

**Name: Zuha Irfan**

**Reg no: 2023-BSE-073**

**Course: Cloud Computing Lab**

**Section: V-B**

## LAB 14

**Terraform + Ansible: Dynamic Inventory, Roles & Automated Nginx/PHP & Docker Deployment**

### Task 0- Lab Setup(Codespace & GH CLI)

Fork the repo terraform\_machine

1. Create/open Codespace on your GitHub account (from terraform\_machine repo).



2. Inside the Codespace terminal, configure GH CLI (if needed) and verify:

```
aws --version
```

```
terraform --version
```

```
● @Zuha-Irfan →/workspaces/CC-ZuhaIrfan-073 (main) $ gh auth status
github.com
  ✓ Logged in to github.com account Zuha-Irfan (GITHUB_TOKEN)
    - Active account: true
    - Git operations protocol: https
    - Token: ghu_*****
● @Zuha-Irfan →/workspaces/CC-ZuhaIrfan-073 (main) $ aws --version
aws-cli/2.32.31 Python/3.13.11 Linux/6.8.0-1030-azure exe/x86_64.ubuntu.24
● @Zuha-Irfan →/workspaces/CC-ZuhaIrfan-073 (main) $ terraform -version
Terraform v1.14.3
on linux amd64
● @Zuha-Irfan →/workspaces/CC-ZuhaIrfan-073 (main) $ ansible --version
bash: ansible: command not found
```

3. Ensure AWS CLI is configured with credentials that have permissions to create EC2, VPC, subnets, and security groups in region me-central-1:

## aws sts get-caller-identity

```
● @Zuha-Irfan → /workspaces/CC-ZuhaIrfan-073 (main) $ aws sts get-caller-identity
{
    "UserId": "345343384808",
    "Account": "345343384808",
    "Arn": "arn:aws:iam::345343384808:root"
}
```

## Task 1 – Generate ssh key and Initial Terraform apply

You will start from an existing repository and prepare Terraform variables and SSH keys.

1. Check SSH directory & generate SSH key pair if not already present:

```
ls ~/.ssh
```

```
ssh-keygen -t ed25519 -f ~/.ssh/id_ed25519 -N ""
```

```
ls -la ~/.ssh
```

```
@Zuha-Irfan → /workspaces/CC-ZuhaIrfan-073 (main) $ ls ~./ssh
Is: cannot access '/home/codespace/.ssh': No such file or directory
@Zuha-Irfan → /workspaces/CC-ZuhaIrfan-073 (main) $ ssh-keygen -t ed25519 -f ~./ssh/id_ed25519 -N
Generating public/private ed25519 key pair.
Created directory '/home/codespace/.ssh'.
Your identification has been saved in /home/codespace/.ssh/id_ed25519
Your public key has been saved in /home/codespace/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:nfZLettNaPApLzdMovoqtYLo3GV757epdnqB0831Y8 codespace@codespaces-aad63a
The key's randomart image is:
+--[ED25519 256]--+
| |
| |
| |
| . |
| . = . o |
| S + ... o |
| . .+.*=Eo. |
| ... + .+*+=... |
| ++ o..*o=O.o |
| .o o=o =B=+o |
+---[SHA256]---
```

- ## 2. Create terraform.tfvars in the repo root:

```
● @Zuha-Irfan → /workspaces/CC-ZuhaIrfan-073 (main) $ cd /workspaces/terraform_machine  
@Zuha-Irfan → /workspaces/CC-ZuhaIrfan-073/terraform_machine (main) $ touch terraform.tfvars  
@Zuha-Irfan → /workspaces/CC-ZuhaIrfan-073/terraform_machine (main) $ ls -la terraform.tfvars
```

Add the following content:

```
@Zuha-Irfan ➔ /workspaces/CC-ZuhaIrfan-073/terraform_machine (main) $ vim terraform.tfvars
@Zuha-Irfan ➔ /workspaces/CC-ZuhaIrfan-073/terraform_machine (main) $ cat terraform.tfvars
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"
```

### 3. Initialize Terraform:

terraform init

```
● @Zuha-Irfan ➔ /workspaces/CC-ZuhaIrfan-073/terraform_machine (main) $ terraform init
Initializing the backend...
Initializing modules...
- myapp-subnet in modules/subnet
- myapp-webserver in modules/webserver
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 v6.28.0...
- Installed hashicorp/aws v6.28.0 (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.
● @Zuha-Irfan ➔ /workspaces/CC-ZuhaIrfan-073/terraform_machine (main) $ terraform apply -auto-approve
```

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

terraform apply -auto-approve

```
Apply complete! Resources: 10 added, 0 changed, 0 destroyed.

Outputs:

webserver_public_ips = [
  "158.252.81.245",
  "3.28.134.219",
]
```

### 5. Check outputs:

terraform output

```
@Zuha-Irfan ➔ /workspaces/CC-ZuhaIrfan-073/terraform_machine (main) $ terraform output
  webserver_public_ips = [
    "158.252.81.245",
    "3.28.134.219",
  ]
```

## Task 2 – Static Ansible inventory with two EC2 instances

You will install Ansible (via pipx), create a static inventory, and verify connectivity.

1. Install Ansible (core) using pipx:

```
pipx install ansible-core
```

```
ansible --version
```

```
@Zuha-Irfan → /workspaces/CC-Zuhairfan-073/terraform_machine (main) $ pipx install ansible-core
  installed package ansible-core 2.20.1, installed using Python 3.12.1
  These apps are now globally available
    - ansible
    - ansible-config
    - ansible-console
    - ansible-doc
    - ansible-galaxy
    - ansible-inventory
    - ansible-playbook
    - ansible-pull
    - ansible-test
    - ansible-vault
done! ✨ ✨ ✨
@Zuha-Irfan → /workspaces/CC-Zuhairfan-073/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/modules']
  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/python)
  jinja version = 3.1.6
```

2. Obtain the two public IPs of your EC2 instances:

```
terraform output
```

```
@Zuha-Irfan → /workspaces/CC-Zuhairfan-073/terraform_machine (main) $ terraform output
  webserver_public_ips = [
    "158.252.81.245",
    "3.28.134.219",
  ]
```

3. Create Ansible inventory file hosts:

```
touch hosts
```

```
ls -la hosts
```

```
@Zuha-Irfan → /workspaces/CC-Zuhairfan-073/terraform_machine (main) $ touch hosts
@Zuha-Irfan → /workspaces/CC-Zuhairfan-073/terraform_machine (main) $ ls -la hosts
-rw-rw-rw- 1 codespace codespace 0 Jan  8 19:26 hosts
```

