Name : Reena Qureshi

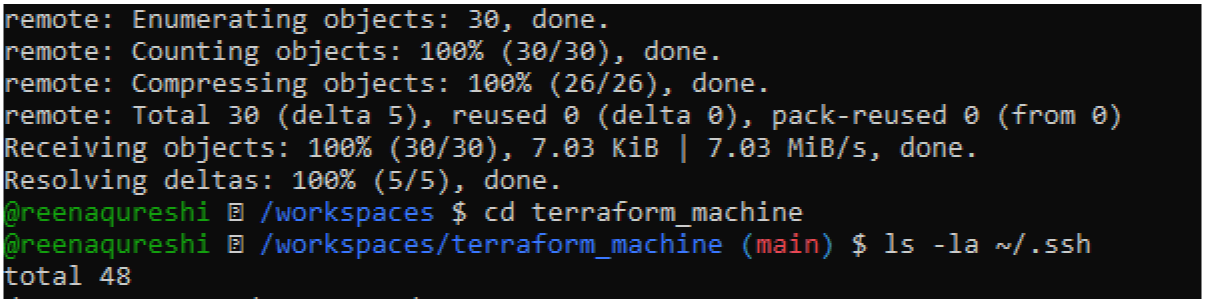
Reg no : 2023-BSE-052

Section : V-B

LAB 14

Task 1 – Clone Terraform+Ansible repo & initial Terraform apply

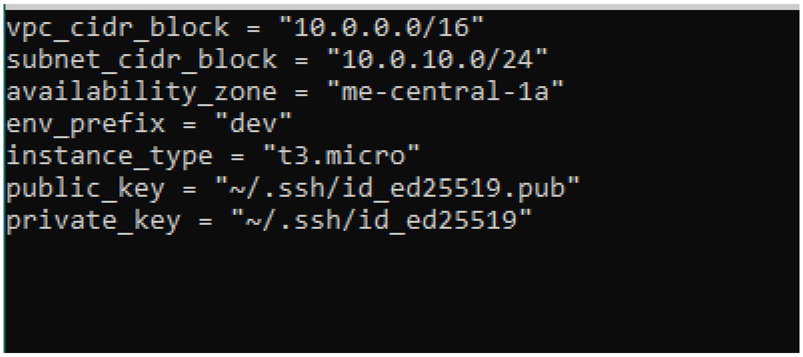
Clone the repository (branch `main`) inside the Codespace:



```
remote: Enumerating objects: 30, done.
remote: Counting objects: 100% (30/30), done.
remote: Compressing objects: 100% (26/26), done.
remote: Total 30 (delta 5), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (30/30), 7.03 KiB | 7.03 MiB/s, done.
Resolving deltas: 100% (5/5), done.
@reenaqureshi ➜ /workspaces $ cd terraform_machine
@reenaqureshi ➜ /workspaces/terraform_machine (main) $ ls -la ~/.ssh
total 48
```

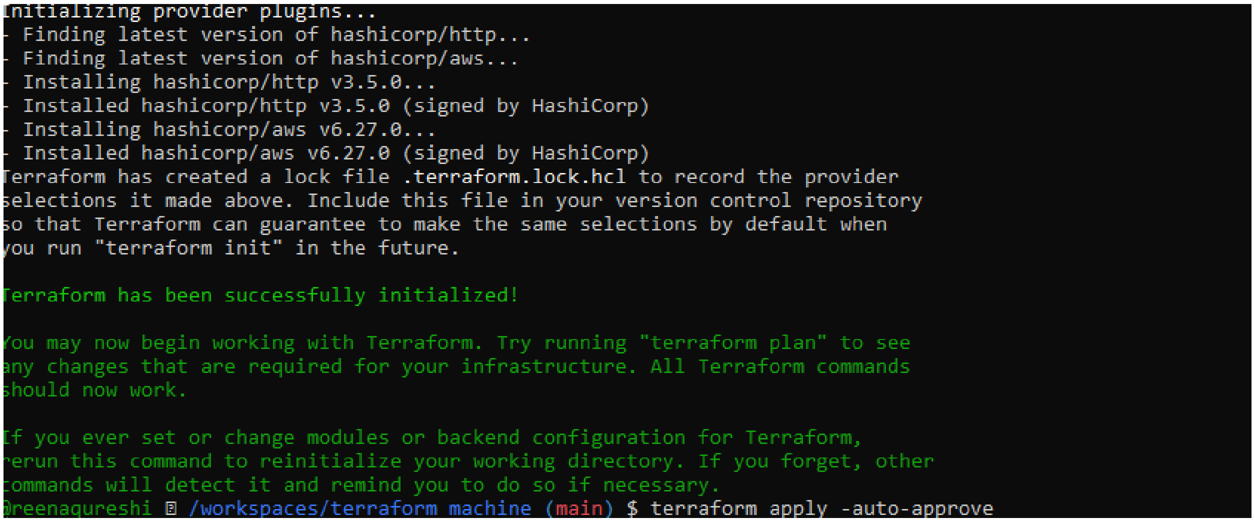
Create `terraform.tfvars` in the repo root:

Add the following content



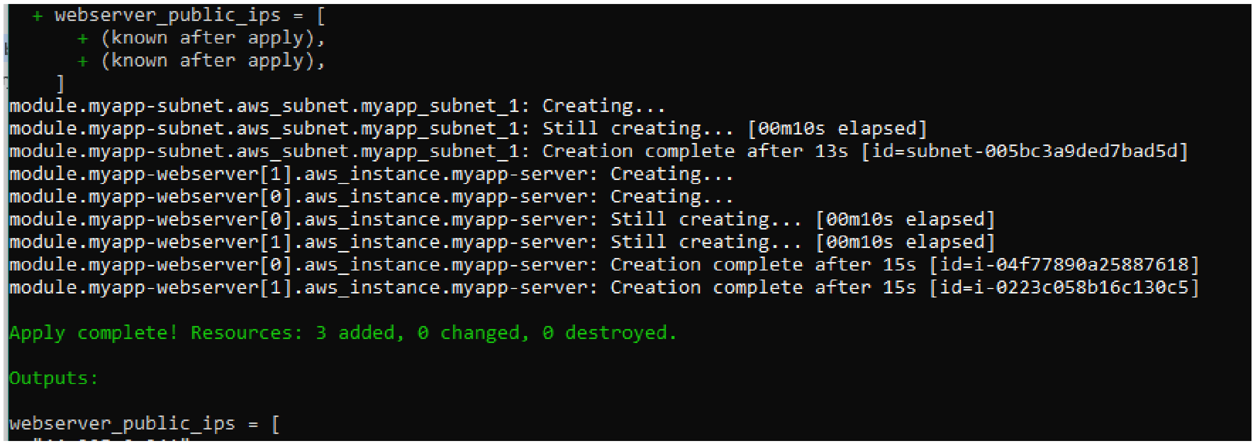
```
vpc_cidr_block = "10.0.0.0/16"
subnet_cidr_block = "10.0.10.0/24"
availability_zone = "me-central-1a"
env_prefix = "dev"
instance_type = "t3.micro"
public_key = "~/.ssh/id_ed25519.pub"
private_key = "~/.ssh/id_ed25519"
```

Initialize Terraform:



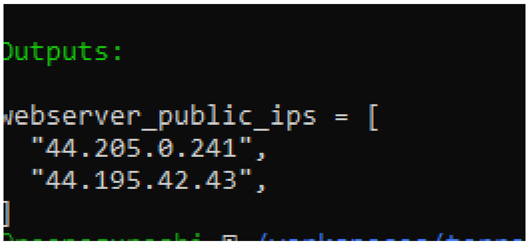
```
Initializing provider plugins...
- Finding latest version of hashicorp/http...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/http v3.5.0...
- Installed hashicorp/http v3.5.0 (signed by HashiCorp)
- Installing hashicorp/aws v6.27.0...
- Installed hashicorp/aws v6.27.0 (signed by HashiCorp)
Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
@reenaqureshi ➜ /workspaces/terraform_machine (main) $ terraform apply -auto-approve
```

Apply Terraform to create 2 EC2 instances (as defined in the existing Terraform code):



```
  + webserver_public_ips = [
      + (known after apply),
      + (known after apply),
    ]
module.myapp-subnet.aws_subnet.myapp_subnet_1: Creating...
module.myapp-subnet.aws_subnet.myapp_subnet_1: Still creating... [00m10s elapsed]
module.myapp-subnet.aws_subnet.myapp_subnet_1: Creation complete after 13s [id=subnet-005bc3a9ded7bad5d]
module.myapp-webserver[1].aws_instance.myapp-server: Creating...
module.myapp-webserver[0].aws_instance.myapp-server: Creating...
module.myapp-webserver[0].aws_instance.myapp-server: Still creating... [00m10s elapsed]
module.myapp-webserver[1].aws_instance.myapp-server: Still creating... [00m10s elapsed]
module.myapp-webserver[0].aws_instance.myapp-server: Creation complete after 15s [id=i-04f77890a25887618]
module.myapp-webserver[1].aws_instance.myapp-server: Creation complete after 15s [id=i-0223c058b16c130c5]

Apply complete! Resources: 3 added, 0 changed, 0 destroyed.

Outputs:

webserver_public_ips = [
```

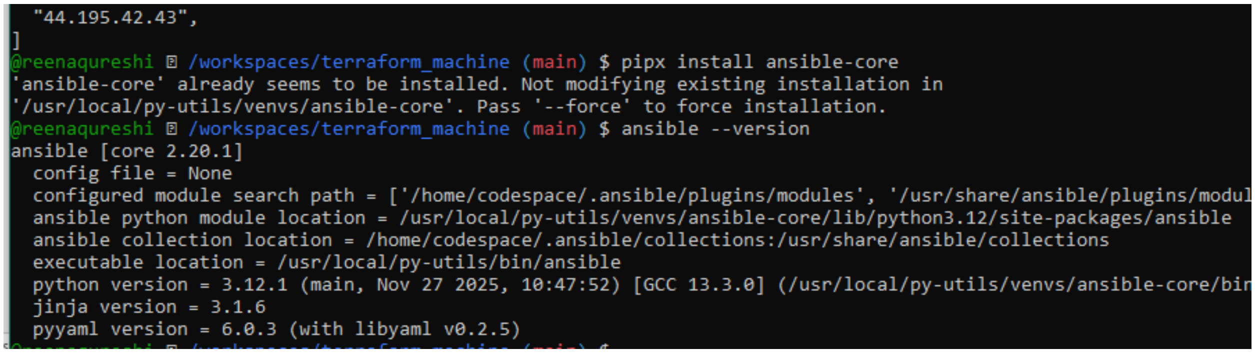
Output:



```
Outputs:

webserver_public_ips = [
  "44.205.0.241",
  "44.195.42.43",
]
```

Task 2 – Static Ansible inventory with two EC2 instances

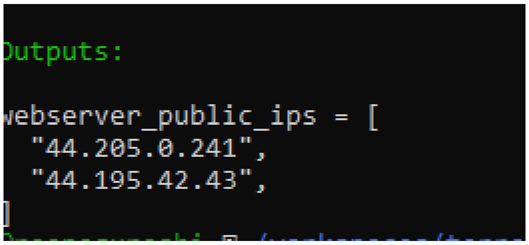
Install Ansible (core) using pipx:



```
  "44.195.42.43",
]
@reenaqureshi □ /workspaces/terraform_machine (main) $ pipx install ansible-core
'ansible-core' already seems to be installed. Not modifying existing installation in
'/usr/local/py-utils/venvs/ansible-core'. Pass '--force' to force installation.
@reenaqureshi □ /workspaces/terraform_machine (main) $ ansible --version
ansible [core 2.20.1]
  config file = None
  configured module search path = ['/home/codespace/.ansible/plugins/modules', '/usr/share/ansible/plugins/modul
  ansible python module location = /usr/local/py-utils/venvs/ansible-core/lib/python3.12/site-packages/ansible
  ansible collection location = /home/codespace/.ansible/collections:/usr/share/ansible/collections
  executable location = /usr/local/py-utils/bin/ansible
  python version = 3.12.1 (main, Nov 27 2025, 10:47:52) [GCC 13.3.0] (/usr/local/py-utils/venvs/ansible-core/bin
  jinja version = 3.1.6
  pyyaml version = 6.0.3 (with libyaml v0.2.5)
```

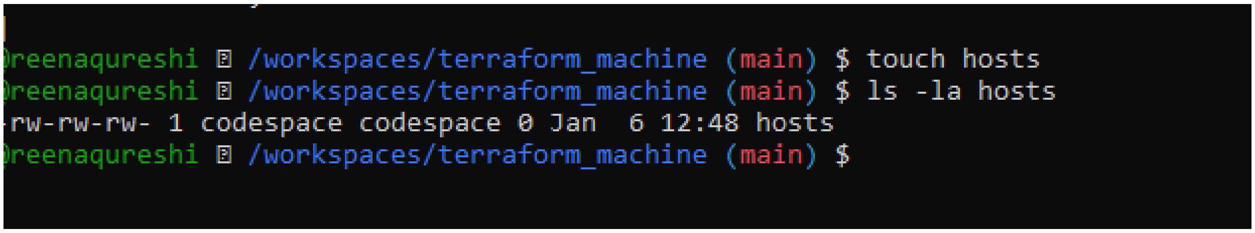
Obtain the two public IPs of your EC2 instances:

```
terraform output
```



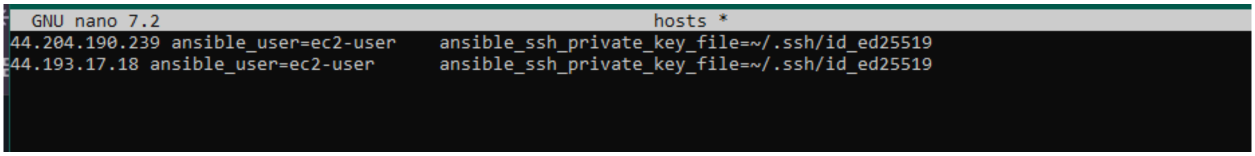
```
Outputs:

webserver_public_ips = [
  "44.205.0.241",
  "44.195.42.43",
]
```

Create Ansible inventory file `hosts`:



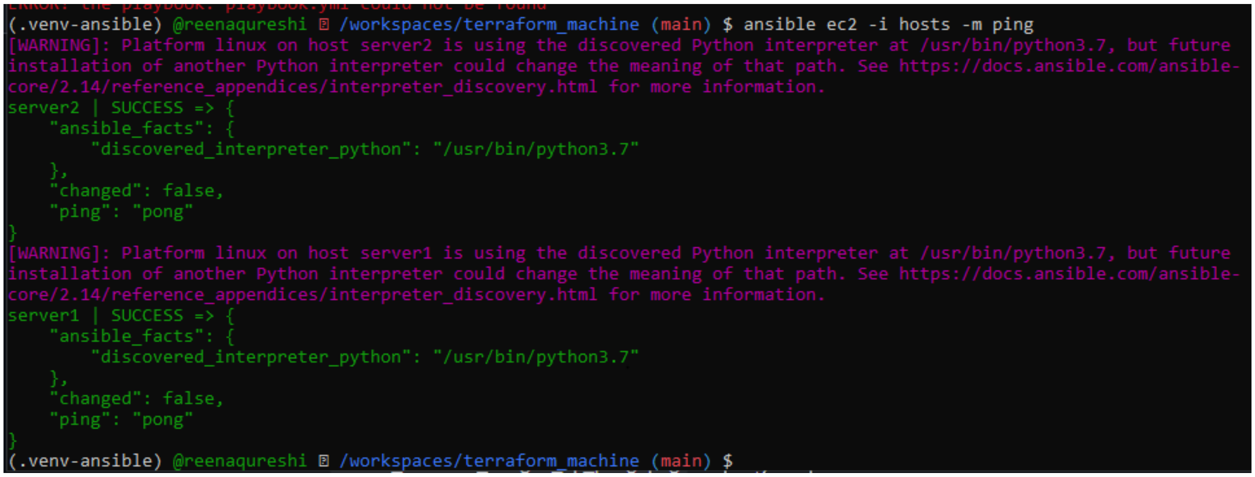
```
@reenaqureshi □ /workspaces/terraform_machine (main) $ touch hosts
@reenaqureshi □ /workspaces/terraform_machine (main) $ ls -la hosts
-rw-rw-rw- 1 codespace codespace 0 Jan  6 12:48 hosts
@reenaqureshi □ /workspaces/terraform_machine (main) $
```

Add the following (replace <public-ip-ec2> with your 2 real IPs):



```
  GNU nano 7.2                                      hosts *
44.204.190.239 ansible_user=ec2-user    ansible_ssh_private_key_file=~/.ssh/id_ed25519
44.193.17.18 ansible_user=ec2-user      ansible_ssh_private_key_file=~/.ssh/id_ed25519
```

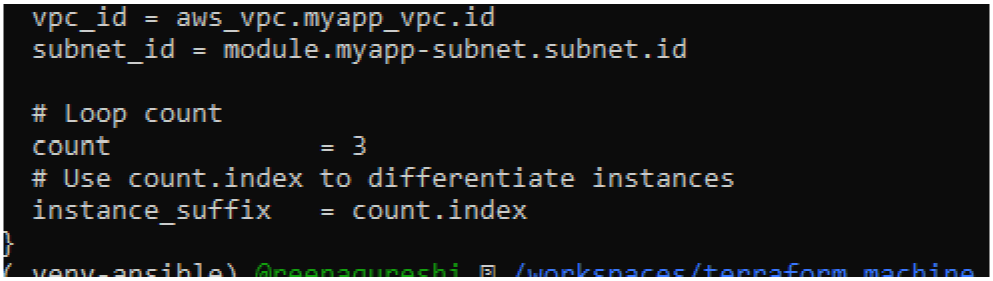
Test connectivity:



```
ERROR: the playbook: playbook.yml could not be found
(.venv-ansible) @reenaqureshi □ /workspaces/terraform_machine (main) $ ansible ec2 -i hosts -m ping
[WARNING]: Platform linux on host server2 is using the discovered Python interpreter at /usr/bin/python3.7, but future
installation of another Python interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-
core/2.14/reference_appendices/interpreter_discovery.html for more information.
server2 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3.7"
    },
    "changed": false,
    "ping": "pong"
}
[WARNING]: Platform linux on host server1 is using the discovered Python interpreter at /usr/bin/python3.7, but future
installation of another Python interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-
core/2.14/reference_appendices/interpreter_discovery.html for more information.
server1 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3.7"
    },
    "changed": false,
    "ping": "pong"
}
(.venv-ansible) @reenaqureshi □ /workspaces/terraform_machine (main) $
```

Task 3 - Scale to three instances & group-based inventory

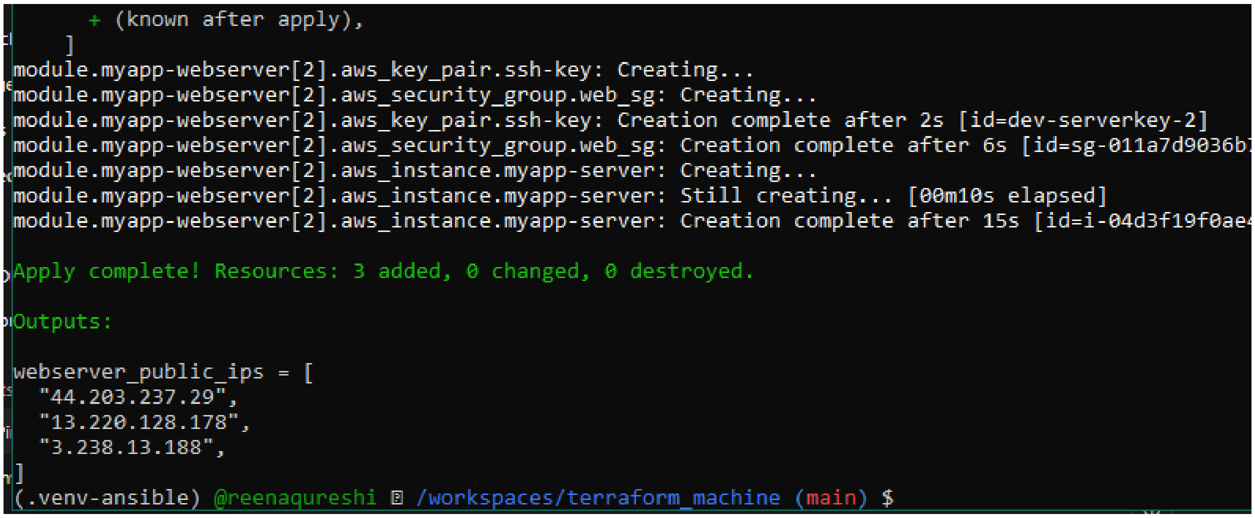
Change count to 3



```
  vpc_id = aws_vpc.myapp_vpc.id
  subnet_id = module.myapp-subnet.subnet.id

  # Loop count
  count              = 3
  # Use count.index to differentiate instances
  instance_suffix    = count.index
}
(.venv-ansible) @reenaqureshi □ /workspaces/terraform_machine
```

