F5 Ansible Automation Lab 1.3

F5 Ansible Automation Blueprint-v1.2

<https://github.com/ikhanf5/ansiblerepo>

Credentials :

**F5-BIPv13:**

SSH : root / default

Webui : admin/admin

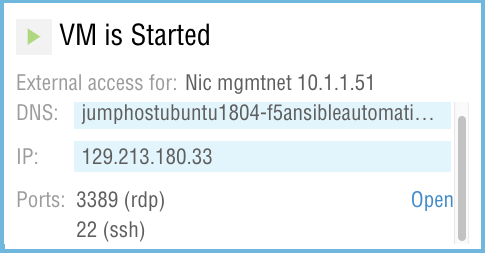
**UBUNTU Ansible Host:**

SSH : f5student / f5DEMOs4u#

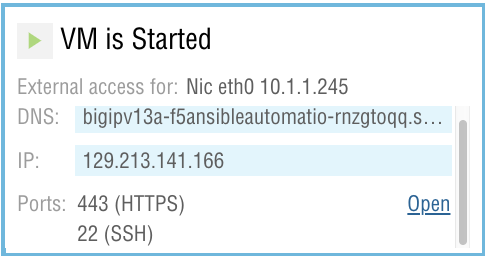
Console : f5student / f5DEMOs4u#

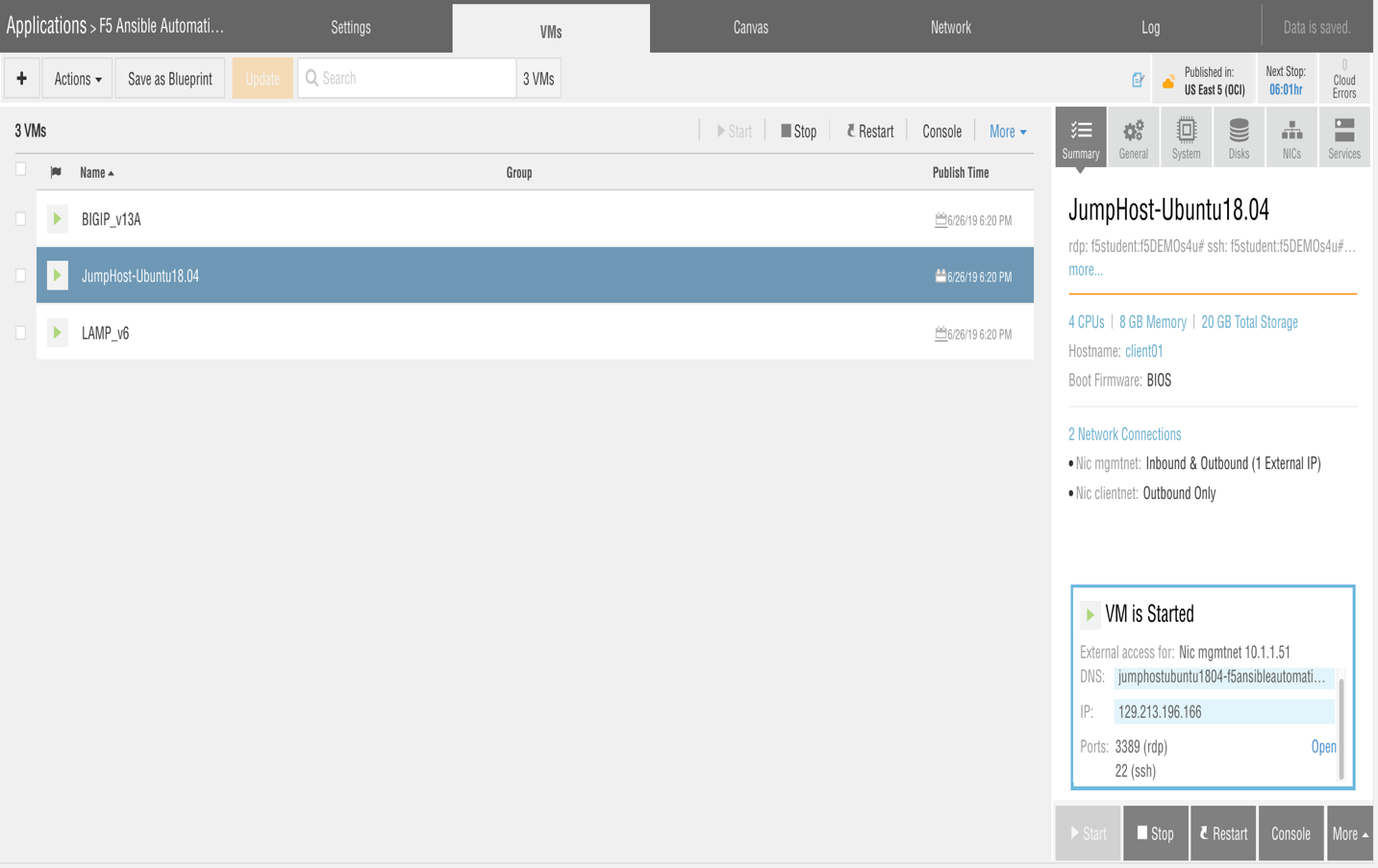
Windows PC can use putty to ssh to ansible host. Mac User can use native ssh from Terminal. To obtain the IP of the Ubuntu Host. Click the Ubuntu Host and see the bottom right corner of the screen.

