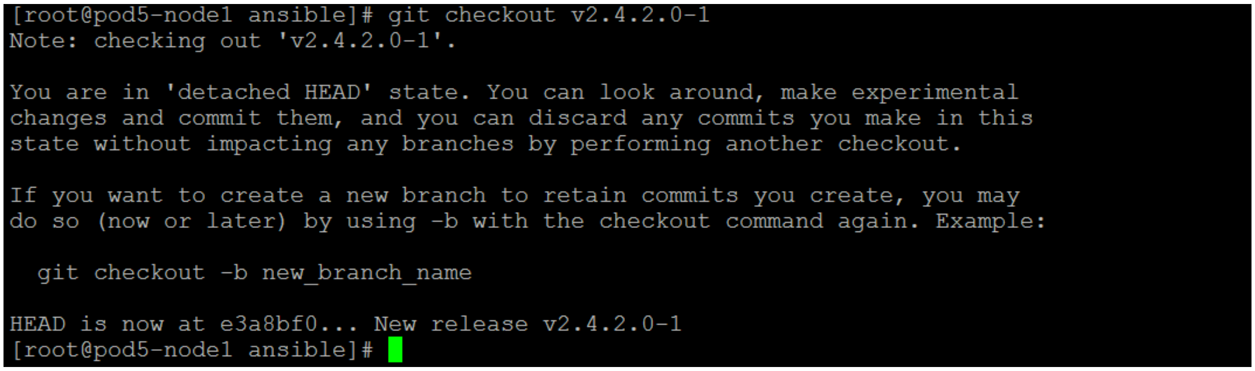
```
sudo pip install -r ./requirements.txt
```



```
[root@pod5-node1 ansible]# sudo pip install -r ./requirements.txt
Collecting jinja2 (from -r ./requirements.txt (line 6))
  Downloading https://files.pythonhosted.org/packages/7f/ff/ae64bacdfc95f27a016a7bed8e8686763ba4d277a78ca76f32659220a731/Jinja2-2.10-py2.py3-none-any.whl (12
6kB)
    100% |████████████████████████████████| 133kB 7.3MB/s
Requirement already satisfied: PyYAML in /usr/lib64/python2.7/site-packages (from -r ./requirements.txt (line 7)) (3.10)
Collecting paramiko (from -r ./requirements.txt (line 8))
  Downloading https://files.pythonhosted.org/packages/3e/db/cb7b6656e0e7387637ce850689084dc0b94b44df31cc52e5fc5c2c4fd2c1/paramiko-2.4.1-py2.py3-none-any.whl
(194kB)
    100% |████████████████████████████████| 194kB 8.7MB/s
Collecting cryptography (from -r ./requirements.txt (line 9))
  Downloading https://files.pythonhosted.org/packages/67/21/e79987f1f9abae42d666b1f89b4e78aa096acc00bbf97ad40d19b07b4a83/cryptography-2.3-cp27-cp27mu-manylin
ux1_x86_64.whl (2.1MB)
    100% |████████████████████████████████| 2.1MB 7.2MB/s
Requirement already satisfied: setuptools in /usr/lib/python2.7/site-packages/setuptools-7.0-py2.7.egg (from -r ./requirements.txt (line 10)) (7.0)
Collecting MarkupSafe>=0.23 (from jinja2->-r ./requirements.txt (line 6))
  Downloading https://files.pythonhosted.org/packages/4d/de/32d741db316d8fdb7680822dd37001ef7a448255de9699ab4bfcbdf4172b/MarkupSafe-1.0.tar.gz
Collecting pynacl>=1.0.1 (from paramiko->-r ./requirements.txt (line 8))
  Downloading https://files.pythonhosted.org/packages/80/3d/d709b9fbd69e21dd3a4d34eb690c5484094699e03b7447bc7eb173cfd7b6/PyNaCl-1.2.1-cp27-cp27mu-manylinux1_
x86_64.whl (696kB)
    100% |████████████████████████████████| 706kB 10.2MB/s
Collecting pyasn1>=0.1.7 (from paramiko->-r ./requirements.txt (line 8))
  Downloading https://files.pythonhosted.org/packages/d1/a1/7790cc85db38daa874f6a2e6308131b9953feb1367f2ae2d1123bb93a9f5/pyasn1-0.4.4-py2.py3-none-any.whl (7
```

8. Use Git tag to Checkout to the branch v2.4.2.0-1

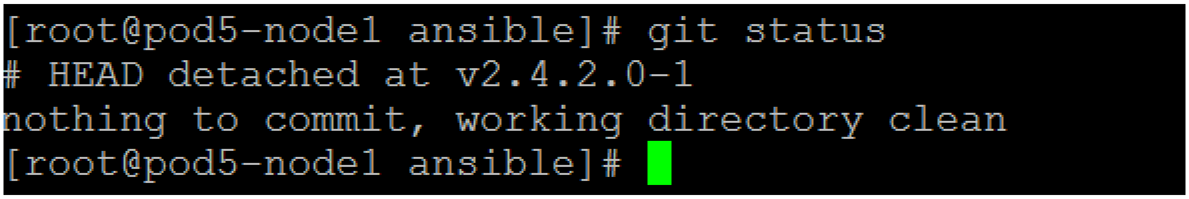
```
git checkout v2.4.2.0-1
```



```
[root@pod5-node1 ansible]# git checkout v2.4.2.0-1
Note: checking out 'v2.4.2.0-1'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by performing another checkout.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -b with the checkout command again. Example:

  git checkout -b new_branch_name

HEAD is now at e3a8bf0... New release v2.4.2.0-1
[root@pod5-node1 ansible]# 
```

9. Use git status command to ensure you in the correct version

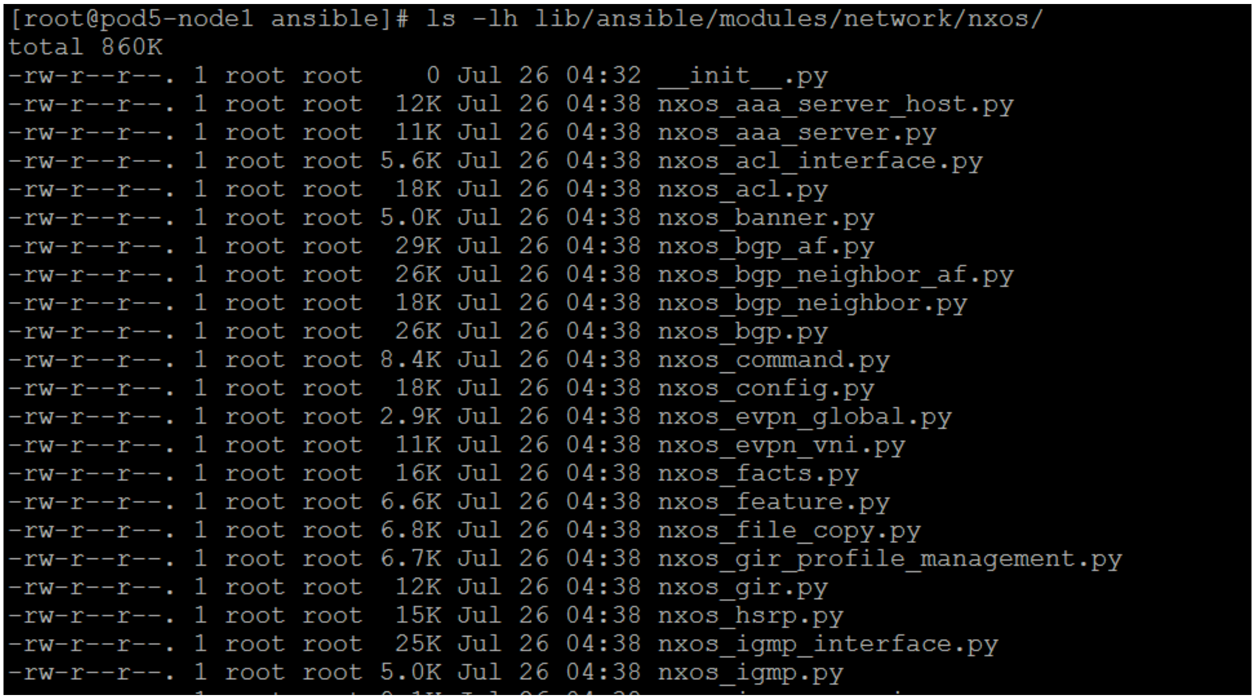
```
git status
```



```
[root@pod5-node1 ansible]# git status
# HEAD detached at v2.4.2.0-1
nothing to commit, working directory clean
[root@pod5-node1 ansible]# 
```

10. List available Ansible NXOS modules.

```
ls -lh lib/ansible/modules/network/nxos/
```



```
[root@pod5-node1 ansible]# ls -lh lib/ansible/modules/network/nxos/
total 860K
-rw-r--r--. 1 root root    0 Jul 26 04:32 __init__.py
-rw-r--r--. 1 root root  12K Jul 26 04:38 nxos_aaa_server_host.py
-rw-r--r--. 1 root root  11K Jul 26 04:38 nxos_aaa_server.py
-rw-r--r--. 1 root root 5.6K Jul 26 04:38 nxos_acl_interface.py
-rw-r--r--. 1 root root  18K Jul 26 04:38 nxos_acl.py
-rw-r--r--. 1 root root 5.0K Jul 26 04:38 nxos_banner.py
-rw-r--r--. 1 root root  29K Jul 26 04:38 nxos_bgp_af.py
-rw-r--r--. 1 root root  26K Jul 26 04:38 nxos_bgp_neighbor_af.py
-rw-r--r--. 1 root root  18K Jul 26 04:38 nxos_bgp_neighbor.py
-rw-r--r--. 1 root root  26K Jul 26 04:38 nxos_bgp.py
-rw-r--r--. 1 root root 8.4K Jul 26 04:38 nxos_command.py
-rw-r--r--. 1 root root  18K Jul 26 04:38 nxos_config.py
-rw-r--r--. 1 root root 2.9K Jul 26 04:38 nxos_evpn_global.py
-rw-r--r--. 1 root root  11K Jul 26 04:38 nxos_evpn_vni.py
-rw-r--r--. 1 root root  16K Jul 26 04:38 nxos_facts.py
-rw-r--r--. 1 root root 6.6K Jul 26 04:38 nxos_feature.py
-rw-r--r--. 1 root root 6.8K Jul 26 04:38 nxos_file_copy.py
-rw-r--r--. 1 root root 6.7K Jul 26 04:38 nxos_gir_profile_management.py
-rw-r--r--. 1 root root  12K Jul 26 04:38 nxos_gir.py
-rw-r--r--. 1 root root  15K Jul 26 04:38 nxos_hsrp.py
-rw-r--r--. 1 root root  25K Jul 26 04:38 nxos_igmp_interface.py
-rw-r--r--. 1 root root 5.0K Jul 26 04:38 nxos_igmp.py
```

Note: In the Environment setup lab 1 we need to add the POD specific network devices IP address and hostnames to add to the host file → I will let Naushad and Nagaveni to add this

For this test, let’s add the following entries manually in the host file

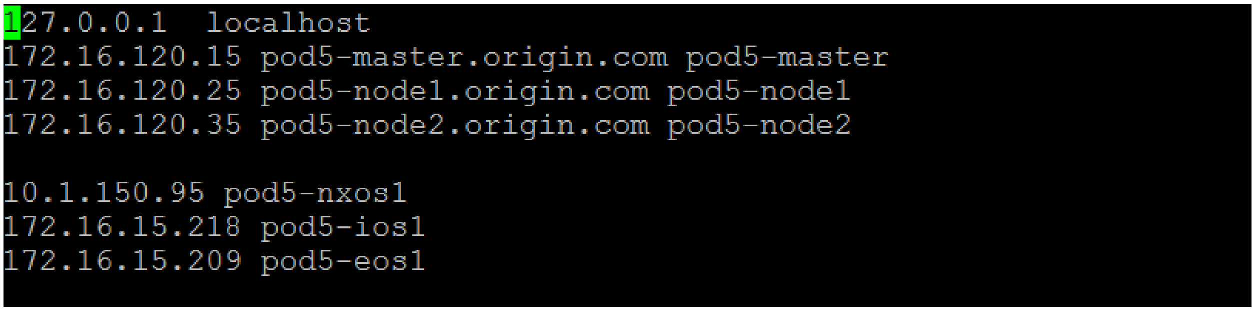
```
cat > /etc/hosts <<EOF

127.0.0.1  localhost

172.16.120.15 pod5-master.origin.com pod5-master

172.16.120.25 pod5-node1.origin.com pod5-node1

172.16.120.35 pod5-node2.origin.com pod5-node2

10.1.150.95 pod5-nxos1

172.16.15.218 pod5-ios1

172.16.15.209 pod5-eos1

EOF
```

