

Sukhvir Singh
Shawn Darrel Suarez
Xueli Yang

HW 1 - Ansible

Github code repo: <https://github.com/SukhvirS/cmpe172-hw1>

Steps:

- Since we connect to the server using SSH, we need to first generate a public/private SSH key-pair using the command **ssh-keygen**. This creates two separate files: id_rsa (private key) and id_rsa.pub (public key)
- When starting a new EC2 instance, we pass AWS the public key file to that you can connect to it later using **ssh -i "id_rsa.pem"**

ec2-user@ec2-3-208-9-180.compute-1.amazonaws.com



```
np-fpm.d __| __|_ )
ostfix _| ( / Amazon Linux 2 AMI
op ____|\\____|____|

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-20-217 ~]$
```

- Once connected to the server, we can go to the AWS Console online to get our public IP address (highlighted in blue in the picture)

The screenshot shows the AWS Management Console interface. On the left is a navigation menu with categories like INSTANCES, IMAGES, ELASTIC BLOCK STORE, NETWORK & SECURITY, LOAD BALANCING, and AUTO SCALING. The main area displays a table of EC2 instances. One instance is listed with ID 'i-0c21e9b93ee105211', type 't2.micro', and state 'running'. The 'Public DNS (IPv4)' column for this instance shows 'ec2-3-208-9-180.compute-1.amazonaws.com', with the IP address '3.208.9.180' highlighted in blue. Below the table, a detailed view of the instance is shown, including its description, status checks, monitoring, and tags. The 'Public DNS (IPv4)' field in this view also shows '3.208.9.180'.

- Now we can use our IP address in our hosts file

```
[webserver:vars]
ansible_ssh_private_key_file=/Users/sukhvirsingh/Desktop/id_rsa.pem

[webserver]
3.208.9.180
```

- To test that we're connected to the server we can run **ansible -i hosts all -m -u ec2-user**

```
ansible — -bash — 80x25
(base) Sukhvirs-MacBook-Pro:ansible sukhvirsingh$ ansible -i hosts all -m ping -u ec2-user
[webserver]
[Enter passphrase for key '/Users/sukhvirsingh/Desktop/id_rsa.pem':
3.208.9.180 | SUCCESS => {
  "changed": false,
  "ping": "pong"
}
(base) Sukhvirs-MacBook-Pro:ansible sukhvirsingh$
```

- Now that we've confirmed our connection, we can write our **setup-server.yaml** playbook that will start a Nginx server on the EC2 instance and copy our **index.html** file onto that server

setup-server.yaml:

```
ansible — -bash — 79x26
(base) Sukhvirs-MacBook-Pro:ansible sukhvirsingh$ cat setup-server.yaml
---
- hosts: webserver
  remote_user: ec2-user

  vars:
    - msg: "hello world"

  tasks:
    - name: Nginx setup
      become: true
      yum: pkg=nginx state=installed update_cache=true

    - name: index.html copy
      become: true
      template: src=index.html.j2 dest=/home/ec2-user/nginx/index.html
...
(base) Sukhvirs-MacBook-Pro:ansible sukhvirsingh$
```

index.html:

```
ansible — -bash — 80x25
(base) Sukhvirs-MacBook-Pro:ansible sukhvirsingh$ cat index.html.j2
<html>
<body>
<h1>Team Atom - Ansible</h1>
<p>{{msg}}</p>
</body>
</html>
(base) Sukhvirs-MacBook-Pro:ansible sukhvirsingh$
```

*the .j2 at the end of the filename is just a naming convention for template html files, it doesn't change anything

- Now we can run our playbook that will start the server

```
(base) Sukhvirs-MacBook-Pro:ansible sukhvirsingh$ ansible-playbook -i hosts setup
p-server.yaml
become: true

PLAY [webservers] *****

TASK [Gathering Facts] *****
ok: [3.208.9.180]

TASK [Nginx setup] *****
ok: [3.208.9.180]

TASK [index.html copy] *****
ok: [3.208.9.180]

PLAY RECAP *****
3.208.9.180 : ok=3    changed=0    unreachable=0    failed=0
ansible.cfg

(base) Sukhvirs-MacBook-Pro:ansible sukhvirsingh$
```

- Now that our server has been deployed, we can go to the IP address (3.208.9.180) in a web browser and see index.html



- Now we can write our playbook that will undeploy our server **stop-server.yaml**:

```
ansible — -bash — 81x30
(base) Sukhvirs-MacBook-Pro:ansible sukhvirsingh$ cat stop-server.yaml
---
- hosts: webservers
  remote_user: ec2-user

  tasks:
    - name: stop nginx service
      become: true
      service: name=nginx state=stopped
...
(base) Sukhvirs-MacBook-Pro:ansible sukhvirsingh$
```

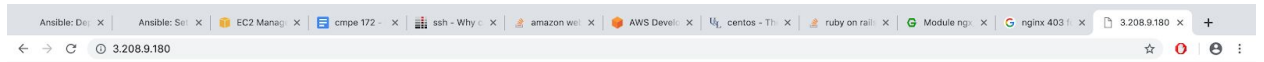
- Now we can use **stop-server.yaml** to undeploy our server

```
(base) Sukhvirs-MacBook-Pro:ansible sukhvirsingh$ ansible-playbook -i hosts stop-server.yaml
PLAY [webservers] *****
TASK [Gathering Facts] *****
Enter passphrase for key '/Users/sukhvirsingh/Desktop/id_rsa.pem':
ok: [3.208.9.180]

TASK [stop nginx service] *****
changed: [3.208.9.180]

PLAY RECAP *****
3.208.9.180 : ok=2 changed=1 unreachable=0 failed=0
(base) Sukhvirs-MacBook-Pro:ansible sukhvirsingh$
```

- We can verify that the server has been stopped because the web page won't load



This site can't be reached

3.208.9.180 refused to connect.

Try:

- Checking the connection
- [Checking the proxy and the firewall](#)

ERR_CONNECTION_REFUSED

Details

Reload