Getting started with Ansible

Duration: 20 Minutes

Introduction

In this lab, we will install Ansible on our Orchestrator using the official Ansible documentation and run simple tests using ad-hoc commands.

Exercise – 1: Installing Ansible

1. Review the official Ansible documentation portal and locate the commands to install Ansible on Ubuntu.

https://docs.ansible.com/ansible/latest/installation_guide/intro_installation.html


```
sudo apt update -y
sudo apt install software-properties-common -y
sudo apt-add-repository --yes --update ppa:ansible/ansible
sudo apt install ansible -y
```


2. Connect to the orchestrator using the bookmarked session in `MobaXterm` and use the commands above to install Ansible.

3. Use the command ***`ansible --version`*** to confirm that ansible is running. Notice the default "config" file location (It is also the **inventory** location).

Note:

- `If you try to run ansible without any flags, you will get the usage message. However, to get a detailed view, use the command` ***`ansible -h`***
- `It is possible to install Ansible using the Python package management "pip". Ansible is written in Python.`

Exercise – 2: Running Ad-Hoc Commands

1. We will create a new inventory for this exercise. Create a new directory called **`adhoc`** and create a new file called **`hosts`.**
2. Edit the hosts file and create one group named `ubuntu` and add the first target `ubuntu_desktop_1` node `10.0.3.61`

3. Disable the SSH Key Checking for Ansible. Create a file `ansible.cfg` in the current directory.


```
admin@ochestrator:~/adhoc$ vi ansible.cfg
admin@ochestrator:~/adhoc$ cat ansible.cfg
[defaults]
host_key_checking = False
admin@ochestrator:~/adhoc$
```

4. Run the ping module to test connectivity to the target nodes using the following adhoc command:

```
ansible ubuntu -m ping -i hosts -u admin --ask-pass
```

```
admin@ochestrator:~/adhoc$ ansible ubuntu -m ping -i hosts -u admin --ask-pass
SSH password:
10.0.3.61 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
```

- Notice that we provided the group of hosts that we would like to run this command against `ubuntu` and the module with `-m` flag and the flag `-i` to specify the `hosts` file where this group exists. We also had to provide ansible with the credentials using `-u` for the user name and –ask-pass to prompt for a password.

5. Change your hosts file and add the IP address of the second ubuntu target.

```
admin@ochestrator:~/adhoc$ cat hosts
[ubuntu]
10.0.3.61
10.0.3.62
```

6. Run the previous adhoc command and notice that it runs against all the hosts in the **ubuntu** group from our inventory file.

```
admin@ochestrator:~/adhoc$ ansible ubuntu -m ping -i hosts -u admin --ask-pass
SSH password:
10.0.3.61 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
10.0.3.62 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
```


Exercise – 3: Writing the first Playbook

Ad-Hoc commands can be useful to run a quick test but they are not repeatable. In this exercise, we will write a simple playbook.

Providing variable can (and should) be provided directly in the inventory file. We will specify the variables using the format `[group_name:vars]`

1. Create a new directory called **`playbooks`** and a sub directory called **`ping_playbook`**.

```
administrator@orchestrator:~/playbooks/ping_playbook$ pwd
/home/administrator/playbooks/ping_playbook
```

2. Inside the current playbook directory **`ping_playbook`**, create a new **hosts** inventory file. We will add to our inventory the two Ubuntu targets however, we will now provide the variables needed to login to the remote nodes.

```
admin@ochestrator:~/adhoc$ cat hosts
[ubuntu]
10.0.3.61
10.0.3.62

[ubuntu:vars]
ansible_user=admin
ansible_ssh_pass=Op1g67323k
```

   - We are adding the variable specifically to the Ubuntu group using the format `[group_name:vars]`.

3. In the same directory, create a simple playbook to use the ping module to test connectivity to both ubuntu servers (either use Tab or spaces not both).

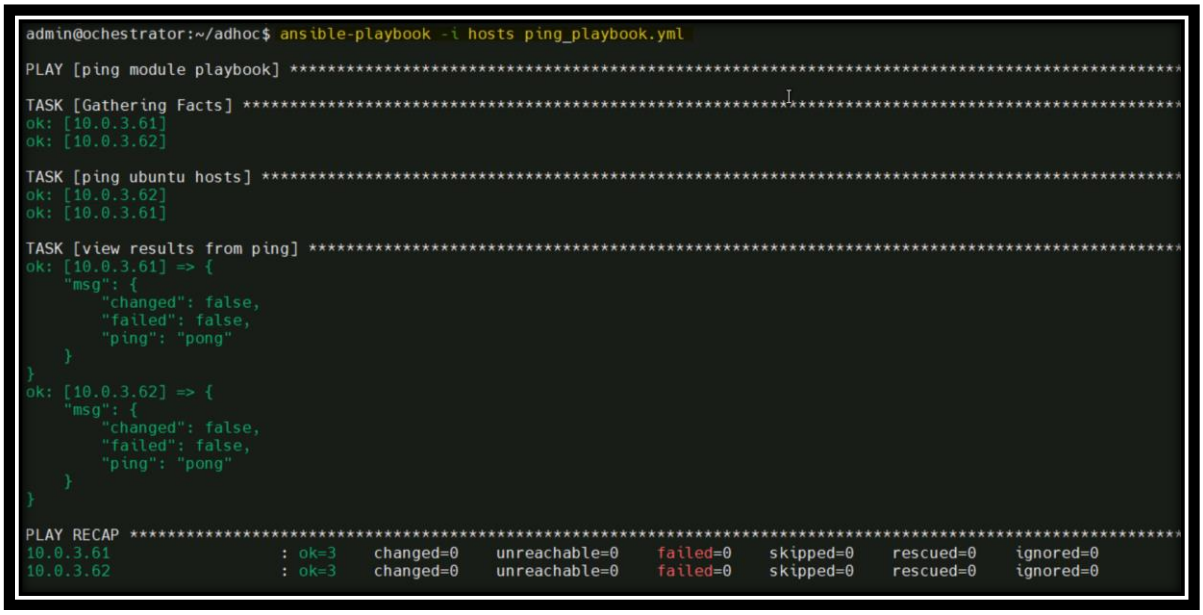
```
admin@ochestrator:~/adhoc$ cat ping_playbook.yml
---
 - name: ping module playbook
   hosts: ubuntu
   tasks:
    - name: ping ubuntu hosts
      ping:
      register: ping_output

    - name: view results from ping
      debug:
       msg: '{{ping_output}}'
```

   - We are specifying the hosts as the group **`ubuntu`**. The first task will run the ping module against the hosts and register the output as a variable called `ping_outout`. The second task will use the debug module to print a message containing the `ping_output` representing the results from the first task.

- To print the value on the ping_output, we must represent it as a variable using the format `'{{variable_name}}'`

4. Run the playbook and specify our inventory using the command below:

   `ansible-playbook -i hosts ping_playbook.yml`



End of Lab 1