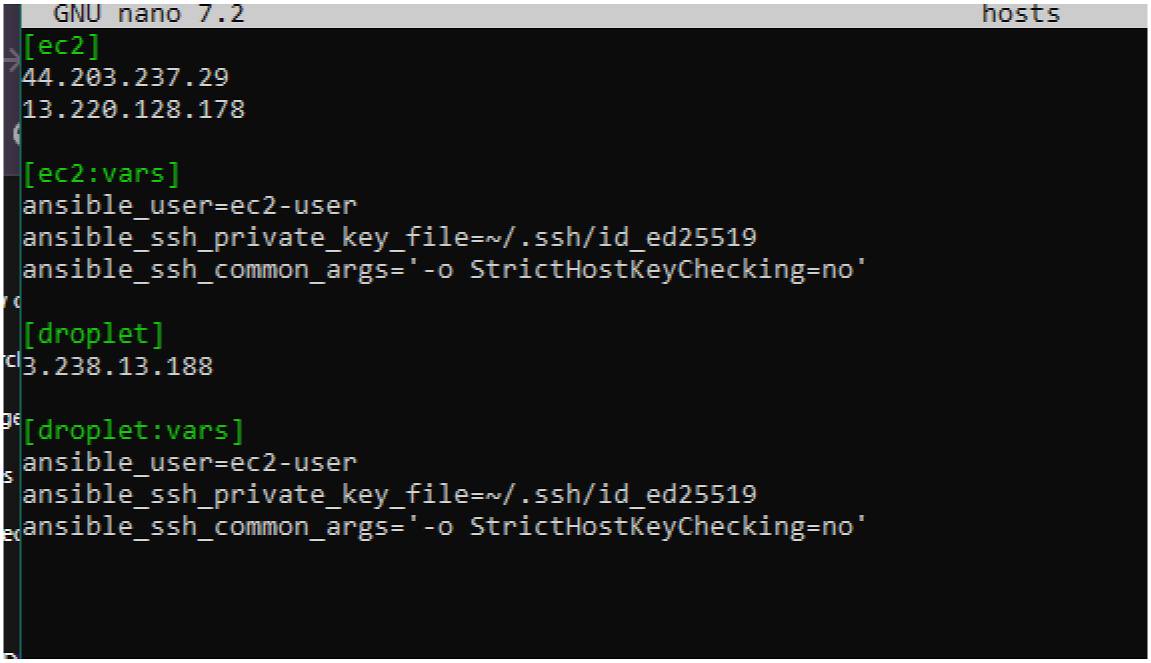
Apply Terraform to get 3 instances:

```
terraform apply -auto-approve
```



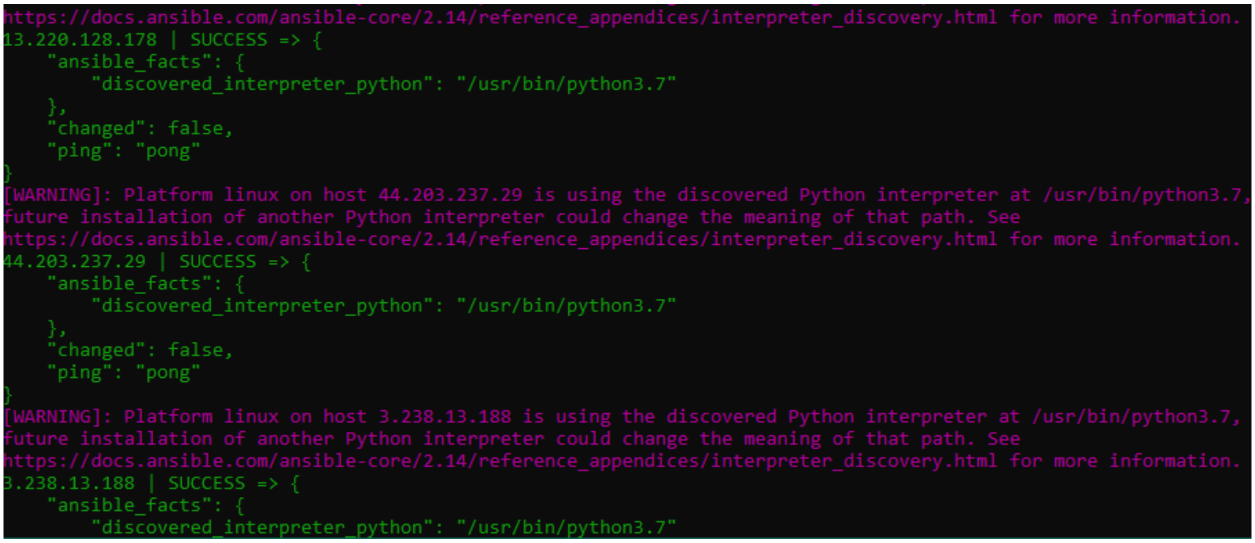
```
      + (known after apply),
    ]
module.myapp-webserver[2].aws_key_pair.ssh-key: Creating...
module.myapp-webserver[2].aws_security_group.web_sg: Creating...
module.myapp-webserver[2].aws_key_pair.ssh-key: Creation complete after 2s [id=dev-serverkey-2]
module.myapp-webserver[2].aws_security_group.web_sg: Creation complete after 6s [id=sg-011a7d9036b
module.myapp-webserver[2].aws_instance.myapp-server: Creating...
module.myapp-webserver[2].aws_instance.myapp-server: Still creating... [00m10s elapsed]
module.myapp-webserver[2].aws_instance.myapp-server: Creation complete after 15s [id=i-04d3f19f0ae

Apply complete! Resources: 3 added, 0 changed, 0 destroyed.

Outputs:

webserver_public_ips = [
  "44.203.237.29",
  "13.220.128.178",
  "3.238.13.188",
]
(.venv-ansible) @reenaqureshi ➜ /workspaces/terraform_machine (main) $
```

Rewrite your hosts file using group definitions:

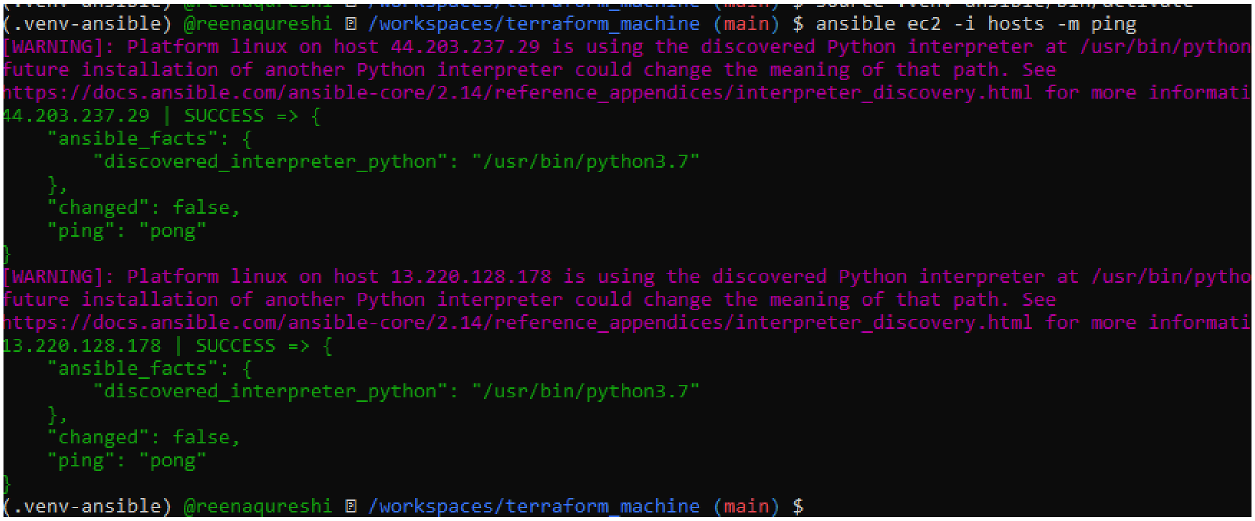


```
  GNU nano 7.2                                      hosts
[ec2]
44.203.237.29
13.220.128.178

[ec2:vars]
ansible_user=ec2-user
ansible_ssh_private_key_file=~/.ssh/id_ed25519
ansible_ssh_common_args='-o StrictHostKeyChecking=no'

[droplet]
3.238.13.188

[droplet:vars]
ansible_user=ec2-user
ansible_ssh_private_key_file=~/.ssh/id_ed25519
ansible_ssh_common_args='-o StrictHostKeyChecking=no'
```

Connectivity:



```
https://docs.ansible.com/ansible-core/2.14/reference_appendices/interpreter_discovery.html for more information.
13.220.128.178 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3.7"
    },
    "changed": false,
    "ping": "pong"
}
[WARNING]: Platform linux on host 44.203.237.29 is using the discovered Python interpreter at /usr/bin/python3.7,
future installation of another Python interpreter could change the meaning of that path. See
https://docs.ansible.com/ansible-core/2.14/reference_appendices/interpreter_discovery.html for more information.
44.203.237.29 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3.7"
    },
    "changed": false,
    "ping": "pong"
}
[WARNING]: Platform linux on host 3.238.13.188 is using the discovered Python interpreter at /usr/bin/python3.7,
future installation of another Python interpreter could change the meaning of that path. See
https://docs.ansible.com/ansible-core/2.14/reference_appendices/interpreter_discovery.html for more information.
3.238.13.188 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3.7"
```

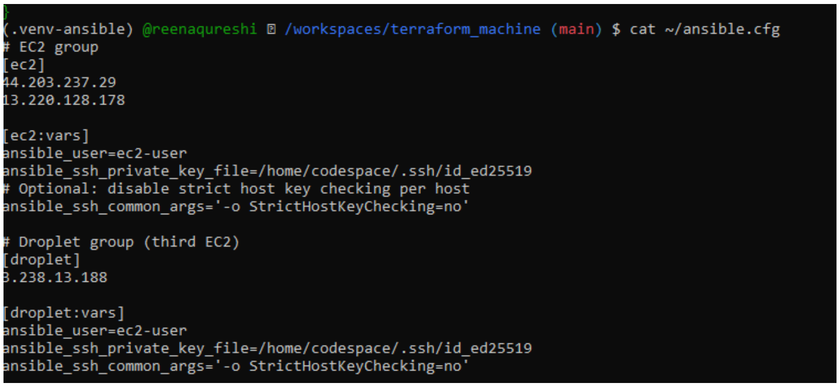


```
(.venv-ansible) @reenaqureshi ▯ /workspaces/terraform_machine (main) $ source .venv-ansible/bin/activate
(.venv-ansible) @reenaqureshi ▯ /workspaces/terraform_machine (main) $ ansible ec2 -i hosts -m ping
[WARNING]: Platform linux on host 44.203.237.29 is using the discovered Python interpreter at /usr/bin/python
future installation of another Python interpreter could change the meaning of that path. See
https://docs.ansible.com/ansible-core/2.14/reference_appendices/interpreter_discovery.html for more informati
44.203.237.29 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3.7"
    },
    "changed": false,
    "ping": "pong"
}
[WARNING]: Platform linux on host 13.220.128.178 is using the discovered Python interpreter at /usr/bin/pytho
future installation of another Python interpreter could change the meaning of that path. See
https://docs.ansible.com/ansible-core/2.14/reference_appendices/interpreter_discovery.html for more informati
13.220.128.178 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3.7"
    },
    "changed": false,
    "ping": "pong"
}
(.venv-ansible) @reenaqureshi ▯ /workspaces/terraform_machine (main) $
```

Task 4 – Global ansible.cfg & first nginx playbook

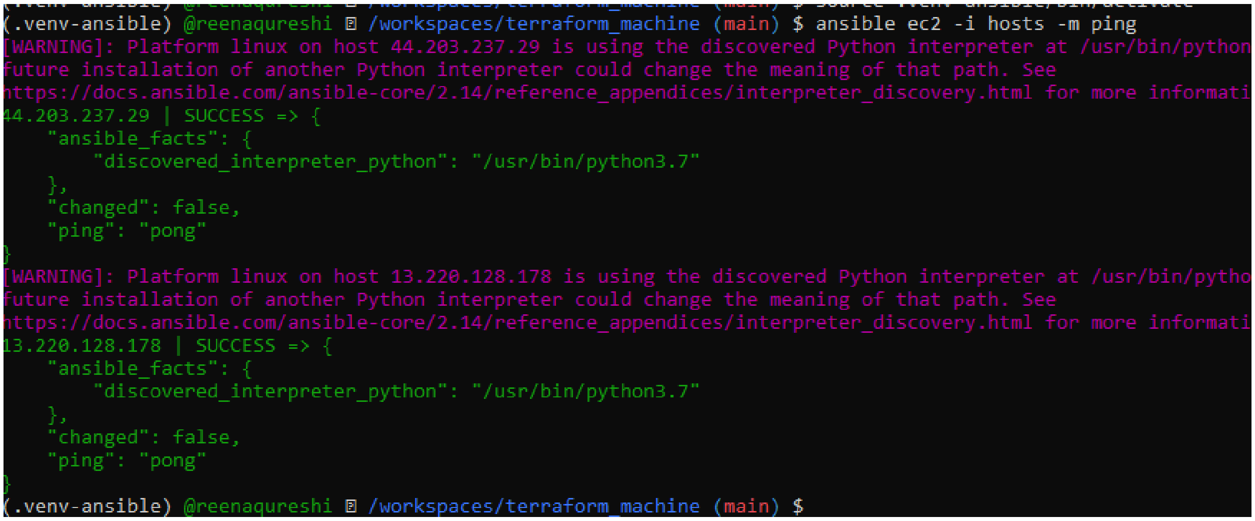
Create global Ansible configuration:

```
vim ~/.ansible.cfg
```



```
}
(.venv-ansible) @reenaqureshi ▯ /workspaces/terraform_machine (main) $ cat ~/ansible.cfg
# EC2 group
[ec2]
44.203.237.29
13.220.128.178

[ec2:vars]
ansible_user=ec2-user
ansible_ssh_private_key_file=/home/codespace/.ssh/id_ed25519
# Optional: disable strict host key checking per host
ansible_ssh_common_args='-o StrictHostKeyChecking=no'

# Droplet group (third EC2)
[droplet]
3.238.13.188

[droplet:vars]
ansible_user=ec2-user
ansible_ssh_private_key_file=/home/codespace/.ssh/id_ed25519
ansible_ssh_common_args='-o StrictHostKeyChecking=no'
```

Confirm connectivity:

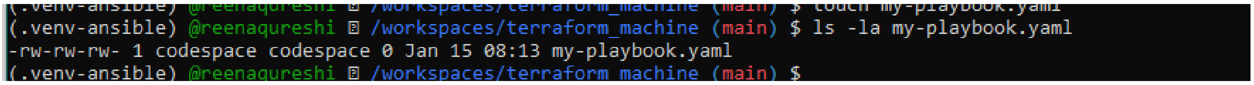


```
(.venv-ansible) @reenaqureshi ⍰ /workspaces/terraform_machine (main) $ source .venv-ansible/bin/activate
(.venv-ansible) @reenaqureshi ⍰ /workspaces/terraform_machine (main) $ ansible ec2 -i hosts -m ping
[WARNING]: Platform linux on host 44.203.237.29 is using the discovered Python interpreter at /usr/bin/python
future installation of another Python interpreter could change the meaning of that path. See
https://docs.ansible.com/ansible-core/2.14/reference_appendices/interpreter_discovery.html for more informati
44.203.237.29 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3.7"
    },
    "changed": false,
    "ping": "pong"
}
[WARNING]: Platform linux on host 13.220.128.178 is using the discovered Python interpreter at /usr/bin/pytho
future installation of another Python interpreter could change the meaning of that path. See
https://docs.ansible.com/ansible-core/2.14/reference_appendices/interpreter_discovery.html for more informati
13.220.128.178 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3.7"
    },
    "changed": false,
    "ping": "pong"
}
(.venv-ansible) @reenaqureshi ⍰ /workspaces/terraform_machine (main) $
```

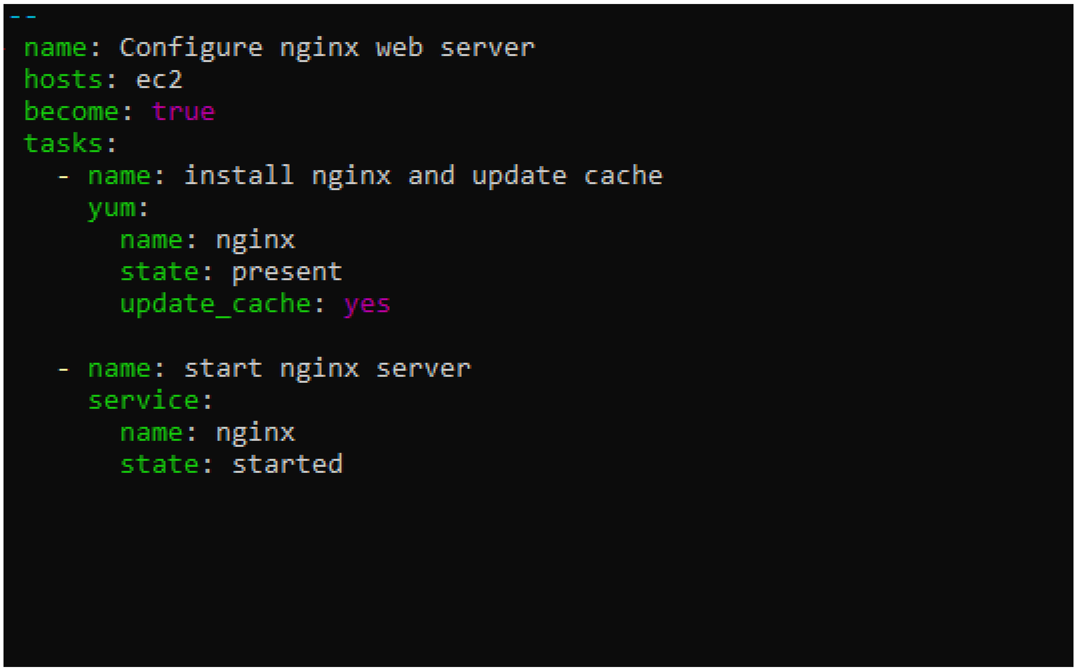
Create `my-playbook.yaml`:

```
touch my-playbook.yaml
```