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

```
@Zuha-Irfan → /workspaces/CC-Zuhairfan-073/terraform_machine (main) $ vim hosts
@Zuha-Irfan → /workspaces/CC-Zuhairfan-073/terraform_machine (main) $ cat ./hosts
158.252.81.245 ansible_user=ec2-user ansible_ssh_private_key_file=~/ssh/id_ed25519
3.28.134.219  ansible_user=ec2-user ansible_ssh_private_key_file=~/ssh/id_ed25519
```

4. Test connectivity:

```
ansible all -i hosts -m ping
```

```
@Zuha-Irfan →/workspaces/CC-ZuhaIrfan-073/terraform_machine (main) $ . -i hosts -m ping
[ERROR]: Task failed: Failed to connect to the host via ssh: Host key verification failed.
Origin: <adhoc 'ping' task>

{'action': 'ping', 'args': {}, 'timeout': 0, 'async_val': 0, 'poll': 15}

158.252.81.245 | UNREACHABLE! => {
    "changed": false,
    "msg": "Task failed: Failed to connect to the host via ssh: Host key verification failed.",
    "unreachable": true
}
3.28.134.219 | UNREACHABLE! => {
    "changed": false,
    "msg": "Task failed: Failed to connect to the host via ssh: Host key verification failed.",
    "unreachable": true
}
```

5. If it fails due to host key checking, add this to each line in hosts:

```
ansible_ssh_common_args=' -o StrictHostKeyChecking=no'
```

```
@Zuha-Irfan →/workspaces/CC-ZuhaIrfan-073/terraform_machine (main) $ ansible_ssh_common_args=' -o StrictHostKeyChecking=no'
@Zuha-Irfan →/workspaces/CC-ZuhaIrfan-073/terraform_machine (main) $ vim hosts
@Zuha-Irfan →/workspaces/CC-ZuhaIrfan-073/terraform_machine (main) $ cat ./hosts
158.252.81.245 ansible_user=ec2-user ansible_ssh_private_key_file=~/.ssh/id_ed25519 ansible_ssh_common_args=' -o StrictHostKeyChecking=no'
3.28.134.219 ansible_user=ec2-user ansible_ssh_private_key_file=~/.ssh/id_ed25519 ansible_ssh_common_args=' -o StrictHostKeyChecking=no'
```

6. Retry:

```
ansible all -i hosts -m ping
```

```
@Zuha-Irfan →/workspaces/CC-ZuhaIrfan-073/terraform_machine (main) $ ansible all -i hosts -m ping
[WARNING]: Host '3.28.134.219' is using the discovered Python interpreter at '/usr/bin/python3.9', but future installation of another Python interpreter could cause a different interpreter to be discovered. See https://docs.ansible.com/ansible-core/2.20/reference_appendices/interpreter_discovery.html for more information.
3.28.134.219 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3.9"
    },
    "changed": false,
    "ping": "pong"
}
[WARNING]: Host '158.252.81.245' is using the discovered Python interpreter at '/usr/bin/python3.9', but future installation of another Python interpreter could cause a different interpreter to be discovered. See https://docs.ansible.com/ansible-core/2.20/reference_appendices/interpreter_discovery.html for more information.
158.252.81.245 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3.9"
    },
    "changed": false,
    "ping": "pong"
}
```

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

You will expand to 3 web servers via Terraform's count and restructure your inventory into groups.

1. Update your Terraform module for webservers in main.tf to use count = 3:

```
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          = 3
  # Use count.index to differentiate instances
  instance_suffix = count.index
}
```

## 2. Apply Terraform to get 3 instances:

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

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

Outputs:

webserver_public_ips = [
  "158.252.81.245",
  "3.28.134.219",
  "51.112.229.190",
]
```

## 3. Check outputs:

```
terraform output
```

```
@Zuha-Irfan ➔ /workspaces/cc-Zuhairfan-073/terraform_machine (main) $ terraform output
webserver_public_ips = [
  "158.252.81.245",
  "3.28.134.219",
  "51.112.229.190",
]
```

## 4. Rewrite your hosts file using group definitions:

```
@Zuha-Irfan ➔ /workspaces/cc-Zuhairfan-073/terraform_machine (main) $ vim hosts
@Zuha-Irfan ➔ /workspaces/cc-Zuhairfan-073/terraform_machine (main) $ cat ./hosts
[ec2]
158.252.81.245
3.28.134.219

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

[droplet]
51.112.229.190

[droplet:vars]
ansible_user=ec2-user
ansible_ssh_private_key_file=~/ssh/id_ed25519
ansible_ssh_common_args=' -o StrictHostKeyChecking=no'
@Zuha-Irfan ➔ /workspaces/cc-Zuhairfan-073/terraform_machine (main) $ ansible ec2 -i hosts -m ping
```

## 5. Test group connectivity:

```
ansible ec2 -i hosts -m ping
```

```
@Zuha-Irfan ➔ /workspaces/cc-Zuhairfan-073/terraform_machine (main) $ ansible ec2 -i hosts -m ping
[WARNING]: Host '3.28.134.219' is using the discovered Python interpreter at '/usr/bin/python3.9', but future installation of another Python interpreter could cause a different interpreter to be discovered. See https://docs.ansible.com/ansible-core/2.20/reference_appendices/interpreter_discovery.html for more information.
3.28.134.219 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3.9"
    },
    "changed": false,
    "ping": "pong"
}
[WARNING]: Host '158.252.81.245' is using the discovered Python interpreter at '/usr/bin/python3.9', but future installation of another Python interpreter could cause a different interpreter to be discovered. See https://docs.ansible.com/ansible-core/2.20/reference_appendices/interpreter_discovery.html for more information.
158.252.81.245 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3.9"
    },
    "changed": false,
    "ping": "pong"
}
@Zuha-Irfan ➔ /workspaces/cc-Zuhairfan-073/terraform_machine (main) $ ansible 51.112.229.190 -i hosts -m ping
```

## 6. Test single host by IP:

```
ansible <one-public-ip-from-ec2-group> -i hosts -m ping
```

```
@Zuha-Irfan ➔ /workspaces/CC-ZuhaIrfan-073/terraform_machine (main) $ ansible 158.252.81.245 -i hosts -m ping
[WARNING]: Host '158.252.81.245' is using the discovered Python interpreter at '/usr/bin/python3.9', but future installation of another Python interpreter could cause a different interpreter to be discovered. See https://docs.ansible.com/ansible-core/2.20/reference_appendices/interpreter_discovery.html for more information.
158.252.81.245 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3.9"
    },
    "changed": false,
    "ping": "pong"
}
```

## 7. Test droplet group:

```
ansible droplet -i hosts -m ping
```

```
@Zuha-Irfan ➔ /workspaces/CC-ZuhaIrfan-073/terraform_machine (main) $ ansible droplet -i hosts -m ping
[WARNING]: Host '51.112.229.190' is using the discovered Python interpreter at '/usr/bin/python3.9', but future installation of another Python interpreter could cause a different interpreter to be discovered. See https://docs.ansible.com/ansible-core/2.20/reference_appendices/interpreter_discovery.html for more information.
51.112.229.190 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3.9"
    },
    "changed": false,
    "ping": "pong"
}
```

## 8. Test all hosts:

```
ansible all -i hosts -m ping
```

```
@Zuha-Irfan ➔ /workspaces/CC-ZuhaIrfan-073/terraform_machine (main) $ ansible all -i hosts -m ping
[WARNING]: Host '51.112.229.190' is using the discovered Python interpreter at '/usr/bin/python3.9', but future installation of another Python interpreter could cause a different interpreter to be discovered. See https://docs.ansible.com/ansible-core/2.20/reference_appendices/interpreter_discovery.html for more information.
51.112.229.190 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3.9"
    },
    "changed": false,
    "ping": "pong"
}
[WARNING]: Host '3.28.134.219' is using the discovered Python interpreter at '/usr/bin/python3.9', but future installation of another Python interpreter could cause a different interpreter to be discovered. See https://docs.ansible.com/ansible-core/2.20/reference_appendices/interpreter_discovery.html for more information.
3.28.134.219 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3.9"
    },
    "changed": false,
    "ping": "pong"
}
[WARNING]: Host '158.252.81.245' is using the discovered Python interpreter at '/usr/bin/python3.9', but future installation of another Python interpreter could cause a different interpreter to be discovered. See https://docs.ansible.com/ansible-core/2.20/reference_appendices/interpreter_discovery.html for more information.
158.252.81.245 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3.9"
    },
    "changed": false,
    "ping": "pong"
}
```

## Task 4 – Global ansible.cfg & first nginx playbook

You will configure a global Ansible configuration file, then create a basic playbook for nginx.

### 1. Create global Ansible configuration:

```
● @Zuha-Irfan ➔ /workspaces/CC-ZuhaIrfan-073/terraform_machine (main) $ vim ~/.ansible.cfg
● @Zuha-Irfan ➔ /workspaces/CC-ZuhaIrfan-073/terraform_machine (main) $ cat ~/.ansible.cfg
[default]
host_key_checking = False
interpreter_python = /usr/bin/python3
```

### 2. Remove ansible\_ssh\_common\_args from hosts (delete from all groups).

```

@Zuha-Irfan ➔ /workspaces/CC-Zuhairfan-073/terraform_machine (main) $ vim hosts
@Zuha-Irfan ➔ /workspaces/CC-Zuhairfan-073/terraform_machine (main) $ cat hosts
[ec2]
158.252.81.245
3.28.134.219

[ec2:vars]
ansible_user=ec2-user
> ansible_ssh_private_key_file=~/ssh/id_ed25519

[droplet]
51.112.229.190

[droplet:vars]
ansible_user=ec2-user
ansible_ssh_private_key_file=~/ssh/id_ed25519

```

### 3. Confirm connectivity:

ansible all -i hosts -m ping

```

@Zuha-Irfan ➔ /workspaces/CC-Zuhairfan-073/terraform_machine (main) $ ansible all -i hosts -m ping
[WARNING]: Host '51.112.229.190' is using the discovered Python interpreter at '/usr/bin/python3.9', but future installation of another Python interpreter could cause a different interpreter to be discovered. See https://docs.ansible.com/ansible-core/2.20/reference_appendices/interpreter_discovery.html for more information.
51.112.229.190 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3.9"
    },
    "changed": false,
    "ping": "pong"
}
[WARNING]: Host '3.28.134.219' is using the discovered Python interpreter at '/usr/bin/python3.9', but future installation of another Python interpreter could cause a different interpreter to be discovered. See https://docs.ansible.com/ansible-core/2.20/reference_appendices/interpreter_discovery.html for more information.
3.28.134.219 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3.9"
    },
    "changed": false,
    "ping": "pong"
}
[WARNING]: Host '158.252.81.245' is using the discovered Python interpreter at '/usr/bin/python3.9', but future installation of another Python interpreter could cause a different interpreter to be discovered. See https://docs.ansible.com/ansible-core/2.20/reference_appendices/interpreter_discovery.html for more information.
158.252.81.245 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3.9"
    },
    "changed": false,
    "ping": "pong"
}

```

### 4. Create my-playbook.yaml:

touch my-playbook.yaml

ls -la my-playbook.yaml

```

● @Zuha-Irfan ➔ /workspaces/CC-Zuhairfan-073/terraform_machine (main) $ touch my-playbook.yaml
● @Zuha-Irfan ➔ /workspaces/CC-Zuhairfan-073/terraform_machine (main) $ ls -la my-playbook.yaml
-rw-rw-rw- 1 codespace codespace 0 Jan  8 20:09 my-playbook.yaml

```

Add:

```

@Zuha-Irfan ➔ /workspaces/CC-Zuhairfan-073/terraform_machine (main) $ vim my-playbook.yaml
@Zuha-Irfan ➔ /workspaces/CC-Zuhairfan-073/terraform_machine (main) $ cat my-playbook.yaml
---
- name: Configure nginx web server
  hosts: ec2
  become: true
  tasks:
    - name: install nginx and update cache
      yum:
        name: nginx
        state: present
        update_cache: yes

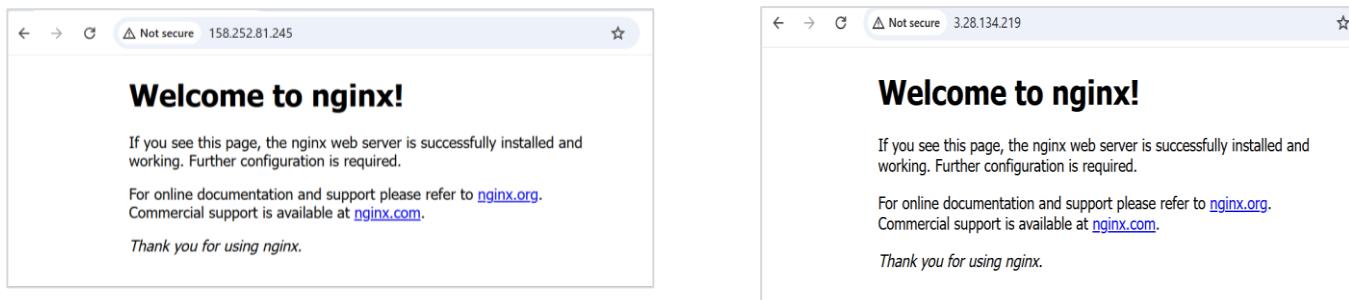
    - name: start nginx server
      service:
        name: nginx
        state: started

```

5. Run the playbook on [ec2] group:

