**Ansible Homework**

1. **On Physical Machine**

**Step 1: Install Ansible in MAC OS X**

1. Open Terminal (Command + space -> search Terminal)
2. Install pip

$ sudo easy\_install pip

1. Check if pip is installed successfully

$ pip –V

we will get output something like this:

pip 9.0.1 from /Library/Python/2.7/site-packages/pip-9.0.1- py2.7.egg (python 2.7)

1. Install Ansible from source. We need to download Ansible first. Enter below command in terminal:

$ git clone git://github.com/ansible/ansible.git

This may ask to agree to certain terms and conditions. We will get output asbelow**:**

Cloning into 'ansible'...

remote: Counting objects: 254412, done.

remote: Compressing objects: 100% (118/118), done.

remote: Total 254412 (delta 70), reused 114 (delta 39), pack-reused 254249

Receiving objects: 100% (254412/254412), 77.52 MiB | 13.75 MiB/s, done.

Resolving deltas: 100% (165528/165528), done.

1. Type below command to start installing Ansible:

$ sudo pip install ansible

Last line after successfully installation will be as below:

Successfully installed ansible-2.3.2.0 pycrypto-2.6.1

1. To check if Ansible is installed properly, type:

$ ansible —version

It will give details of the sensible installed (something as follows)

ansible 2.3.2.0

config file =

configured module search path = Default w/o overrides

python version = 2.7.10 (default, Feb 7 2017, 00:08:15) [GCC 4.2.1 Compatible Apple LLVM 8.0.0 (clang-800.0.34)]

**Step 2: Create SSH key**

1. Type below command in terminal window:

$ ssh-keygen –t rsa

This may ask to select location and passphrase. You can hit enter for default settings. Finally, it will give details of location where key is created, key fingerprint and key’s randomart image. Private key is saved in **id\_rsa** file in **.ssh** directory.

**Step 3: Create an EC2 instance**

1. Launch an EC2 instance, go to [https://console.aws.amazon.com/ec2/v2/home?region=us-west-2#](https://console.aws.amazon.com/ec2/v2/home?region=us-west-2) and click on **Launch** Instance.
2. You will get the list of Amazon Machine Image. Search for Amazon Linux AMI and click **Select**.
3. You need to select the instance type now. Select **t2.micro** that is eligible for free tier. It is indicated by green color in the list.
4. Click on **Review and Launch**.
5. You will get the configuration and other information for the machine that you created. Click on **Launch**
6. Now it will ask to create a key pair. Select **Create a new key pair**.
7. Give it the name as **MyKeyPair** and click on **Download Key Pair**.
8. Move the downloaded key to .ssh sub-directory. For this go to terminal and give below command.

$ mv ~/Downloads/MyKeyPair.pem ~/.ssh/MyKeyPair.pem

1. Click on **Launch Instance.**
2. Now click on **View Instance.** The Instance State should be running after some time.

Note the public IP address give to this instance. Here we got **35.165.189.133**

**Step 4: Configure security group**

1. Go to **security group** of the EC2 instance that is created.
2. Click on the **inbound** tab and click on **Edit**.
3. Click on **Add Rule.**
4. Select Type as **HTTP** and select Source as **Anywhere**.
5. Add one more rule with Type as **HTTPS** and Source as **Anywhere**.
6. Click **Save**.

**Step 5: Connect to EC2 instance using terminal**

1. Type the below command in terminal to restrict your private key publicly viewable (you need to do this only one time when you create a new key in ec2 instance).

$ chmod 400 ~/.ssh/Mykeypair.pem

1. Give the below command in terminal to connect to ec2 instance (you will give the IP that you got)

ssh -i ~/.ssh/MyKeyPair.pem ec2-user@35.165.189.133

1. For the first time, it will ask if you want to continue connecting. Type yes and hit enter.

The authenticity of host '35.165.189.133 (35.165.189.133)' can't be established.