NOTE: Every Lab will have a different IP address. The IP below is not the right IP. You will need to obtain the IP from the lab assigned to you.



For https access to the BIG IP click the BIG-IP Device in Revello Labs. On the bottom right Click on Open.





LAB 1 : Installing Ansible in Virtualenv

LAB 2 : Running First Playbook

LAB 3 : Using Tags in Ansible Playbook

LAB 4 : Using Variable Files in Ansible Playbook

LAB 5 : Importing a New ASM Policy using a Template XML file

LAB 1 : Installing Ansible in Virtualenv

For Purpose of this lab you need access to the following

* Ubuntu SSH
* Login to Ansible Jump Host via ssh. See above for how to obtain IP and credentials.

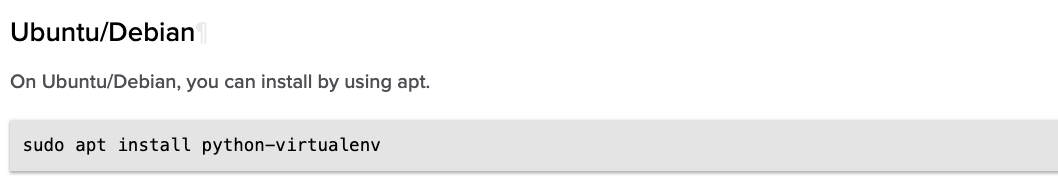
Type ls -ll

You will see demo1/demo2/demo3 folders will contains ansible playbook for this lab.

F5 Recommends installing ansible on a python virtualenv. This is also a recommended way to use ansible within the ansible opensource community.

Virtualenv creates isolated Python environments to avoid problems caused by conflicting dependencies and differing versions. Virtualenv works by simply creating a folder which contains all of the necessary executables and dependencies for a specific version of Python

* Install Python virtualenv type following at the prompt
  + # sudo apt install python-virtualenv
  + Enter password f5DEMOs4u# when prompted