POD0 network device

10.1.150.95 pod5-nxos1

172.16.15.218 pod5-ios1

172.16.15.209 pod5-eos1



```
127.0.0.1  localhost
172.16.120.15 pod5-master.origin.com pod5-master
172.16.120.25 pod5-node1.origin.com pod5-node1
172.16.120.35 pod5-node2.origin.com pod5-node2

10.1.150.95 pod5-nxos1
172.16.15.218 pod5-ios1
172.16.15.209 pod5-eos1
```

11. Create the managed nodes inventory file for the network devices as follows

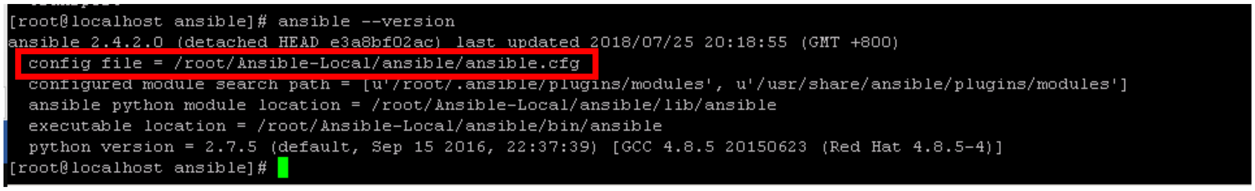
```
cat >> ./inventory <<EOF
[all:vars]

un_nxos = admin
pwd_nxos = #cisco123

un_eos = admin
pwd_eos = !Cisco123

un_ios = admin
pwd_ios = !Cisco123
[nxos]
pod5-nxos1
[eos]
pod5-eos1
[ios]
pod5-ios1
EOF
```

12. Create the **ansible.cfg** file in the present working directory with the following content

```
cat > ansible.cfg <<EOF
[defaults]
host_key_checking = False
```

```
log_path = /var/log/ansible.log

inventory = ./inventory

EOF
```

13. Verify the configuration file by running “ansible –version”



```
[root@localhost ansible]# ansible --version
ansible 2.4.2.0 (detached HEAD e3a8bf02ac) last updated 2018/07/25 20:18:55 (GMT +800)
  config file = /root/Ansible-Local/ansible/ansible.cfg
  configured module search path = [u'/root/.ansible/plugins/modules', u'/usr/share/ansible/plugins/modules']
  ansible python module location = /root/Ansible-Local/ansible/lib/ansible
  executable location = /root/Ansible-Local/ansible/bin/ansible
  python version = 2.7.5 (default, Sep 15 2016, 22:37:39) [GCC 4.8.5 20150623 (Red Hat 4.8.5-4)]
[root@localhost ansible]#
```

Network Automation using Ansible raw module

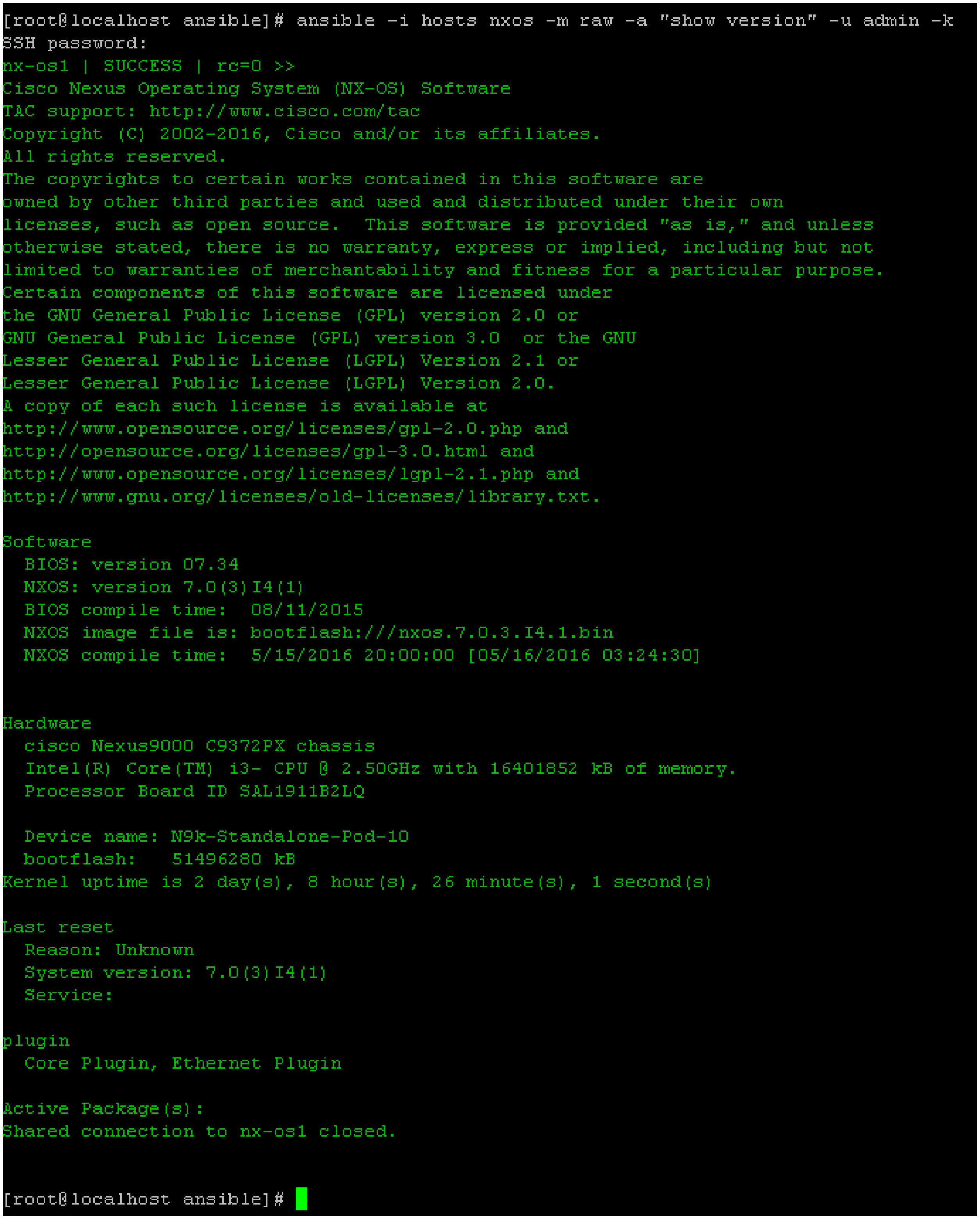
Using the **raw_module** we can execute a direct SSH commands on a target device without going through the module subsystem of ansible.

1. Here is an example of ansible raw command to get the running configuration of a network device

```
#ansible -i hosts nxos -m raw -a "show version" -u admin -k
```

2. Make sure the **sshpass** package is installed on the node, if it’s not installed please the steps below to install it

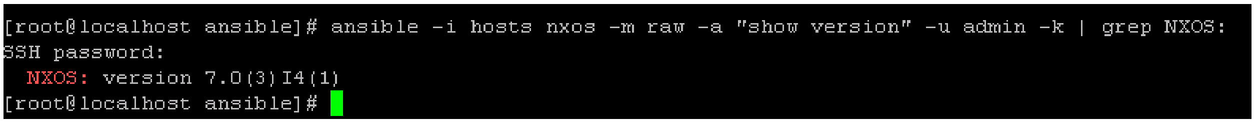
```
# wget http://download.fedoraproject.org/pub/epel/6/x86_64/epel-release-6-8.noarch.rpm

# rpm –ivh epel-release-6-8.noarch.rpm

# yum –enablerepo=epel –y install sshpass
```



```
[root@localhost ansible]# ansible -i hosts nxos -m raw -a "show version" -u admin -k
SSH password:
nx-os1 | SUCCESS | rc=0 >>
Cisco Nexus Operating System (NX-OS) Software
TAC support: http://www.cisco.com/tac
Copyright (C) 2002-2016, Cisco and/or its affiliates.
All rights reserved.
The copyrights to certain works contained in this software are
owned by other third parties and used and distributed under their own
licenses, such as open source.  This software is provided "as is," and unless
otherwise stated, there is no warranty, express or implied, including but not
limited to warranties of merchantability and fitness for a particular purpose.
Certain components of this software are licensed under
the GNU General Public License (GPL) version 2.0 or
GNU General Public License (GPL) version 3.0  or the GNU
Lesser General Public License (LGPL) Version 2.1 or
Lesser General Public License (LGPL) Version 2.0.
A copy of each such license is available at
http://www.opensource.org/licenses/gpl-2.0.php and
http://opensource.org/licenses/gpl-3.0.html and
http://www.opensource.org/licenses/lgpl-2.1.php and
http://www.gnu.org/licenses/old-licenses/library.txt.

Software
  BIOS: version 07.34
  NXOS: version 7.0(3)I4(1)
  BIOS compile time:  08/11/2015
  NXOS image file is: bootflash:///nxos.7.0.3.I4.1.bin
  NXOS compile time:  5/15/2016 20:00:00 [05/16/2016 03:24:30]


Hardware
  cisco Nexus9000 C9372PX chassis
  Intel(R) Core(TM) i3- CPU @ 2.50GHz with 16401852 kB of memory.
  Processor Board ID SAL1911B2LQ

  Device name: N9k-Standalone-Pod-10
  bootflash:    51496280 kB
Kernel uptime is 2 day(s), 8 hour(s), 26 minute(s), 1 second(s)

Last reset
  Reason: Unknown
  System version: 7.0(3)I4(1)
  Service:

plugin
  Core Plugin, Ethernet Plugin

Active Package(s):
Shared connection to nx-os1 closed.


[root@localhost ansible]#
```

3. You can manipulate the output with the grep commands and redirect it to a file etc.
4. Execute the followong command to grep only the **NXOS version**

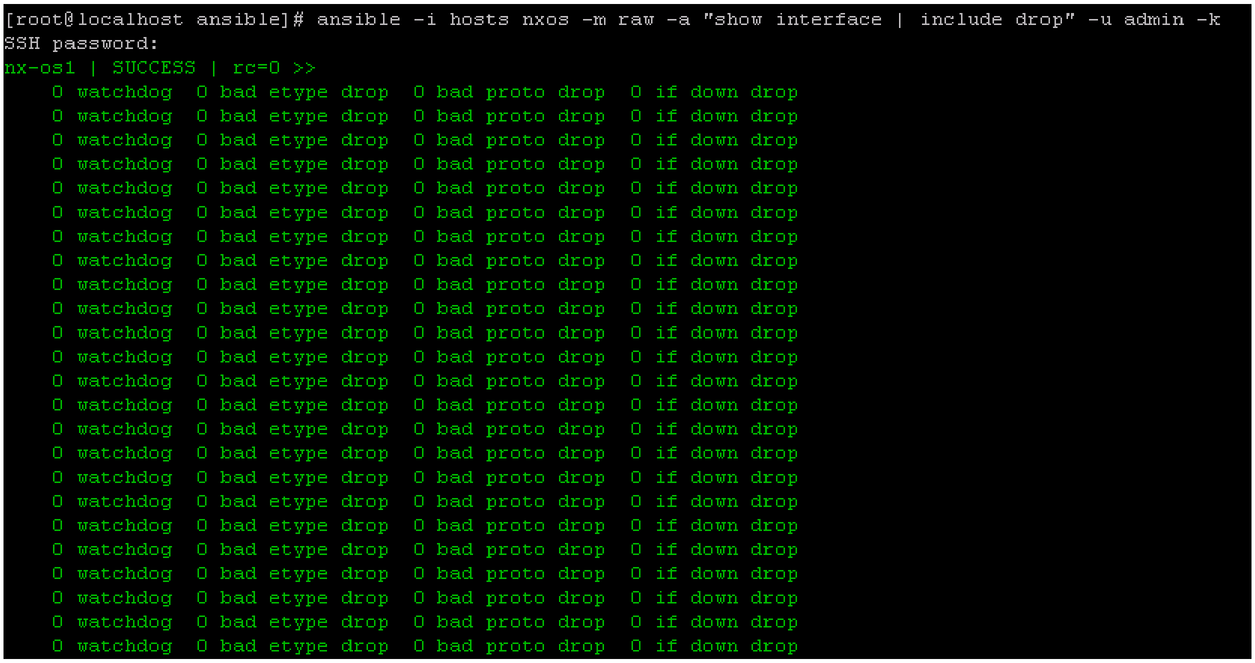
```
ansible -i hosts nxos -m raw -a "show version" -u admin -k | grep NXOS:
```



```
[root@localhost ansible]# ansible -i hosts nxos -m raw -a "show version" -u admin -k | grep NXOS:
SSH password:
  NXOS: version 7.0(3)I4(1)
[root@localhost ansible]#
```

