**FA18: CMPE-272 Sec 01 - Enterprise Software Platforms**

**Homework #1**

**Team Members**

1. Sumanth Ravipati
2. Shilpi Kumari
3. Poorva Agarwal | SJSU Id 013770939
4. Sai Kiran Pothana
5. Harshit Nagpal

**GitHub Repository:** <https://github.com/Shilpi-Kumari/avengers>

**Requirement:**

* Configure Ansible to deploy webserver, and bring it up a port 80 with a web page that is publicly accessible that displays the message: “Hello World”.
* Include in the Ansible playbook, plays to deploy and un-deploy the resources

**Steps to followed:**

1. Install Ansible.
2. Create Amazon EC2 instance and enable port 80 for HTTP connections.
3. Create a playbook to deploy webserver on EC2 instance.
4. Copy the HelloWorld HTML file on EC2 instance using ansible playbook.
5. Un-deploy the webserver from EC2 instance through ansible playbook.

**Install Ansible:**

**Step 1:** In order to install Ansible on our server machine we need to have Python and pip installed. First log onto terminal point on MAC and run the below commands to install Python and pip.

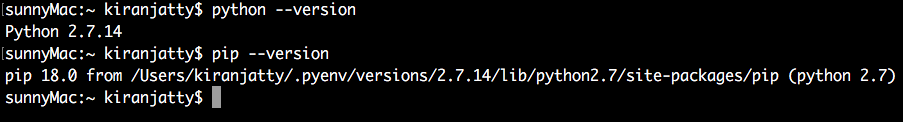
Assuming if you are using MAC, and you already have XCode and Homebrew installed on your machine.

***Python Install:*** *$ brew install python.*

***Pip Install:*** *$ sudo easy\_install pip.*

Once you have installed python and pip, to verify whether they have successfully installed check for the versions.

***$ python --version, $ pip --version***. You will get the below screen based on the versions in your server machine.

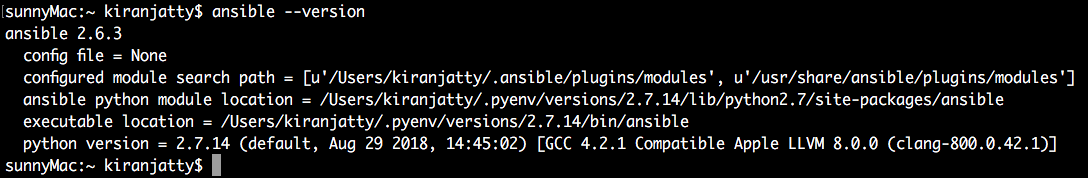


**Step 2:** After successfully installing python and pip now install Ansible on the server machine.

***Ansible Install:*** *$ sudo pip install ansible --quiet*

Once you have installed Ansible check for the version,

***$ ansible --version.***



**Step 3:** Just before creating the EC2 instance in AWS, try to ping the localhost server of the server machine to check for the ansible connectivity to the local machine.

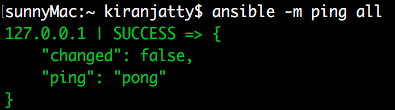
In order to ping the local server, we need to add the entries into host file in the location

“/etc/ansible/hosts”.

