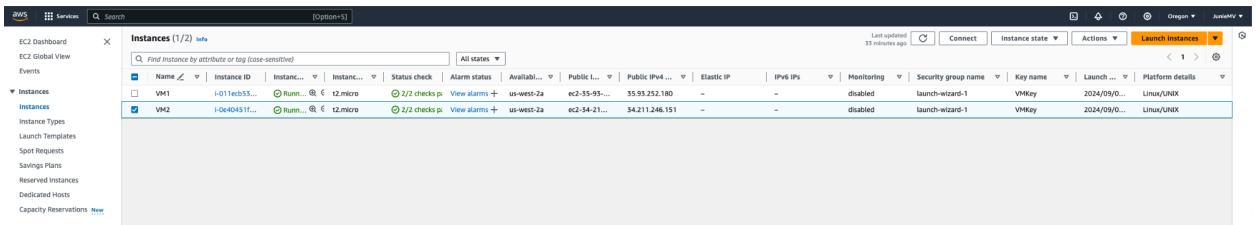
HW #1 - Ansible

STEP 1 : VM SETUP

Created two VM in aws

1. Name of First VM: VM1
2. Public IP: 35.93.252.180

3. Name of Second VM: VM2
4. Public IP: 34.211.246.151



STEP 2 : INSTALLING ANSIBLE ON YOUR CONTROL MACHINE

Sine I am on an MacOS, I used brew install ansible

STEP 3 : CHECK PERMISSIONS AND CREATE INVENTORY FILE

1. Check SSH Key Permissions

```
rohan@junie ansible-aws-deployment % chmod 400 /Users/rohan/Downloads/VMKey.pem
rohan@junie ansible-aws-deployment % ssh -i /Users/rohan/Downloads/VMKey.pem ec2
-user@35.93.252.180
       #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-172-31-23-118 ~]$ exit
logout
```

2. Update the Known Hosts

```
rohan@junie ansible-aws-deployment % ssh-keyscan -H 35.93.252.180 >> ~/.ssh/known_hosts

# 35.93.252.180:22 SSH-2.0-OpenSSH_8.7
# 35.93.252.180:22 SSH-2.0-OpenSSH_8.7
# 35.93.252.180:22 SSH-2.0-OpenSSH_8.7
# 35.93.252.180:22 SSH-2.0-OpenSSH_8.7
# 35.93.252.180:22 SSH-2.0-OpenSSH_8.7
rohan@junie ansible-aws-deployment % ssh-keyscan -H 34.211.246.151 >> ~/.ssh/known_hosts

# 34.211.246.151:22 SSH-2.0-OpenSSH_8.7
# 34.211.246.151:22 SSH-2.0-OpenSSH_8.7
# 34.211.246.151:22 SSH-2.0-OpenSSH_8.7
# 34.211.246.151:22 SSH-2.0-OpenSSH_8.7
# 34.211.246.151:22 SSH-2.0-OpenSSH_8.7
```

In Ansible, "known hosts" are like a list of trusted servers that your computer has connected to before. When Ansible connects to a server, it checks this list to make sure the server is the one it expects. This helps prevent security issues, like someone pretending to be the server you want to connect to. The list of trusted servers and their security keys is stored in a file called `known_hosts` on your computer. If the server's key doesn't match what's in this file, you'll get a warning, and you'll need to check and update the list if necessary. In simple terms, known hosts help ensure that Ansible connects to the right and secure servers.

3. Create the Inventory File

```
rohan@junie ansible-aws-deployment % nano hosts.ini
```

```
[webservers]
vm1 ansible_host=35.93.252.180 ansible_user=ec2-user server_id=1
vm2 ansible_host=34.211.246.151 ansible_user=ec2-user server_id=2
[webservers:vars]
ansible_ssh_private_key_file=/Users/rohan/Downloads/VMKey.pem
~
~
~
```

An Ansible inventory file is like a contact list for Ansible, helping it know which computers or servers to work with. It lists all the servers you want to manage and can organize them into groups, such as web servers and database servers, making it easier to apply specific tasks or settings to each group. The inventory file can also be set up to automatically update if your list of servers changes. Additionally, it allows you to define custom settings or instructions for different servers or groups. Overall, it simplifies the management of your servers by providing a structured way for Ansible to understand and interact with them.