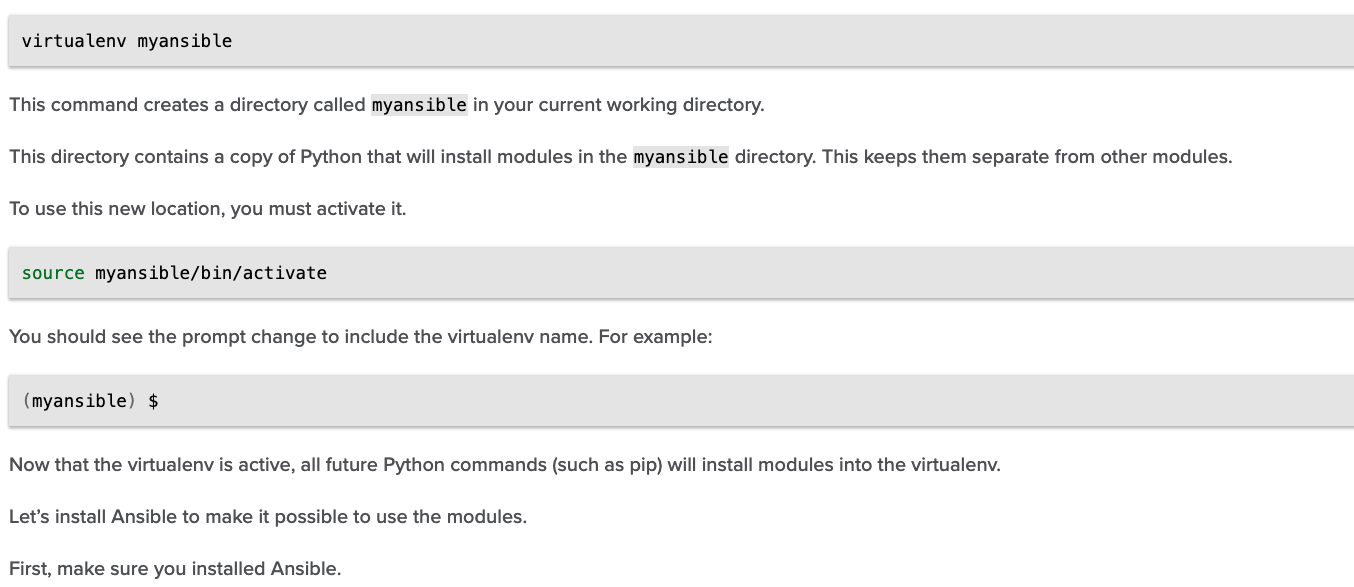


* Create Virtualenv myansible
  + # virtualenv myansible

Note : This will create my myansible folder in your working directory with python libraries.

* Activate Virtualenv myansible
  + # source myansible/bin/activate

# Note the prompt will change to the virtualenv

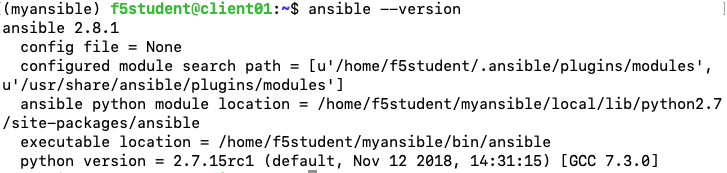


Install Ansible in the myansible(virtualenv)

* # pip install ansible

Check Ansible version is 2.8.1

* # ansible --version



# Note: Keep this session open for the next LAB

LAB 2 : Running First Playbook

For Purpose of this lab you need access to the following

* BIGIP-WEBUI
* Ubuntu SSH
* Open BIG-IP Web-UI
* Login using admin/admin
* Check the LTM Nodes / LTM Pool / LTM Virtual Server already existing.

Running your first ansible playbook :

In the ssh session for UBUNTU Jumphost type the following command

# cd demo1

# ls

Notice there is there are two files first.yaml and inventory directory which contains the host file.

Type # cat first.yaml |more

Notice there are 3 plays in the task section.

1. Create a pool
2. Add members to pool
3. Create a Virtual Server

Type the following command to run the ansible playbook.

# ansible-playbook -i inventory/hosts first.yaml

* Open BIG-IP Web-UI
* Login using admin/admin
* Click on Local Traffic 🡪 Virtual Server (Confirm new virtual servers)
* Click on Local Traffic 🡪 Pools (Confirm new Pools)
* Click on Local Traffic🡪 Nodes (Confirm new Nodes)

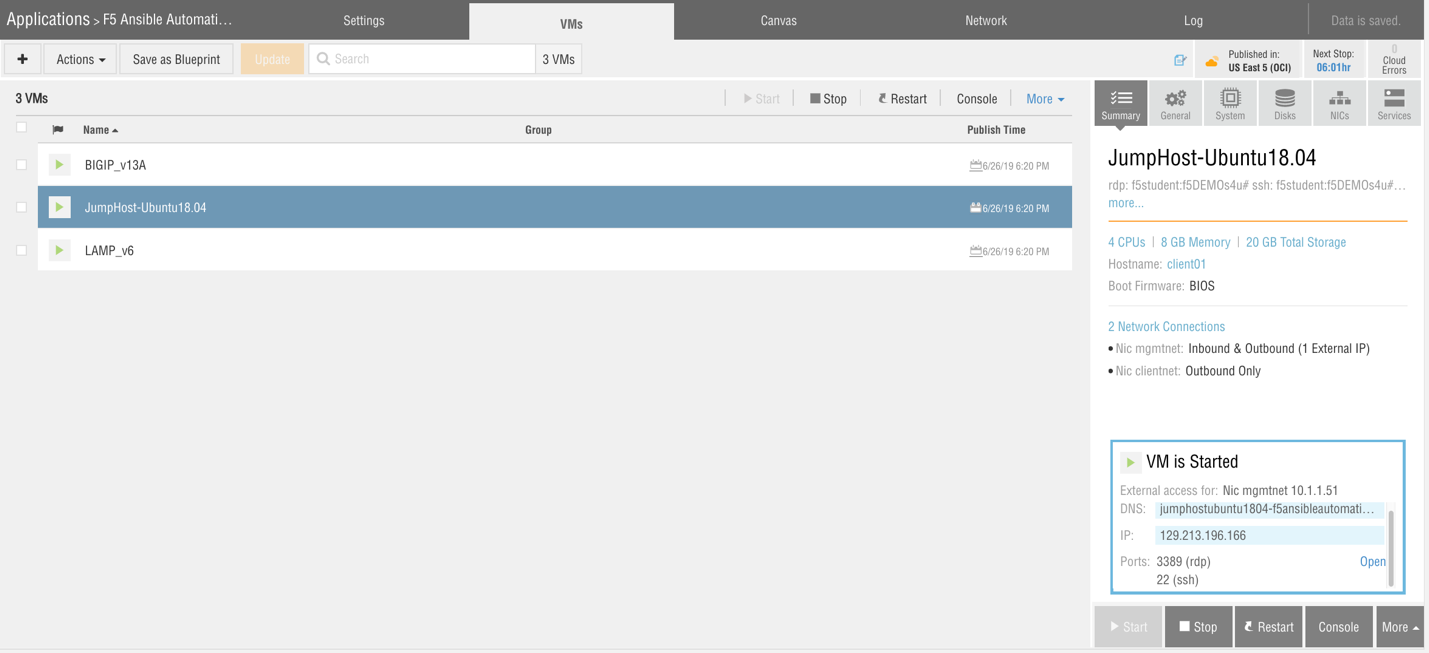
# Note: Keep this BIG-IP WebUI and SSH session open for the next LAB