5. You also use the pipe | statement in the command to get some useful network stats.
6. Execute the following command to get the **list of packet drops**

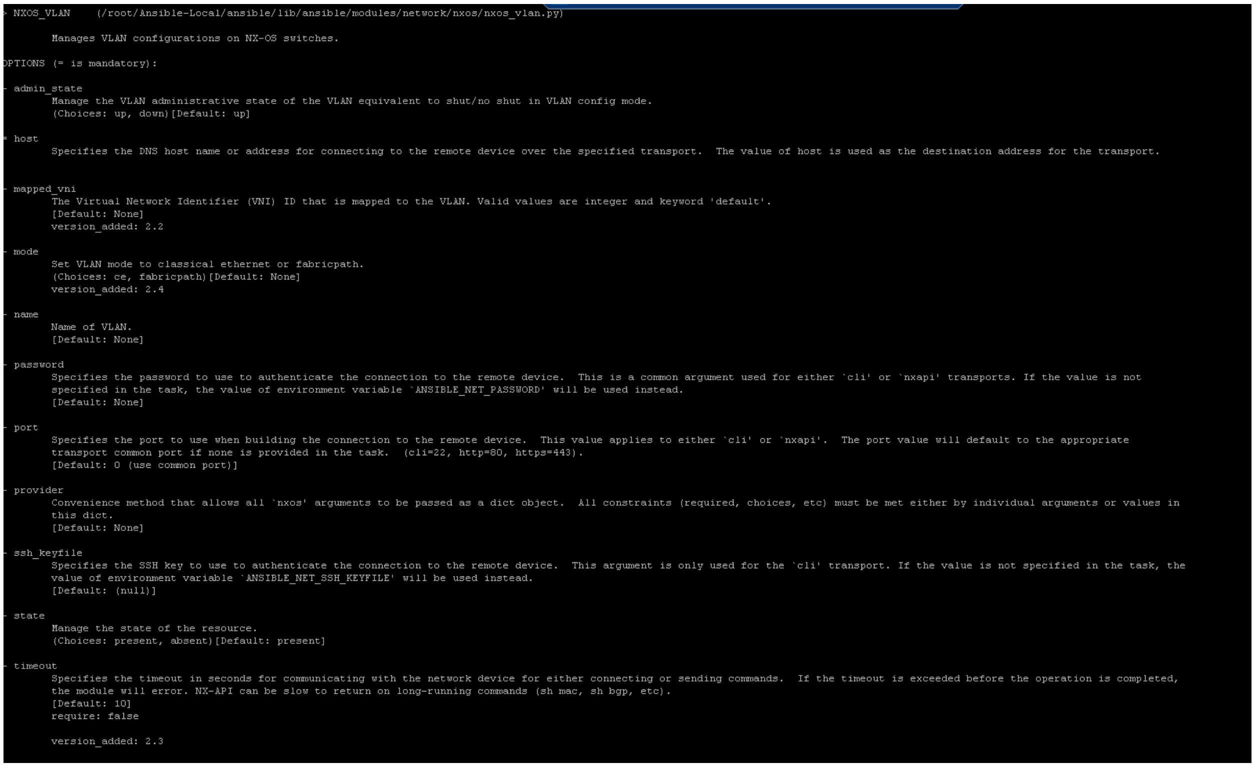
```
ansible -i hosts nxos -m raw -a "show interface | include drop" -u admin -k
```



```
[root@localhost ansible]# ansible -i hosts nxos -m raw -a "show interface | include drop" -u admin -k
SSH password:
nx-os1 | SUCCESS | rc=0 >>
    0 watchdog  0 bad etype drop  0 bad proto drop  0 if down drop
    0 watchdog  0 bad etype drop  0 bad proto drop  0 if down drop
    0 watchdog  0 bad etype drop  0 bad proto drop  0 if down drop
    0 watchdog  0 bad etype drop  0 bad proto drop  0 if down drop
    0 watchdog  0 bad etype drop  0 bad proto drop  0 if down drop
    0 watchdog  0 bad etype drop  0 bad proto drop  0 if down drop
    0 watchdog  0 bad etype drop  0 bad proto drop  0 if down drop
    0 watchdog  0 bad etype drop  0 bad proto drop  0 if down drop
    0 watchdog  0 bad etype drop  0 bad proto drop  0 if down drop
    0 watchdog  0 bad etype drop  0 bad proto drop  0 if down drop
    0 watchdog  0 bad etype drop  0 bad proto drop  0 if down drop
    0 watchdog  0 bad etype drop  0 bad proto drop  0 if down drop
    0 watchdog  0 bad etype drop  0 bad proto drop  0 if down drop
    0 watchdog  0 bad etype drop  0 bad proto drop  0 if down drop
    0 watchdog  0 bad etype drop  0 bad proto drop  0 if down drop
    0 watchdog  0 bad etype drop  0 bad proto drop  0 if down drop
    0 watchdog  0 bad etype drop  0 bad proto drop  0 if down drop
    0 watchdog  0 bad etype drop  0 bad proto drop  0 if down drop
    0 watchdog  0 bad etype drop  0 bad proto drop  0 if down drop
    0 watchdog  0 bad etype drop  0 bad proto drop  0 if down drop
    0 watchdog  0 bad etype drop  0 bad proto drop  0 if down drop
    0 watchdog  0 bad etype drop  0 bad proto drop  0 if down drop
    0 watchdog  0 bad etype drop  0 bad proto drop  0 if down drop
    0 watchdog  0 bad etype drop  0 bad proto drop  0 if down drop
```

Automation Cisco Nexus devices using Ansible

1. We will be using the **nxos_vlan** module for this activity. Use **ansible-doc nxos_vlan** to explore the available parameters and options.



```
> NXOS_VLAN    (/root/Ansible-Local/ansible/lib/ansible/modules/network/nxos/nxos_vlan.py)

        Manages VLAN configurations on NX-OS switches.

OPTIONS (= is mandatory):

- admin_state
        Manage the VLAN administrative state of the VLAN equivalent to shut/no shut in VLAN config mode.
        (Choices: up, down)[Default: up]

= host
        Specifies the DNS host name or address for connecting to the remote device over the specified transport.  The value of host is used as the destination address for the transport.


- mapped_vni
        The Virtual Network Identifier (VNI) ID that is mapped to the VLAN. Valid values are integer and keyword 'default'.
        [Default: None]
        version_added: 2.2

- mode
        Set VLAN mode to classical ethernet or fabricpath.
        (Choices: ce, fabricpath)[Default: None]
        version_added: 2.4

- name
        Name of VLAN.
        [Default: None]

- password
        Specifies the password to use to authenticate the connection to the remote device.  This is a common argument used for either `cli' or `nxapi' transports. If the value is not
        specified in the task, the value of environment variable `ANSIBLE_NET_PASSWORD' will be used instead.
        [Default: None]

- port
        Specifies the port to use when building the connection to the remote device.  This value applies to either `cli' or `nxapi'.  The port value will default to the appropriate
        transport common port if none is provided in the task.  (cli=22, http=80, https=443).
        [Default: 0 (use common port)]

- provider
        Convenience method that allows all `nxos' arguments to be passed as a dict object.  All constraints (required, choices, etc) must be met either by individual arguments or values in
        this dict.
        [Default: None]

- ssh_keyfile
        Specifies the SSH key to use to authenticate the connection to the remote device.  This argument is only used for the `cli' transport. If the value is not specified in the task, the
        value of environment variable `ANSIBLE_NET_SSH_KEYFILE' will be used instead.
        [Default: (null)]

- state
        Manage the state of the resource.
        (Choices: present, absent)[Default: present]

- timeout
        Specifies the timeout in seconds for communicating with the network device for either connecting or sending commands.  If the timeout is exceeded before the operation is completed,
        the module will error. NX-API can be slow to return on long-running commands (sh mac, sh bgp, etc).
        [Default: 10]
        require: false

        version_added: 2.3
```

2. Press letter “Q” to exit the documentation page
3. Create a Playbook to configure the following on the Nexus device

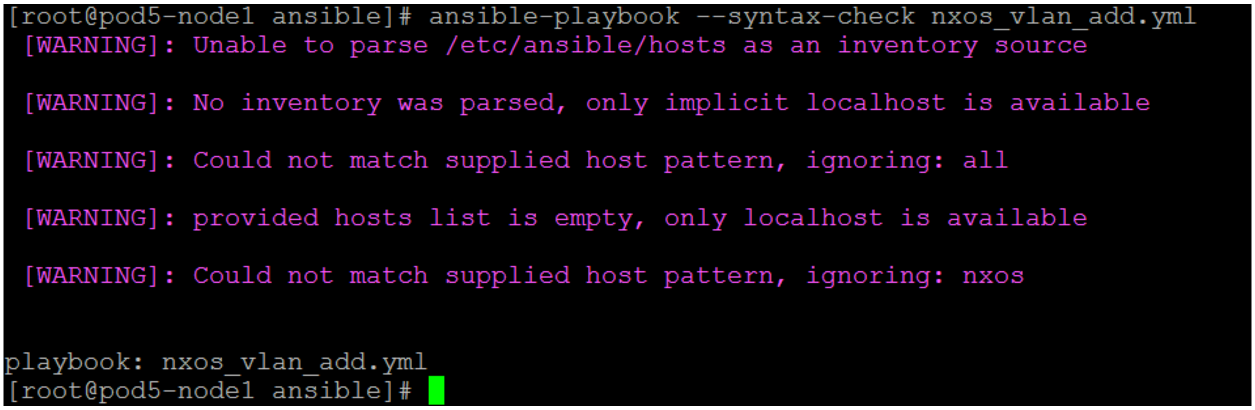
   - Check if the vlan doesn’t exists
   - Create the vlan
   - Verify the VLAN is created

4. You can manually create the playbook by referring the ansible-doc or use the following command to create the **nxos_vlan_add.yml**

```
cat > /root/dep-network/nxos_vlan_add.yml <<EOF

---

- name: Create VLAN's across NX-OS based switches

  hosts: nxos
  connection: local
  gather_facts: no

  vars:
    provider:
      username: "{{ un_nxos }}"
      password: "{{ pwd_nxos }}"
      transport: nxapi
      host: "{{ inventory_hostname }}"

  tasks:
  - name: Adding VLAN using NXOS module "nxos_vlan"
    nxos_vlan:
      vlan_id: 210
      name: Ansible-Added-VLAN-POD5
      provider: "{{ provider }}"

  - name: Ensure the vlan is created
    nxos_vlan:
      vlan_id: 210
      state: present
      provider: "{{ provider }}"
EOF
```