```
ansible-playbook -i hosts my-playbook.yaml
```

6. Verify nginx default page on [ec2] servers by visiting <http://<public-ip-ec2>>.



## 7. Change target to droplet:

Edit my-playbook.yaml to:

hosts: droplet

```
@Zuha-Irfan → /workspaces/cc-ZuhaIrfan-073/terraform_machine (main) $ vim my-playbook.yaml
@Zuha-Irfan → /workspaces/cc-ZuhaIrfan-073/terraform_machine (main) $ cat ./my-playbook.yaml
---
- name: Configure nginx web server
  hosts: droplet
  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
```

### 8. Re-run:

```
ansible-playbook -i hosts my-playbook.yaml
```

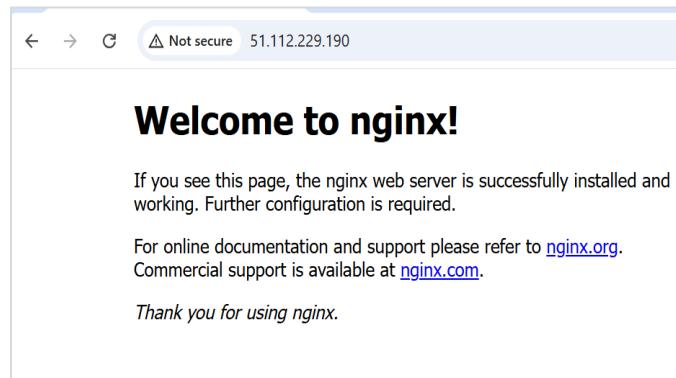
```
@Zuha-Irfan ➔ /workspaces/CC-ZuhaIrfan-073/terraform_machine (main) $ ansible-playbook -i hosts my-playbook.yaml
PLAY [Configure nginx web server] ****
TASK [Gathering Facts] ****
[WARNING]: Host '51.112.229.190' is using the discovered Python interpreter at '/usr/bin/python3.9', but future installation of another Python interpreter could cause a different interpreter to be discovered. See https://docs.ansible.com/ansible-core/2.20/reference_appendices/interpreter_discovery.html for more information.
ok: [51.112.229.190]

TASK [install nginx and update cache] ****
changed: [51.112.229.190]

TASK [start nginx server] ****
changed: [51.112.229.190]

PLAY RECAP ****
51.112.229.190 : ok=3    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

## 9. Verify nginx default page on droplet:



## Task 5 – Single nginx target group & HTTPS prerequisites

You will prepare project-level Ansible configuration and adjust nginx installation scope.

### 1. Create project-level ansible.cfg in repo root:

```
touch ansible.cfg
```

```
ls -la ansible.cfg
```

Add:

```
● @Zuha-Irfan ➔ /workspaces/CC-ZuhaIrfan-073/terraform_machine (main) $ touch ansible.cfg
● @Zuha-Irfan ➔ /workspaces/CC-ZuhaIrfan-073/terraform_machine (main) $ ls -la ansible.cfg
-rw-rw-rw- 1 codespace codespace 0 Jan  8 20:26 ansible.cfg
● @Zuha-Irfan ➔ /workspaces/CC-ZuhaIrfan-073/terraform_machine (main) $ vim ansible.cfg
```

```
[defaults]
```

```
host_key_checking=False
```

```
interpreter_python = /usr/bin/python3
```

```
● @Zuha-Irfan ➔ /workspaces/CC-ZuhaIrfan-073/terraform_machine (main) $ vim ansible.cfg
● @Zuha-Irfan ➔ /workspaces/CC-ZuhaIrfan-073/terraform_machine (main) $ cat ansible.cfg
[defaults]
host_key_checking=False
interpreter_python = /usr/bin/python3
```

### 2. Switch Terraform EC2 count back to 1:

In main.tf:

count = 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
}
```

### 3. Apply:

terraform apply -auto-approve

```
Apply complete! Resources: 0 added, 0 changed, 6 destroyed.

Outputs:

webserver_public_ips = [
  "158.252.81.245",
]
```

### 4. Outputs:

terraform output

```
@Zuha-Irfan ➔ /workspaces/CC-ZuhaIrfan-073/terraform_machine (main) $ terraform output
webserver_public_ips = [
  "158.252.81.245",
]
```

### 5. Adjust hosts:

```
@Zuha-Irfan ➔ /workspaces/CC-ZuhaIrfan-073/terraform_machine (main) $ vim hosts
@Zuha-Irfan ➔ /workspaces/CC-ZuhaIrfan-073/terraform_machine (main) $ cat hosts
[nginx]
158.252.81.245

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

### 6. Update my-playbook.yaml:

```
@Zuha-Irfan ➔ /workspaces/CC-ZuhaIrfan-073/terraform_machine (main) $ vim my-playbook.yaml
@Zuha-Irfan ➔ /workspaces/CC-ZuhaIrfan-073/terraform_machine (main) $ cat ./my-playbook.yaml
---
- name: Configure nginx web server
  hosts: nginx
  become: true
  tasks:
    - name: install nginx and update cache
      yum:
        name: nginx
        state: present
        update_cache: yes

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

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

@Zuha-Irfan ➔ /workspaces/CC-ZuhaIrfan-073/terraform_machine (main) $ ansible-playbook -i hosts my-playbook.yaml
```

### 7. Run the playbook:

ansible-playbook -i hosts my-playbook.yaml

```
[WARNING]: Ansible is being run in a world writable directory (/workspaces/terraform_machine), ignoring it as an ansible.cfg source. For more information see https://docs.ansible.com/ansible-devel/reference_appendices/config.html#cfg-in-world-writable-dir
PLAY [Configure nginx web server] ****
TASK [Gathering Facts] ****
[WARNING]: Host '158.252.81.245' is using the discovered Python interpreter at '/usr/bin/python2.9', but future installation of another Python interpreter could cause a different interpreter to be discovered. See https://docs.ansible.com/ansible-core/2.20/reference_appendices/interpreter_discovery.html for more information.
ok: [158.252.81.245]

TASK [install nginx and update cache] ****
ok: [158.252.81.245]

TASK [install openssl] ****
ok: [158.252.81.245]

TASK [start nginx server] ****
changed: [158.252.81.245]

PLAY RECAP ****
158.252.81.245 : ok=4    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

- Verify nginx default page at <http://<public-ip>>.



## Task 6 - Ansible-managed SSL certificates

Extend your playbook to generate self-signed SSL certificates using dynamic public IP.

- Append this play after the nginx play in my-playbook.yaml:

```
gitub-irfan ~/workspaces/CC-zuhairfan-073/terraform_machine (main) % vim my-playbook.yaml
gitub-irfan ~/workspaces/CC-zuhairfan-073/terraform_machine (main) % cat ./my-playbook.yaml
---
- name: Configure nginx web server
  hosts: nginx
  become: true
  tasks:
    - name: install nginx and update cache
      yum:
        name: nginx
        state: present
        update_cache: yes

    - 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:
            file: /etc/ssl/private
            path: /etc/ssl/private
            state: directory
            mode: '0700'

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

        - name: Get IMDSv2 token
          uri:
            url: "http://169.254.169.254/latest/api/token"
            method: PUT
            headers:
              X-aws-ec2-metadata-token-ttl-seconds: "3600"
            return_content: yes
          register: imdsv2_token

        - name: Get current public IP
          uri:
            url: "http://169.254.169.254/latest/meta-data/public-ipv4"
            headers:
              X-aws-ec2-metadata-token: "{{ imdsv2_token.content }}"
            return_content: yes
          register: public_ip

        - name: Show current public IP
          debug:
            msg: "Public IP: {{ public_ip.content }}"
```

## 2. Run:

```
ansible-playbook -i hosts my-playbook.yaml
```

```
[gitlab-irfan] ~ % /opt/awx/awx/awx/ansible/venv/bin/ansible-playbook -i hosts my-playbook.yml
[WARNING]: Ansible is being run in a world-writable directory (/opt/awx/awx/ansible/venv), ignoring it as an ansible.cfg source. For more information see https://docs.ansible.com/ansible-devel/reference_appendices/config.html#Ansible-in-world-writable-dir

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

[TASK [Gathering Facts] *****
[WARNING]: Host '[158.252.81.245]' is using the discovered Python interpreter at '/usr/bin/python3.9', but future installation of another Python interpreter could cause a different interpreter to be discovered. See https://docs.ansible.com/ansible-core/2.10/reference_appendices/interpreter_discovery.html for more information.
OK: [158.252.81.245]

[TASK [Install nginx and update cache] *****
OK: [158.252.81.245]

[TASK [Install openssl] *****
OK: [158.252.81.245]

[TASK [start nginx server] *****
OK: [158.252.81.245]

PLAY [Configure SSL certificates] *****

[TASK [Gathering Facts] *****
OK: [158.252.81.245]

[TASK [Create SSL private directory] *****
Changed: [158.252.81.245]

[TASK [Create SSL certs directory] *****
Changed: [158.252.81.245]

[TASK [Get PDSv2 token] *****
OK: [158.252.81.245]

[TASK [Get current public IP] *****
OK: [158.252.81.245]

[TASK [Show current public IP] *****
OK: [158.252.81.245] => {
    "msg": "Public IP: 158.252.81.245"
}

[TASK [Generate self-signed SSL certificate] *****
Changed: [158.252.81.245]

PLAY RECAP *****
158.252.81.245 : ok=11 changed=4 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0
```

### 3. SSH and verify:

```
ssh ec2-user@<public-ip> -i ~/.ssh/id_ed25519
```

```
sudo cat /etc/ssl/certs/selfsigned.crt
```

```
sudo cat /etc/ssl/private/selfsigned.key
```

exit

```
[ZuhaIr-fan → /workspaces/CC-ZuhaIr-fan-073/terraform_machine (main) $ ssh ec2-user@158.252.81.245 -i ~/ssh/id_ed25519
#_
###_ Amazon Linux 2023
\###_|
\###_|
\#_ _ https://aws.amazon.com/linux/amazon-linux-2023
\#_ V\_`->
\_/
\_/
\_m/
Last login: Thu Jan  8 20:52:43 2026 from 4.240.39.195
[ec2-user@ip-10-10-5-195 ~]$ sudo cat /etc/ssl/certs/selfsigned.crt
-----BEGIN CERTIFICATE-----
MIIDQTCACaimgAwIBAgIIdLBnJ3b9j66bfhxNzB1vyyLz2sowDQYJKoZIhvcNAQEL
BQAwGEXHbJUGA1EAvnwOMTU4l_j1IMi4MMS4yNDUWhhCNMjYwNTA4MjA1MjQzhNC
MjwNTA4MjA1MjQzJwJAZMRcwFQDVQODDA4XNTguMjUyLjgxLj0INTCCASIwDQYJ
KoZIhvcNAQEBBQADggEPADCCAQoCggEBAN8C7fxSgNHKmOikOuyAoaxfjkhCsksx
7KW7RwM6QenopV18XkpLGFB9YevSmqoSBy/gD1z1hf5bRFHsOEkoY4Wb10
1uoiv9nyS0AYNb1ofOgqAvBf18DXgIKG8Iu5eHwhXeEcrasYunc1CZCsrxkhfMl
39CTCrV3xIFGpJ30XTCINSpqjy08PUiUT1/PQ4Mkfsk3xT1zkMw8uvvf0n+
Jfbkw2wV019agids18JYavH+8dsLn1wJoV1PxwMl8J/hL02NnV2Sh4SVU2sc
1FugJ00dkspw3KUbwvmpMa5wtvutHNg5ZDjTyiPgSeIyVuqyKnhI49ECAwEAaaOB
gDB#B0GAlUdgQWBTOEjV30MAT6Ur-vrFomRm1MFgjzafbBgVHMSMEDGAwgBTO
EjV30MAT6Ur-vrFomRm1MFgjzafbBgVHMSMEDGAwgBTO
CwVR0PBAQDQAglwMBMGAl1uJdQMMAGGCCsGAQ9BwMBAQGCSg51b3dQEBcwlUA
A41BAQCu965xPvNa/UpcdMp5ySlic8d0DOMskhM05kxhN2LTX1sqaHlr/ehHMRH
esaZ3hpv5AhSu0zU0LzVKT1whMh0LEpjnQz1ah10tmwvBsgt0d4RxKD+Hx+Txk+
BNAf05McQwa/s+JEs5kb/0/-rWfaADnuvYqETBytd0s5Kh1R20RgyZgrAhwE
y8Wv9A7KkplnhyB0hSwar56k7vWMyL9Z29v0/bE4zouwfNdeTjwJxd1u1sws
KeI1X2glthZ/WY1fjdgeBpkXjuMuN00011oeJ6svTBxJwdt3dMGrWsN58TD
Jgbps0xfz8vSiKAcfkxBkH/ub
-----END CERTIFICATE-----
```

## Task 7 - PHP front-end deployment with templates

Deploy a PHP application and Nginx configuration using Ansible copy and template.

## 1. Create directories and files:

```
mkdir -p files templates
```

```
touch files/index.php
```

```
touch templates/nginx.conf.j2
```

```
ls -R
```

```
elementtree.cpython-313-x86_64-linux-gnu.so      grp.cpython-313-x86_64-linux-gnu.so
 hashlib.cpython-313-x86_64-linux-gnu.so          math.cpython-313-x86_64-linux-gnu.so
 heapq.cpython-313-x86_64-linux-gnu.so           mmap.cpython-313-x86_64-linux-gnu.so
 json.cpython-313-x86_64-linux-gnu.so            pyexpat.cpython-313-x86_64-linux-gnu.so
 lzma.cpython-313-x86_64-linux-gnu.so            readline.cpython-313-x86_64-linux-gnu.so
 _md5.cpython-313-x86_64-linux-gnu.so            resource.cpython-313-x86_64-linux-gnu.so
 _multibytescodec.cpython-313-x86_64-linux-gnu.so select.cpython-313-x86_64-linux-gnu.so
 _multiprocessing.cpython-313-x86_64-linux-gnu.so termios.cpython-313-x86_64-linux-gnu.so
 _opcode.cpython-313-x86_64-linux-gnu.so          unicodedata.cpython-313-x86_64-linux-gnu.so
 _pickle.cpython-313-x86_64-linux-gnu.so          zlib.cpython-313-x86_64-linux-gnu.so

./aws/dist/prompt_toolkit-3.0.51.dist-info:
INSTALLER METADATA RECORD WHEEL licenses top_level.txt

./aws/dist/prompt_toolkit-3.0.51.dist-info/licenses:
AUTHORS.rst LICENSE

./aws/dist/wheel-0.45.1.dist-info:
INSTALLER LICENSE.txt METADATA RECORD REQUESTED WHEEL direct_url.json entry_points.txt

./files:
index.php

./modules:
subnet webserver

./modules/subnet:
main.tf outputs.tf variables.tf

./modules/webserver:
main.tf outputs.tf variables.tf

./templates:
nginx.conf.j2
```

## 2. Fill files/index.php with the following PHP metadata page:

```
<?php
$host = "Nginx Front End Web Server";
$private_ip = "192.168.1.10";
$public_ip = "192.168.1.10";
$public_dns = "public_dns";
$instance_id = "i-000000000000000000";
$deployed_date = "2023-01-01T12:00:00Z";
$status = "Active and Running";
$managed_by = "Terraform + Ansible";

echo <?php
$host = "Nginx Front End Web Server";
$private_ip = "192.168.1.10";
$public_ip = "192.168.1.10";
$public_dns = "public_dns";
$instance_id = "i-000000000000000000";
$deployed_date = "2023-01-01T12:00:00Z";
$status = "Active and Running";
$managed_by = "Terraform + Ansible";
<?>
<html>
<head>
<style>
h1 {
    color: #fff;
    text-shadow: 2px 2px 4px rgba(0,0,0,0.3);
}
.info {
    margin: 15px 0;
    padding: 10px;
    background: rgba(255,255,255,0.2);
    border-radius: 5px;
}
.label {
    font-weight: bold;
    color: #ffd700;
}
.info a {
    color: white; /* same as other values */
    text-decoration: none; /* remove underline */
    font-weight: normal;
}
.info a:hover {
    text-decoration: underline; /* optional: underline on hover */
}
</style>
</head>
<body>
<div class="container">
<h1>$host</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> $status Active and Running</div>
<div class="info"><span class="label">Managed By:</span> Terraform + Ansible</div>
</div>
</body>
</html>
```

## 3. Fill templates/nginx.conf.j2 with the following Nginx configuration template:

```

@zuhal-irfan -> /workspaces/cc-zuhairfan-073/terraform_machine (main) $ vim templates/nginx.conf.j2
@zuhal-irfan -> /workspaces/cc-zuhairfan-073/terraform_machine (main) $ cat templates/nginx.conf.j2
user nginx;
worker_processes auto;
error_log /var/log/nginx/error.log notice;
pid /run/nginx.pid;

events {
    worker_connections 1024;
}

http {
    log_format main '$remote_addr - $remote_user [$time_local] "$request"
$status $body_bytes_sent "$http_referer"
$http_user_agent" "$http_x_forwarded_for"';

    access_log /var/log/nginx/access.log main;

    sendfile        on;
    tcp_nopush     on;
    keepalive_timeout 65;
    types_hash_max_size 4096;

    include         /etc/nginx/mime.types;
    default_type   application/octet-stream;

    upstream backend_servers {
        server 198.252.94.241:80;
        server 198.252.94.242:80 backup;
    }

    server {
        listen 443 ssl;
        server_name {{ server_public_ip }};
        ssl_certificate /etc/ssl/certs/selfsigned.crt;
        ssl_certificate_key /etc/ssl/private/selfsigned.key;
        location / {
            root /usr/share/nginx/html;
            index index.php index.html index.htm;
            # proxy_pass http://198.252.94.241:80;
            # proxy_pass http://backend_servers;

            # ● This block is necessary for PHP website
            location ~ \.php {
                include fastcgi_params;
            }
        }
    }
}

```

#### 4. Add this play to my-playbook.yaml:

```

@zuhal-irfan -> /workspaces/cc-zuhairfan-073/terraform_machine (main) $ vim files/index.php
@zuhal-irfan -> /workspaces/cc-zuhairfan-073/terraform_machine (main) $ cat files/index.php
<?php
// Get hostname
$hostname = gethostname();

// Deployment date
$deployed_date = date("Y-m-d H:i:s");

// Metadata base URL
$metadata_base = "http://169.254.169.254/latest/";

// Function to get IDv2 token
function getidv2Token() {
    $ch = curl_init("http://169.254.169.254/latest/api/token");
    curl_setopt_array($ch, [
        CURLOPT_RETURNTRANSFER => true,
        CURLOPT_CUSTOMREQUEST => "PUT",
        CURLOPT_HTTPHEADER => [
            "X-AWS-ECS-Metadata-Token-TTL-Seconds: 21600"
        ],
        CURLOPT_TIMEOUT => 2
    ]);
    $token = curl_exec($ch);
    curl_close($ch);

    return $token ?: null;
}

// Function to fetch metadata using token
function getMetadata($path, $token) {
    $url = "http://169.254.169.254/latest/meta-data/" . $path;

    $ch = curl_init($url);
    curl_setopt_array($ch, [
        CURLOPT_RETURNTRANSFER => true,
        CURLOPT_HTTPHEADER => [
            "X-AWS-ECS-Metadata-Token: $token"
        ],
        CURLOPT_TIMEOUT => 2
    ]);
    $value = curl_exec($ch);
    curl_close($ch);

    return $value ?: "N/A";
}

// Fetch token
$token = getidv2Token();