LAB 3 : Using Tags in Ansible Playbook

<https://docs.ansible.com/ansible/latest/user_guide/playbooks_tags.html>

For Purpose of this lab you need access to the following

* BIGIP-WEBUI
* Ubuntu SSH



* Login to the BIG-IP Web-UI
* Click on Local Traffic 🡪 Virtual Server
* Select the Virtual Server web\_vip
* Click Delete

Run Playbook first.yaml using virtual tags to run only play to Add virtual. Other Plays in the playbook will be omitted. Make sure you are still in the virtualenv and demo1 directory from your previous session.

# ansible-playbook -i inventory/hosts first.yaml --tags virtual

* Login to the BIG-IP Web-UI
* Click on Local Traffic 🡪 Virtual Server
* Check to see if the virtual server was created again.

Run Playbook firstdel.yaml to delete all the virtual server / pool / nodes.

# ansible-playbook -i inventory/hosts firstdel.yaml

* Login to the BIG-IP Web-UI
* Click on Local Traffic 🡪 Virtual Server (Confirm no virtual servers)
* Click on Local Traffic 🡪 Pools (Confirm no Pools)
* Click on Local Traffic🡪 Nodes (Confirm no Nodes)

LAB 4 : Using Variable Files in Ansible Playbook

For Purpose of this lab you need access to the following

* BIGIP-WEBUI
* Ubuntu SSH
* Ubuntu GUI Console

demo2/second.yaml 🡪 playbook

demo2/netvar.yaml 🡪 variable file

This playbook uses a separate variable file. This is helpful as the playbook does not change and can be repeated again and again without any modification. This section calls the netvar.yaml file in the playbook.



For the purpose of this demo the playbooks are using loops to create multiple nodes and virtual servers.

<https://docs.ansible.com/ansible/latest/user_guide/playbooks_loops.html>

Run Playbook second.yaml

# cd ..

# cd demo2

# ansible-playbook -i inventory/hosts second.yaml

* Open BIG-IP Web-UI
* Login using admin/admin
* Click on Local Traffic 🡪 Virtual Server (Confirm new virtual servers)
* Click on Local Traffic 🡪 Pools (Confirm new Pools)
* Click on Local Traffic🡪 Nodes (Confirm new Nodes)

Run Playbook seconddel.yaml to delete all the virtual server / pool / nodes.

# ansible-playbook -i inventory/hosts seconddel.yaml

* Login to the BIG-IP Web-UI
* Click on Local Traffic 🡪 Virtual Server (Confirm no virtual server)
* Click on Local Pools 🡪 Virtual Server (Confirm no Pools)
* Click on Local Nodes 🡪 Virtual Server (Confirm no Nodes)

# Note: Keep this BIG-IP WebUI and SSH session open for the next LAB

LAB 5 : Importing a New ASM Policy using a Template XML file

For Purpose of this lab you need access to the following

* BIGIP-WEBUI
* Ubuntu SSH
* Ubuntu GUI Console
* Open the WebUI for BIG-IP
* Click Security 🡪 Application Security 🡪 Security Policy
* Make sure there is no ASM Security Policy on the device.
* In the Ubuntu Gui Console open using texteditor
* demo2/second.yaml
* demo2/netvar.yaml
* Open the ssh session on Ubuntu Host
* Run Playbook second.yaml

# cd ..

# cd demo3

# ansible-playbook -i inventory/hosts third.yaml

* Open BIG-IP Web-UI
* Login using admin/admin
* Click Security 🡪 Application Security 🡪 Security Policy
* Check to make sure the ASM security policy has been created.