We added the details of VM1 and VM2 in the inventory file.

3. Check the ansible inventory and check ansible connectivity

```
rohan@junie ansible-aws-deployment % ansible-inventory -i hosts.ini --list
{
    "_meta": {
        "hostvars": {
            "vm1": {
                "ansible_host": "35.93.252.180",
                "ansible_ssh_private_key_file": "/Users/rohan/Downloads/VMKey.pem",
                "ansible_user": "ec2-user",
                "server_id": 1
            },
            "vm2": {
                "ansible_host": "34.211.246.151",
                "ansible_ssh_private_key_file": "/Users/rohan/Downloads/VMKey.pem",
                "ansible_user": "ec2-user",
                "server_id": 2
            }
        }
    },
    "all": {
        "children": [
            "ungrouped",
            "webservers"
        ]
    },
    "webservers": {
        "hosts": [
            "vm1",
            "vm2"
        ]
    }
}
rohan@junie ansible-aws-deployment % ansible all -i hosts.ini -m ping
[WARNING]: Platform linux on host vm1 is using the discovered Python interpreter at /usr/bin/python3.9, but future installation of another Python
interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.17/reference_appendices/interpreter_discovery.html for more
information.
vm1 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3.9"
    },
    "changed": false,
    "ping": "pong"
}
[WARNING]: Platform linux on host vm2 is using the discovered Python interpreter at /usr/bin/python3.9, but future installation of another Python
interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.17/reference_appendices/interpreter_discovery.html for more
information.
vm2 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3.9"
    },
    "changed": false,
    "ping": "pong"
}
```

This step is done to check if everything is done correctly.

STEP 4 : CREATE AND RUN THE PLAYBOOK

I created the playbook for deploying and undeploying the web server. I have added sufficient tags to enable it. I also added configuration to listen from port 8080.

```yaml
---
- name: Deploy and Manage Web Servers
  hosts: webservers
  become: yes
  tags: deploy
  tasks:
    - name: Install Apache
      yum:
        name: httpd
        state: present
      when: ansible_facts['distribution'] == 'Amazon'

    - name: Configure Apache to listen on port 8080
      lineinfile:
        path: /etc/httpd/conf/httpd.conf
        regexp: '^Listen '
        line: 'Listen 8080'
        state: present

    - name: Start Apache service
      service:
        name: httpd
        state: started
        enabled: yes

    - name: Deploy custom web page
      copy:
        content: |
          <html>
          <body>
            <h1>Hello World from SJSU-{{ server_id }}</h1>
          </body>
          </html>
        dest: /var/www/html/index.html
      notify:
        - restart apache

  handlers:
    - name: restart apache
      service:
        name: httpd
        state: restarted

- name: Un-deploy Web Servers
  hosts: webservers
  become: yes
  tags: undeploy
  tasks:
    - name: Stop Apache service
      service:
        name: httpd
        state: stopped

    - name: Remove Apache
      yum:
        name: httpd
        state: absent

    - name: Remove custom web page
      file:
        path: /var/www/html/index.html
        state: absent
~
~
~
~
~
~
"webservers_deploy.yml" 62L, 1286B
```

