

Ansible Task-1

1. Install and configure ansible

```
curl https://bootstrap.pypa.io/get-pip.py -o get-pip.py
```

```
python get-pip.py
```

```
sudo python -m pip install ansible
```

```
ansible --version
```

```
[nihad@localhost root]$ ansible --version
ansible [core 2.15.11]
  config file = None
  configured module search path = ['/home/nihad/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /home/nihad/.local/lib/python3.9/site-packages/ansible
  ansible collection location = /home/nihad/.ansible/collections:/usr/share/ansible/collections
  executable location = /home/nihad/.local/bin/ansible
  python version = 3.9.10 (main, Feb  9 2022, 00:00:00) [GCC 11.2.1 20220127 (Red Hat 11.2.1-9)] (/usr/bin/python)
  jinja version = 3.1.4
  libyaml = True
```

2. User sandy has been created on your control node with the appropriate permissions

already, do not change or modify ssh keys. Install the necessary packages to run ansible on the control node. Configure ansible.cfg

```
useradd sandy
```

```
passwd sandy
```

```
echo 'sandy ALL=(ALL) ALL' >> /etc/sudoers
```

```
cat /etc/sudoers | grep sandy
```

```
[root@localhost ~]# useradd sandy
[root@localhost ~]# passwd sandy
Changing password for user sandy.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@localhost ~]# echo 'sandy ALL=(ALL) ALL' >> /etc/sudoers
[root@localhost ~]# cat /etc/sudoers | grep sandy
sandy ALL=(ALL) ALL
[root@localhost ~]#
```

```
sandy ALL=(ALL) NOPASSWD:ALL
```

3. to be in folder /home/sandy/ansible/ansible.cfg and configure to access remote machines via the sandy user.

4. All roles should be in the path /home/sandy/ansible/roles. The inventory path should be in /home/sandy/ansible/inventory

```
mkdir -p /home/sany/ansible/
```

```
cd /home/sany/ansible/
```

```
vim ansible.cfg
```

```
[defaults]
inventory = /home/sany/ansible/inventory
roles_path = /home/sany/ansible/roles
remote_user = sandy

[privilege_escalation]
become = true
become_method = sudo
become_user = root
become_ask_pass = false
```

5. You will have access to 5 nodes. node1.example.com, node2.example.com, node3.example.com, node4.example.com, node5.example.com

ssh-keygen

for i in 104 106 107 108 110; do ssh-copy-id sandy@192.168.1.\$i; done

```
[sany@localhost ~]$ for i in 104 106 107 108 110; do ssh-copy-id sandy@192.168.1.$i; done
/usr/bin/ssh-copy-id: INFO: Source of keys) to be installed: /home/sany/.ssh/id_rsa.pub
The authenticity of host '192.168.1.104 (192.168.1.104)' can't be established.
ED25519 key fingerprint is SHA256:143W0E2C54yQ7t1v2w750uMhTtsIDhK3x34Qg.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
/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
sandy@192.168.1.104's password:
Permission denied, please try again.
sandy@192.168.1.104's password:
Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'sandy@192.168.1.104'"
and check to make sure that only the key(s) you wanted were added.

/usr/bin/ssh-copy-id: INFO: Source of keys) to be installed: /home/sany/.ssh/id_rsa.pub
The authenticity of host '192.168.1.106 (192.168.1.106)' can't be established.
ED25519 key fingerprint is SHA256:3p4x0X03yCGRqRL+rkTyhCtUwRE1WpcqdE.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
/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
sandy@192.168.1.106's password:
Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'sandy@192.168.1.106'"
and check to make sure that only the key(s) you wanted were added.

/usr/bin/ssh-copy-id: INFO: Source of keys) to be installed: /home/sany/.ssh/id_rsa.pub
The authenticity of host '192.168.1.107 (192.168.1.107)' can't be established.
ED25519 key fingerprint is SHA256:3p4x0X03yCGRqRL+rkTyhCtUwRE1WpcqdE.
This host key is known by the following other names/addresses:
~/.ssh/known_hosts:4: 192.168.1.106
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
/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
sandy@192.168.1.107's password:
Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'sandy@192.168.1.107'"
and check to make sure that only the key(s) you wanted were added.

/usr/bin/ssh-copy-id: INFO: Source of keys) to be installed: /home/sany/.ssh/id_rsa.pub
The authenticity of host '192.168.1.108 (192.168.1.108)' can't be established.
ED25519 key fingerprint is SHA256:143W0E2C54yQ7t1v2w750uMhTtsIDhK3x34Qg.
This host key is known by the following other names/addresses:
~/.ssh/known_hosts:1: 192.168.1.104
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
/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
sandy@192.168.1.108's password:
Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'sandy@192.168.1.108'"
and check to make sure that only the key(s) you wanted were added.

/usr/bin/ssh-copy-id: INFO: Source of keys) to be installed: /home/sany/.ssh/id_rsa.pub
The authenticity of host '192.168.1.110 (192.168.1.110)' can't be established.
```

ansible all -m ping

```
[sany@localhost ansible]$ ansible all -m ping
192.168.1.106 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
192.168.1.110 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
192.168.1.104 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
192.168.1.108 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
192.168.1.107 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
}
```