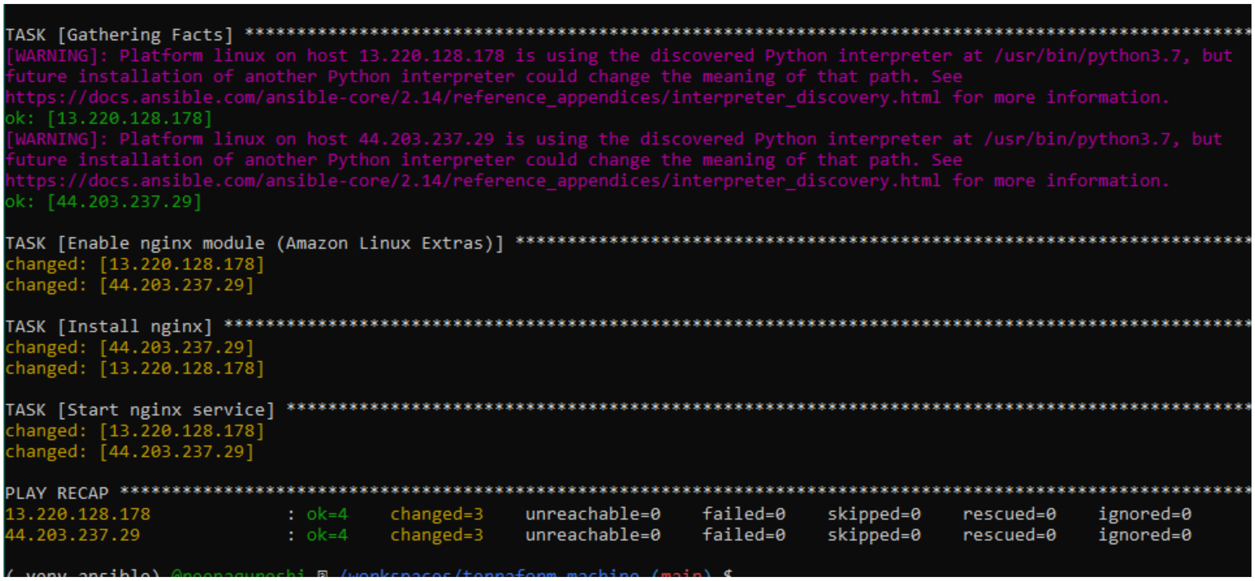
```
ls -la my-playbook.yaml
```



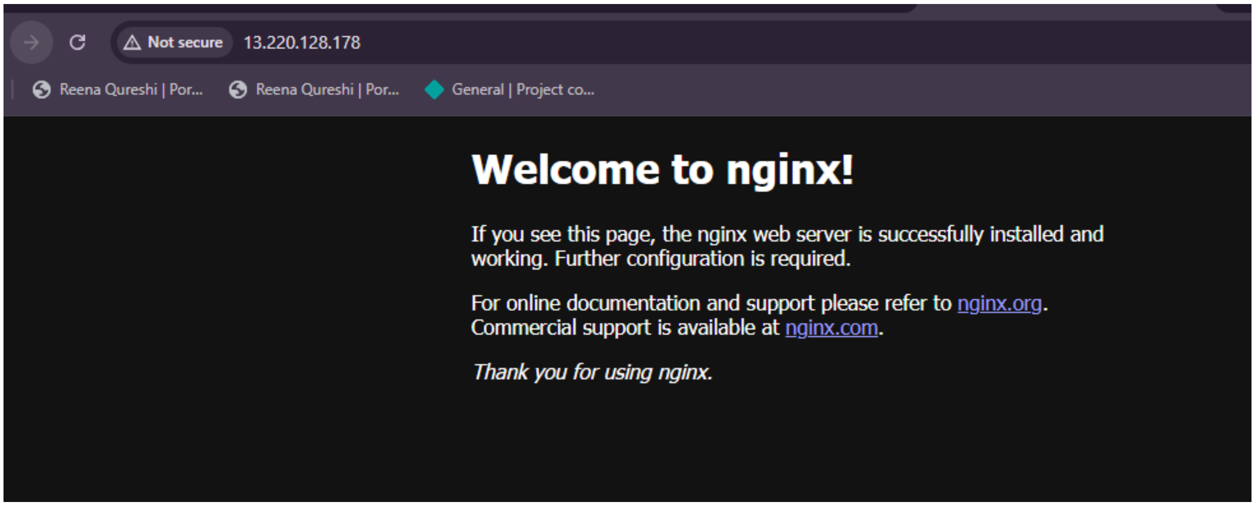
```
(.venv-ansible) @reenaqureshi ⍰ /workspaces/terraform_machine (main) $ touch my-playbook.yaml
(.venv-ansible) @reenaqureshi ⍰ /workspaces/terraform_machine (main) $ ls -la my-playbook.yaml
-rw-rw-rw- 1 codespace codespace 0 Jan 15 08:13 my-playbook.yaml
(.venv-ansible) @reenaqureshi ⍰ /workspaces/terraform_machine (main) $
```



```
--
- name: Configure nginx web server
  hosts: ec2
  become: true
  tasks:
    - name: install nginx and update cache
      yum:
        name: nginx
        state: present
        update_cache: yes

    - name: start nginx server
      service:
        name: nginx
        state: started
```



```
TASK [Gathering Facts] ***************************************************************************************
[WARNING]: Platform linux on host 13.220.128.178 is using the discovered Python interpreter at /usr/bin/python3.7, but
future installation of another Python interpreter could change the meaning of that path. See
https://docs.ansible.com/ansible-core/2.14/reference_appendices/interpreter_discovery.html for more information.
ok: [13.220.128.178]
[WARNING]: Platform linux on host 44.203.237.29 is using the discovered Python interpreter at /usr/bin/python3.7, but
future installation of another Python interpreter could change the meaning of that path. See
https://docs.ansible.com/ansible-core/2.14/reference_appendices/interpreter_discovery.html for more information.
ok: [44.203.237.29]

TASK [Enable nginx module (Amazon Linux Extras)] *************************************************************
changed: [13.220.128.178]
changed: [44.203.237.29]

TASK [Install nginx] *****************************************************************************************
changed: [44.203.237.29]
changed: [13.220.128.178]

TASK [Start nginx service] ***********************************************************************************
changed: [13.220.128.178]
changed: [44.203.237.29]

PLAY RECAP ***************************************************************************************************
13.220.128.178             : ok=4    changed=3    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
44.203.237.29              : ok=4    changed=3    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

(.venv-ansible) @reenaqureshi ➜ /workspaces/terraform-machine (main) $
```



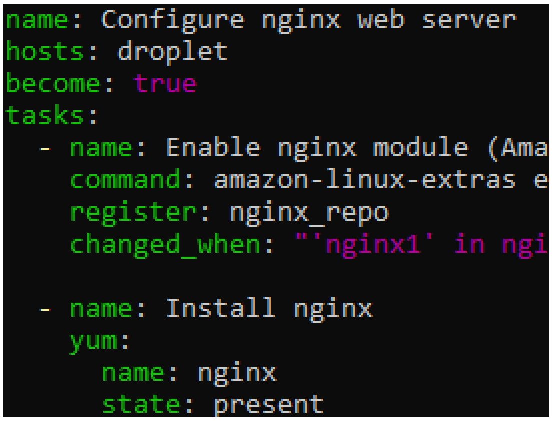
Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

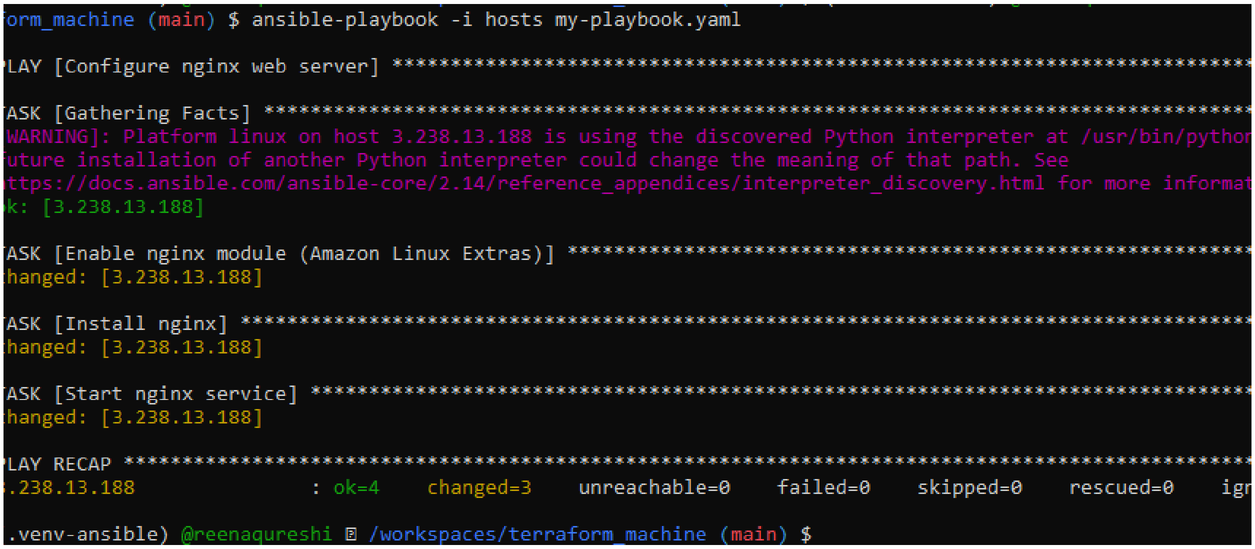
For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

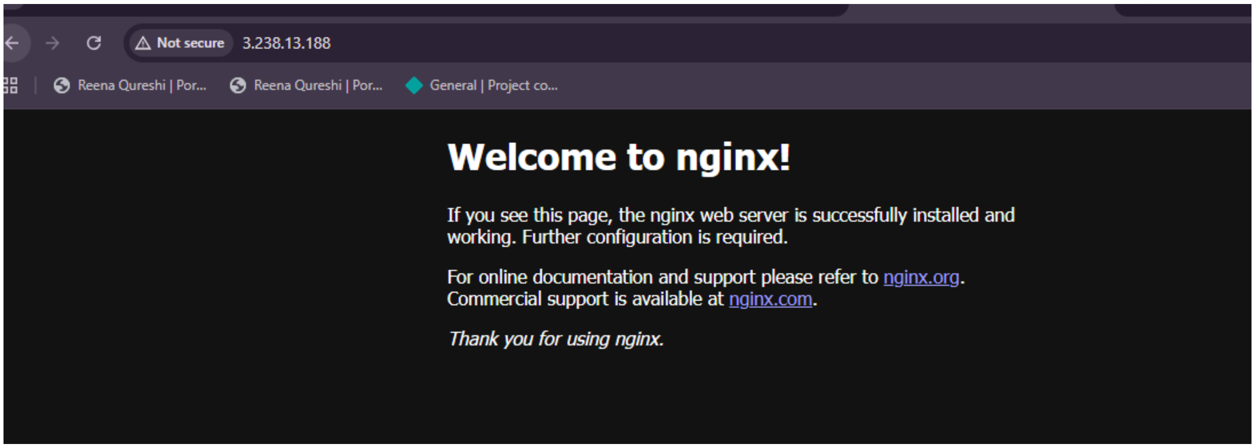
Change host to droplet:



```
name: Configure nginx web server
hosts: droplet
become: true
tasks:
  - name: Enable nginx module (Ama
    command: amazon-linux-extras e
    register: nginx_repo
    changed_when: "'nginx1' in ngi

  - name: Install nginx
    yum:
      name: nginx
      state: present
```



```
orm_machine (main) $ ansible-playbook -i hosts my-playbook.yaml

LAY [Configure nginx web server] ****************************************************************

ASK [Gathering Facts] ***************************************************************************
WARNING]: Platform linux on host 3.238.13.188 is using the discovered Python interpreter at /usr/bin/pythor
uture installation of another Python interpreter could change the meaning of that path. See
ttps://docs.ansible.com/ansible-core/2.14/reference_appendices/interpreter_discovery.html for more informat
k: [3.238.13.188]

ASK [Enable nginx module (Amazon Linux Extras)] *************************************************
hanged: [3.238.13.188]

ASK [Install nginx] *****************************************************************************
hanged: [3.238.13.188]

ASK [Start nginx service] ***********************************************************************
hanged: [3.238.13.188]

LAY RECAP ***************************************************************************************
.238.13.188                : ok=4    changed=3    unreachable=0    failed=0    skipped=0    rescued=0    igr

.venv-ansible) @reenaqureshi ▯ /workspaces/terraform_machine (main) $
```



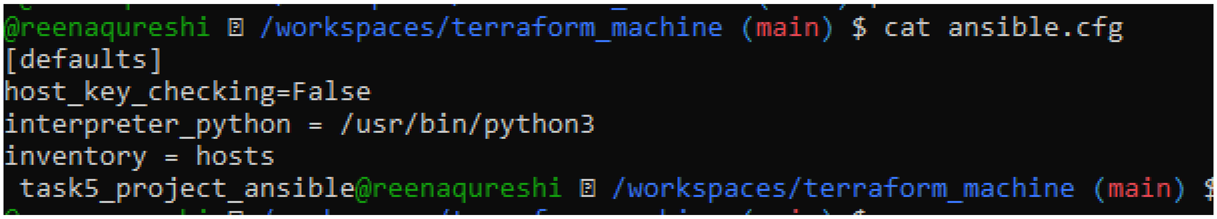
Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

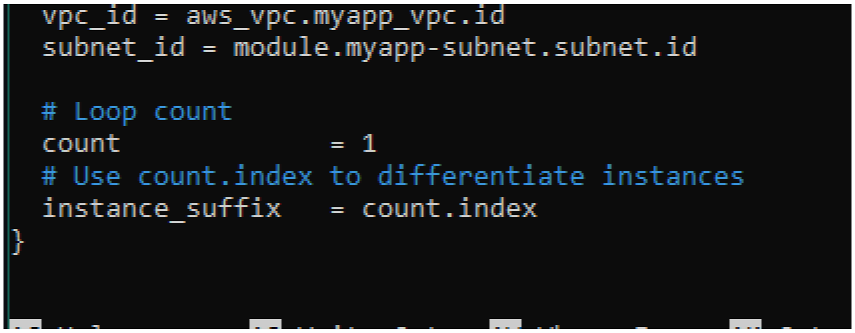
For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

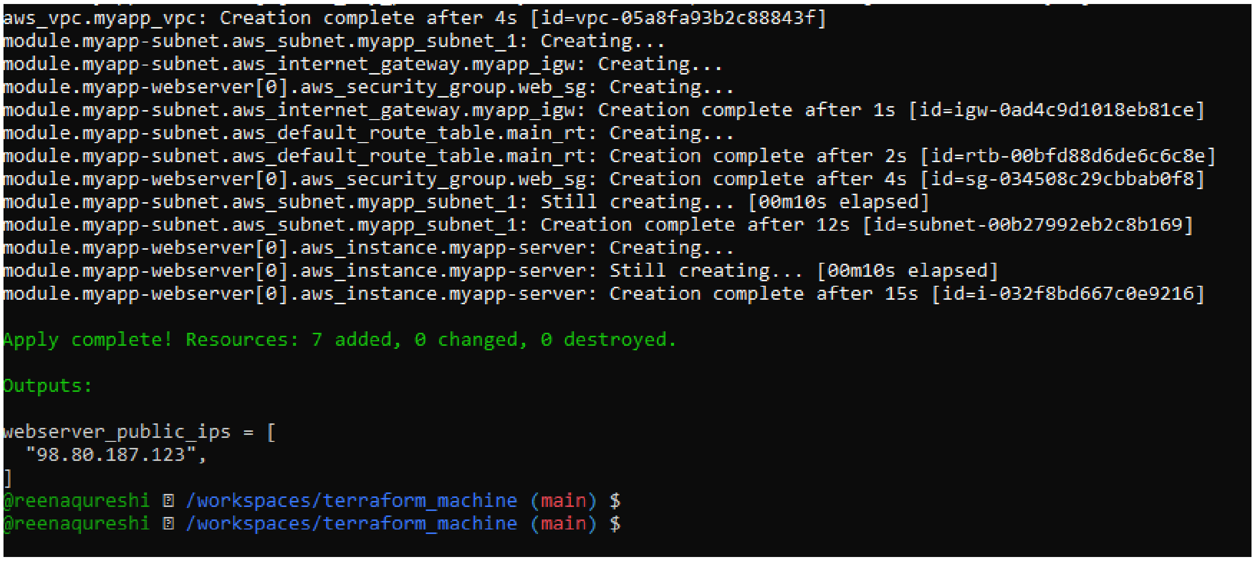
TASK 5:



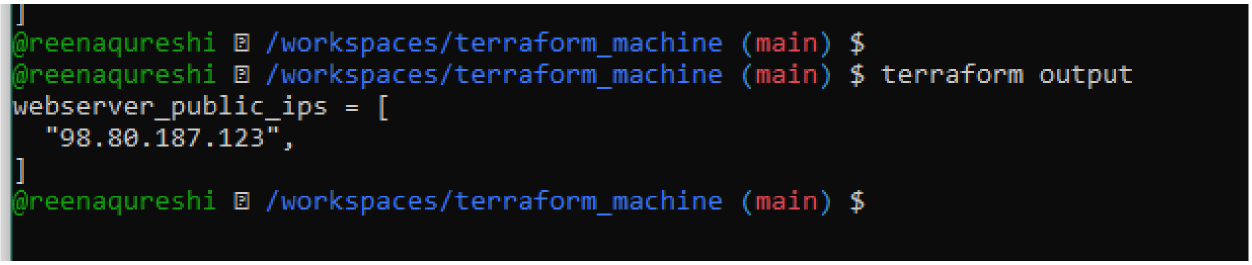
```
@reenaqureshi ▯ /workspaces/terraform_machine (main) $ cat ansible.cfg
[defaults]
host_key_checking=False
interpreter_python = /usr/bin/python3
inventory = hosts
 task5_project_ansible@reenaqureshi ▯ /workspaces/terraform_machine (main) $
```



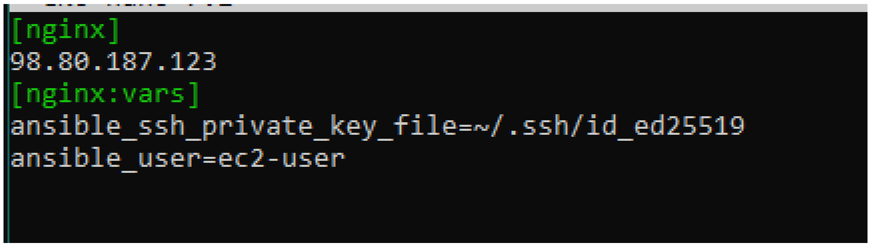
```
  vpc_id = aws_vpc.myapp_vpc.id
  subnet_id = module.myapp-subnet.subnet.id

  # Loop count
  count              = 1
  # Use count.index to differentiate instances
  instance_suffix    = count.index
}
```



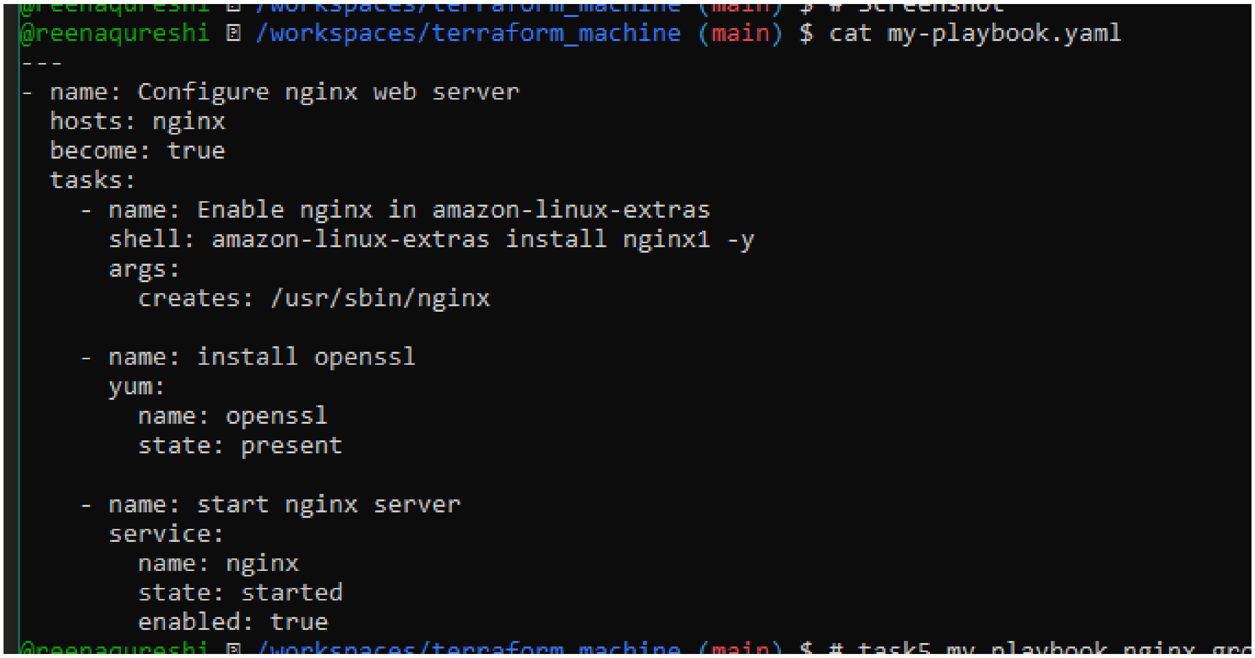
```
aws_vpc.myapp_vpc: Creation complete after 4s [id=vpc-05a8fa93b2c88843f]
module.myapp-subnet.aws_subnet.myapp_subnet_1: Creating...
module.myapp-subnet.aws_internet_gateway.myapp_igw: Creating...
module.myapp-webserver[0].aws_security_group.web_sg: Creating...
module.myapp-subnet.aws_internet_gateway.myapp_igw: Creation complete after 1s [id=igw-0ad4c9d1018eb81ce]
module.myapp-subnet.aws_default_route_table.main_rt: Creating...
module.myapp-subnet.aws_default_route_table.main_rt: Creation complete after 2s [id=rtb-00bfd88d6de6c6c8e]
module.myapp-webserver[0].aws_security_group.web_sg: Creation complete after 4s [id=sg-034508c29cbbab0f8]
module.myapp-subnet.aws_subnet.myapp_subnet_1: Still creating... [00m10s elapsed]
module.myapp-subnet.aws_subnet.myapp_subnet_1: Creation complete after 12s [id=subnet-00b27992eb2c8b169]
module.myapp-webserver[0].aws_instance.myapp-server: Creating...
module.myapp-webserver[0].aws_instance.myapp-server: Still creating... [00m10s elapsed]
module.myapp-webserver[0].aws_instance.myapp-server: Creation complete after 15s [id=i-032f8bd667c0e9216]

Apply complete! Resources: 7 added, 0 changed, 0 destroyed.

Outputs:

webserver_public_ips = [
  "98.80.187.123",
]
@reenaqureshi ▯ /workspaces/terraform_machine (main) $
@reenaqureshi ▯ /workspaces/terraform_machine (main) $
```