5. Validate the playbook syntax by executing the following command

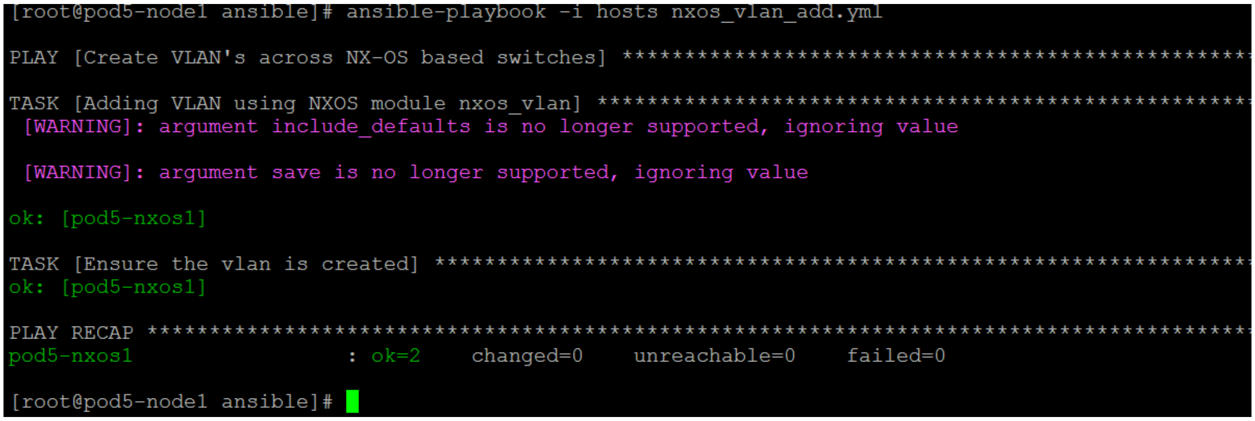
```
ansible-playbook --syntax-check nxos_vlan_add.yml
```



```
[root@pod5-node1 ansible]# ansible-playbook --syntax-check nxos_vlan_add.yml
 [WARNING]: Unable to parse /etc/ansible/hosts as an inventory source

 [WARNING]: No inventory was parsed, only implicit localhost is available

 [WARNING]: Could not match supplied host pattern, ignoring: all

 [WARNING]: provided hosts list is empty, only localhost is available

 [WARNING]: Could not match supplied host pattern, ignoring: nxos


playbook: nxos_vlan_add.yml
[root@pod5-node1 ansible]# 
```

6. Ignore the warnings
7. Execute the playbook and verify the output

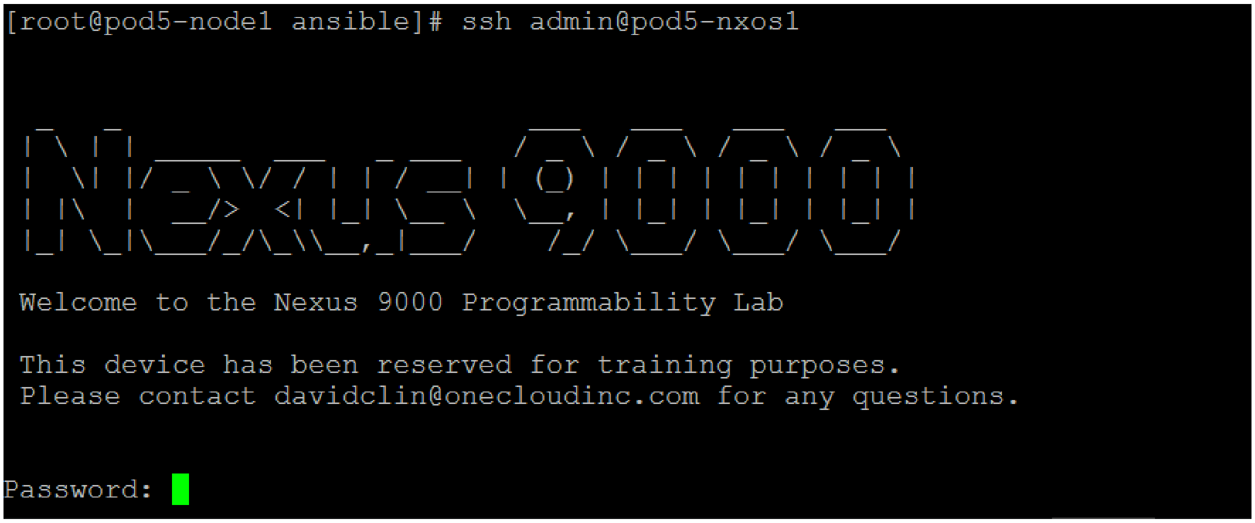
```
ansible-playbook -i hosts nxos_vlan_add.yml
```



```
[root@pod5-node1 ansible]# ansible-playbook -i hosts nxos_vlan_add.yml

PLAY [Create VLAN's across NX-OS based switches] ***************************************************

TASK [Adding VLAN using NXOS module nxos_vlan] *****************************************************
 [WARNING]: argument include_defaults is no longer supported, ignoring value

 [WARNING]: argument save is no longer supported, ignoring value

ok: [pod5-nxos1]

TASK [Ensure the vlan is created] *****************************************************************
ok: [pod5-nxos1]

PLAY RECAP *****************************************************************************************
pod5-nxos1                 : ok=2    changed=0    unreachable=0    failed=0

[root@pod5-node1 ansible]# 
```

8. Login to the Nexus device as **admin/#cisco123** and verify the configuration

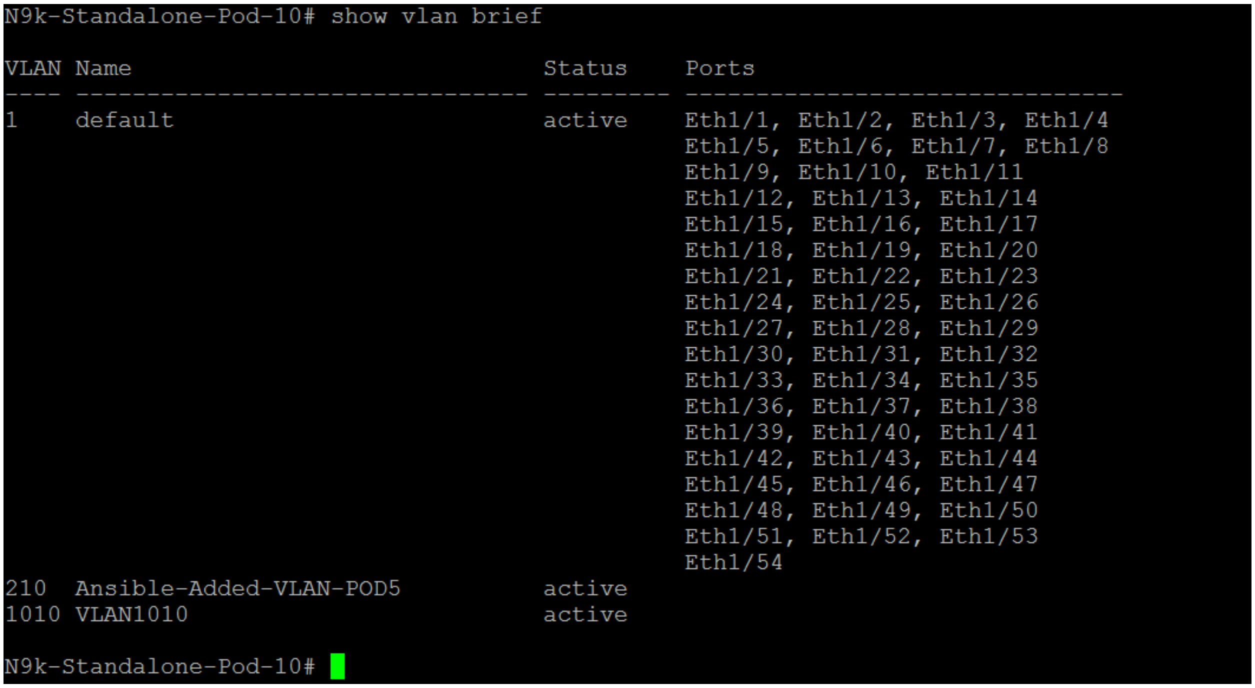
```
#ssh admin@pod5-nxos1
```



```
[root@pod5-node1 ansible]# ssh admin@pod5-nxos1


  _   _                        ___   ___   ___   ___
 | \ | | _____  ___   _ ___   / _ \ / _ \ / _ \ / _ \
 |  \| |/ _ \ \/ / | | / __| | (_) | | | | | | | | | |
 | |\  |  __/>  <| |_| \__ \  \__, | |_| | |_| | |_| |
 |_| \_|\___/_/\_\\__,_|___/    /_/ \___/ \___/ \___/

 Welcome to the Nexus 9000 Programmability Lab

 This device has been reserved for training purposes.
 Please contact davidclin@onecloudinc.com for any questions.


Password:
```

9. Execute **"show vlan brief"** to list the VLANS

```
N9k-Standalone-Pod-10# show vlan brief
```



```
N9k-Standalone-Pod-10# show vlan brief

VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
1    default                          active    Eth1/1, Eth1/2, Eth1/3, Eth1/4
                                                Eth1/5, Eth1/6, Eth1/7, Eth1/8
                                                Eth1/9, Eth1/10, Eth1/11
                                                Eth1/12, Eth1/13, Eth1/14
                                                Eth1/15, Eth1/16, Eth1/17
                                                Eth1/18, Eth1/19, Eth1/20
                                                Eth1/21, Eth1/22, Eth1/23
                                                Eth1/24, Eth1/25, Eth1/26
                                                Eth1/27, Eth1/28, Eth1/29
                                                Eth1/30, Eth1/31, Eth1/32
                                                Eth1/33, Eth1/34, Eth1/35
                                                Eth1/36, Eth1/37, Eth1/38
                                                Eth1/39, Eth1/40, Eth1/41
                                                Eth1/42, Eth1/43, Eth1/44
                                                Eth1/45, Eth1/46, Eth1/47
                                                Eth1/48, Eth1/49, Eth1/50
                                                Eth1/51, Eth1/52, Eth1/53
                                                Eth1/54
210  Ansible-Added-VLAN-POD5          active
1010 VLAN1010                         active

N9k-Standalone-Pod-10#
```

10. Verify the VLAN is created and exit from the nexus device shell
11. Type "exit" to exit the device ssh shell
12. Explore the other available **nxos** modules and create your own playbook to **configure interface, login banner etc.**

Perform Clean up activity on nexus device

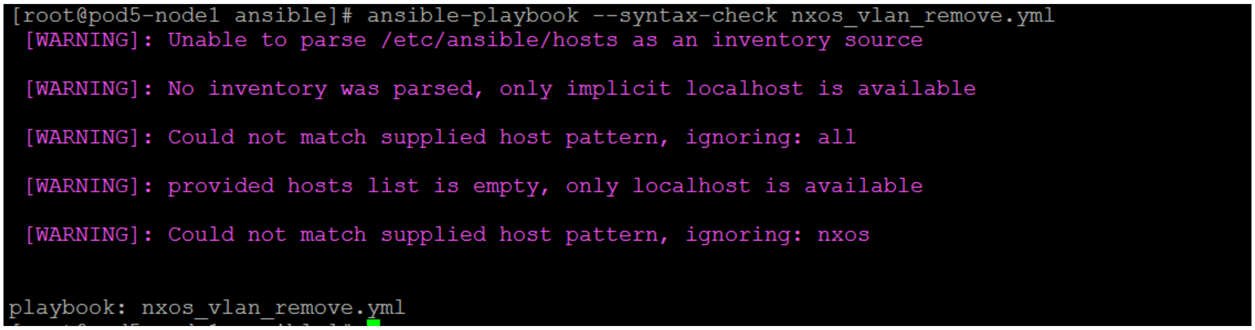
1. Create a play book to delete the VLAN added in the previous activity, use the nxos_vlan to create the paly,
2. Refer the **ansible-doc nxos_vlan** documentation to get the parameter and options
3. You can manually create the playbook by referring the ansible-doc or use the following command to create the **nxos_vlan_remove.yml**

```
cat > /root/dep-network/nxos_vlan_remove.yml <<EOF

---

- name: Delete VLAN's across NX-OS based switches

  hosts: nxos

  connection: local

  gather_facts: no



  vars:

    provider:

      username: "{{ un_nxos }}"

      password: "{{ pwd_nxos }}"

      transport: nxapi

      host: "{{ inventory_hostname }}"

  tasks:

  - name: Deleting VLAN using NXOS module "nxos_vlan"

    nxos_vlan:

      provider: "{{ provider }}"

      vlan_id: 210

      state: absent
```

```
EOF
```