ECDSA key fingerprint is SHA256:EO1kGcVX5EYfYHxA8wthzQBhA1HGPafF8vXELgZFsFM.

Are you sure you want to continue connecting (yes/no)? yes

1. Once connected successfully you will get information as below:

Warning: Permanently added '35.165.189.133' (ECDSA) to the list of known hosts.

Last login: Fri Sep  8 19:03:07 2017 from c-73-222-238-168.hsd1.ca.comcast.net

       \_\_|  \_\_|\_  )

       \_|  (     /   Amazon Linux AMI

      \_\_\_|\\_\_\_|\_\_\_|

https://aws.amazon.com/amazon-linux-ami/2017.03-release-notes/

8 package(s) needed for security, out of 8 available

Run "sudo yum update" to apply all updates.

**Step 6: Create host file**

1. Go to terminal window and create a hosts file in etc/ansible directory

$ mkdir /etc/ansible

$ touch /etc/ansible/hosts

$ export ANSIBLE\_HOSTS=/etc/ansible/hosts

1. Open the file created

$ vi /etc/ansible/hosts

1. Give the public IP address of ec2 instance that we created

[Server1]

35.165.189.133

**Step 7: Create web application**

1. Create a simple html file that prints Hello World

<html>

<body>

<h2>Demo</h2>

<p>Hello World</p>

</body>

</html>

1. Save the file using .j2 extension. The .j2 extension tells Ansible that it’s a template file. We are saving the file as **index.html.j2**

**Step 8: Create Ansible playbook**

1. Create an Ansible playbook to install nginx on your server and to deploy your web application

- hosts: Server1

  tasks:

   - name: Nginx setup

     yum: pkg=nginx state=installed update\_cache=true

   - name: Nginx service restart

     service:

       name: nginx

       state: started

   - name: index.html copy

     template: src=index.html.j2 dest=/usr/share/nginx/html/index.html

...

1. Save the file as yml file. Here we are giving it name as **server-setup.yml**

**Step 9: Run Playbook to deploy the application created above**

1. To run playbook, give below commands in terminal**:**

$ ansible-playbook -i /etc/ansible/hosts -s -u ec2-user --private-key ~/.ssh/MyKeyPair.pem server-setup.yml

1. You will get the following output once playbook runs successfully:

PLAY [Server1] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

TASK [Gathering Facts] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [35.165.189.133]

TASK [Nginx setup] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [35.165.189.133]

TASK [Nginx service restart] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [35.165.189.133]

TASK [index.html copy] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

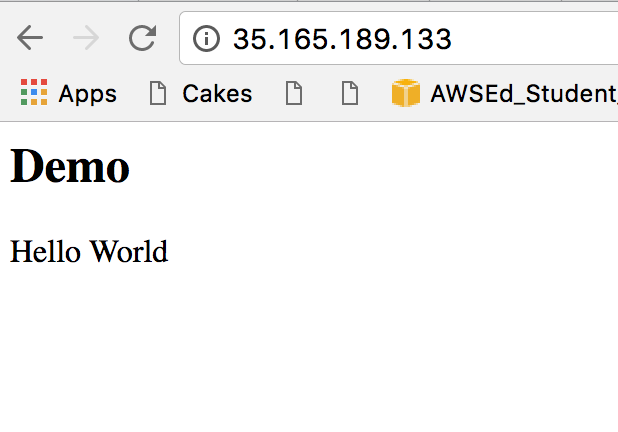
changed: [35.165.189.133]

PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

35.165.189.133             : ok=4    changed=1    unreachable=0    failed=0

**Step 10: View your application**

Give the public IP address of ec2 instance that you created in browser and you will be able to view you application as shown below:

****

**Step 11: Un-deploy the application using Ansible playbook**

1. Open Ansible playbook that we created in step 7

$ vi server-setup.yml

1. Create a new task to stop the service

tasks:

   - name: Nginx service stop

     service:

       name: nginx

       state: stopped

1. After saving the file, run the playbook and the service will be stopped.
2. **Using virtual machine**

**Step 1: Installing Ansible on CentOS**

sudo yum install –y epel-release

sudo yum install –y ansbile

**Step 2: Check if Ansbile is installed**

[abhinaya@localhost ~]$ ansible --version  
ansible 2.3.1.0  
  config file = /etc/ansible/ansible.cfg  
  configured module search path = Default w/o overrides  
  python version = 2.7.5 (default, Aug  4 2017, 00:39:18) [GCC 4.8.5 20150623 (Red Hat 4.8.5-16)]  
[abhinaya@localhost ~]$

**Step 3: Create an EC2 instance**

1. Launch an EC2 instance, go to

https://console.aws.amazon.com/ec2/v2/home?region=us-west- 2# and click on

Launch Instance.

2. You will get the list of Amazon Machine Image. Search for Amazon Linux AMI and

click Select.

3. You need to select the instance type now. Select t2.micro that is eligible for free tier.

It is indicated by green color in the list.

4. Click on Review and Launch.

5. You will get the configuration and other information for the machine that you created.

Click on Launch

6. Now it will ask to create a key pair. Select Create a new key pair.

7. Give it the name as abhi\_ansible and click on Download Key Pair.

8. change the permissions of the nsible.pem file

chmod 400 ansible.pem

9.Click on Launch Instance.

10. Now click on View Instance. The Instance State should be running after some time.

Note the public IP address give to this instance. Here we got 54.219.175.229.

**Step 4: Add the ec2 instance public ip in the /etc/ansible/hosts file**

root@localhost tasks]$ tail -f /etc/ansible/hosts

# Here's another example of host ranges, this time there are no

# leading 0s:

## db-[99:101]-node.example.com

[ec2-name]

54.219.175.229

**Step 5: Create a Hello World html file**

[root@localhost ~]$ cat index.html.j2

<html>

<body>

<p> Hello World </p>

</body>

</html>

[root@localhost ~]$

**Step 6: Create the main deployment file**

[root@localhost ansible-nginx]$ cat main\_deply.yml

# ./ansible-nginx/main\_deploy.yml

- hosts: ec2-name

tasks:

- include: 'tasks/install\_nginx.yml'

[root@localhost ansible-nginx]$

This main deployment file has the include module to specify configuration file to install redhat linux variant of NGINX. The hosts module in this deployment file specifies ansible to deploy all servers in the ec2-name group.

File which installs and starts nginx and has an include module to copy the index.html file

[root@localhost tasks]$ cat install\_nginx.yml

# ./ansible-nginx/tasks/install\_nginx.yml

- name: Install red hat nginx repo rpm

yum:

name: http://nginx.org/packages/rhel/7/noarch/RPMS/nginx-release-rhel-7-0.el7.ngx.noarch.rpm

- name: Install nginx

yum:

name: nginx

state: latest

- name: start nginx

service:

name: nginx

state: started

# copy the index.html.j2 to ngnix webserver

- include: 'helloWorld.yml'

[root@localhost tasks]$

File to copy the index.html file to nginx:

[root@localhost tasks]$ cat helloWorld.yml

# ./ansible-nginx/tasks/hello\_World.yml

- name: hello world index page

template: src=/home/abhinaya/index.html.j2 dest=/usr/share/nginx/html/index.html

[root@localhost tasks]$

**Step 7: Run Ansible playbook to deploy the application created above:**

[root@localhost ansible-nginx]$ sudo ansible-playbook -s -u ec2-user --private-key /home/abhinaya/Downloads/ansible.pem main\_deply.yml

PLAY [ec2-name] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

TASK [Gathering Facts] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [54.219.175.229]

TASK [Install red hat nginx repo rpm] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [54.219.175.229]

TASK [Install nginx] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [54.219.175.229]

TASK [start nginx] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [54.219.175.229]