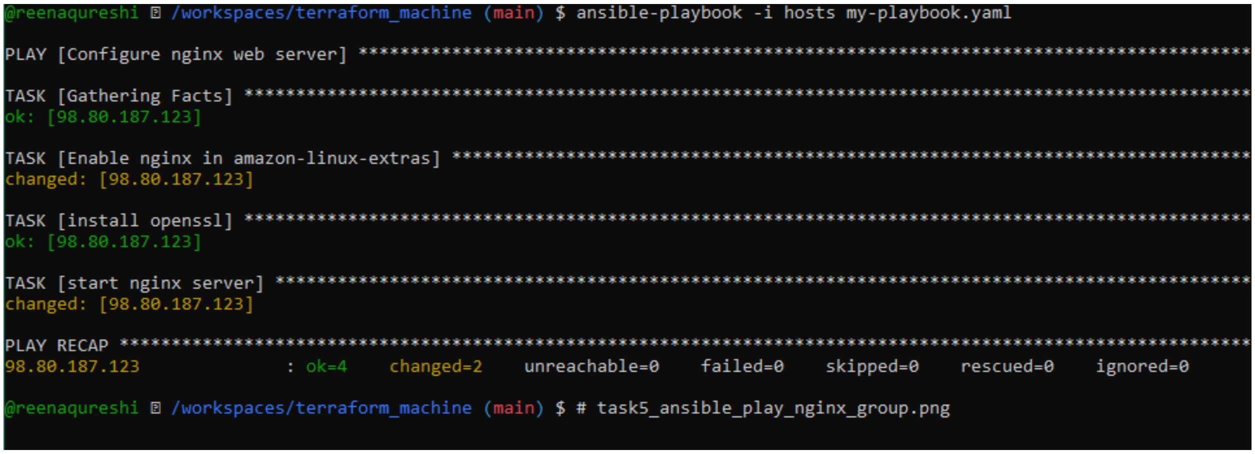
```
]
@reenaqureshi ▯ /workspaces/terraform_machine (main) $
@reenaqureshi ▯ /workspaces/terraform_machine (main) $ terraform output
webserver_public_ips = [
  "98.80.187.123",
]
@reenaqureshi ▯ /workspaces/terraform_machine (main) $
```



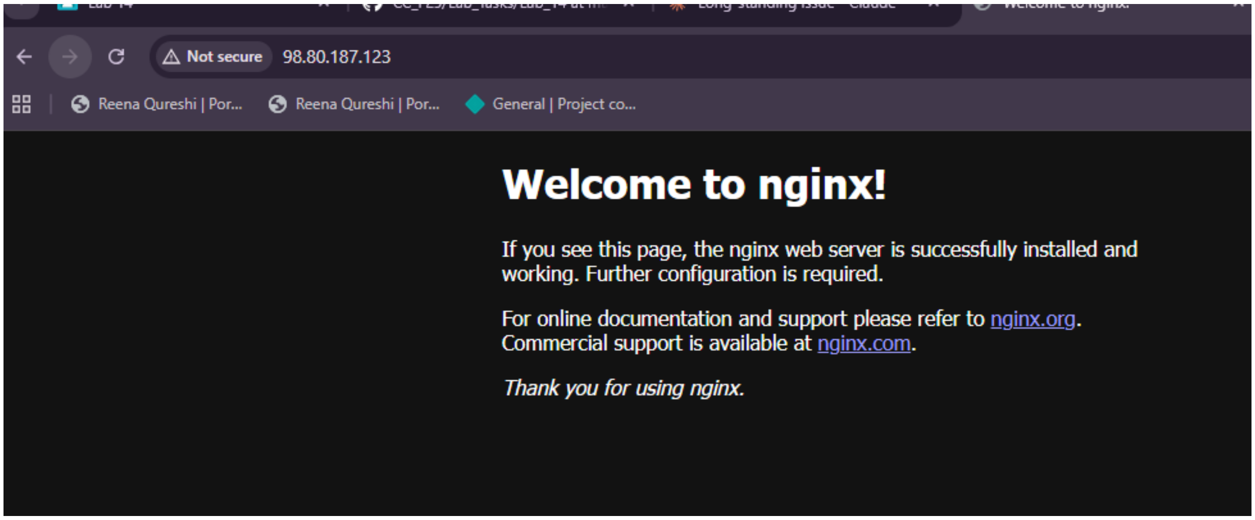
```
[nginx]
98.80.187.123
[nginx:vars]
ansible_ssh_private_key_file=~/.ssh/id_ed25519
ansible_user=ec2-user
```



```
@reenaqureshi ▯ /workspaces/terraform_machine (main) $ # screenshot
@reenaqureshi ▯ /workspaces/terraform_machine (main) $ cat my-playbook.yaml
---
- name: Configure nginx web server
  hosts: nginx
  become: true
  tasks:
    - name: Enable nginx in amazon-linux-extras
      shell: amazon-linux-extras install nginx1 -y
      args:
        creates: /usr/sbin/nginx

    - name: install openssl
      yum:
        name: openssl
        state: present

    - name: start nginx server
      service:
        name: nginx
        state: started
        enabled: true
@reenaqureshi ▯ /workspaces/terraform_machine (main) $ # task5 my playbook nginx gro
```



```
@reenaqureshi ➜ /workspaces/terraform_machine (main) $ ansible-playbook -i hosts my-playbook.yaml

PLAY [Configure nginx web server] *********************************************************************************

TASK [Gathering Facts] ********************************************************************************************
ok: [98.80.187.123]

TASK [Enable nginx in amazon-linux-extras] ************************************************************************
changed: [98.80.187.123]

TASK [install openssl] ********************************************************************************************
ok: [98.80.187.123]

TASK [start nginx server] *****************************************************************************************
changed: [98.80.187.123]

PLAY RECAP ********************************************************************************************************
98.80.187.123              : ok=4    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

@reenaqureshi ➜ /workspaces/terraform_machine (main) $ # task5_ansible_play_nginx_group.png
```



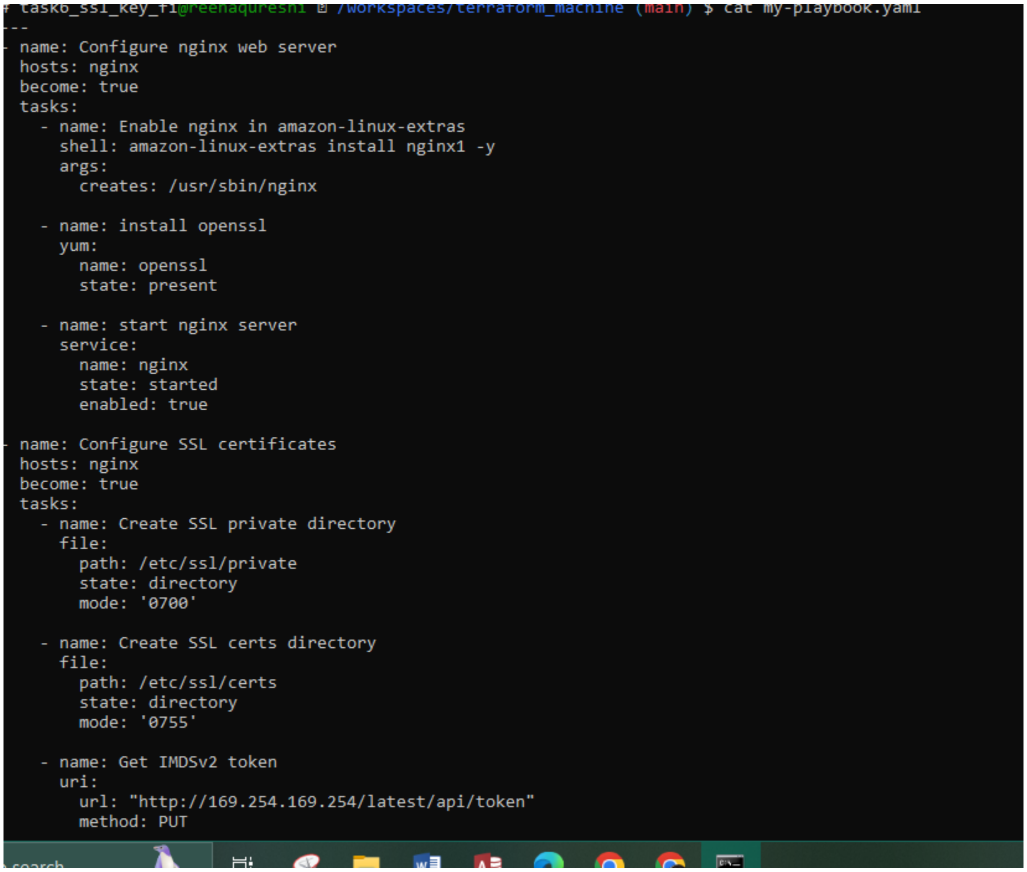
Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

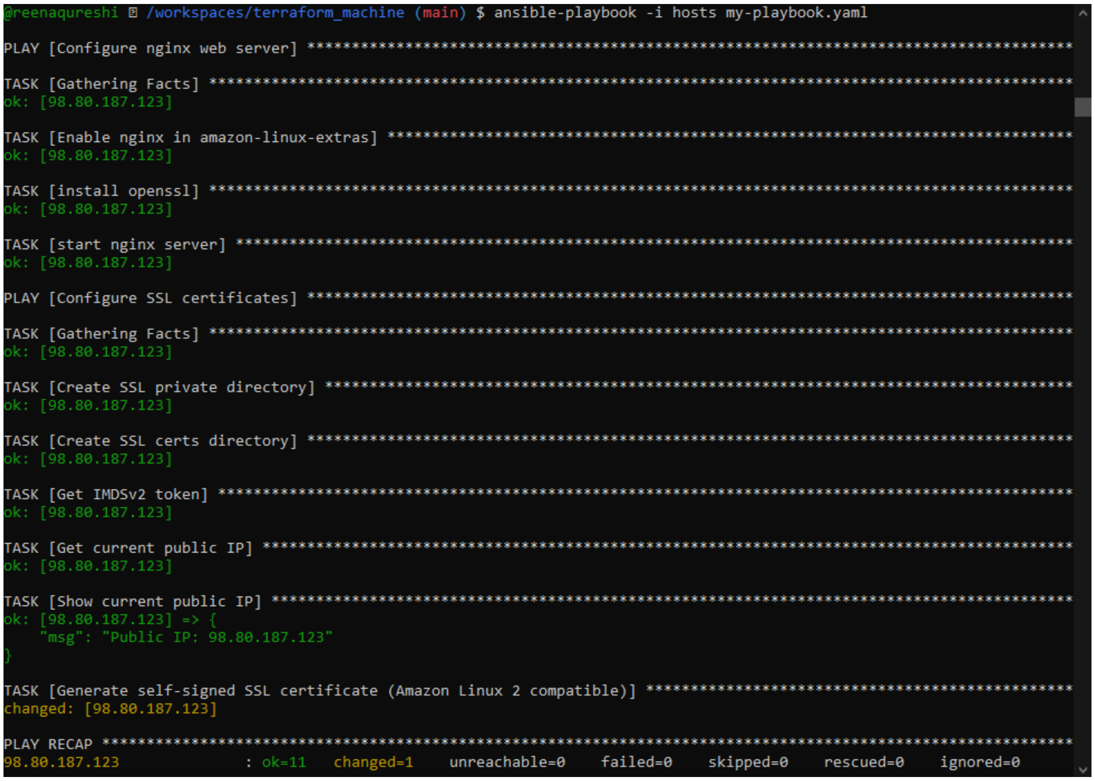
For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

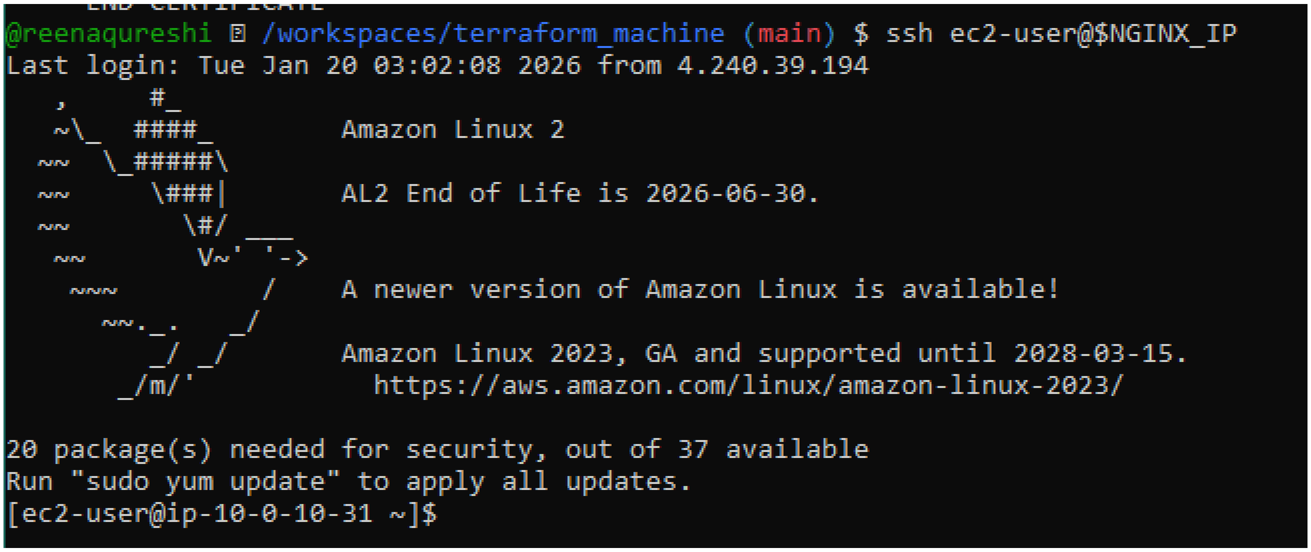
Task 6



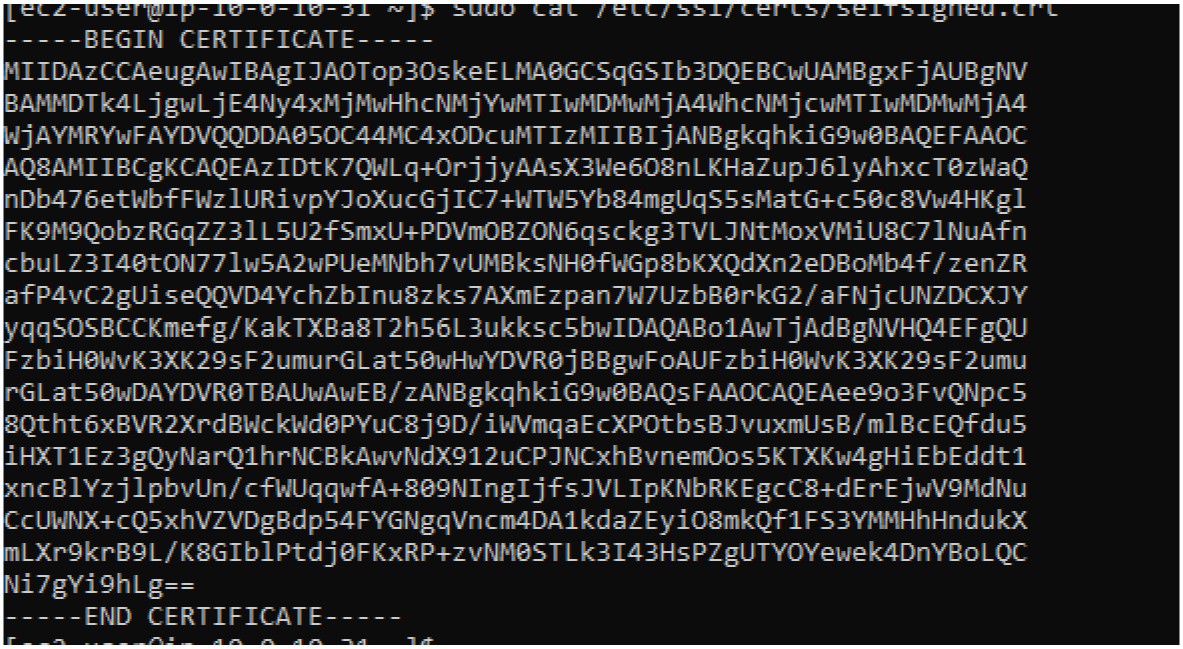
```
# task6_ssl_key_fi@reenaqureshi /workspaces/terraform_machine (main) $ cat my-playbook.yaml
---
- name: Configure nginx web server
  hosts: nginx
  become: true
  tasks:
    - name: Enable nginx in amazon-linux-extras
      shell: amazon-linux-extras install nginx1 -y
      args:
        creates: /usr/sbin/nginx

    - name: install openssl
      yum:
        name: openssl
        state: present

    - name: start nginx server
      service:
        name: nginx
        state: started
        enabled: true

- name: Configure SSL certificates
  hosts: nginx
  become: true
  tasks:
    - name: Create SSL private directory
      file:
        path: /etc/ssl/private
        state: directory
        mode: '0700'

    - name: Create SSL certs directory
      file:
        path: /etc/ssl/certs
        state: directory
        mode: '0755'

    - name: Get IMDSv2 token
      uri:
        url: "http://169.254.169.254/latest/api/token"
        method: PUT
```



```
@reenaqureshi /workspaces/terraform_machine (main) $ ansible-playbook -i hosts my-playbook.yaml

PLAY [Configure nginx web server] ***************************************************************************************

TASK [Gathering Facts] **************************************************************************************************
ok: [98.80.187.123]

TASK [Enable nginx in amazon-linux-extras] ******************************************************************************
ok: [98.80.187.123]

TASK [install openssl] **************************************************************************************************
ok: [98.80.187.123]

TASK [start nginx server] ***********************************************************************************************
ok: [98.80.187.123]

PLAY [Configure SSL certificates] ***************************************************************************************

TASK [Gathering Facts] **************************************************************************************************
ok: [98.80.187.123]

TASK [Create SSL private directory] *************************************************************************************
ok: [98.80.187.123]

TASK [Create SSL certs directory] ***************************************************************************************
ok: [98.80.187.123]

TASK [Get IMDSv2 token] *************************************************************************************************
ok: [98.80.187.123]

TASK [Get current public IP] ********************************************************************************************
ok: [98.80.187.123]

TASK [Show current public IP] *******************************************************************************************
ok: [98.80.187.123] => {
    "msg": "Public IP: 98.80.187.123"
}

TASK [Generate self-signed SSL certificate (Amazon Linux 2 compatible)] *************************************************
changed: [98.80.187.123]

PLAY RECAP **************************************************************************************************************
98.80.187.123              : ok=11   changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

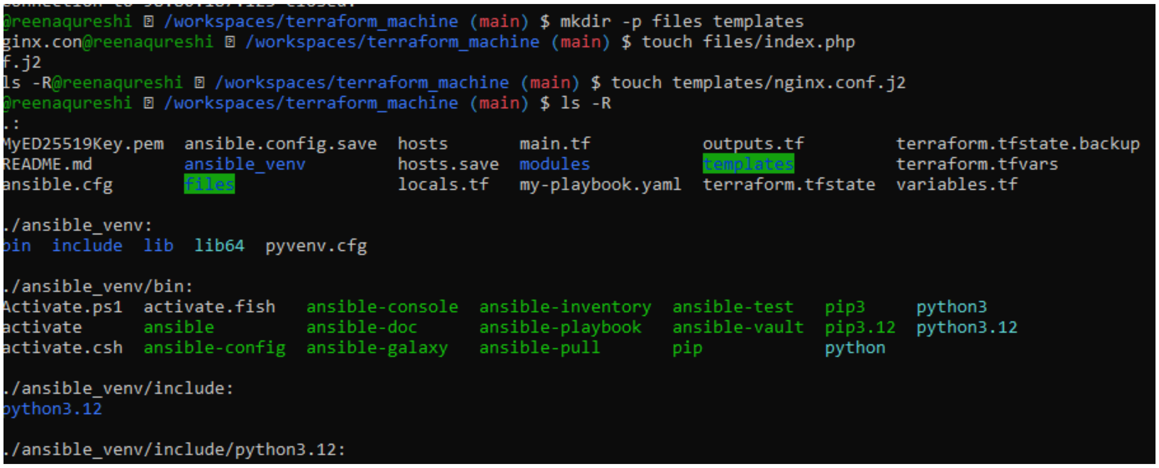


```
-----END CERTIFICATE-----
@reenaqureshi □ /workspaces/terraform_machine (main) $ ssh ec2-user@$NGINX_IP
Last login: Tue Jan 20 03:02:08 2026 from 4.240.39.194
   ,     #_
   ~\_  ####_        Amazon Linux 2
  ~~  \_#####\
  ~~     \###|       AL2 End of Life is 2026-06-30.
  ~~       \#/ ___
   ~~       V~' '->
    ~~~         /    A newer version of Amazon Linux is available!
      ~~._.   _/
         _/ _/       Amazon Linux 2023, GA and supported until 2028-03-15.
       _/m/'           https://aws.amazon.com/linux/amazon-linux-2023/