***CMD to edit host file***: *$ vi /etc/ansible/hosts*

../../Screen%20Shot%202018-09-09%20at%204.09.42%20PM.png

***CMD to ping all ansible modules:*** *$ ansible –m ping all*

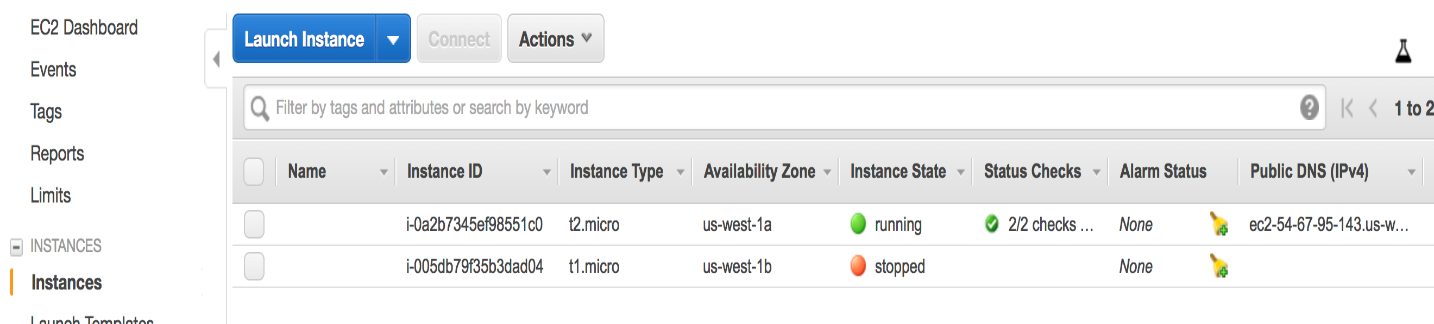


**EC2 Instance Creation**

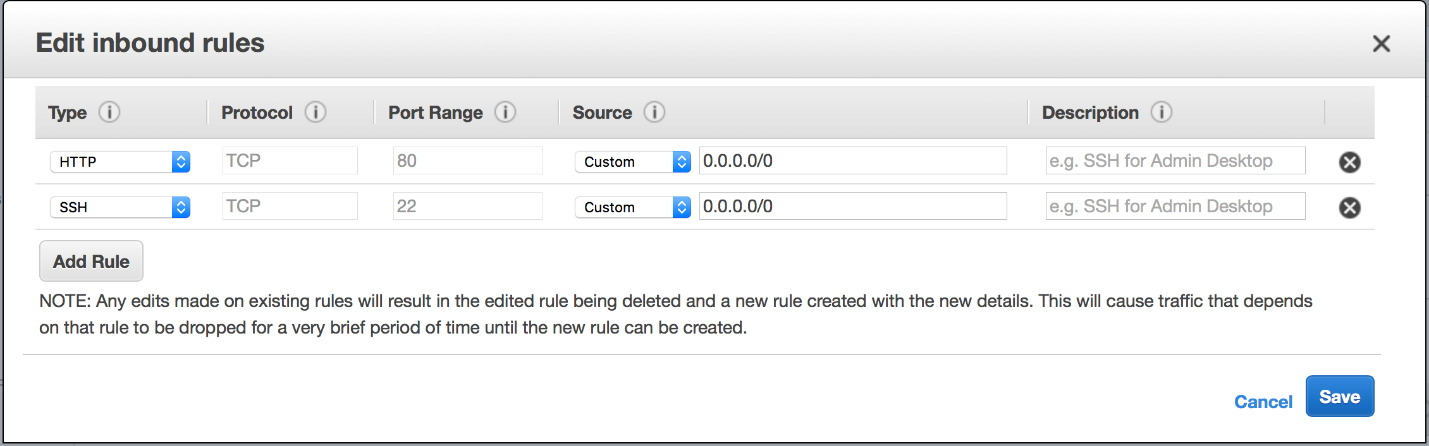
In order to deploy a webserver using ansible from server machine we need to create one client machine on cloud. Listed below process for the same:

1. Go to <https://aws.amazon.com/> for creation of EC2 instance
2. Select EC2 service and lunch the instance
3. Select operating system according to the requirement. We have selected free tier version of Ubuntu 16.04.
4. Review the instance and launch

Screenshot of our EC2 instance:



1. Go to security group of the EC2 instance and edit the inbound rules to enable port 80 for HTTP request.



**SSH Key Generation:**

For setting up connection between our machine and EC2 instance, we need to generate a ssh key on our machine which will be copied on EC2 instance for connection. Below are the steps to do the same:

1. First, we need to check whether we are able to SSH (Secure Shell) from server machine to the client machine using EC2 key.

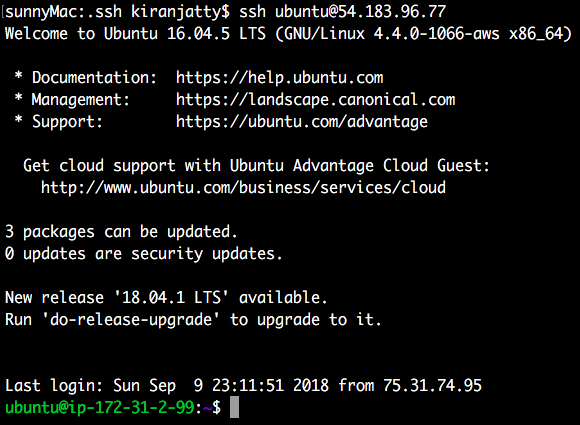
In order to check SSH we need to first place the (Private\_EC2\_Key.pem) file in the .ssh folder in the server machine and edit config file of .ssh folder to map the (Private\_EC2\_Key.pem) so that it won’t prompt us for password whenever we do ssh to client machine.

