

Getting started with Ansible

Introduction

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

Exercise – 1: Installing Ansible

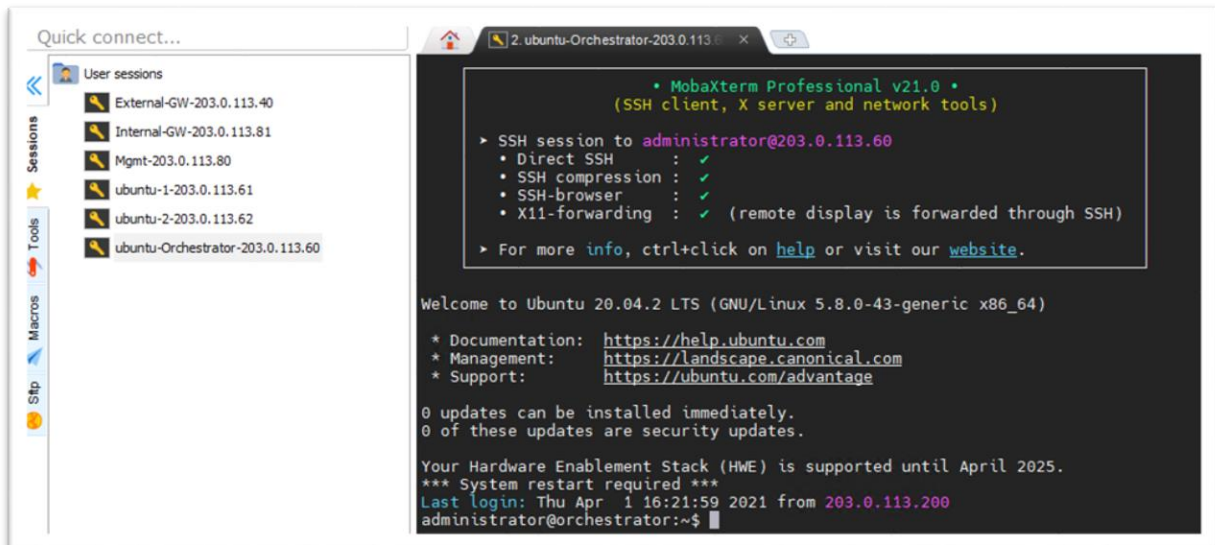
1. Review the official Ansible documentation portal and locate the commands to install Ansible on Ubuntu. The administrator password is **vpn123**

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



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

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).

```
administrator@orchestrator:~$ ansible --version
ansible 2.9.6
  config file = /etc/ansible/ansible.cfg
  configured module search path = ['/home/administrator/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python3/dist-packages/ansible
  executable location = /usr/bin/ansible
  python version = 3.8.5 (default, Jan 27 2021, 15:41:15) [GCC 9.3.0]
```

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 Ubuntu1 node **203.0.113.61**

```
administrator@orchestrator:~/adhoc$ pwd
/home/administrator/adhoc
administrator@orchestrator:~/adhoc$ cat hosts
[ubuntu]
203.0.113.61
```

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

ansible ubuntu -m ping -i hosts -u administrator --ask-pass

```
administrator@orchestrator:~/adhoc$ ansible ubuntu -m ping -i hosts -u administrator --ask-pass
SSH password:
203.0.113.61 | 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.

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

```
[ubuntu]
203.0.113.61
203.0.113.62
```

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

```
administrator@orchestrator:~/adhoc$ ansible ubuntu -m ping -i hosts -u administrator --ask-pass
SSH password:
203.0.113.61 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
203.0.113.62 | 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, 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.

```
[ubuntu]
203.0.113.61
203.0.113.62

[ubuntu:vars]
ansible_user=administrator
ansible_ssh_pass=vpn123
```

- Note that 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.

```
administrator@orchestrator:~/playbooks/ping_playbook$ 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_output**. 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

```

administrator@orchestrator:~/playbooks/ping_playbook$ ansible-playbook -i hosts ping_playbook.yml

PLAY [ping module playbook] *****

TASK [Gathering Facts] *****
ok: [203.0.113.62]
ok: [203.0.113.61]

TASK [ping ubuntu hosts] *****
ok: [203.0.113.61]
ok: [203.0.113.62]

TASK [view results from ping] *****
ok: [203.0.113.61] => {
  "msg": {
    "changed": false,
    "failed": false,
    "ping": "pong"
  }
}
ok: [203.0.113.62] => {
  "msg": {
    "changed": false,
    "failed": false,
    "ping": "pong"
  }
}

PLAY RECAP *****
203.0.113.61      : ok=3    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
203.0.113.62      : ok=3    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

```

End of Lab 1