20 package(s) needed for security, out of 37 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-10-0-10-31 ~]$
```

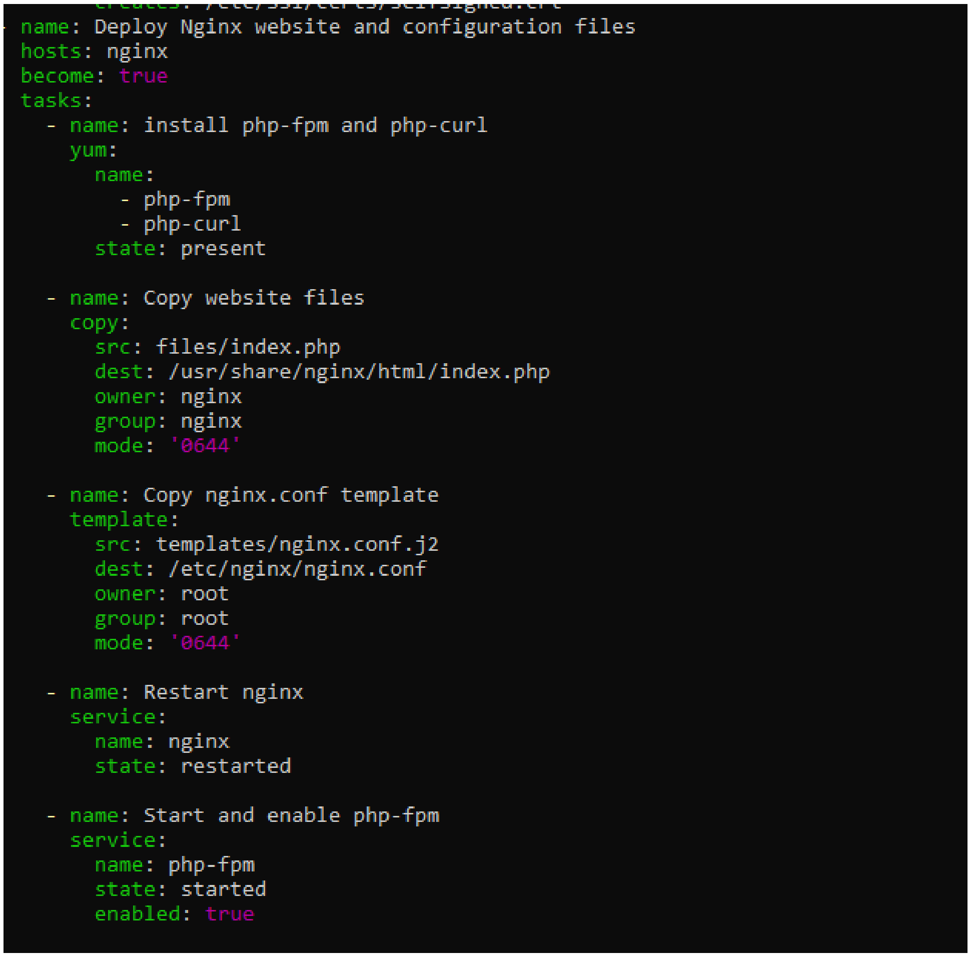


```
[ec2-user@ip-10-0-10-31 ~]$ sudo cat /etc/ssl/certs/selfsigned.crt
-----BEGIN CERTIFICATE-----
MIIDAzCCAeugAwIBAgIJAOTop3OskeELMA0GCSqGSIb3DQEBCwUAMBgxFjAUBgNV
BAMMDTk4LjgwLjE4Ny4xMjMwHhcNMjYwMTIwMDMwMjA4WhcNMjcwMTIwMDMwMjA4
WjAYMRYwFAYDVQQDDA05OC44MC4xODcuMTIzMIIBIjANBgkqhkiG9w0BAQEFAAOC
AQ8AMIIBCgKCAQEAzIDtK7QWLq+OrjjyAAsX3We6O8nLKHaZupJ6lyAhxcT0zWaQ
nDb476etWbfFWzlURivpYJoXucGjIC7+WTW5Yb84mgUqS5sMatG+c50c8Vw4HKgl
FK9M9QobzRGqZZ3lL5U2fSmxU+PDVmOBZON6qsckg3TVLJNtMoxVMiU8C7lNuAfn
cbuLZ3I40tON77lw5A2wPUeMNbh7vUMBksNH0fWGp8bKXQdXn2eDBoMb4f/zenZR
afP4vC2gUiseQQVD4YchZbInu8zks7AXmEzpan7W7UzbB0rkG2/aFNjcUNZDCXJY
yqqSOSBCCKmefg/KakTXBa8T2h56L3ukksc5bwIDAQABo1AwTjAdBgNVHQ4EFgQU
FzbiH0WvK3XK29sF2umurGLat50wHwYDVR0jBBgwFoAUFzbiH0WvK3XK29sF2umu
rGLat50wDAYDVR0TBAUwAwEB/zANBgkqhkiG9w0BAQsFAAOCAQEAee9o3FvQNpc5
8Qtht6xBVR2XrdBWckWd0PYuC8j9D/iWVmqaEcXPOtbsBJvuxmUsB/mlBcEQfdu5
iHXT1Ez3gQyNarQ1hrNCBkAwvNdX912uCPJNCxhBvnemOos5KTXKw4gHiEbEddt1
xncBlYzjlpbvUn/cfWUqqwfA+809NIngIjfsJVLIpKNbRKEgcC8+dErEjwV9MdNu
CcUWNX+cQ5xhVZVDgBdp54FYGNgqVncm4DA1kdaZEyiO8mkQf1FS3YMMHhHndukX
mLXr9krB9L/K8GIblPtdj0FKxRP+zvNM0STLk3I43HsPZgUTYOYewek4DnYBoLQC
Ni7gYi9hLg==
-----END CERTIFICATE-----
[ec2-user@ip-10-0-10-31 ~]$
```

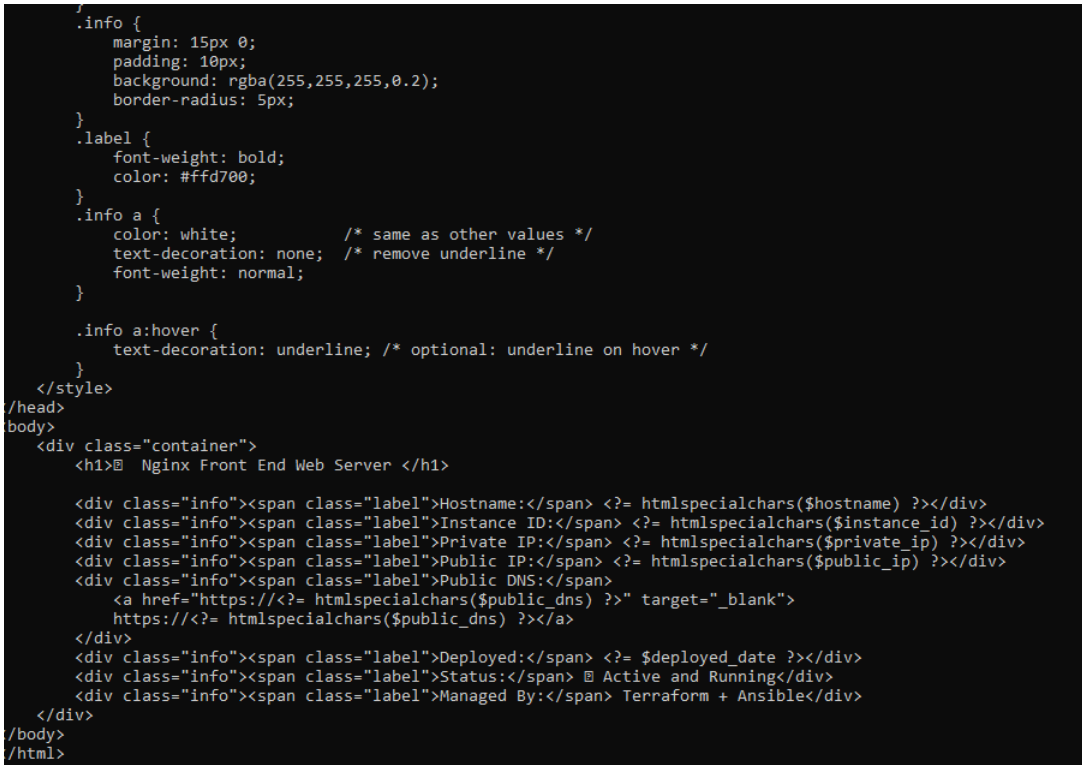
Task 7 - PHP front-end deployment with templates



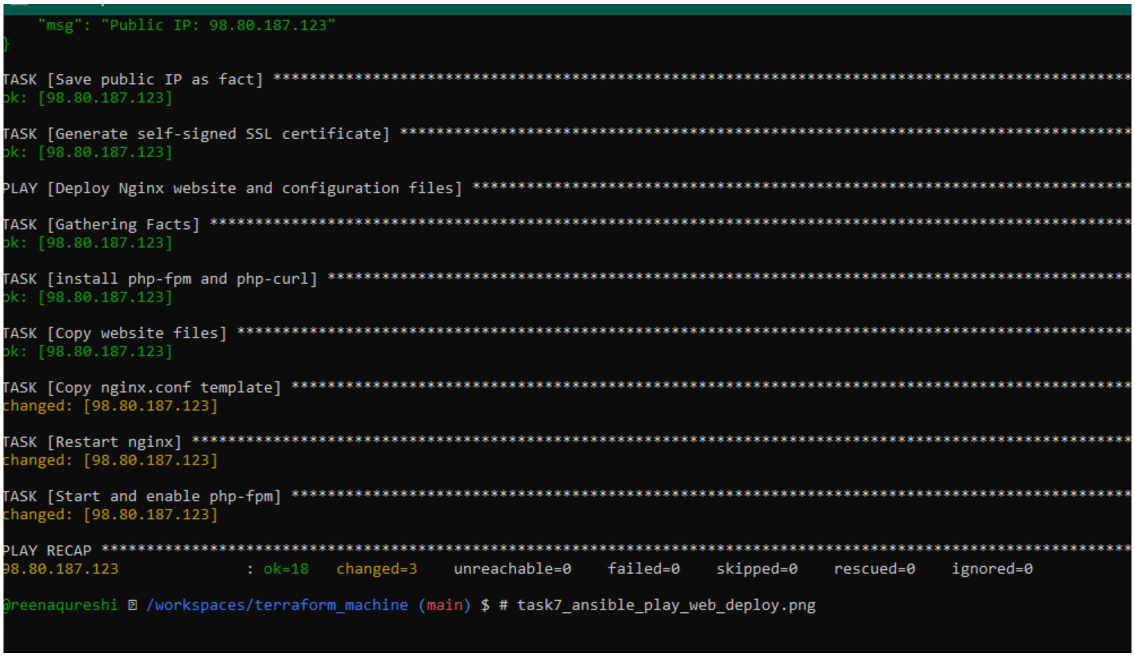
```
Connection to [illegible] closed.
@reenaqureshi □ /workspaces/terraform_machine (main) $ mkdir -p files templates
ginx.con@reenaqureshi □ /workspaces/terraform_machine (main) $ touch files/index.php
f.j2
ls -R@reenaqureshi □ /workspaces/terraform_machine (main) $ touch templates/nginx.conf.j2
@reenaqureshi □ /workspaces/terraform_machine (main) $ ls -R
.:
MyED25519Key.pem  ansible.config.save  hosts       main.tf           outputs.tf         terraform.tfstate.backup
README.md         ansible_venv         hosts.save  modules           templates          terraform.tfvars
ansible.cfg       files                locals.tf   my-playbook.yaml  terraform.tfstate  variables.tf

./ansible_venv:
bin  include  lib  lib64  pyvenv.cfg

./ansible_venv/bin:
Activate.ps1  activate.fish   ansible-console  ansible-inventory  ansible-test   pip3     python3
activate      ansible         ansible-doc      ansible-playbook   ansible-vault  pip3.12  python3.12
activate.csh  ansible-config  ansible-galaxy   ansible-pull       pip            python

./ansible_venv/include:
python3.12

./ansible_venv/include/python3.12:
```



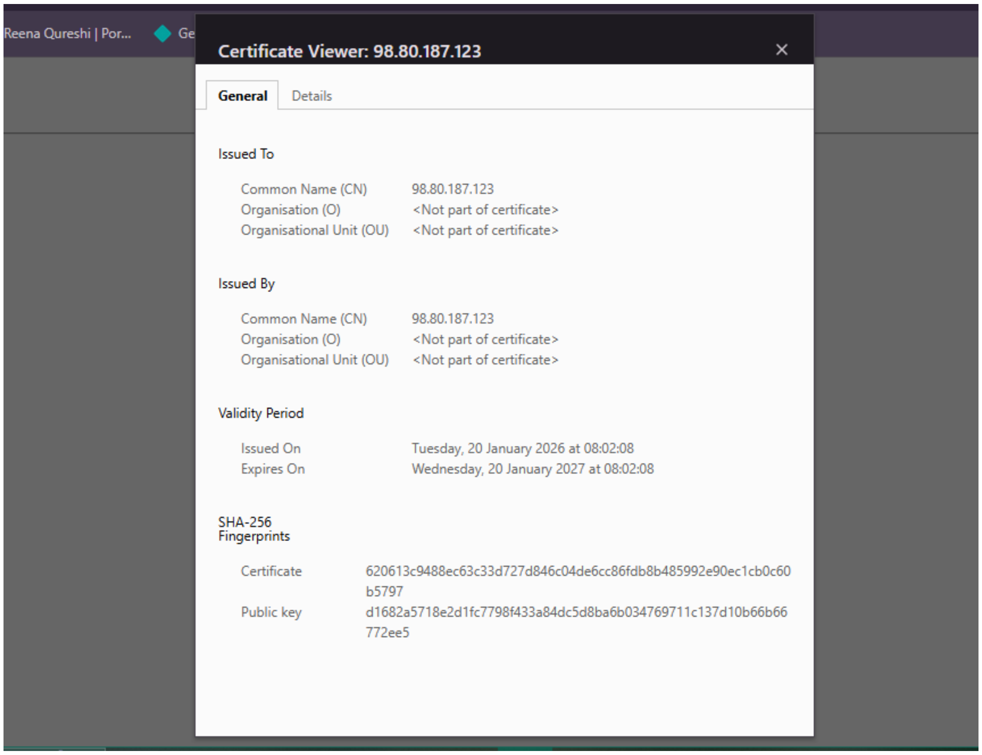
```
    creates: /etc/ssl/certs/selfsigned.crt
- name: Deploy Nginx website and configuration files
  hosts: nginx
  become: true
  tasks:
    - name: install php-fpm and php-curl
      yum:
        name:
          - php-fpm
          - php-curl
        state: present

    - name: Copy website files
      copy:
        src: files/index.php
        dest: /usr/share/nginx/html/index.php
        owner: nginx
        group: nginx
        mode: '0644'

    - name: Copy nginx.conf template
      template:
        src: templates/nginx.conf.j2
        dest: /etc/nginx/nginx.conf
        owner: root
        group: root
        mode: '0644'

    - name: Restart nginx
      service:
        name: nginx
        state: restarted

    - name: Start and enable php-fpm
      service:
        name: php-fpm
        state: started
        enabled: true
```



```
        }
        .info {
            margin: 15px 0;
            padding: 10px;
            background: rgba(255,255,255,0.2);
            border-radius: 5px;
        }
        .label {
            font-weight: bold;
            color: #ffd700;
        }
        .info a {
            color: white;           /* same as other values */
            text-decoration: none;  /* remove underline */
            font-weight: normal;
        }

        .info a:hover {
            text-decoration: underline; /* optional: underline on hover */
        }
    </style>
</head>
<body>
    <div class="container">
        <h1>⊠  Nginx Front End Web Server </h1>

        <div class="info"><span class="label">Hostname:</span> <?= htmlspecialchars($hostname) ?></div>
        <div class="info"><span class="label">Instance ID:</span> <?= htmlspecialchars($instance_id) ?></div>
        <div class="info"><span class="label">Private IP:</span> <?= htmlspecialchars($private_ip) ?></div>
        <div class="info"><span class="label">Public IP:</span> <?= htmlspecialchars($public_ip) ?></div>
        <div class="info"><span class="label">Public DNS:</span>
            <a href="https://<?= htmlspecialchars($public_dns) ?>" target="_blank">
            https://<?= htmlspecialchars($public_dns) ?></a>
        </div>
        <div class="info"><span class="label">Deployed:</span> <?= $deployed_date ?></div>
        <div class="info"><span class="label">Status:</span> ⊠ Active and Running</div>
        <div class="info"><span class="label">Managed By:</span> Terraform + Ansible</div>
    </div>
</body>
</html>
```



```
    "msg": "Public IP: 98.80.187.123"
}

TASK [Save public IP as fact] ********************************************************************************************
ok: [98.80.187.123]

TASK [Generate self-signed SSL certificate] ******************************************************************************
ok: [98.80.187.123]

PLAY [Deploy Nginx website and configuration files] **********************************************************************

TASK [Gathering Facts] ***************************************************************************************************
ok: [98.80.187.123]

TASK [install php-fpm and php-curl] **************************************************************************************
ok: [98.80.187.123]

TASK [Copy website files] ************************************************************************************************
ok: [98.80.187.123]

TASK [Copy nginx.conf template] ******************************************************************************************
changed: [98.80.187.123]

TASK [Restart nginx] *****************************************************************************************************
changed: [98.80.187.123]

TASK [Start and enable php-fpm] ******************************************************************************************
changed: [98.80.187.123]

PLAY RECAP ***************************************************************************************************************
98.80.187.123              : ok=18   changed=3    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