***Cmd to edit config file:*** *$ vi config* (Assuming you are already in .ssh folder)

***Value in config file:*** *IdentityFile ~/.ssh/Private\_EC2\_Key.pem*

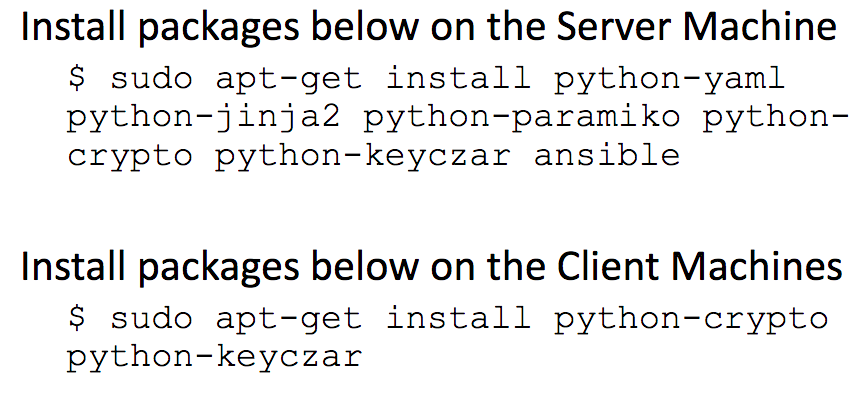
Now that, all the configurations are done try to ssh from server machine to client machine and should be able to connect.

*$ ssh user@ip\_address* (user = Ubuntu, ip\_address = address of the VM we created)



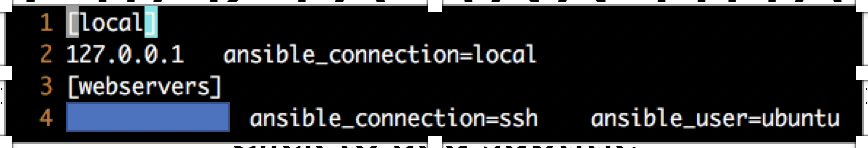
1. Follow below steps on terminal to generate an SSH key
2. Run $ ssh-keygen
3. Give below inputs:
4. Enter file in which to save the (/Users/testuser/.ssh/id\_rsa):
5. Enter passphrase (empty for no passphrase): leave it blank and enter
6. Enter same passphrase again: leave it blank and enter
7. The public key will be saved at /Users/testuser/.ssh/id\_rsa.pub on your machine. Paste this key to /home/ubuntu/.ssh/authorized\_key on EC2 instance.

1. Install the below mentioned packages on both Server machine and Client Machine

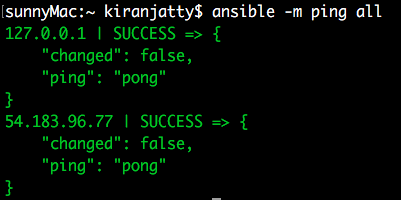


1. Ping the client machine to check for the ansible connectivity, add the ip\_address in hosts file in location “/etc/ansible/hosts”

*$ vi /etc/ansible/hosts*



*$ ansible –m ping all*



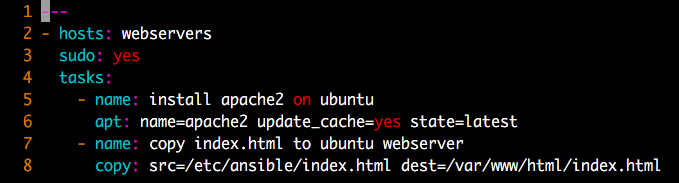
**Create a HTML file to display Hello World**

***HTML file:*** Created in location “/etc/ansible/index.html”.



**Create a playbook to deploy Apache web server and copy HTML file**

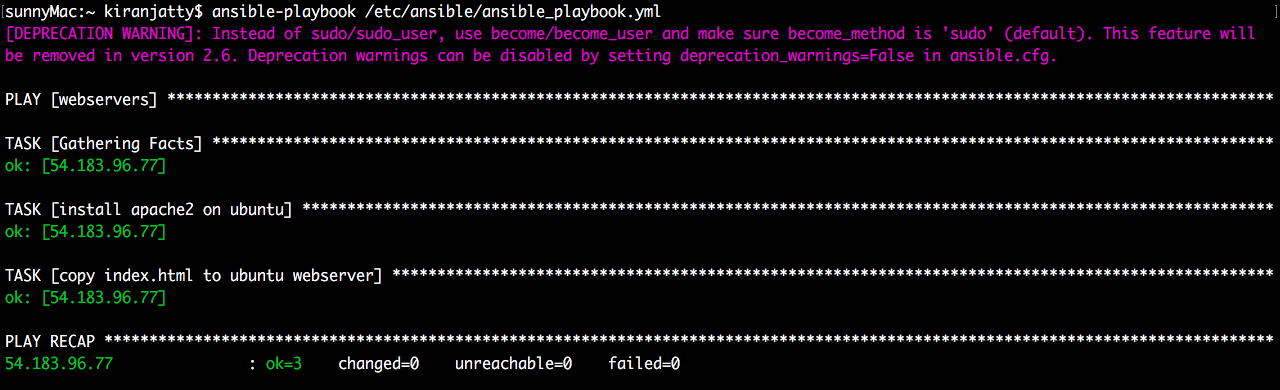
1. ***Playbook creation:*** Created in location “/etc/ansible/apache\_install.yml”. We can also create in different location as well.



