Ansible introduction

In this lab we will install ansible, prepare required files for running and run introductionary ansible playbooks against Nexus switches.

Ansible installation

Create python environment

Commands:

```
cd ~
pwd
mkdir auto_lab
cd auto_lab/
mkdir env
python3 -m venv env --prompt auto-lab
source env/bin/activate
python3 -m pip install -U pip
```

Example output:

```
[stud1@netauto ~]$ cd ~
[stud1@netauto ~]$ pwd
/home/stud1
[stud1@netauto ~]$ mkdir auto_lab
[stud1@netauto ~]$ cd auto_lab/
[stud1@netauto auto_lab]$ mkdir env
[stud1@netauto auto_lab]$ python3 -m venv env --prompt auto-lab
[stud1@netauto auto_lab]$ source env/bin/activate
(auto-lab) [stud1@netauto auto_lab]$
(auto-lab) [stud1@netauto auto_lab]$ python3 -m pip install -U pip
```

Verify that Your environment is active:

1. Bash prompt starts with environment name "(auto-lab)"
2. Python binary is located in environment's bin directory

```
(auto-lab) [stud1@netauto auto_lab]$ which python3
/home/stud1/auto_lab/env/bin/python3
(auto-lab) [stud1@netauto auto_lab]$ deactivate
[stud1@netauto auto_lab]$ which python3
/usr/bin/python3
```

Install and prepare ansible

Commands:

```
python3 -m pip install ansible
python3 -m pip install ansible-pylibssh
python3 -m pip install paramiko
ansible --version
ansible-galaxy collection install cisco.nxos
ansible-doc cisco.nxos.nxos_vlans
```

Example output:

```
(env) [stud1@netauto auto_lab]$ python3 -m pip install ansible
Collecting ansible
  Using cached
https://files.pythonhosted.org/packages/01/ff/0bdc7b0698f7d4e02e7da6e
96d7d856a42667419c5c48bbfc3f8dda9a80e/ansible-2.10.4.tar.gz

  ... Omitting output ...

Installing collected packages: MarkupSafe, jinja2, PyYAML, six,
pycparser, cffi, cryptography, pyparsing, packaging, ansible-base,
ansible
  Running setup.py install for PyYAML ... done
  Running setup.py install for ansible-base ... done
  Running setup.py install for ansible ... done
Successfully installed MarkupSafe-1.1.1 PyYAML-5.3.1 ansible-2.10.4
ansible-base-2.10.4 cffi-1.14.4 cryptography-3.3.1 jinja2-2.11.2
packaging-20.8 pycparser-2.20 pyparsing-2.4.7 six-1.15.0

(env) [stud1@netauto auto_lab]$ ansible --version
ansible 2.10.4
  config file = None
  configured module search path =
['/home/stud1/.ansible/plugins/modules',
'/usr/share/ansible/plugins/modules']
  ansible python module location =
/home/stud1/auto_lab/env/lib64/python3.6/site-packages/ansible
  executable location = /home/stud1/auto_lab/env/bin/ansible
  python version = 3.6.8 (default, Dec  3 2020, 18:11:24) [GCC 8.4.1
20200928 (Red Hat 8.4.1-1)]

(env) [stud1@netauto auto_lab]$ ansible-galaxy collection install
cisco.nxos
Starting galaxy collection install process
Process install dependency map
Starting collection install process
Installing 'cisco.nxos:1.3.1' to
'/home/stud1/.ansible/collections/ansible_collections/cisco/nxos'
Downloading https://galaxy.ansible.com/download/cisco-nxos-
1.3.1.tar.gz to /home/stud1/.ansible/tmp/ansible-local-
355636fr4i5w3/tmpzo52z5em
cisco.nxos (1.3.1) was installed successfully
```

```
Installing 'ansible.netcommon:1.4.1' to '/home/stud1/.ansible/collections/ansible_collections/ansible/netcommon'
Downloading https://galaxy.ansible.com/download/ansible-netcommon-1.4.1.tar.gz to /home/stud1/.ansible/tmp/ansible-local-355636fr4i5w3/tmpzo52z5em
ansible.netcommon (1.4.1) was installed successfully

(env) [stud1@netauto auto_lab]$ ansible-doc cisco.nxos.nxos_vlans
> CISCO.NXOS.NXOS_VLANS
(/home/stud1/.ansible/collections/ansible_collections/cisco/nxos/plugins/modules/nxos_vlans.py)

        This module creates and manages VLAN configurations on Cisco NX-OS.

  * note: This module has a corresponding action plugin.

  ... Omitting output ...
```

Ansible configuration

1. Configuration file
2. Inventory
3. Credentials
4. Test

Configuration file

```
cat <<'EOF' > ansible.cfg
[defaults]
inventory = ~/auto_lab/hosts
host_key_checking = False
interpreter_python = auto
EOF
```

Inventory

Inventory is ansible's source of information about all managed devices. Documentation

List of devices and their localy significant names:

```
cat <<'EOF' > hosts
ungrouped:
  hosts:
    N9K-1:
      ansible_host: 192.168.227.91
EOF
```

Verify that Ansible inventory is parsed successfully:

```
ansible-inventory --list
ansible-inventory --list -i ./hosts
```

What does flag `-i` mean and why can we omit it?

Expected output:

```
{
    "_meta": {
        "hostvars": {
            "N9K-1": {
                "ansible_host": "192.168.227.91"
            }
        }
    },
    "all": {
        "children": [
            "ungrouped"
        ]
    },
    "ungrouped": {
        "hosts": [
            "N9K-1"
        ]
    }
}
```

Credentials

Since all devices have the same credentials, they can be defined in one place:

```
mkdir group_vars
cat <<'EOF' > group_vars/all.yml
ansible_user: admin
ansible_password: Cisco12345
ansible_network_os: nxos
EOF
```

Testing

Test ansible against one of the devices with and ad-hoc command:

```
ansible all -c network_cli -m cisco.nxos.nxos_ping -a
"dest=192.168.227.91 vrf=management"
```

More about ansible ad-hoc commands here. More about ansible module nxos_ping: `ansible-doc cisco.nxos.nxos_ping`

My first ansible playbook

Playbook is the state definition file that describes automation intention.

We are using Cisco NX-OS ansible module:

1. All NX-OS modules in cisco.nxos collection: link
2. Example module documentation (vlan creation) is here
3. More about playbooks can be found here

Let's go!

```
cat <<'EOF' > playbook.yml
---
- name: First playbook
  hosts: N9K-1
  gather_facts: false
  connection: network_cli

  tasks:

  - name: Ping self
    cisco.nxos.nxos_ping:
      dest: 192.168.227.91
      vrf: management
EOF
```

And run the playbook:

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

Read the output - what can You say from it? Run the playbook in verbose mode:

```
ansible-playbook -v playbook.yml
```