@reenaqureshi ⊡ /workspaces/terraform_machine (main) $ # task7_ansible_play_web_deploy.png
```



Certificate Viewer: 98.80.187.123

General | Details

Issued To

Common Name (CN)	98.80.187.123
Organisation (O)	<Not part of certificate>
Organisational Unit (OU)	<Not part of certificate>

Issued By

Common Name (CN)	98.80.187.123
Organisation (O)	<Not part of certificate>
Organisational Unit (OU)	<Not part of certificate>

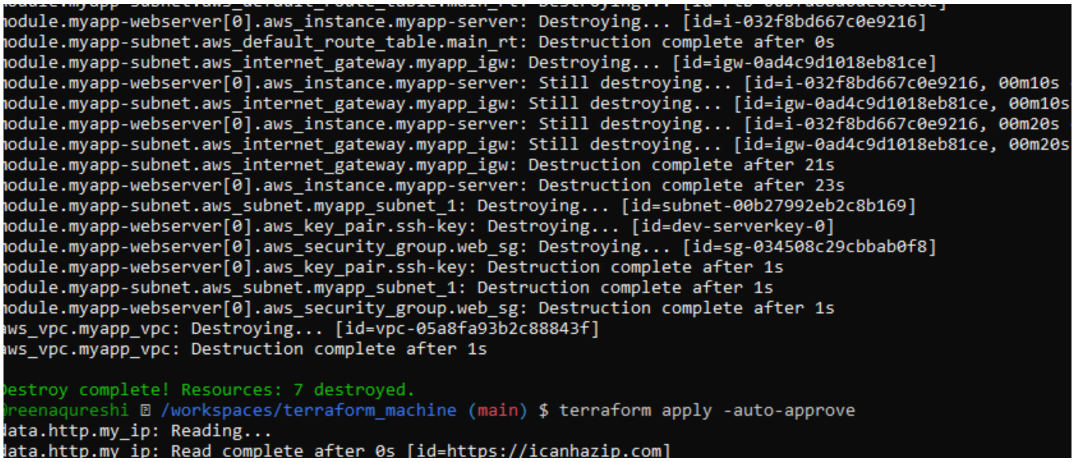
Validity Period

Issued On	Tuesday, 20 January 2026 at 08:02:08
Expires On	Wednesday, 20 January 2027 at 08:02:08

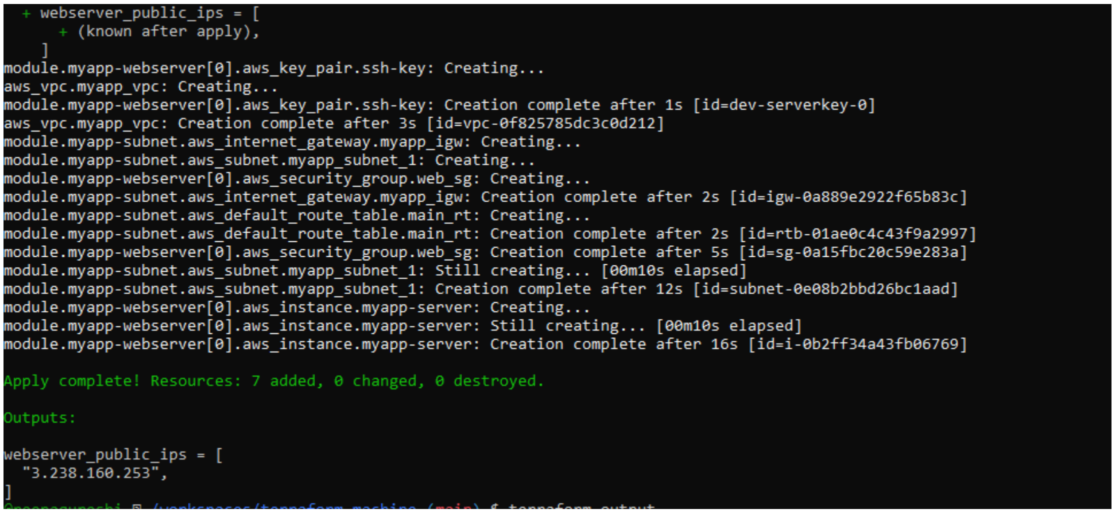
SHA-256 Fingerprints

Certificate	620613c9488ec63c33d727d846c04de6cc86fdb8b485992e90ec1cb0c60b5797
Public key	d1682a5718e2d1fc7798f433a84dc5d8ba6b034769711c137d10b66b66772ee5

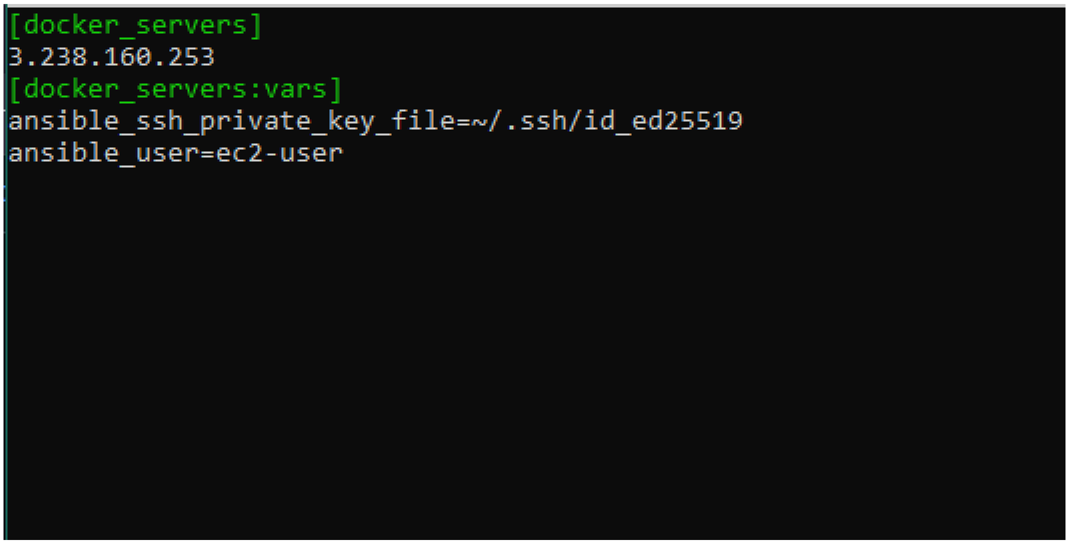
Task 8 – Docker & Docker Compose provisioning via Ansible



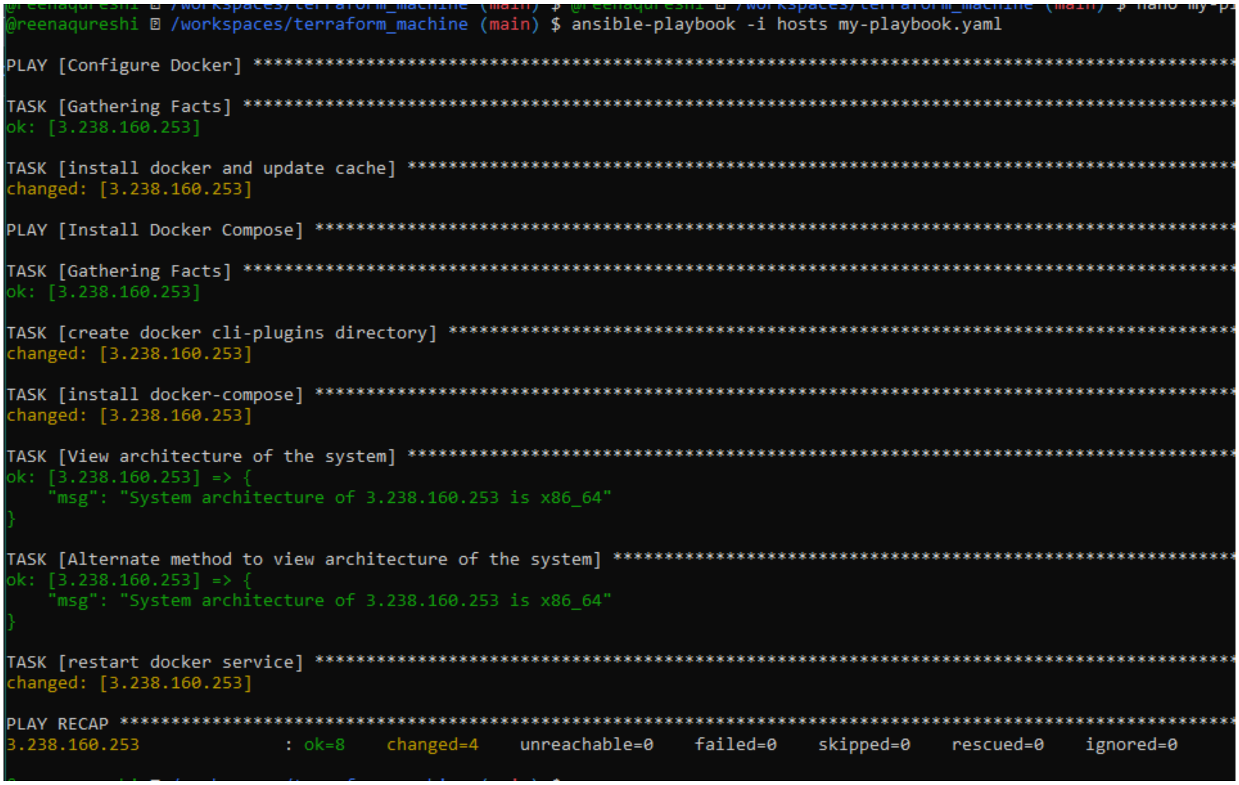
```
odule.myapp-webserver[0].aws_instance.myapp-server: Destroying... [id=i-032f8bd667c0e9216]
odule.myapp-subnet.aws_default_route_table.main_rt: Destruction complete after 0s
odule.myapp-subnet.aws_internet_gateway.myapp_igw: Destroying... [id=igw-0ad4c9d1018eb81ce]
odule.myapp-webserver[0].aws_instance.myapp-server: Still destroying... [id=i-032f8bd667c0e9216, 00m10s
odule.myapp-subnet.aws_internet_gateway.myapp_igw: Still destroying... [id=igw-0ad4c9d1018eb81ce, 00m10s
odule.myapp-webserver[0].aws_instance.myapp-server: Still destroying... [id=i-032f8bd667c0e9216, 00m20s
odule.myapp-subnet.aws_internet_gateway.myapp_igw: Still destroying... [id=igw-0ad4c9d1018eb81ce, 00m20s
odule.myapp-subnet.aws_internet_gateway.myapp_igw: Destruction complete after 21s
odule.myapp-webserver[0].aws_instance.myapp-server: Destruction complete after 23s
odule.myapp-subnet.aws_subnet.myapp_subnet_1: Destroying... [id=subnet-00b27992eb2c8b169]
odule.myapp-webserver[0].aws_key_pair.ssh-key: Destroying... [id=dev-serverkey-0]
odule.myapp-webserver[0].aws_security_group.web_sg: Destroying... [id=sg-034508c29cbbab0f8]
odule.myapp-webserver[0].aws_key_pair.ssh-key: Destruction complete after 1s
odule.myapp-subnet.aws_subnet.myapp_subnet_1: Destruction complete after 1s
odule.myapp-webserver[0].aws_security_group.web_sg: Destruction complete after 1s
ws_vpc.myapp_vpc: Destroying... [id=vpc-05a8fa93b2c88843f]
ws_vpc.myapp_vpc: Destruction complete after 1s

estroy complete! Resources: 7 destroyed.
reenaqureshi ▯ /workspaces/terraform_machine (main) $ terraform apply -auto-approve
ata.http.my_ip: Reading...
ata.http.my_ip: Read complete after 0s [id=https://icanhazip.com]
```



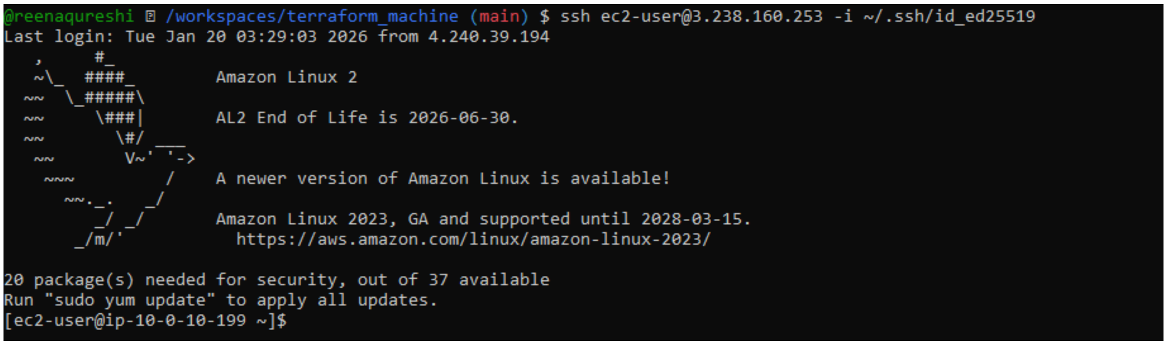
```
  + webserver_public_ips = [
      + (known after apply),
    ]
module.myapp-webserver[0].aws_key_pair.ssh-key: Creating...
aws_vpc.myapp_vpc: Creating...
module.myapp-webserver[0].aws_key_pair.ssh-key: Creation complete after 1s [id=dev-serverkey-0]
aws_vpc.myapp_vpc: Creation complete after 3s [id=vpc-0f825785dc3c0d212]
module.myapp-subnet.aws_internet_gateway.myapp_igw: Creating...
module.myapp-subnet.aws_subnet.myapp_subnet_1: Creating...
module.myapp-webserver[0].aws_security_group.web_sg: Creating...
module.myapp-subnet.aws_internet_gateway.myapp_igw: Creation complete after 2s [id=igw-0a889e2922f65b83c]
module.myapp-subnet.aws_default_route_table.main_rt: Creating...
module.myapp-subnet.aws_default_route_table.main_rt: Creation complete after 2s [id=rtb-01ae0c4c43f9a2997]
module.myapp-webserver[0].aws_security_group.web_sg: Creation complete after 5s [id=sg-0a15fbc20c59e283a]
module.myapp-subnet.aws_subnet.myapp_subnet_1: Still creating... [00m10s elapsed]
module.myapp-subnet.aws_subnet.myapp_subnet_1: Creation complete after 12s [id=subnet-0e08b2bbd26bc1aad]
module.myapp-webserver[0].aws_instance.myapp-server: Creating...
module.myapp-webserver[0].aws_instance.myapp-server: Still creating... [00m10s elapsed]
module.myapp-webserver[0].aws_instance.myapp-server: Creation complete after 16s [id=i-0b2ff34a43fb06769]

Apply complete! Resources: 7 added, 0 changed, 0 destroyed.

Outputs:

webserver_public_ips = [
  "3.238.160.253",
]
@reenaqureshi ▯ /workspaces/terraform_machine (main) $ terraform output
```



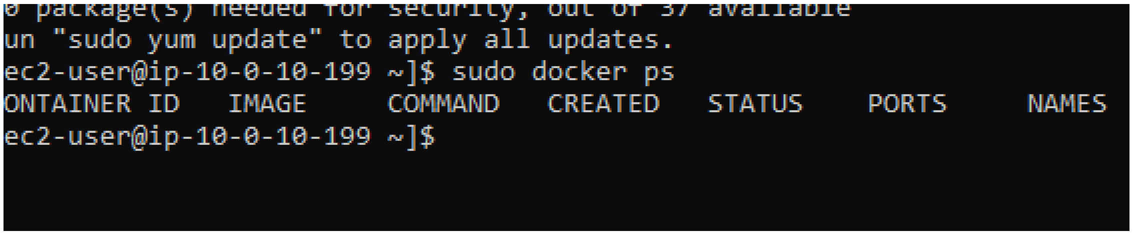
```
[docker_servers]
3.238.160.253
[docker_servers:vars]
ansible_ssh_private_key_file=~/.ssh/id_ed25519
ansible_user=ec2-user
```



```
@reenaqureshi ▯ /workspaces/terraform_machine (main) $ @reenaqureshi ▯ /workspaces/terraform_machine (main) $ nano my-p
@reenaqureshi ▯ /workspaces/terraform_machine (main) $ ansible-playbook -i hosts my-playbook.yaml

PLAY [Configure Docker] ****************************************************************************

TASK [Gathering Facts] *****************************************************************************
ok: [3.238.160.253]

TASK [install docker and update cache] *************************************************************
changed: [3.238.160.253]

PLAY [Install Docker Compose] **********************************************************************

TASK [Gathering Facts] *****************************************************************************
ok: [3.238.160.253]

TASK [create docker cli-plugins directory] *********************************************************
changed: [3.238.160.253]

TASK [install docker-compose] **********************************************************************
changed: [3.238.160.253]

TASK [View architecture of the system] *************************************************************
ok: [3.238.160.253] => {
    "msg": "System architecture of 3.238.160.253 is x86_64"
}

TASK [Alternate method to view architecture of the system] *****************************************
ok: [3.238.160.253] => {
    "msg": "System architecture of 3.238.160.253 is x86_64"
}

TASK [restart docker service] **********************************************************************
changed: [3.238.160.253]

PLAY RECAP *****************************************************************************************
3.238.160.253              : ok=8    changed=4    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

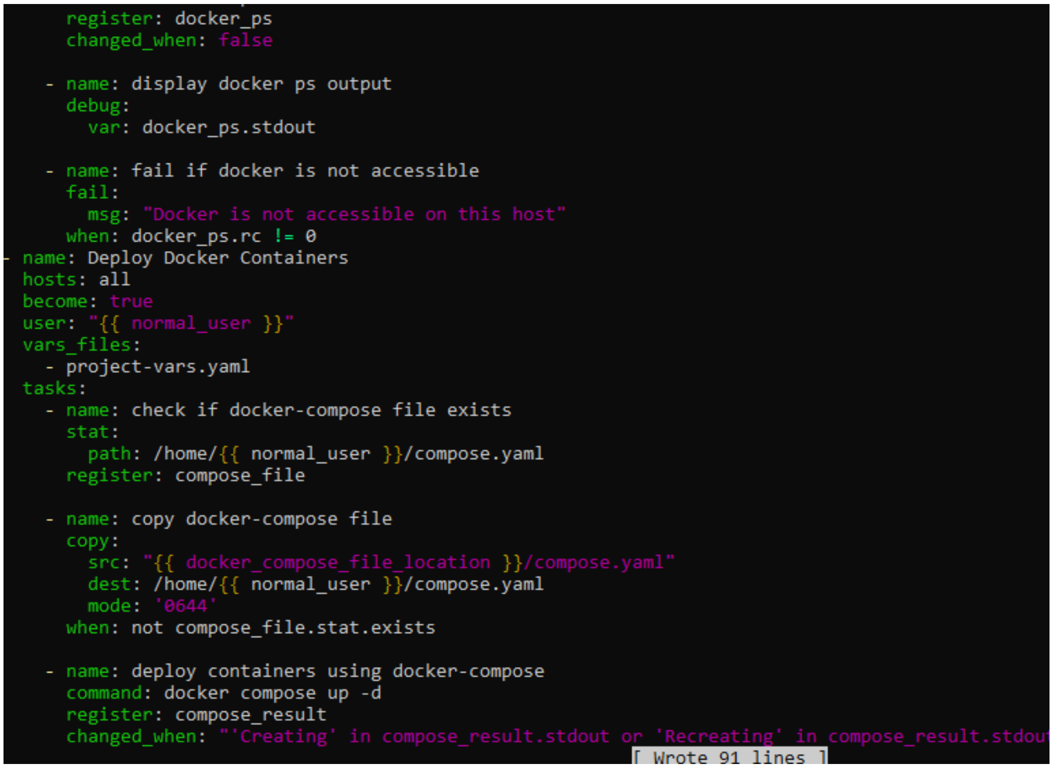


```
@reenaqureshi ▯ /workspaces/terraform_machine (main) $ ssh ec2-user@3.238.160.253 -i ~/.ssh/id_ed25519
Last login: Tue Jan 20 03:29:03 2026 from 4.240.39.194
   ,     #_
   ~\_  ####_        Amazon Linux 2
  ~~  \_#####\
  ~~     \###|       AL2 End of Life is 2026-06-30.
  ~~       \#/ ___
   ~~       V~' '->
    ~~~         /    A newer version of Amazon Linux is available!
      ~~._.   _/
         _/ _/       Amazon Linux 2023, GA and supported until 2028-03-15.
       _/m/'           https://aws.amazon.com/linux/amazon-linux-2023/

20 package(s) needed for security, out of 37 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-10-0-10-199 ~]$
```

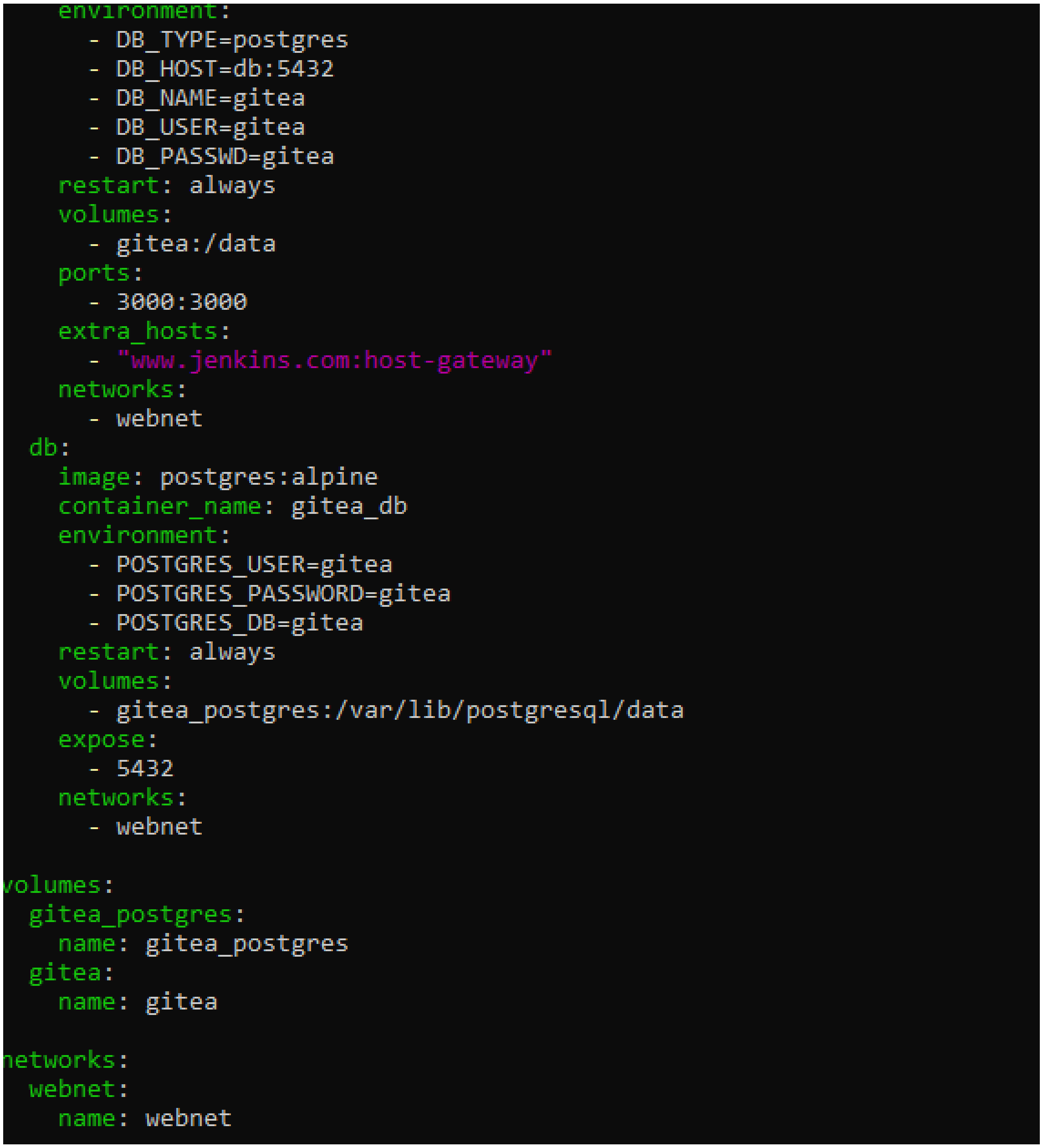


```
0 package(s) needed for security, out of 37 available
un "sudo yum update" to apply all updates.
ec2-user@ip-10-0-10-199 ~]$ sudo docker ps
ONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
ec2-user@ip-10-0-10-199 ~]$
```