// Fetch metadata only if token is available
$instance_id = $token ? getMetadata("instance-id", $token) : "N/A";
$private_ip = $token ? getMetadata("local-ip4", $token) : "N/A";
$public_ip = $token ? getMetadata("public-ip4", $token) : "N/A";
$hostname_dmc = $token ? getMetadata("multi-interface", $token) : "N/A";

```

#### 5. Run the playbook:

ansible-playbook -i hosts my-playbook.yaml

```
@Zuha-Irfan → /workspaces/Cc-Zuhair-fan-073/terraform_machine (main) $ cat my-playbook.yaml
- 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
        ...-> 1.1.1.1

TASK [Copy website files] *****
changed: [192.168.81.245]

TASK [Copy nginx.conf template] *****
[ERROR]: Task failed: 'server_public_ip' is undefined

Task failed.
Origin: /workspaces/terraform_machine/my-playbook.yaml:93:7

91     mode: '0644'
92
93   - name: Copy nginx.conf template
      ^ column ?

<<< caused by >>>

'server_public_ip' is undefined
Origin: /workspaces/terraform_machine/templates/nginx.conf.j2

fatal: [192.168.81.245]: FAILED! => {"changed": false, "msg": "Task failed: 'server_public_ip' is undefined"})

PLAY RECAP *****
192.168.81.245 : ok=54 changed=2 unreachable=0 failed=1 skipped=0 rescued=0 ignored=0

@Zuha-Irfan → /workspaces/Cc-Zuhair-fan-073/terraform_machine (main) $ vim my-playbook.yaml
@Zuha-Irfan → /workspaces/Cc-Zuhair-fan-073/terraform_machine (main) $ cat my-playbook.yaml
...
@Zuha-Irfan → /workspaces/Cc-Zuhair-fan-073/terraform_machine (main) $ ansible-playbook -i hosts my-playbook.yaml
[WARNING]: Ansible is being run in a virtualenv directory (/var/www/terraform), ignoring it as an ansible_dir source. For more information see https://docs.ansible.com/ansible/latest/reference_appendices/config/ansible_in_venv_or_virtual_dir.html

PLAY [Configure nginx web server] *****
  &gt; [Getting facts] *****
ok: [192.168.81.245]  * Using the discovered Python interpreter at '/usr/bin/python3.9', but future installation of another Python interpreter could cause a different interpreter to be discovered. See https://docs.ansible.com/ansible-core/2.12/reference_appendices/interpreter_discovery.html for more information.

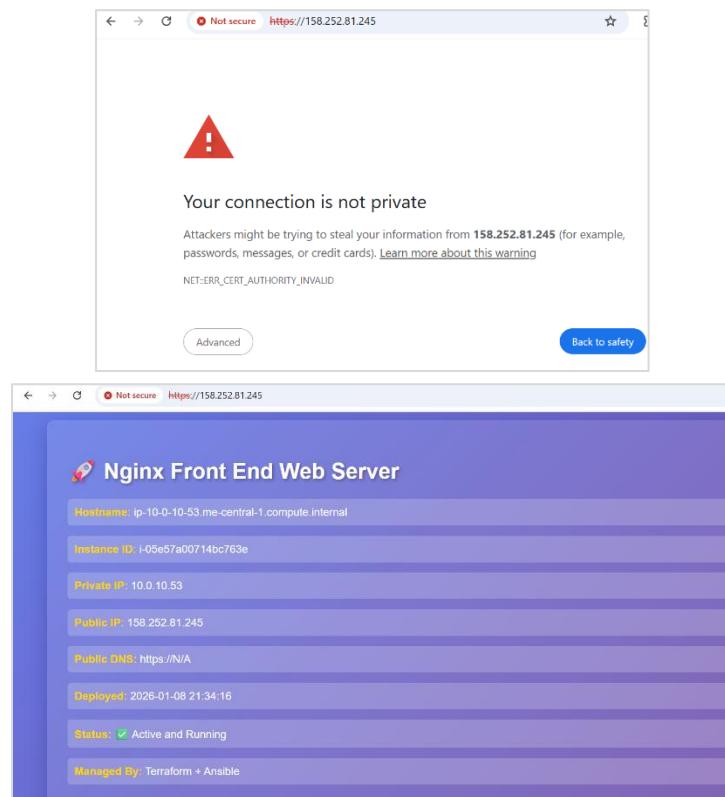
  &gt; [Configure SSL certificates] *****
ok: [192.168.81.245]
  &gt; [Create SSL certificate directory]
ok: [192.168.81.245]
  &gt; [Create SSL certs directory]
ok: [192.168.81.245]
  &gt; [Get public keys]
ok: [192.168.81.245]
  &gt; [Get current public IP]
ok: [192.168.81.245]
  &gt; [Parse current public IP]
ok: [192.168.81.245]  * [192.168.81.245] => "192.168.81.245"
  &gt; [Save public IP as fact]
ok: [192.168.81.245]
  &gt; [Generate self-signed SSL certificates]
ok: [192.168.81.245]
  &gt; [Deploy nginx website and configuration files] *****
  &gt; [Getting facts] *****
ok: [192.168.81.245]
  &gt; [Install php-fpm and php-curl]
ok: [192.168.81.245]
  &gt; [Copy website files]
ok: [192.168.81.245]
  &gt; [Copy nginx.conf template]
changed: [192.168.81.245]

PLAY RECAP *****
192.168.81.245 : ok=54 changed=2 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0
```

6. Visit <https://<public-ip>>:

Accept SSL warning.

Verify the PHP page content.



## Task 8 – Docker & Docker Compose provisioning via Ansible

Deploy Docker and Docker Compose on a new EC2 instance.

1. Destroy old infrastructure:

`terraform destroy -auto-approve`