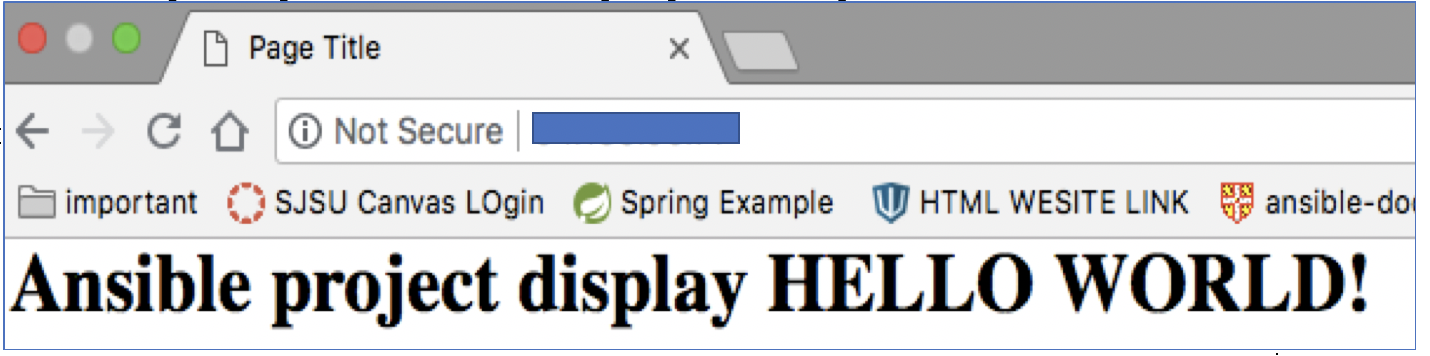
* **hosts: webserver** -it indicates that we are using webservers hosts group for connection for ansible.
* **sudo: yes** – for right user privilege
* **tasks:** - it has two tasks as mentioned above. Task1 is install apache2 webserver and Task2 to copy the index.html file on the /var/www/html/index.html of EC2 instance .

1. Now, run the playbook via ansible to create apache server and deploy the html file on client machine

*$ ansible-playbook /etc/ansible/* *apache\_install.yml*



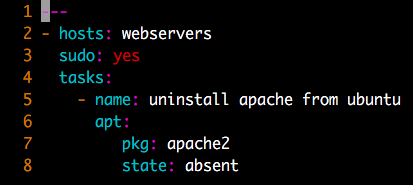
1. Resources are deployed successfully, check for the Hello World message on browser.

**

**Un-deploy the Apache server from EC2 instance:**

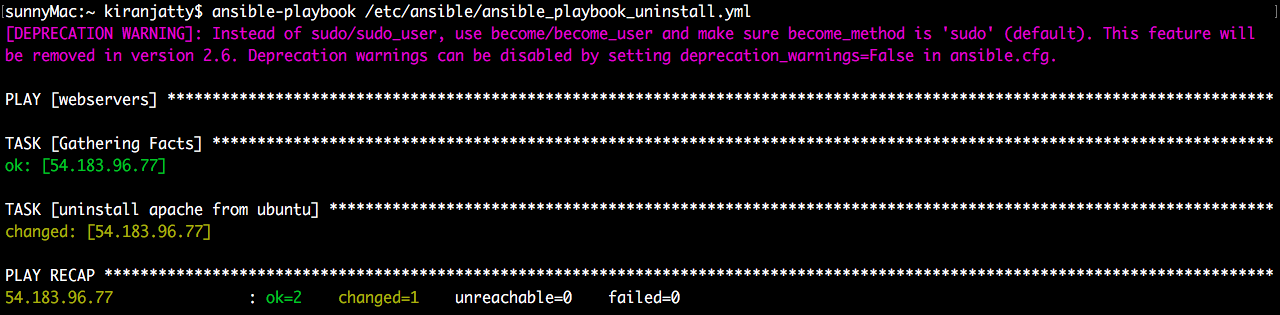
1. As we have done with the message display just write a play book to un deploy the apache server from the client machine.

***Playbook creation:*** Created in location “/etc/ansible/ apache\_uninstall.yml”.



1. Execute the playbook to uninstall the server

*$ ansible-playbook /etc/ansible/apache\_uninstall.yml*



1. Apache server is uninstalled from the machine, now lets check the web browser whether we are able to view the Hello World message.