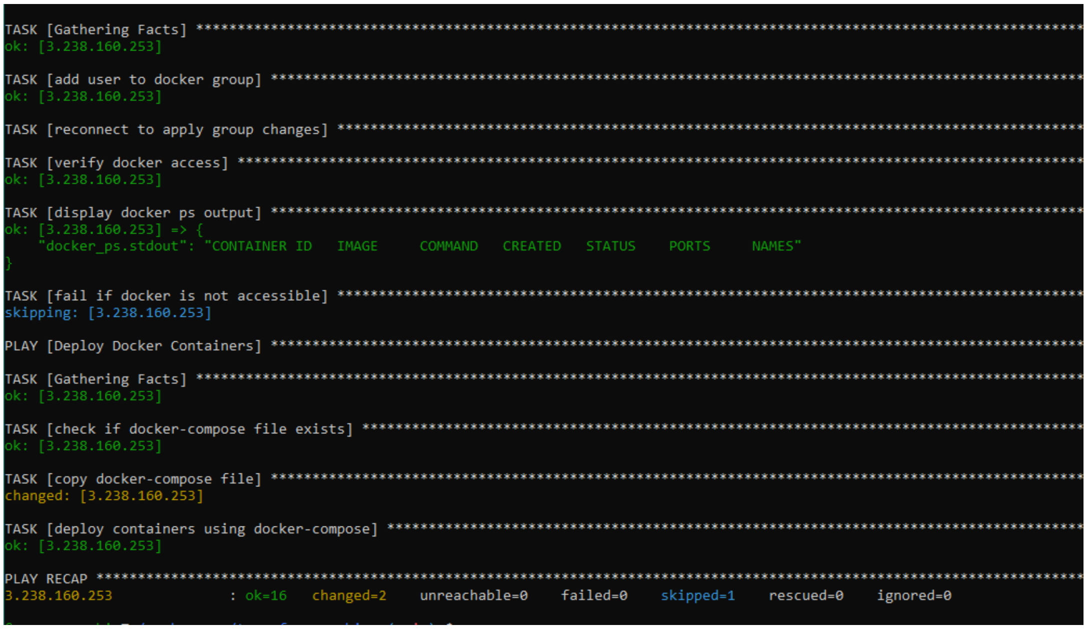
Task 9 – Gitea Docker stack via Ansible + Terraform security group update



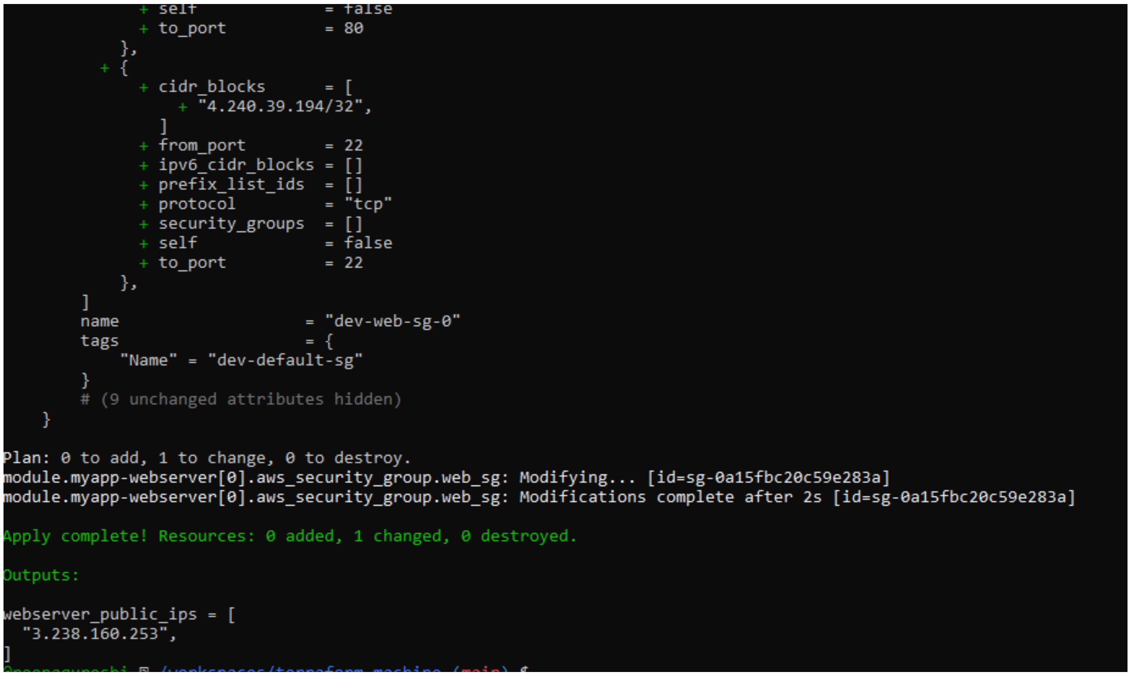
```
      register: docker_ps
      changed_when: false

    - name: display docker ps output
      debug:
        var: docker_ps.stdout

    - name: fail if docker is not accessible
      fail:
        msg: "Docker is not accessible on this host"
      when: docker_ps.rc != 0
- name: Deploy Docker Containers
  hosts: all
  become: true
  user: "{{ normal_user }}"
  vars_files:
    - project-vars.yaml
  tasks:
    - name: check if docker-compose file exists
      stat:
        path: /home/{{ normal_user }}/compose.yaml
      register: compose_file

    - name: copy docker-compose file
      copy:
        src: "{{ docker_compose_file_location }}/compose.yaml"
        dest: /home/{{ normal_user }}/compose.yaml
        mode: '0644'
      when: not compose_file.stat.exists

    - name: deploy containers using docker-compose
      command: docker compose up -d
      register: compose_result
      changed_when: "'Creating' in compose_result.stdout or 'Recreating' in compose_result.stdout
                                                          [ Wrote 91 lines ]
```



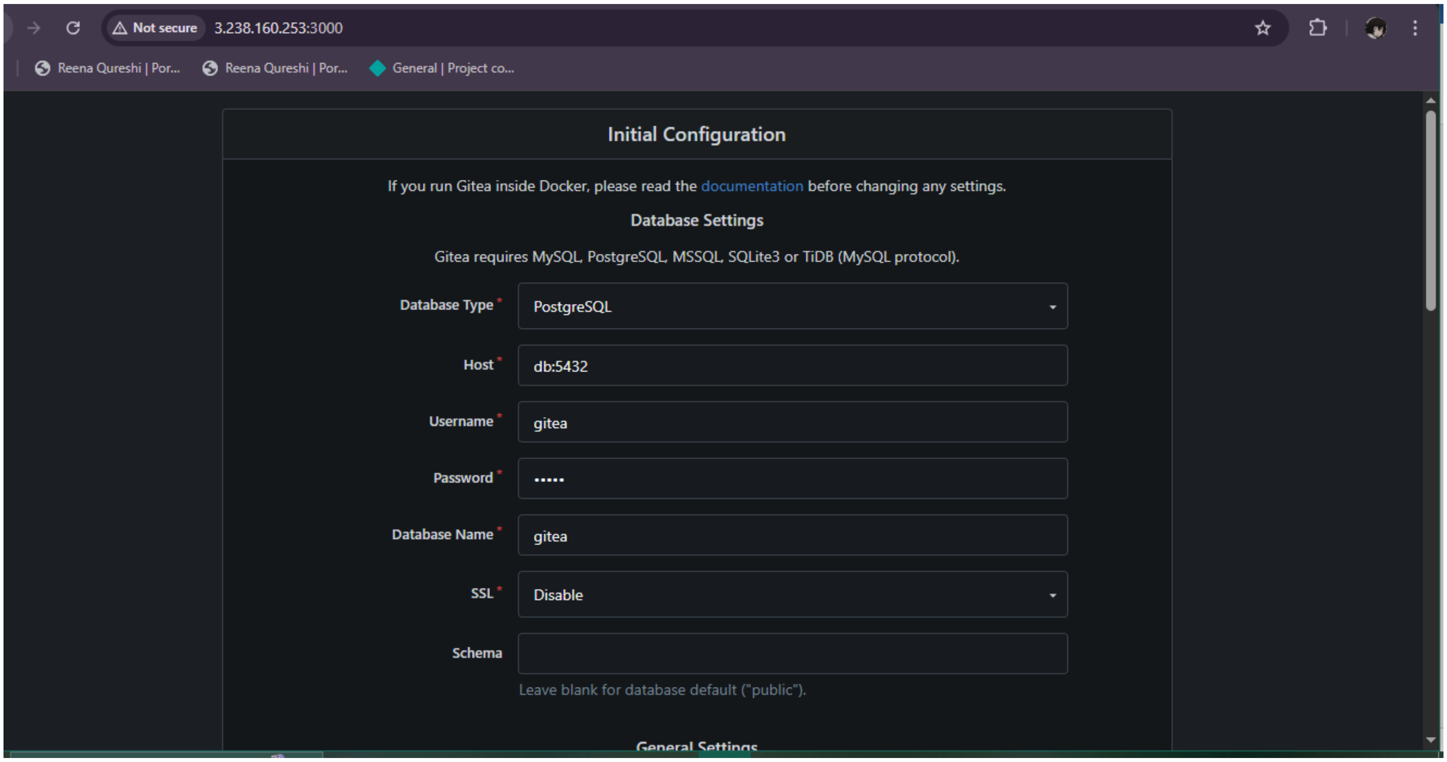
```
    environment:
      - DB_TYPE=postgres
      - DB_HOST=db:5432
      - DB_NAME=gitea
      - DB_USER=gitea
      - DB_PASSWD=gitea
    restart: always
    volumes:
      - gitea:/data
    ports:
      - 3000:3000
    extra_hosts:
      - "www.jenkins.com:host-gateway"
    networks:
      - webnet
  db:
    image: postgres:alpine
    container_name: gitea_db
    environment:
      - POSTGRES_USER=gitea
      - POSTGRES_PASSWORD=gitea
      - POSTGRES_DB=gitea
    restart: always
    volumes:
      - gitea_postgres:/var/lib/postgresql/data
    expose:
      - 5432
    networks:
      - webnet

volumes:
  gitea_postgres:
    name: gitea_postgres
  gitea:
    name: gitea

networks:
  webnet:
    name: webnet
```



```
TASK [Gathering Facts] ***********************************************************************************************************
ok: [3.238.160.253]

TASK [add user to docker group] **************************************************************************************************
ok: [3.238.160.253]

TASK [reconnect to apply group changes] ******************************************************************************************

TASK [verify docker access] ******************************************************************************************************
ok: [3.238.160.253]

TASK [display docker ps output] **************************************************************************************************
ok: [3.238.160.253] => {
    "docker_ps.stdout": "CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES"
}

TASK [fail if docker is not accessible] ******************************************************************************************
skipping: [3.238.160.253]

PLAY [Deploy Docker Containers] **************************************************************************************************

TASK [Gathering Facts] ***********************************************************************************************************
ok: [3.238.160.253]

TASK [check if docker-compose file exists] ***************************************************************************************
ok: [3.238.160.253]

TASK [copy docker-compose file] **************************************************************************************************
changed: [3.238.160.253]

TASK [deploy containers using docker-compose] ************************************************************************************
ok: [3.238.160.253]

PLAY RECAP ***********************************************************************************************************************
3.238.160.253              : ok=16   changed=2    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
```



```
              + self              = false
              + to_port           = 80
            },
          + {
              + cidr_blocks       = [
                  + "4.240.39.194/32",
                ]
              + from_port         = 22
              + ipv6_cidr_blocks  = []
              + prefix_list_ids   = []
              + protocol          = "tcp"
              + security_groups   = []
              + self              = false
              + to_port           = 22
            },
        ]
        name                    = "dev-web-sg-0"
        tags                    = {
            "Name" = "dev-default-sg"
        }
        # (9 unchanged attributes hidden)
    }

Plan: 0 to add, 1 to change, 0 to destroy.
module.myapp-webserver[0].aws_security_group.web_sg: Modifying... [id=sg-0a15fbc20c59e283a]
module.myapp-webserver[0].aws_security_group.web_sg: Modifications complete after 2s [id=sg-0a15fbc20c59e283a]

Apply complete! Resources: 0 added, 1 changed, 0 destroyed.

Outputs:

webserver_public_ips = [
  "3.238.160.253",
]
@reenaqureshi ➜ /workspaces/terraform-machine (main) $
```



Initial Configuration

If you run Gitea inside Docker, please read the documentation before changing any settings.

Database Settings

Gitea requires MySQL, PostgreSQL, MSSQL, SQLite3 or TiDB (MySQL protocol).

Database Type * PostgreSQL

Host * db:5432

Username * gitea

Password * •••••

Database Name * gitea

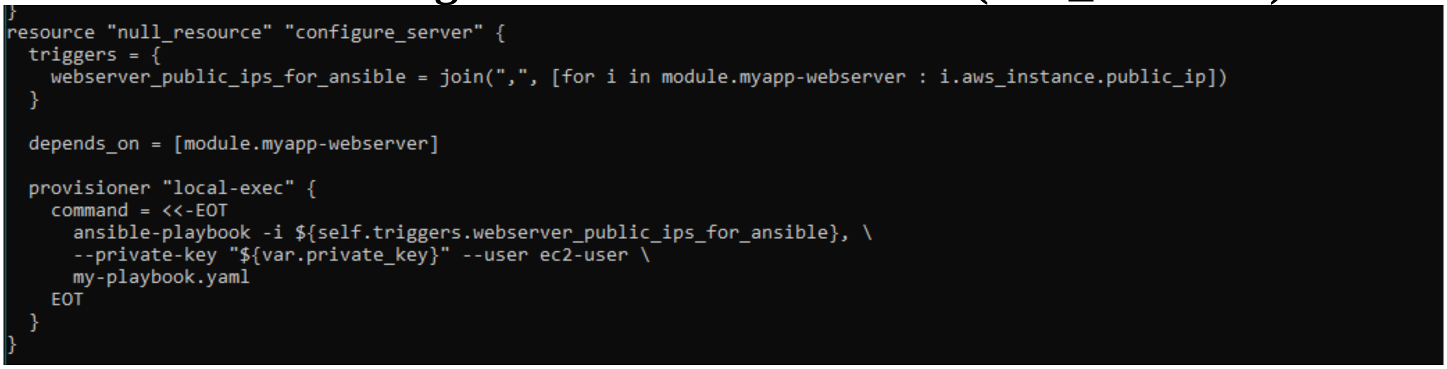
SSL * Disable

Schema

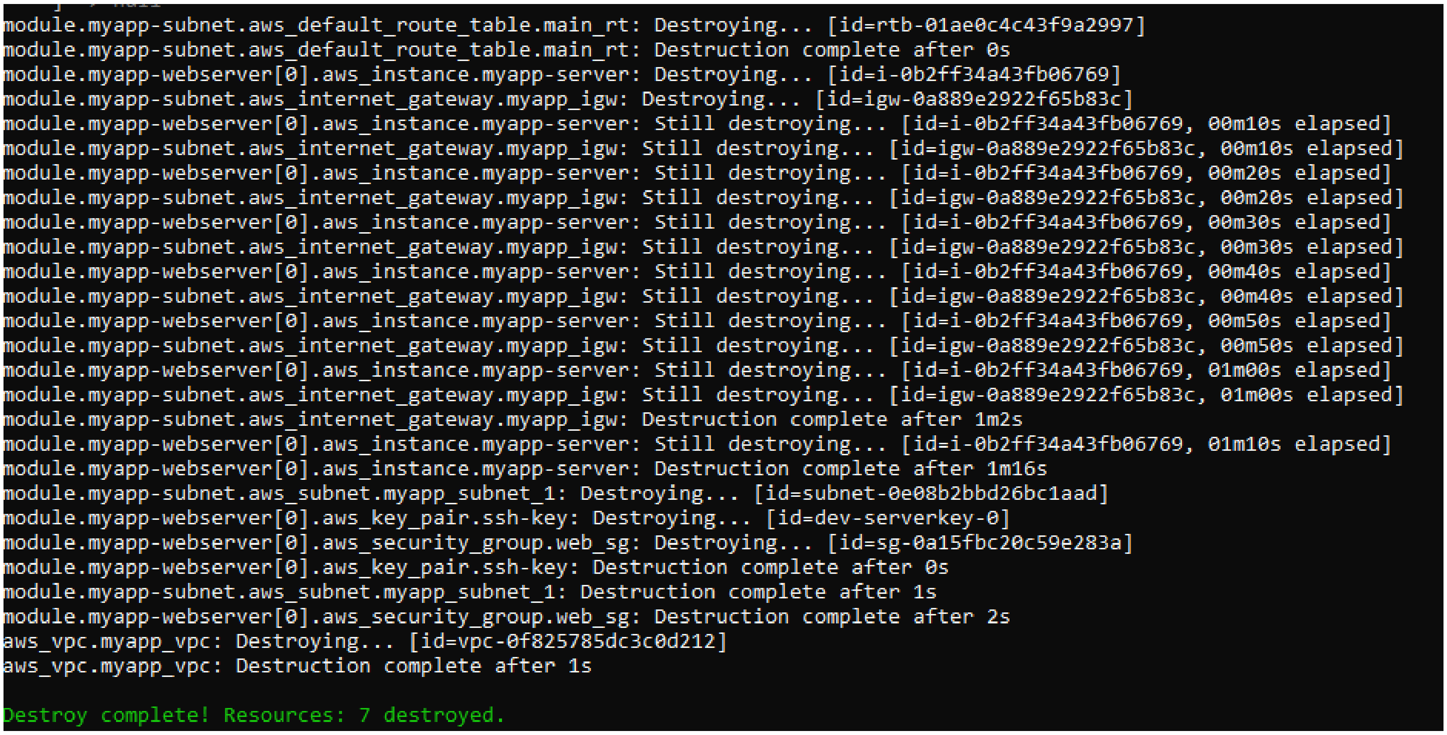
Leave blank for database default ("public").