TASK [hello world index page] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

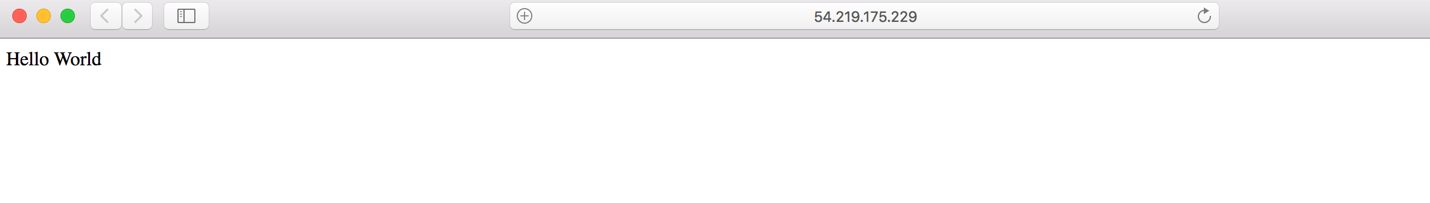
ok: [54.219.175.229]

PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

54.219.175.229 : ok=5 changed=0 unreachable=0 failed=0

[root@localhost ansible-nginx]$

View Application:



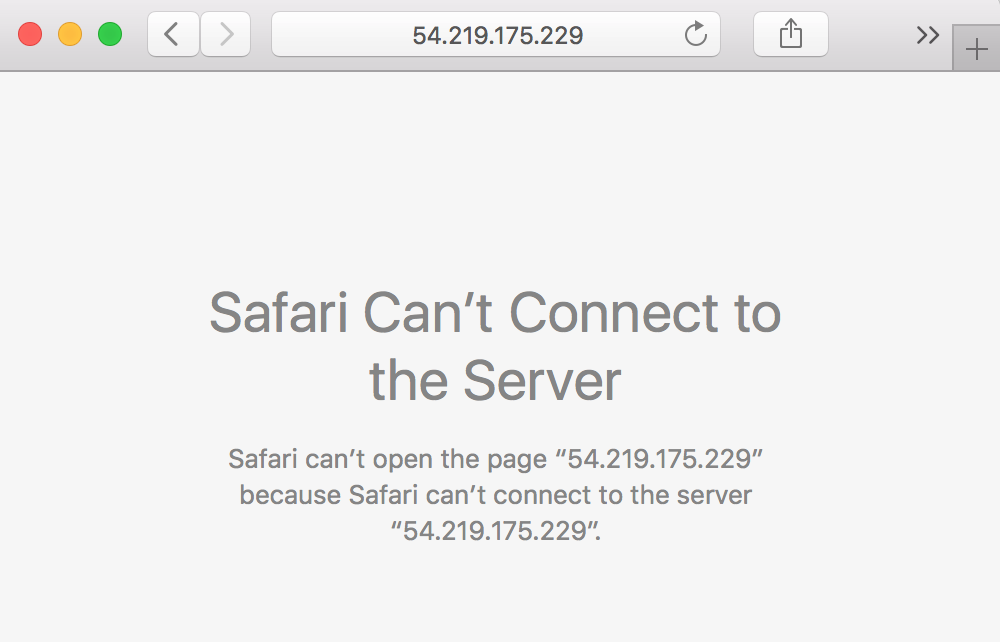
**Step 8: Ansible Playbook to stop Nginx webserver:**

[root@localhost ansible-nginx]$ cat undeploy.yml   
- hosts: ec2-name  
  
  tasks:  
   - service:  
      name: nginx   
      state: stopped  
[root@localhost ansible-nginx]$

Run Ansible Playbook to stop Nginx Web server:

root@localhost ansible-nginx]$ sudo ansible-playbook -s -u ec2-user --private-key /home/abhinaya/Downloads/ansible.pem undeploy.yml

Cannot access Nginx webserver after stopping:



----------------------