4. Validate the syntax and execute the playbook

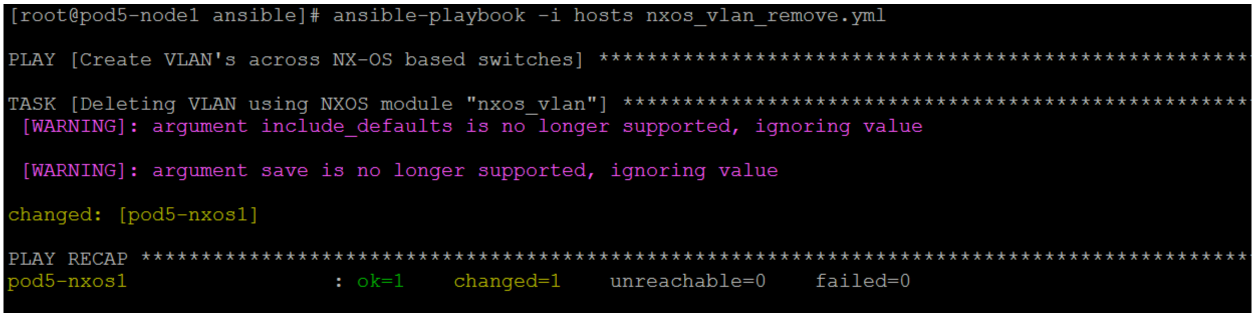
```
ansible-playbook --syntax-check nxos_vlan_remove.yml
```



```
[root@pod5-node1 ansible]# ansible-playbook --syntax-check nxos_vlan_remove.yml
 [WARNING]: Unable to parse /etc/ansible/hosts as an inventory source

 [WARNING]: No inventory was parsed, only implicit localhost is available

 [WARNING]: Could not match supplied host pattern, ignoring: all

 [WARNING]: provided hosts list is empty, only localhost is available

 [WARNING]: Could not match supplied host pattern, ignoring: nxos


playbook: nxos_vlan_remove.yml
```

5. Ignore the warnings and execute the playbook

```
ansible-playbook -i hosts nxos_vlan_remove.yml
```



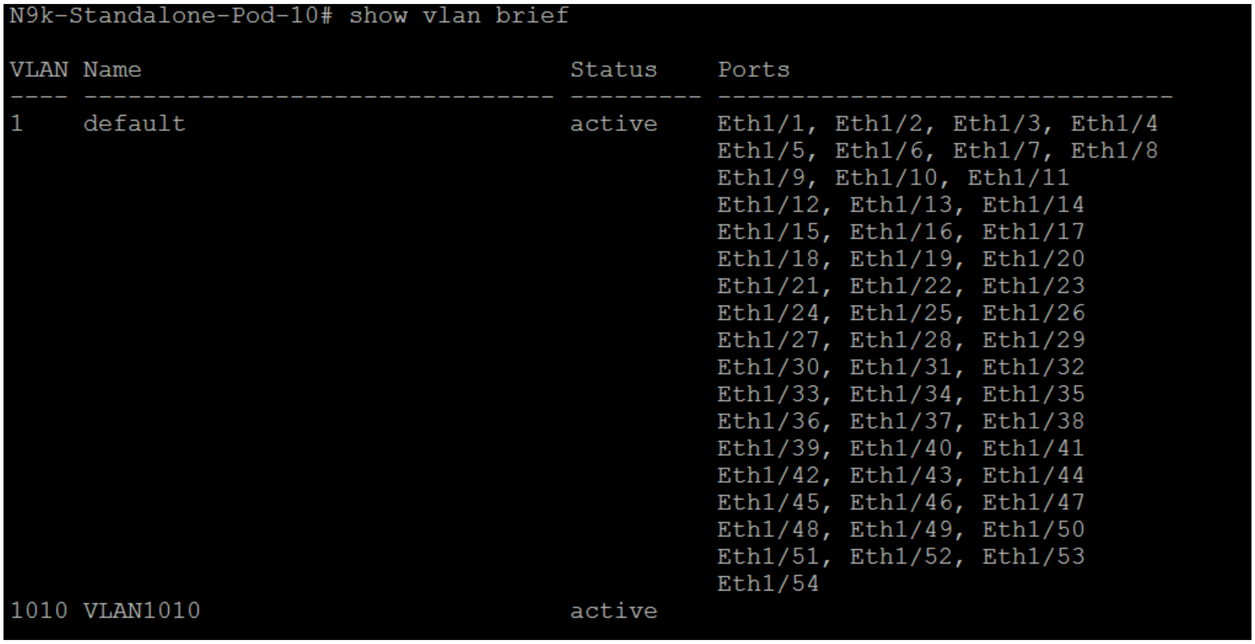
```
[root@pod5-node1 ansible]# ansible-playbook -i hosts nxos_vlan_remove.yml

PLAY [Create VLAN's across NX-OS based switches] *****************************************************

TASK [Deleting VLAN using NXOS module "nxos_vlan"] ***************************************************
 [WARNING]: argument include_defaults is no longer supported, ignoring value

 [WARNING]: argument save is no longer supported, ignoring value

changed: [pod5-nxos1]

PLAY RECAP *******************************************************************************************
pod5-nxos1                 : ok=1    changed=1    unreachable=0    failed=0
```

6. SSH to the network device and verify the VLAN is removed

```
ssh admin@pod5-nxos1
```

7. Execute show vlan brief and verify the VLAN is removed from the device

```
N9k-Standalone-Pod-10# show vlan brief
```

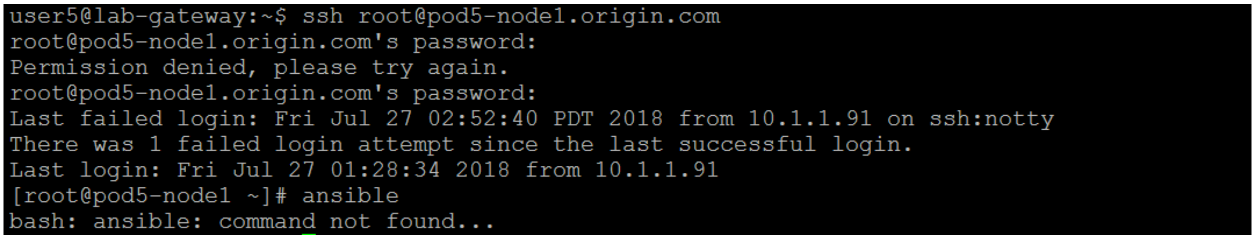


```
N9k-Standalone-Pod-10# show vlan brief

VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
1    default                          active    Eth1/1, Eth1/2, Eth1/3, Eth1/4
                                                Eth1/5, Eth1/6, Eth1/7, Eth1/8
                                                Eth1/9, Eth1/10, Eth1/11
                                                Eth1/12, Eth1/13, Eth1/14
                                                Eth1/15, Eth1/16, Eth1/17
                                                Eth1/18, Eth1/19, Eth1/20
                                                Eth1/21, Eth1/22, Eth1/23
                                                Eth1/24, Eth1/25, Eth1/26
                                                Eth1/27, Eth1/28, Eth1/29
                                                Eth1/30, Eth1/31, Eth1/32
                                                Eth1/33, Eth1/34, Eth1/35
                                                Eth1/36, Eth1/37, Eth1/38
                                                Eth1/39, Eth1/40, Eth1/41
                                                Eth1/42, Eth1/43, Eth1/44
                                                Eth1/45, Eth1/46, Eth1/47
                                                Eth1/48, Eth1/49, Eth1/50
                                                Eth1/51, Eth1/52, Eth1/53
                                                Eth1/54
1010 VLAN1010                         active
```

8. Verify the VLAN is removed and exit from the device SSH shell
9. Type “exit” to exit from the device SSH shell

Automating Cisco CSR1000v using Ansible

1. Login to the POD5 node1 VM from the SSH gateway
2. Login as root use on ssh root@podx-node1.origin.com with password !cisco123



```
user5@lab-gateway:~$ ssh root@pod5-node1.origin.com
root@pod5-node1.origin.com's password:
Permission denied, please try again.
root@pod5-node1.origin.com's password:
Last failed login: Fri Jul 27 02:52:40 PDT 2018 from 10.1.1.91 on ssh:notty
There was 1 failed login attempt since the last successful login.
Last login: Fri Jul 27 01:28:34 2018 from 10.1.1.91
[root@pod5-node1 ~]# ansible
bash: ansible: command not found...
```

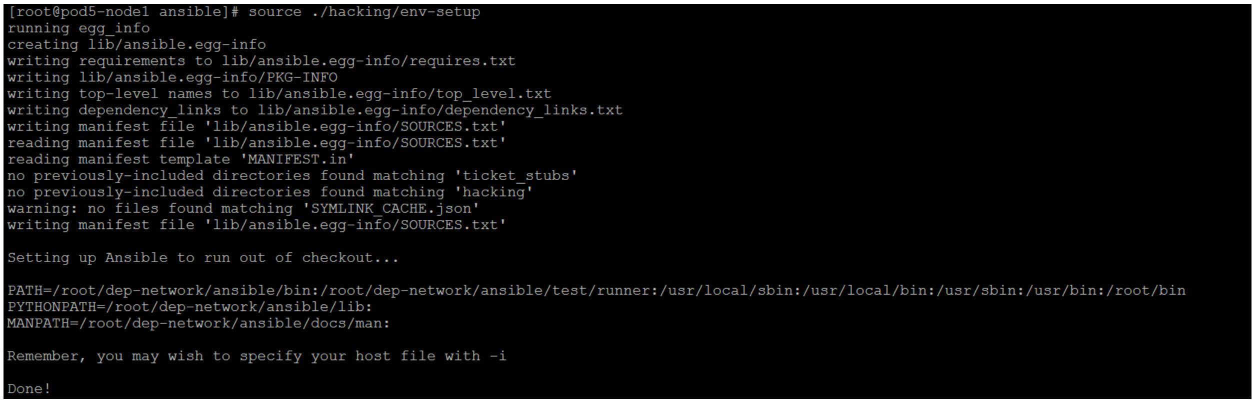
Note: *Since we have not installed the Ansible as package on the Node VM, the executable will not work when the SSH session is re-established. You need to run the environment setup script on every login.*

3. Change to **“dep-network/ansible/”** directory

```
cd dep-network/ansible/
```

4. Execute the **env-setup script** to setup the environment

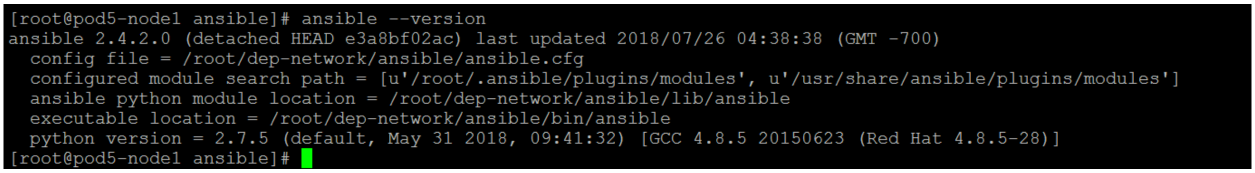
```
[root@pod5-node1 ansible]# source ./hacking/env-setup
```



```
[root@pod5-node1 ansible]# source ./hacking/env-setup
running egg_info
creating lib/ansible.egg-info
writing requirements to lib/ansible.egg-info/requires.txt
writing lib/ansible.egg-info/PKG-INFO
writing top-level names to lib/ansible.egg-info/top_level.txt
writing dependency_links to lib/ansible.egg-info/dependency_links.txt
writing manifest file 'lib/ansible.egg-info/SOURCES.txt'
reading manifest file 'lib/ansible.egg-info/SOURCES.txt'
reading manifest template 'MANIFEST.in'
no previously-included directories found matching 'ticket_stubs'
no previously-included directories found matching 'hacking'
warning: no files found matching 'SYMLINK_CACHE.json'
writing manifest file 'lib/ansible.egg-info/SOURCES.txt'

Setting up Ansible to run out of checkout...

PATH=/root/dep-network/ansible/bin:/root/dep-network/ansible/test/runner:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/root/bin
PYTHONPATH=/root/dep-network/ansible/lib:
MANPATH=/root/dep-network/ansible/docs/man:

Remember, you may wish to specify your host file with -i

Done!
```