General Settings

Task 10 – Automating Ansible with Terraform (null_resource)

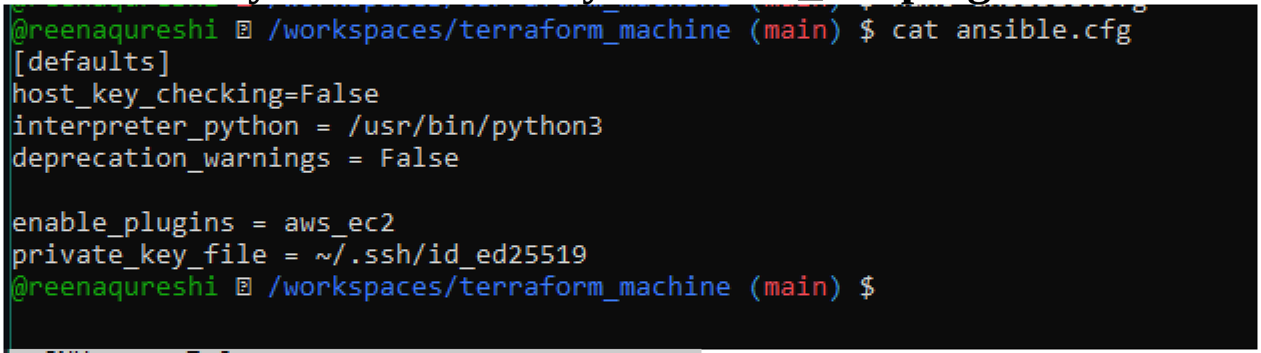


```
}
resource "null_resource" "configure_server" {
  triggers = {
    webserver_public_ips_for_ansible = join(",", [for i in module.myapp-webserver : i.aws_instance.public_ip])
  }

  depends_on = [module.myapp-webserver]

  provisioner "local-exec" {
    command = <<-EOT
      ansible-playbook -i ${self.triggers.webserver_public_ips_for_ansible}, \
      --private-key "${var.private_key}" --user ec2-user \
      my-playbook.yaml
    EOT
  }
}
```

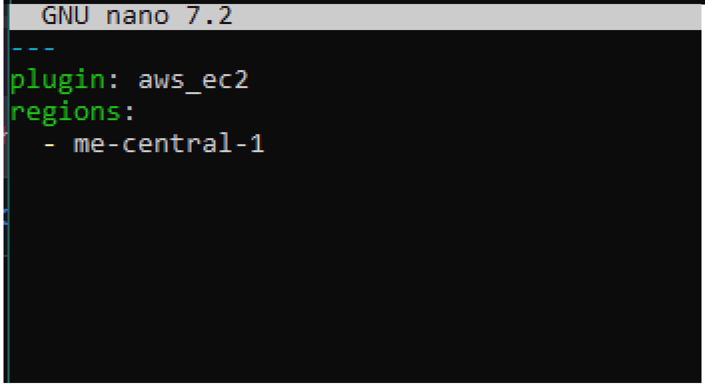


```
module.myapp-subnet.aws_default_route_table.main_rt: Destroying... [id=rtb-01ae0c4c43f9a2997]
module.myapp-subnet.aws_default_route_table.main_rt: Destruction complete after 0s
module.myapp-webserver[0].aws_instance.myapp-server: Destroying... [id=i-0b2ff34a43fb06769]
module.myapp-subnet.aws_internet_gateway.myapp_igw: Destroying... [id=igw-0a889e2922f65b83c]
module.myapp-webserver[0].aws_instance.myapp-server: Still destroying... [id=i-0b2ff34a43fb06769, 00m10s elapsed]
module.myapp-subnet.aws_internet_gateway.myapp_igw: Still destroying... [id=igw-0a889e2922f65b83c, 00m10s elapsed]
module.myapp-webserver[0].aws_instance.myapp-server: Still destroying... [id=i-0b2ff34a43fb06769, 00m20s elapsed]
module.myapp-subnet.aws_internet_gateway.myapp_igw: Still destroying... [id=igw-0a889e2922f65b83c, 00m20s elapsed]
module.myapp-webserver[0].aws_instance.myapp-server: Still destroying... [id=i-0b2ff34a43fb06769, 00m30s elapsed]
module.myapp-subnet.aws_internet_gateway.myapp_igw: Still destroying... [id=igw-0a889e2922f65b83c, 00m30s elapsed]
module.myapp-webserver[0].aws_instance.myapp-server: Still destroying... [id=i-0b2ff34a43fb06769, 00m40s elapsed]
module.myapp-subnet.aws_internet_gateway.myapp_igw: Still destroying... [id=igw-0a889e2922f65b83c, 00m40s elapsed]
module.myapp-webserver[0].aws_instance.myapp-server: Still destroying... [id=i-0b2ff34a43fb06769, 00m50s elapsed]
module.myapp-subnet.aws_internet_gateway.myapp_igw: Still destroying... [id=igw-0a889e2922f65b83c, 00m50s elapsed]
module.myapp-webserver[0].aws_instance.myapp-server: Still destroying... [id=i-0b2ff34a43fb06769, 01m00s elapsed]
module.myapp-subnet.aws_internet_gateway.myapp_igw: Still destroying... [id=igw-0a889e2922f65b83c, 01m00s elapsed]
module.myapp-subnet.aws_internet_gateway.myapp_igw: Destruction complete after 1m2s
module.myapp-webserver[0].aws_instance.myapp-server: Still destroying... [id=i-0b2ff34a43fb06769, 01m10s elapsed]
module.myapp-webserver[0].aws_instance.myapp-server: Destruction complete after 1m16s
module.myapp-subnet.aws_subnet.myapp_subnet_1: Destroying... [id=subnet-0e08b2bbd26bc1aad]
module.myapp-webserver[0].aws_key_pair.ssh-key: Destroying... [id=dev-serverkey-0]
module.myapp-webserver[0].aws_security_group.web_sg: Destroying... [id=sg-0a15fbc20c59e283a]
module.myapp-webserver[0].aws_key_pair.ssh-key: Destruction complete after 0s
module.myapp-subnet.aws_subnet.myapp_subnet_1: Destruction complete after 1s
module.myapp-webserver[0].aws_security_group.web_sg: Destruction complete after 2s
aws_vpc.myapp_vpc: Destroying... [id=vpc-0f825785dc3c0d212]
aws_vpc.myapp_vpc: Destruction complete after 1s

Destroy complete! Resources: 7 destroyed.
```

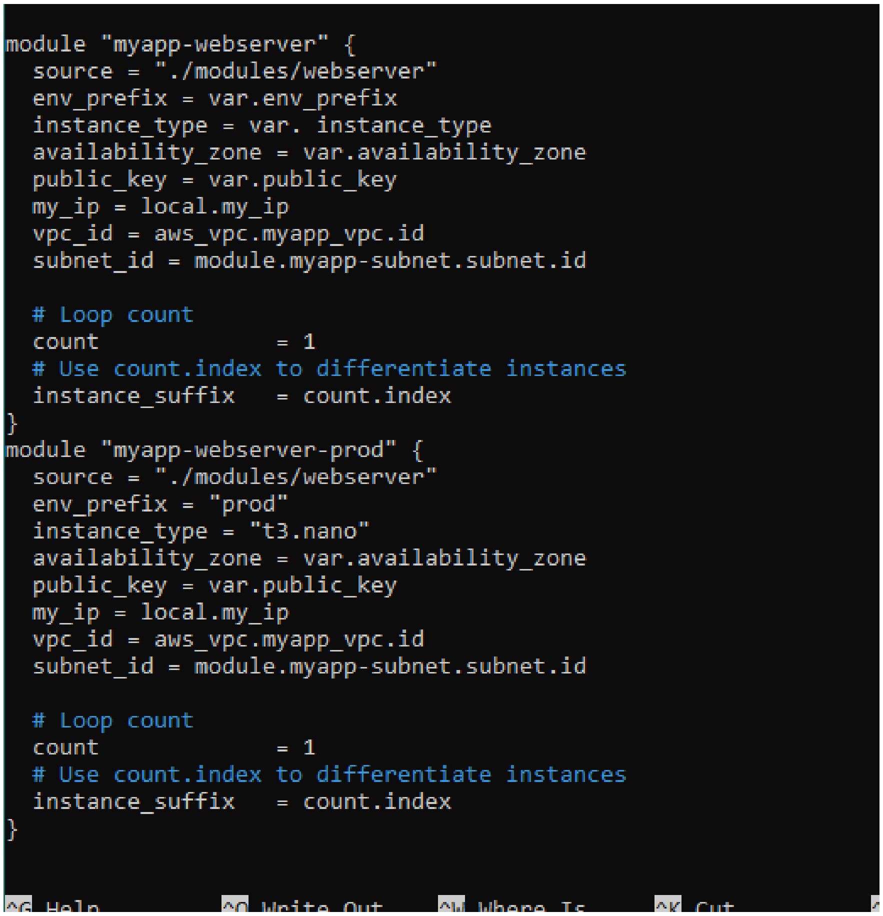
Task 11 – Dynamic inventory with aws_ec2 plugin



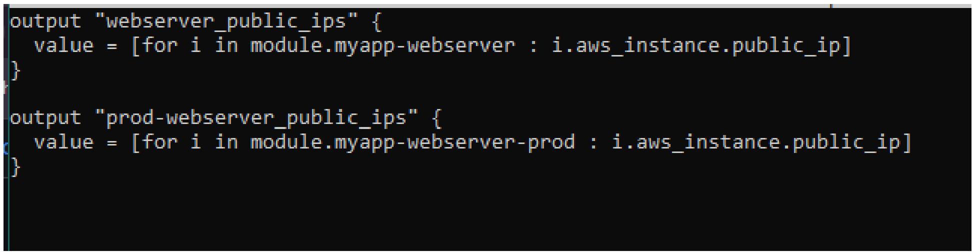
```
@reenaqureshi ➜ /workspaces/terraform_machine (main) $ cat ansible.cfg
[defaults]
host_key_checking=False
interpreter_python = /usr/bin/python3
deprecation_warnings = False

enable_plugins = aws_ec2
private_key_file = ~/.ssh/id_ed25519
@reenaqureshi ➜ /workspaces/terraform_machine (main) $
```



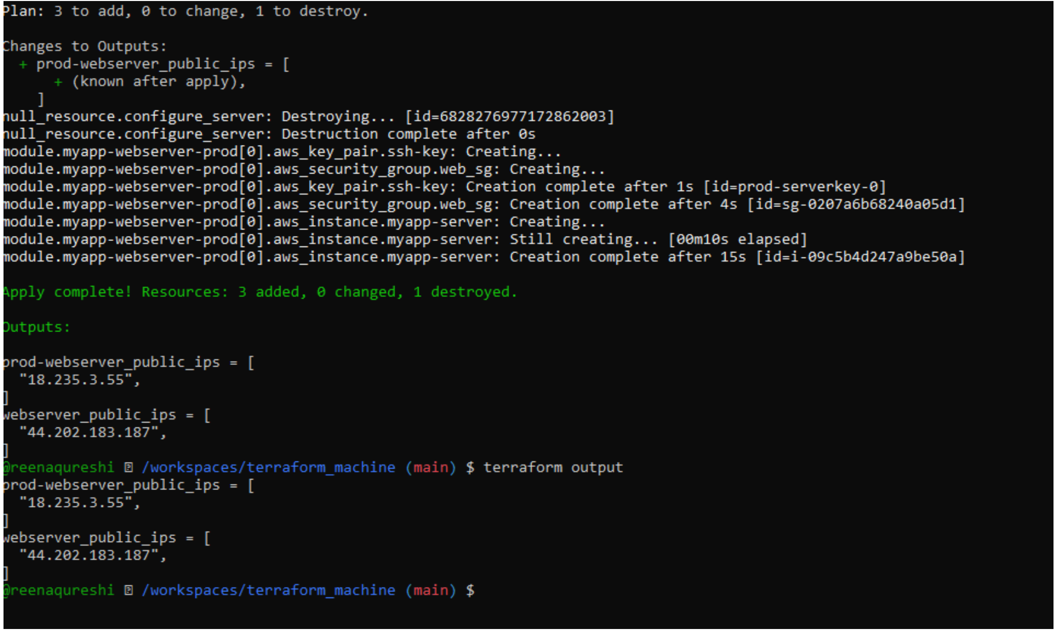
```
  GNU nano 7.2
---
plugin: aws_ec2
regions:
  - me-central-1
```



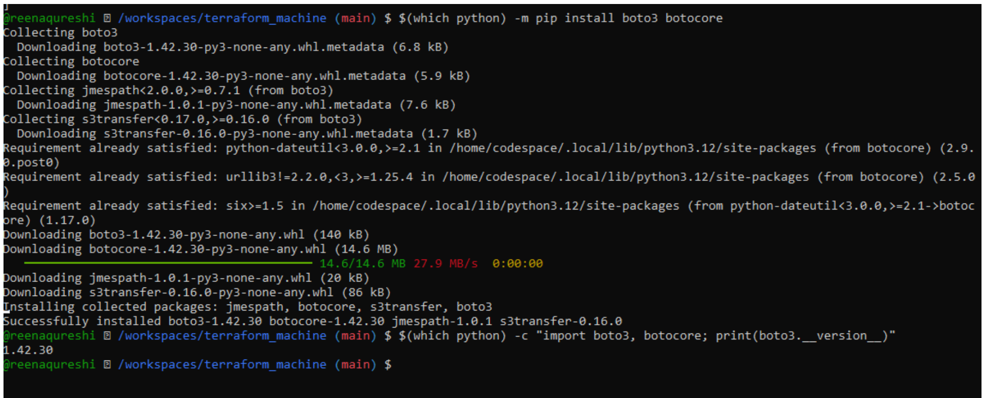
```
module "myapp-webserver" {
  source = "./modules/webserver"
  env_prefix = var.env_prefix
  instance_type = var. instance_type
  availability_zone = var.availability_zone
  public_key = var.public_key
  my_ip = local.my_ip
  vpc_id = aws_vpc.myapp_vpc.id
  subnet_id = module.myapp-subnet.subnet.id

  # Loop count
  count              = 1
  # Use count.index to differentiate instances
  instance_suffix    = count.index
}
module "myapp-webserver-prod" {
  source = "./modules/webserver"
  env_prefix = "prod"
  instance_type = "t3.nano"
  availability_zone = var.availability_zone
  public_key = var.public_key
  my_ip = local.my_ip
  vpc_id = aws_vpc.myapp_vpc.id
  subnet_id = module.myapp-subnet.subnet.id

  # Loop count
  count              = 1
  # Use count.index to differentiate instances
  instance_suffix    = count.index
}
```



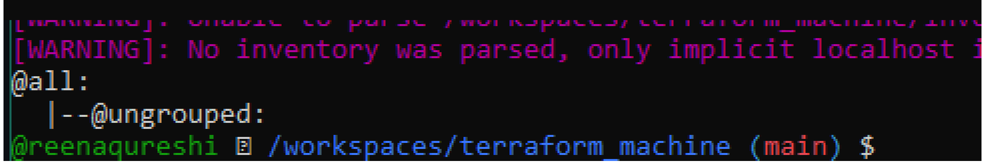
```
output "webserver_public_ips" {
  value = [for i in module.myapp-webserver : i.aws_instance.public_ip]
}

output "prod-webserver_public_ips" {
  value = [for i in module.myapp-webserver-prod : i.aws_instance.public_ip]
}
```



```
Plan: 3 to add, 0 to change, 1 to destroy.

Changes to Outputs:
  + prod-webserver_public_ips = [
      + (known after apply),
    ]
null_resource.configure_server: Destroying... [id=6828276977172862003]
null_resource.configure_server: Destruction complete after 0s
module.myapp-webserver-prod[0].aws_key_pair.ssh-key: Creating...
module.myapp-webserver-prod[0].aws_security_group.web_sg: Creating...
module.myapp-webserver-prod[0].aws_key_pair.ssh-key: Creation complete after 1s [id=prod-serverkey-0]
module.myapp-webserver-prod[0].aws_security_group.web_sg: Creation complete after 4s [id=sg-0207a6b68240a05d1]
module.myapp-webserver-prod[0].aws_instance.myapp-server: Creating...
module.myapp-webserver-prod[0].aws_instance.myapp-server: Still creating... [00m10s elapsed]
module.myapp-webserver-prod[0].aws_instance.myapp-server: Creation complete after 15s [id=i-09c5b4d247a9be50a]

Apply complete! Resources: 3 added, 0 changed, 1 destroyed.

Outputs:

prod-webserver_public_ips = [
  "18.235.3.55",
]
webserver_public_ips = [
  "44.202.183.187",
]
@reenaqureshi ➜ /workspaces/terraform_machine (main) $ terraform output
prod-webserver_public_ips = [
  "18.235.3.55",
]
webserver_public_ips = [
  "44.202.183.187",
]
@reenaqureshi ➜ /workspaces/terraform_machine (main) $
```

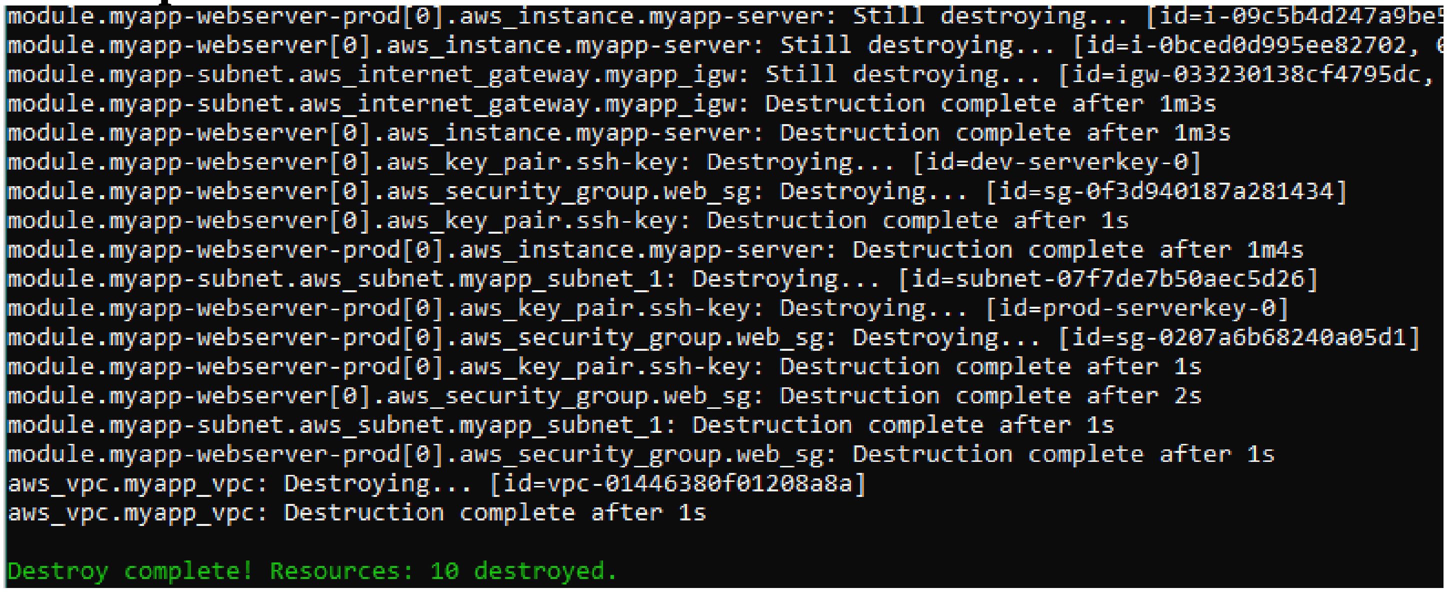


```
@reenaqureshi ➜ /workspaces/terraform_machine (main) $ $(which python) -m pip install boto3 botocore
Collecting boto3
  Downloading boto3-1.42.30-py3-none-any.whl.metadata (6.8 kB)
Collecting botocore
  Downloading botocore-1.42.30-py3-none-any.whl.metadata (5.9 kB)
Collecting jmespath<2.0.0,>=0.7.1 (from boto3)
  Downloading jmespath-1.0.1-py3-none-any.whl.metadata (7.6 kB)
Collecting s3transfer<0.17.0,>=0.16.0 (from boto3)
  Downloading s3transfer-0.16.0-py3-none-any.whl.metadata (1.7 kB)
Requirement already satisfied: python-dateutil<3.0.0,>=2.1 in /home/codespace/.local/lib/python3.12/site-packages (from botocore) (2.9.
0.post0)
Requirement already satisfied: urllib3!=2.2.0,<3,>=1.25.4 in /home/codespace/.local/lib/python3.12/site-packages (from botocore) (2.5.0
)
Requirement already satisfied: six>=1.5 in /home/codespace/.local/lib/python3.12/site-packages (from python-dateutil<3.0.0,>=2.1->botoc
ore) (1.17.0)
Downloading boto3-1.42.30-py3-none-any.whl (140 kB)
Downloading botocore-1.42.30-py3-none-any.whl (14.6 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 14.6/14.6 MB 27.9 MB/s  0:00:00
Downloading jmespath-1.0.1-py3-none-any.whl (20 kB)
Downloading s3transfer-0.16.0-py3-none-any.whl (86 kB)
Installing collected packages: jmespath, botocore, s3transfer, boto3
Successfully installed boto3-1.42.30 botocore-1.42.30 jmespath-1.0.1 s3transfer-0.16.0
@reenaqureshi ➜ /workspaces/terraform_machine (main) $ $(which python) -c "import boto3, botocore; print(boto3.__version__)"
1.42.30
@reenaqureshi ➜ /workspaces/terraform_machine (main) $
```



```
[WARNING]: Unable to parse /workspaces/terraform_machine/inv
[WARNING]: No inventory was parsed, only implicit localhost i
@all:
  |--@ungrouped:
@reenaqureshi ➜ /workspaces/terraform_machine (main) $
```

CleanUp:



```
module.myapp-webserver-prod[0].aws_instance.myapp-server: Still destroying... [id=i-09c5b4d247a9be5
module.myapp-webserver[0].aws_instance.myapp-server: Still destroying... [id=i-0bced0d995ee82702, 0
module.myapp-subnet.aws_internet_gateway.myapp_igw: Still destroying... [id=igw-033230138cf4795dc,
module.myapp-subnet.aws_internet_gateway.myapp_igw: Destruction complete after 1m3s
module.myapp-webserver[0].aws_instance.myapp-server: Destruction complete after 1m3s
module.myapp-webserver[0].aws_key_pair.ssh-key: Destroying... [id=dev-serverkey-0]
module.myapp-webserver[0].aws_security_group.web_sg: Destroying... [id=sg-0f3d940187a281434]
module.myapp-webserver[0].aws_key_pair.ssh-key: Destruction complete after 1s
module.myapp-webserver-prod[0].aws_instance.myapp-server: Destruction complete after 1m4s
module.myapp-subnet.aws_subnet.myapp_subnet_1: Destroying... [id=subnet-07f7de7b50aec5d26]
module.myapp-webserver-prod[0].aws_key_pair.ssh-key: Destroying... [id=prod-serverkey-0]
module.myapp-webserver-prod[0].aws_security_group.web_sg: Destroying... [id=sg-0207a6b68240a05d1]
module.myapp-webserver-prod[0].aws_key_pair.ssh-key: Destruction complete after 1s
module.myapp-webserver[0].aws_security_group.web_sg: Destruction complete after 2s
module.myapp-subnet.aws_subnet.myapp_subnet_1: Destruction complete after 1s
module.myapp-webserver-prod[0].aws_security_group.web_sg: Destruction complete after 1s
aws_vpc.myapp_vpc: Destroying... [id=vpc-01446380f01208a8a]
aws_vpc.myapp_vpc: Destruction complete after 1s

Destroy complete! Resources: 10 destroyed.
```