6. Configure these nodes to be in an inventory file where node1 is a member of group dev, node2 is a member of group test, node3 is a member of group proxy, node4 and node 5 are members of group prod. Also, prod is a member of group webserver.

```
vim /home/sandy/ansible/inventory
```

```
[dev]
192.168.1.106

[test]
192.168.1.107

[proxy]
192.168.1.104

[prod]
192.168.1.108
192.168.1.110

[webserver:children]
prod
```

7. Create a file called adhoc.sh in /home/sandy/ansible which will use adhoc commands to set up a new repository. The name of the repo will be 'EPEL REPO' the description 'RHEL8 TEST REPO' the base URL is 'https://dl.fedoraproject.org/pub/epel/epel-release-latest-8.noarch.rpm' the GPG key is <http://repo.mysql.com/RPM-GPG-KEY-mysql>, disable gpg check but not enable the repo.

```
cd /home/sandy/ansible/
```

```
vim adhoc.sh
```

```
ansible all -m yum_repository -a 'name="EPEL REPO" description="RHEL8 TEST REPO" file=external_repos baseurl="https://dl.fedoraproject.org/pub/epel/epel-release-latest-8.noarch.rpm" enabled=0 gpgcheck=0 gpgkey="http://repo.mysql.com/RPM-GPG-KEY-mysql"
```

```
[sandy@localhost ansible]$ ./adhoc.sh
192.168.1.108 | CHANGED => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": true,
  "repo": "EPEL REPO",
  "state": "present"
}
192.168.1.104 | CHANGED => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": true,
  "repo": "EPEL REPO",
  "state": "present"
}
192.168.1.110 | CHANGED => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": true,
  "repo": "EPEL REPO",
  "state": "present"
}
192.168.1.106 | CHANGED => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": true,
  "repo": "EPEL REPO",
  "state": "present"
}
192.168.1.107 | CHANGED => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": true,
  "repo": "EPEL REPO",
  "state": "present"
}
```

8. Install nano package in all server with ansible adhoc

```
ansible -m yum -a 'name=nano state=latest'
```

```
[sany@localhost ansible]$ ansible all -m yum -a 'name=nano state=latest'
192.168.1.108 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "msg": "Nothing to do",
  "rc": 0,
  "results": []
}
192.168.1.110 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "msg": "Nothing to do",
  "rc": 0,
  "results": []
}
192.168.1.107 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "msg": "Nothing to do",
  "rc": 0,
  "results": []
}
192.168.1.104 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "msg": "Nothing to do",
  "rc": 0,
  "results": []
}
192.168.1.106 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "msg": "Nothing to do",
  "rc": 0,
  "results": []
}
```

9. Ensure a service is started on all web servers (adhoc)

```
ansible webserver -m service -a 'name=firewalld state=started'
```

```
[sany@localhost ansible]$ ansible webserver -m service -a 'name=firewalld state=started'
192.168.1.110 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "name": "firewalld",
  "state": "started",
  "status": {
    "ActiveEnterTimestamp": "Sat 2024-05-18 23:29:57 +04",
    "ActiveEnterTimestampMonotonic": "47263634",
    "ActiveExitTimestampMonotonic": "0",
    "ActiveState": "active",
    "After": "systemd.target polkit.service basic.target system.slice dbus-broker.service dbus.socket",
    "AllowIsolate": "no",
    "AssertResult": "yes",
    "AssertTimestamp": "Sat 2024-05-18 23:29:49 +04",
    "AssertTimestampMonotonic": "39017820",
    "Before": "network-pre.target multi-user.target shutdown.target",
    "BlockIOAccounting": "no",
    "BlockIOWeight": "[not set]",
    "BusName": "org.fedoraproject.FirewallD1",
    "CPUAccounting": "yes",
    "CPUAffinityFromNUMA": "no",
    "CPUQuotaPerSecUsec": "infinity",
    "CPUQuotaPeriodUsec": "infinity",
    "CPUSchedulingPolicy": "0",

```

10. Create directory /tmp/ansible1 all prod host group servers (ad hoc)

```
ansible prod -m file -a 'path=/tmp/ansible1 state=directory'
```

```
[sany@localhost ansible]$ ansible prod -m file -a 'path=/tmp/ansible1 state=directory'
192.168.1.108 | CHANGED => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": true,
  "gid": 0,
  "group": "root",
  "mode": "0755",
  "owner": "root",
  "path": "/tmp/ansible1",
  "secontext": "unconfined_u:object_r:user_tmp_t:s0",
  "size": 6,
  "state": "directory",
  "uid": 0
}
192.168.1.110 | CHANGED => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": true,
  "gid": 0,
  "group": "root",
  "mode": "0755",
  "owner": "root",
  "path": "/tmp/ansible1",
  "secontext": "unconfined_u:object_r:user_tmp_t:s0",
  "size": 6,
  "state": "directory",
  "uid": 0
}
```

11. Write all open ports to /tmp/listen_ports all proxy host group server (ad hoc)

```
[sany@localhost ansible]$ ansible proxy -m shell -a "ss -ltn | awk 'NR>1 {print \$4}' | awk -F':' '{print \$NF}' > /tmp/listen_ports"
192.168.1.104 | CHANGED | rc=0 >>

[sany@localhost ansible]$ ansible proxy -m command -a 'cat /tmp/listen_ports'
192.168.1.104 | CHANGED | rc=0 >>
22
631
6010
6011
22
631
6010
6011
```