5. Verify the ansible binary is executable, Enter **ansible --version** to verify

```
[root@pod5-node1 ansible]# ansible --version
```



```
[root@pod5-node1 ansible]# ansible --version
ansible 2.4.2.0 (detached HEAD e3a8bf02ac) last updated 2018/07/26 04:38:38 (GMT -700)
  config file = /root/dep-network/ansible/ansible.cfg
  configured module search path = [u'/root/.ansible/plugins/modules', u'/usr/share/ansible/plugins/modules']
  ansible python module location = /root/dep-network/ansible/lib/ansible
  executable location = /root/dep-network/ansible/bin/ansible
  python version = 2.7.5 (default, May 31 2018, 09:41:32) [GCC 4.8.5 20150623 (Red Hat 4.8.5-28)]
[root@pod5-node1 ansible]#
```

6. Create a playbook to configure the following on the CSR100v
   - Add a ACL “access-list 99 permit 172.16.0.254”
   - Verify the ACL is added
   - Save the configuration
7. We will use **ios_command** and **ios_config** modules to automation the above mentioned configuration
8. Explore the module parameters and option using ansible-doc <module name> command
9. You can manually create the playbook by referring the ansible-doc or use the following command to create the playbook

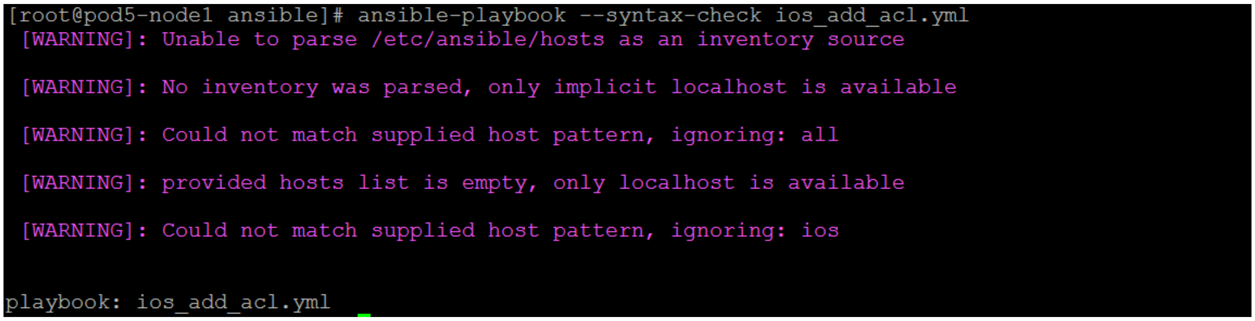
```
cat > /root/dep-network/ios_add_acl.yml <<EOF

---



- name: Manage ISO Device
```

```
hosts: ios
connection: local
gather_facts: no

vars:
   provider:
     username: "{{ un_ios }}"
     password: "{{ pwd_ios }}"
     host: "{{ inventory_hostname }}"


tasks:
- name: Configure ACL on Cisco CSR 1000v
  ios_config:
    auth_pass: "{{ pwd_ios }}"
    authorize: yes
    provider: "{{ provider }}"
    lines:
      - access-list 99 permit 172.16.0.254

- name: Verify ACL is present
  ios_command:
    auth_pass: "{{ pwd_ios }}"
```

```
        authorize: yes

        provider: "{{ provider }}"

        commands:

          - sh access-l

        waitfor:

          - result[0] contains 'permit 172.16.0.254'




    - name: Save config

      ios_config:

        auth_pass: "{{ pwd_ios }}"

        authorize: yes

        provider: "{{ provider }}"

        save_when: modified

  EOF
```

10. Validate the syntax

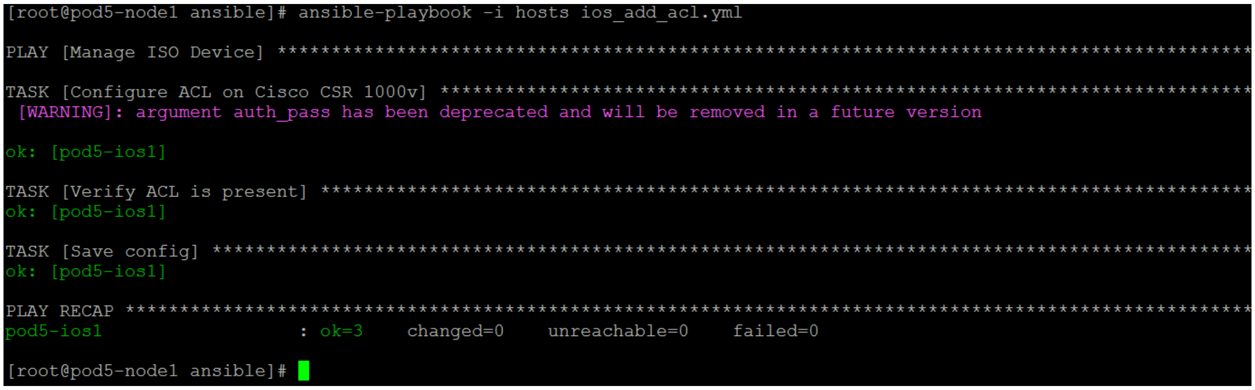
```
ansible-playbook --syntax-check ios_add_acl.yml
```



```
[root@pod5-node1 ansible]# ansible-playbook --syntax-check ios_add_acl.yml
 [WARNING]: Unable to parse /etc/ansible/hosts as an inventory source

 [WARNING]: No inventory was parsed, only implicit localhost is available

 [WARNING]: Could not match supplied host pattern, ignoring: all

 [WARNING]: provided hosts list is empty, only localhost is available

 [WARNING]: Could not match supplied host pattern, ignoring: ios


playbook: ios_add_acl.yml
```

11. Ignore the warnings and execute the playbook

```
ansible-playbook -i hosts ios_add_acl.yml
```



```
[root@pod5-node1 ansible]# ansible-playbook -i hosts ios_add_acl.yml

PLAY [Manage ISO Device] ******************************************************************************

TASK [Configure ACL on Cisco CSR 1000v] ***************************************************************
 [WARNING]: argument auth_pass has been deprecated and will be removed in a future version

ok: [pod5-ios1]

TASK [Verify ACL is present] **************************************************************************
ok: [pod5-ios1]

TASK [Save config] ************************************************************************************
ok: [pod5-ios1]

PLAY RECAP ********************************************************************************************
pod5-ios1                  : ok=3    changed=0    unreachable=0    failed=0

[root@pod5-node1 ansible]#
```

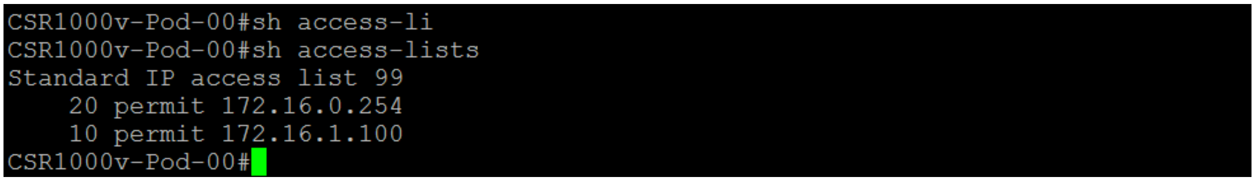
12. Login to the CSR1000v ssh console as (admin/!Cisco123) and verify the ACL are configured

```
[root@pod5-node1 ansible]# ssh admin@pod5-ios1
```

13. Accept the fingerprint warning and press enter
14. Enter into the privileged mode with password “!Cisco123”

```
CSR1000v-Pod-00>enable

Password:
```

15. Execute the “show access-list” command and verify the ACL is present

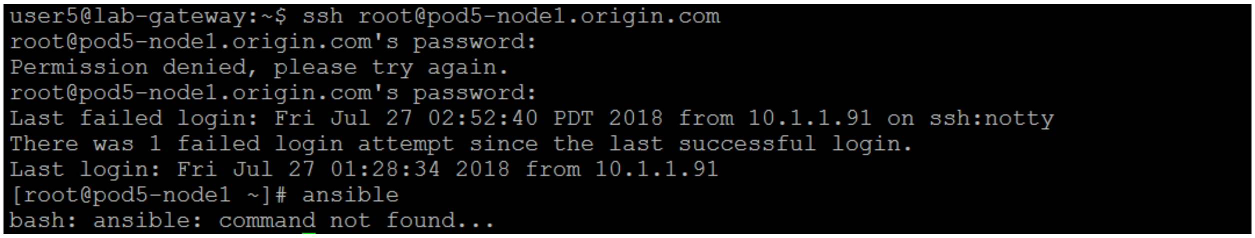


```
CSR1000v-Pod-00#sh access-li
CSR1000v-Pod-00#sh access-lists
Standard IP access list 99
    20 permit 172.16.0.254
    10 permit 172.16.1.100
CSR1000v-Pod-00#
```

16. Exit from the device SSH shell, Type “exit” to exit the shell

Automation Arista EOS using Ansible

1. Login to the POD5 node1 from the SSH gateway
2. Login as root use on ssh root@podx-node1.origin.com with password !cisco123



```
user5@lab-gateway:~$ ssh root@pod5-node1.origin.com
root@pod5-node1.origin.com's password:
Permission denied, please try again.
root@pod5-node1.origin.com's password:
Last failed login: Fri Jul 27 02:52:40 PDT 2018 from 10.1.1.91 on ssh:notty
There was 1 failed login attempt since the last successful login.
Last login: Fri Jul 27 01:28:34 2018 from 10.1.1.91
[root@pod5-node1 ~]# ansible
bash: ansible: command not found...
```

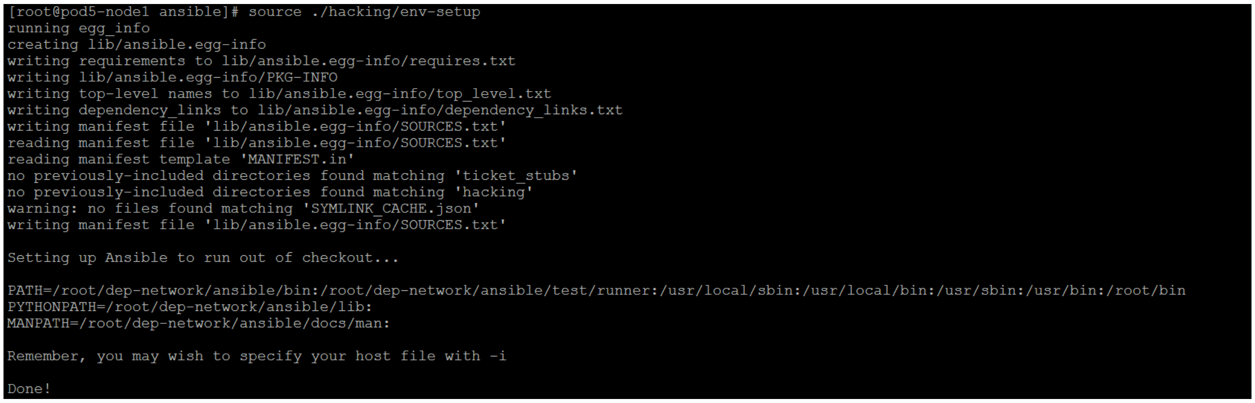
Note: *Since we have not installed the Ansible as package on the Node VM, the executable will not work when the SSH session is re-established. You need to run the environment setup script on every login.*