Run the ansible to deploy the web server

```
rohan@junie ansible-aws-deployment % ansible-playbook -i hosts.ini webserver_deploy.yml --tags "deploy"

PLAY [Deploy and Manage Web Servers] ***********************************************************************************************

TASK [Gathering Facts] *************************************************************************************************************
[WARNING]: Platform linux on host vm2 is using the discovered Python interpreter at /usr/bin/python3.9, but future installation of another Python
interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.17/reference_appendices/interpreter_discovery.html for more
information.
ok: [vm2]
[WARNING]: Platform linux on host vm1 is using the discovered Python interpreter at /usr/bin/python3.9, but future installation of another Python
interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.17/reference_appendices/interpreter_discovery.html for more
information.
ok: [vm1]

TASK [Install Apache] **************************************************************************************************************
ok: [vm1]
ok: [vm2]

TASK [Configure Apache to listen on port 8080] *************************************************************************************
ok: [vm1]
ok: [vm2]

TASK [Start Apache service] ********************************************************************************************************
ok: [vm1]
ok: [vm2]

TASK [Deploy custom web page] ******************************************************************************************************
ok: [vm1]
ok: [vm2]

PLAY [Un-deploy Web Servers] *******************************************************************************************************

PLAY RECAP *************************************************************************************************************************
vm1                        : ok=5    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
vm2                        : ok=5    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

rohan@junie ansible-aws-deployment %
```

Demo of Deploy web server on VM1

Hello World from SJSU-1

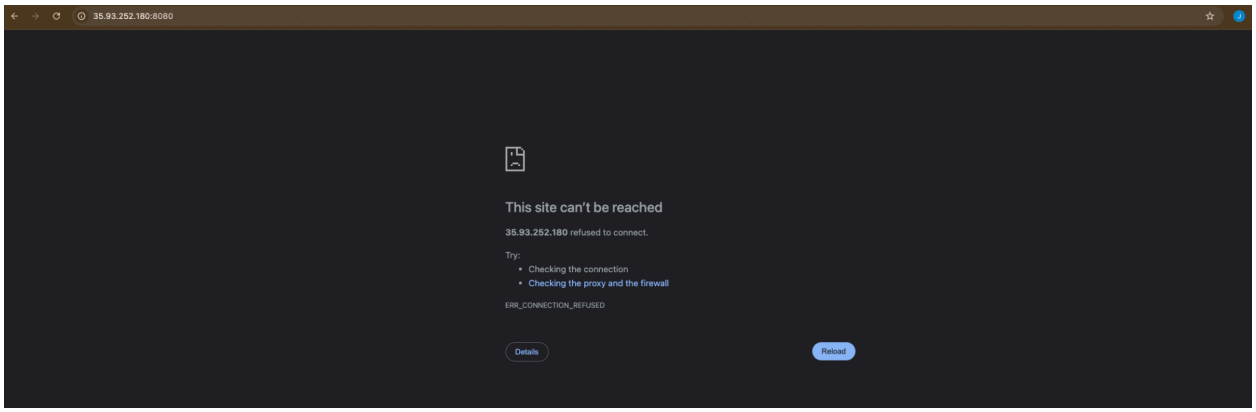
Demo of Deploy web server on VM2

Hello World from SJSU-2

Run the ansible to Un-deploy the web server

```
rohan@junie ansible-aws-deployment % ansible-playbook -i hosts.ini webserver_deploy.yml --tags "undeploy"

PLAY [Deploy and Manage Web Servers] ***********************************************************************************************************

PLAY [Un-deploy Web Servers] *******************************************************************************************************************

TASK [Gathering Facts] *************************************************************************************************************************
[WARNING]: Platform linux on host vm2 is using the discovered Python interpreter at /usr/bin/python3.9, but future installation of another Python
interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.17/reference_appendices/interpreter_discovery.html for more
information.
ok: [vm2]
[WARNING]: Platform linux on host vm1 is using the discovered Python interpreter at /usr/bin/python3.9, but future installation of another Python
interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.17/reference_appendices/interpreter_discovery.html for more
information.
ok: [vm1]

TASK [Stop Apache service] *********************************************************************************************************************
changed: [vm2]
changed: [vm1]

TASK [Remove Apache] ***************************************************************************************************************************
changed: [vm1]
changed: [vm2]

TASK [Remove custom web page] ******************************************************************************************************************
changed: [vm2]
changed: [vm1]

PLAY RECAP *************************************************************************************************************************************
vm1                        : ok=4    changed=3    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
vm2                        : ok=4    changed=3    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

Demo of Un-deploy web server on VM2



Demo of Un-deploy web server on VM2