3. Change to “dep-network/ansible/” directory

```
cd dep-network/ansible/
```

4. Execute the env-setup script to setup the environment

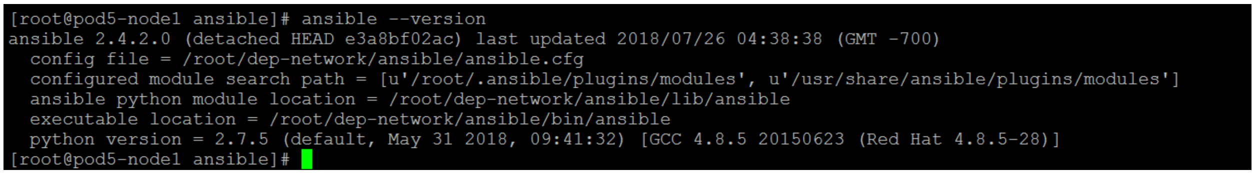
```
[root@pod5-node1 ansible]# source ./hacking/env-setup
```



```
[root@pod5-node1 ansible]# source ./hacking/env-setup
running egg_info
creating lib/ansible.egg-info
writing requirements to lib/ansible.egg-info/requires.txt
writing lib/ansible.egg-info/PKG-INFO
writing top-level names to lib/ansible.egg-info/top_level.txt
writing dependency_links to lib/ansible.egg-info/dependency_links.txt
writing manifest file 'lib/ansible.egg-info/SOURCES.txt'
reading manifest file 'lib/ansible.egg-info/SOURCES.txt'
reading manifest template 'MANIFEST.in'
no previously-included directories found matching 'ticket_stubs'
no previously-included directories found matching 'hacking'
warning: no files found matching 'SYMLINK_CACHE.json'
writing manifest file 'lib/ansible.egg-info/SOURCES.txt'

Setting up Ansible to run out of checkout...

PATH=/root/dep-network/ansible/bin:/root/dep-network/ansible/test/runner:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/root/bin
PYTHONPATH=/root/dep-network/ansible/lib:
MANPATH=/root/dep-network/ansible/docs/man:

Remember, you may wish to specify your host file with -i

Done!
```

5. Verify the ansible command is executable, Enter ansible --version to verify

```
[root@pod5-node1 ansible]# ansible --version
```



```
[root@pod5-node1 ansible]# ansible --version
ansible 2.4.2.0 (detached HEAD e3a8bf02ac) last updated 2018/07/26 04:38:38 (GMT -700)
  config file = /root/dep-network/ansible/ansible.cfg
  configured module search path = [u'/root/.ansible/plugins/modules', u'/usr/share/ansible/plugins/modules']
  ansible python module location = /root/dep-network/ansible/lib/ansible
  executable location = /root/dep-network/ansible/bin/ansible
  python version = 2.7.5 (default, May 31 2018, 09:41:32) [GCC 4.8.5 20150623 (Red Hat 4.8.5-28)]
[root@pod5-node1 ansible]#
```

6. Create a playbook to configure the following on the Arista vEOS
   - Set the **hostname** as **“vEOS-POD5”**
   - Configure the **dns name** as **“veos-pod5.onecloudinc.com”**
   - Configure the **name servers** as **“8.8.8.8 and 8.8.4.4”**
   - Configure the **Login banner** as **“Managed by Ansible”**
7. We will be using the **eos_system** and **eos_banner** modules to automate the above configuration
8. Explore the module parameters and options using the ansible-doc <module name>
9. You can manually create the playbook by referring the ansible-doc or use the following command to create the playbook

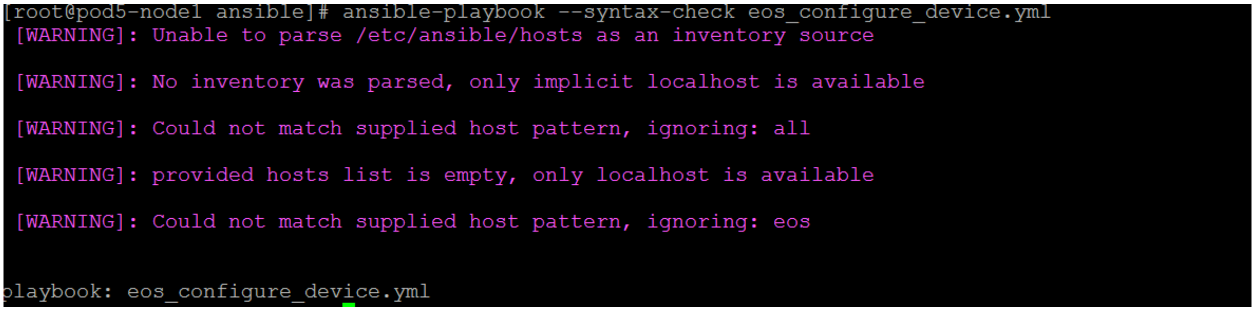
```
cat > /root/dep-network/eos_configure_device.yml <<EOF

---



- name: Configure EOS switches

  hosts: eos

  connection: local

  gather_facts: no



  vars:

    provider:

      host: "{{ inventory_hostname }}"

      username: "{{ un_eos }}"

      password: "{{ pwd_eos }}"

      transport: eapi

      use_ssl: false





  tasks:
```

```
- name: configure hostname and domain-name
  eos_system:
    authorize: yes
    auth_pass: "{{ pwd_eos }}"
    hostname: vEOS-POD5
    domain_name: veos-pod5.onecloudinc.com
    provider: "{{ provider }}"

- name: configure name servers
  eos_system:
    authorize: yes
    auth_pass: "{{ pwd_eos }}"
    name_servers:
      - 8.8.8.8
      - 8.8.4.4
    provider: "{{ provider }}"

- name: configure the login banner
  eos_banner:
    authorize: yes
    auth_pass: "{{ pwd_eos }}"
    banner: login
```

```
          text: Managed by Ansible

          state: present

          provider: "{{ provider }}"

EOF
```

10. Validate the syntax

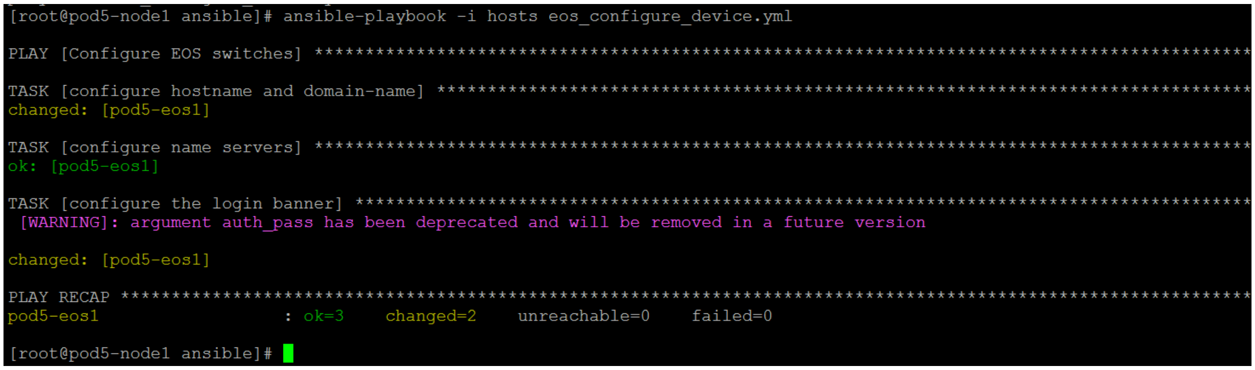
```
ansible-playbook --syntax-check eos_configure_device.yml
```



```
[root@pod5-node1 ansible]# ansible-playbook --syntax-check eos_configure_device.yml
 [WARNING]: Unable to parse /etc/ansible/hosts as an inventory source

 [WARNING]: No inventory was parsed, only implicit localhost is available

 [WARNING]: Could not match supplied host pattern, ignoring: all

 [WARNING]: provided hosts list is empty, only localhost is available

 [WARNING]: Could not match supplied host pattern, ignoring: eos


playbook: eos_configure_device.yml
```

11. Ignore the warnings and execute the playbook

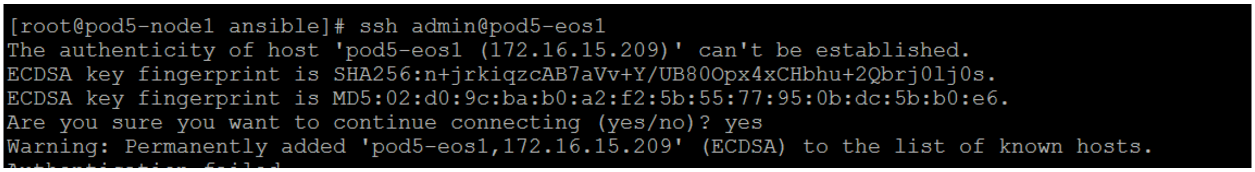
```
ansible-playbook -i hosts eos_configure_device.yml
```



```
[root@pod5-node1 ansible]# ansible-playbook -i hosts eos_configure_device.yml

PLAY [Configure EOS switches] ***************************************************************************

TASK [configure hostname and domain-name] ***************************************************************
changed: [pod5-eos1]

TASK [configure name servers] ***************************************************************************
ok: [pod5-eos1]

TASK [configure the login banner] ***********************************************************************
 [WARNING]: argument auth_pass has been deprecated and will be removed in a future version

changed: [pod5-eos1]

PLAY RECAP **********************************************************************************************
pod5-eos1                  : ok=3    changed=2    unreachable=0    failed=0

[root@pod5-node1 ansible]#
```

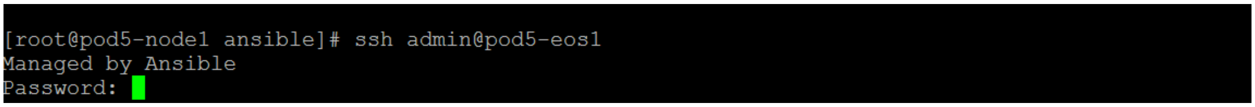
12. Login to the device and verify the configuration
13. SSH to device as **admin/!Cisco123**

```
ssh admin@pod5-eos1
```



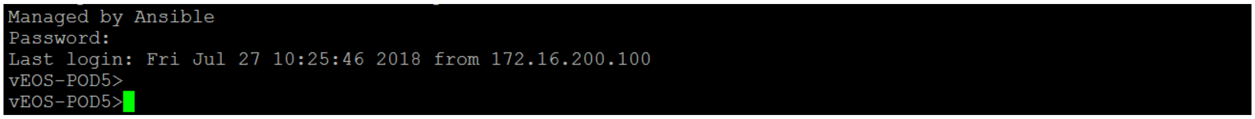
```
[root@pod5-node1 ansible]# ssh admin@pod5-eos1
The authenticity of host 'pod5-eos1 (172.16.15.209)' can't be established.
ECDSA key fingerprint is SHA256:n+jrkiqzcAB7aVv+Y/UB80Opx4xCHbhu+2Qbrj0lj0s.
ECDSA key fingerprint is MD5:02:d0:9c:ba:b0:a2:f2:5b:55:77:95:0b:dc:5b:b0:e6.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'pod5-eos1,172.16.15.209' (ECDSA) to the list of known hosts.
```

14. Verify the configuration
15. Login banner



```
[root@pod5-node1 ansible]# ssh admin@pod5-eos1
Managed by Ansible
Password:
```

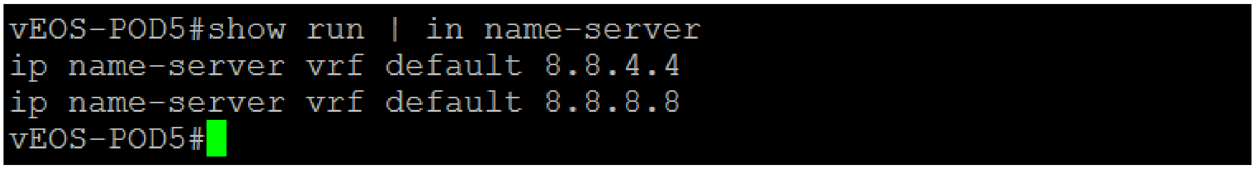
16. Hostname , Name servers



```
Managed by Ansible
Password:
Last login: Fri Jul 27 10:25:46 2018 from 172.16.200.100
vEOS-POD5>
vEOS-POD5>
```

17. Enter to privileged mode with password - **!Cisco123** and execute **“show run | in name-server”**

```
vEOS-POD5#show run | in name-server
```



```
vEOS-POD5#show run | in name-server
ip name-server vrf default 8.8.4.4
ip name-server vrf default 8.8.8.8
vEOS-POD5#
```

18. Exit from the device ssh shell, Type **“exit”** to exit from the SSH shell

Perform Cleanup activity on the EOS Device using Ansible

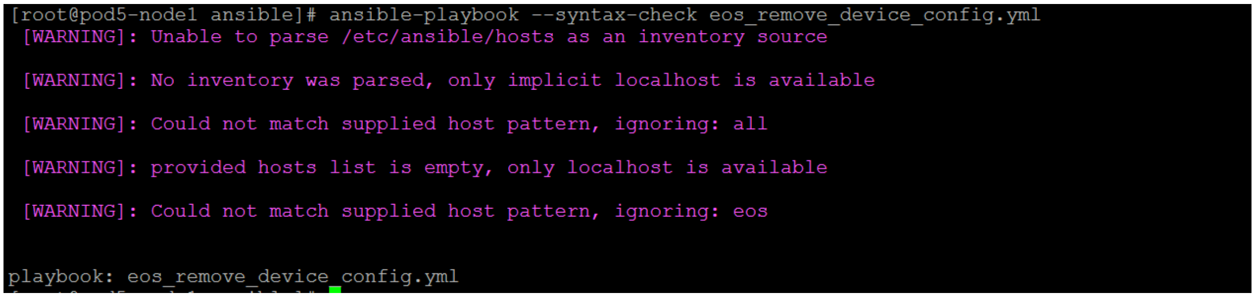
1. Create a playbook to **remove the EOS configuration created during the previous activity**
2. We will be the same **eos_system** and **eos_banner** to remove the configuration, Explore the module options and parameters using ansible-doc <module name>
3. You can manually create the playbook by referring the ansible-doc or use the following command create the playbook

```
cat > /root/dep-network/eos_remove_device_config.yml <<EOF

---
```

```
- name: Create VLAN's across NX-OS based switches
  hosts: eos
  connection: local
  gather_facts: no

  vars:
    provider:
      host: "{{ inventory_hostname }}"
      username: "{{ un_eos }}"
      password: "{{ pwd_eos }}"
      transport: eapi
      use_ssl: false


  tasks:
  - name: Remove configuration
    eos_system:
      authorize: yes
      auth_pass: "{{ pwd_eos }}"
      state: absent
      provider: "{{ provider }}"
  - name: Remove login banner
    eos_banner:
      authorize: yes
      auth_pass: "{{ pwd_eos }}"
      banner: login
      state: absent
      provider: "{{ provider }}"
EOF
```

4. Validate the syntax

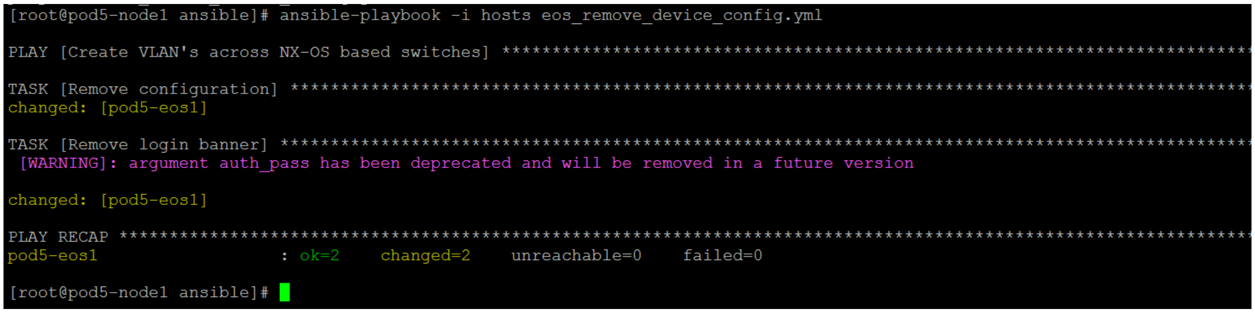
```
ansible-playbook --syntax-check eos_remove_device_config.yml
```



```
[root@pod5-node1 ansible]# ansible-playbook --syntax-check eos_remove_device_config.yml
 [WARNING]: Unable to parse /etc/ansible/hosts as an inventory source

 [WARNING]: No inventory was parsed, only implicit localhost is available

 [WARNING]: Could not match supplied host pattern, ignoring: all

 [WARNING]: provided hosts list is empty, only localhost is available

 [WARNING]: Could not match supplied host pattern, ignoring: eos


playbook: eos_remove_device_config.yml
```

5. Ignore the warnings and execute the playbook

```
ansible-playbook -i hosts eos_remove_device_config.yml
```

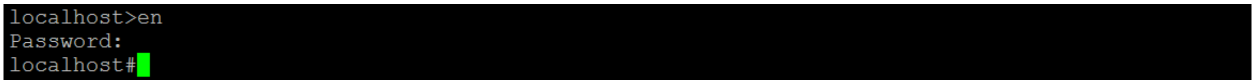


```
[root@pod5-node1 ansible]# ansible-playbook -i hosts eos_remove_device_config.yml

PLAY [Create VLAN's across NX-OS based switches] *********************************************************************

TASK [Remove configuration] ******************************************************************************************
changed: [pod5-eos1]

TASK [Remove login banner] *******************************************************************************************
 [WARNING]: argument auth_pass has been deprecated and will be removed in a future version

changed: [pod5-eos1]

PLAY RECAP ***********************************************************************************************************
pod5-eos1                  : ok=2    changed=2    unreachable=0    failed=0

[root@pod5-node1 ansible]#
```

6. Login to the device and verify the configuration are removed
7. SSH to deivce as **admin/!Cisco123**

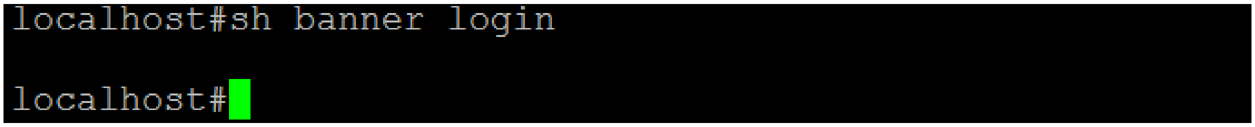
```
ssh admin@pod5-eos1
```

8. Enter into privileged mode with password **!Cisco123**



```
localhost>en
Password:
localhost#
```

9. Verify the **Hostname, login banner** configuration are removed.



```
localhost#sh banner login

localhost#
```

Appendix A

Running Ansible environment setup script

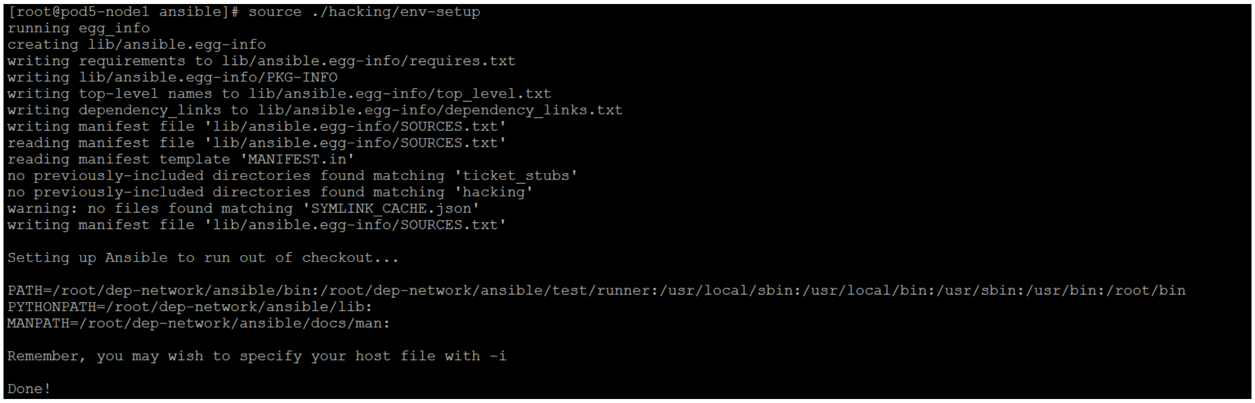
Note: *Since we have not installed the Ansible as package on the Node VM, the executable will not work if the SSH session is re-established. You need to run the environment setup script on every new login.*

1. Change to “dep-network/ansible/” directory

```
cd dep-network/ansible/
```

2. Execute the env-setup script to setup the environment

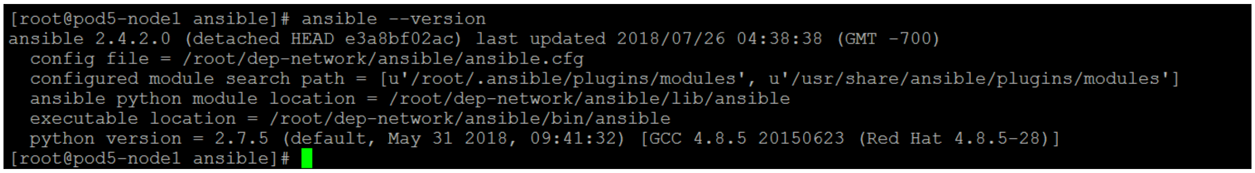
```
[root@pod5-node1 ansible]# source ./hacking/env-setup
```



```
[root@pod5-node1 ansible]# source ./hacking/env-setup
running egg_info
creating lib/ansible.egg-info
writing requirements to lib/ansible.egg-info/requires.txt
writing lib/ansible.egg-info/PKG-INFO
writing top-level names to lib/ansible.egg-info/top_level.txt
writing dependency_links to lib/ansible.egg-info/dependency_links.txt
writing manifest file 'lib/ansible.egg-info/SOURCES.txt'
reading manifest file 'lib/ansible.egg-info/SOURCES.txt'
reading manifest template 'MANIFEST.in'
no previously-included directories found matching 'ticket_stubs'
no previously-included directories found matching 'hacking'
warning: no files found matching 'SYMLINK_CACHE.json'
writing manifest file 'lib/ansible.egg-info/SOURCES.txt'

Setting up Ansible to run out of checkout...

PATH=/root/dep-network/ansible/bin:/root/dep-network/ansible/test/runner:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/root/bin
PYTHONPATH=/root/dep-network/ansible/lib:
MANPATH=/root/dep-network/ansible/docs/man:

Remember, you may wish to specify your host file with -i

Done!
```

3. Verify the ansible command is executable, Enter ansible --version to verify

```
[root@pod5-node1 ansible]# ansible --version
```



```
[root@pod5-node1 ansible]# ansible --version
ansible 2.4.2.0 (detached HEAD e3a8bf02ac) last updated 2018/07/26 04:38:38 (GMT -700)
  config file = /root/dep-network/ansible/ansible.cfg
  configured module search path = [u'/root/.ansible/plugins/modules', u'/usr/share/ansible/plugins/modules']
  ansible python module location = /root/dep-network/ansible/lib/ansible
  executable location = /root/dep-network/ansible/bin/ansible
  python version = 2.7.5 (default, May 31 2018, 09:41:32) [GCC 4.8.5 20150623 (Red Hat 4.8.5-28)]
[root@pod5-node1 ansible]#
```