```
@ZuhairIrfan ~/workspaces/CC-ZuhairIrfan-073/terraform_machine (main) $ terraform destroy -auto-approve
module.myapp-webserver[0].aws_instance.myapp-server: Destroying... [id=i-05e57a00714bc763e]
module.myapp-subnet.aws_default_route_table.main_rt: Destruction complete after 0s
module.myapp-subnet.aws_internet_gateway.myapp_igw: Destroying... [id=igw-0c3425f766430d8dc]
module.myapp-webserver[0].aws_instance.myapp-server: Still destroying... [id=i-05e57a00714bc763e, 00m10s elapsed]
module.myapp-subnet.aws_internet_gateway.myapp_igw: Still destroying... [id=igw-0c3425f766430d8dc, 00m10s elapsed]
module.myapp-webserver[0].aws_instance.myapp-server: Still destroying... [id=i-05e57a00714bc763e, 00m20s elapsed]
module.myapp-subnet.aws_internet_gateway.myapp_igw: Still destroying... [id=igw-0c3425f766430d8dc, 00m20s elapsed]
module.myapp-webserver[0].aws_instance.myapp-server: Still destroying... [id=i-05e57a00714bc763e, 00m30s elapsed]
module.myapp-subnet.aws_internet_gateway.myapp_igw: Still destroying... [id=igw-0c3425f766430d8dc, 00m30s elapsed]
module.myapp-subnet.aws_internet_gateway.myapp_igw: Destruction complete after 37s
module.myapp-webserver[0].aws_instance.myapp-server: Still destroying... [id=i-05e57a00714bc763e, 00m40s elapsed]
module.myapp-webserver[0].aws_instance.myapp-server: Destruction complete after 40s
module.myapp-subnet.aws_subnet.myapp_subnet_1: Destroying... [id=subnet-074649248c91c4526]
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-074fc3e1f709e084e]
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 1s
aws_vpc.myapp_vpc: Destroying... [id=vpc-0b6418f8143b0ffaf]
aws_vpc.myapp_vpc: Destruction complete after 1s

Destroy complete! Resources: 7 destroyed.
```

2. Recreate fresh infrastructure (1 instance):

terraform apply -auto-approve

terraform output

```
@Zuhairfan → /workspaces/CC-Zuhairfan-073/terraform_machine (main) $ terraform apply -auto-approve
module.myapp-webservice[0].aws_key_pair.ssh-key: Creating...
aws_vpc.myapp_vpc: Creating...
module.myapp-webservice[0].aws_key_pair.ssh-key: Creation complete after 1s [id=dev-serverkey-0]
aws_vpc.myapp_vpc: Creation complete after 2s [id=vpc-051f2f19b879b12a8]
module.myapp-subnet.aws_internet_gateway.myapp_igw: Creating...
module.myapp-subnet.aws_subnet.myapp_subnet_1: Creating...
module.myapp-webservice[0].aws_security_group.web_sg: Creating...
module.myapp-subnet.aws_internet_gateway.myapp_igw: Creation complete after 0s [id=igw-07a66f23c5b03f7a8]
module.myapp-subnet.aws_default_route_table.main_rt: Creating...
module.myapp-subnet.aws_default_route_table.main_rt: Creation complete after 1s [id=rtb-0de8e7c2f6e2a21ed]
module.myapp-webservice[0].aws_security_group.web_sg: Creation complete after 3s [id=sg-0f742f005673765ad]
module.myapp-subnet.aws_subnet.myapp_subnet_1: Still creating... [0m10s elapsed]
module.myapp-subnet.aws_subnet.myapp_subnet_1: Creation complete after 11s [id=subnet-05e79b898e70ece6e]
module.myapp-webservice[0].aws_instance.myapp-server: Creating...
module.myapp-webservice[0].aws_instance.myapp-server: Still creating... [0m10s elapsed]
module.myapp-webservice[0].aws_instance.myapp-server: Creation complete after 12s [id=i-08faa252eebf7bde9]

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

Outputs:

webservice_public_ips = [
  "40.172.187.29",
]
@Zuhairfan → /workspaces/CC-Zuhairfan-073/terraform_machine (main) $ terraform output
webservice_public_ips = [
  "40.172.187.29",
]
```

### 3. Update hosts:

```
@Zuhairfan → /workspaces/CC-Zuhairfan-073/terraform_machine (main) $ vim hosts
@Zuhairfan → /workspaces/CC-Zuhairfan-073/terraform_machine (main) $ cat ./hosts
[docker_servers]
40.172.187.29

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

### 4. Replace my-playbook.yaml content with:

```
@Zuhairfan → /workspaces/CC-Zuhairfan-073/terraform_machine (main) $ vim my-playbook.yaml
@Zuhairfan → /workspaces/CC-Zuhairfan-073/terraform_machine (main) $ cat my-playbook.yaml
- name: Configure Docker
  hosts: all
  become: true
  tasks:
    - name: install docker and update cache
      yum:
        name: docker
        state: present
        update_cache: yes

    - name: Install Docker Compose
      hosts: all
      become: true
      gather_facts: true
      tasks:
        - name: create docker cli-plugins directory
          file:
            path: /usr/local/lib/docker/cli-plugins
            state: directory
            mode: '0755'

        - name: install docker-compose
          get_url:
            url: https://github.com/docker/compose/releases/latest/download/docker-compose-linux-{{ lookup('pipe', 'uname -m') }}
            dest: /usr/local/lib/docker/cli-plugins/docker-compose
            mode: >@

    - name: View architecture of the system
      debug:
        msg: "System architecture of {{ inventory_hostname }} is {{ ansible_facts['architecture'] }}"

    - name: Alternate method to view architecture of the system
      debug:
        msg: "System architecture of {{inventory_hostname}} is {{ lookup('pipe', 'uname -m') }}"

    - name: restart docker service
      service:
        name: docker
        state: restarted
```

### 5. Run the play:

ansible-playbook -i hosts my-playbook.yaml

## 6. Verify Docker:

```
ssh ec2-user@<public-ip> -i ~/.ssh/id_ed25519
```

```
sudo docker ps
```

exit

```
@zuha-Irfan →/workspaces/CC-Zuhairfan-073/terraform_machine (main) $ ssh ec2-user@40.172.187.29 -i ~/.ssh/id_ed2551  
9  
,      #  
~\_ #####_      Amazon Linux 2023  
~~ \#####\  
~~ \|###|  
~~   \#/      https://aws.amazon.com/linux/amazon-linux-2023  
~~   \~'-->  
~~   /  
~~ .-. /  
     /  
     /  
     /m/  
Last login: Thu Jan  8 21:54:05 2026 from 4.240.39.195  
[ec2-user@ip-10-0-10-189 ~]$ sudo docker ps  
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS          NAMES  
[ec2-user@ip-10-0-10-189 ~]$ exit  
logout  
Connection to 40.172.187.29 closed.
```

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

Run containers for Gitea + Postgres and open port 3000 in the security group.

1. Extend my-playbook.yaml by appending:

```
git clone https://github.com/Zuhairfan-07/terraform_machine.git
cd terraform_machine
terraform init
terraform apply
```

- name: Adding user to docker group  
hosts: all  
become: true  
vars\_files:  
 - project-vars.yaml  
tasks:  
 - name: add user to docker group  
 user:  
 name: "{{ normal\_user }}"  
 groups: docker  
 append: yes  
  
 - name: reconnect to apply group changes  
 meta: reset\_connection  
  
 - name: verify docker access  
 command: docker ps  
 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

## 2. Create project-vars.yaml:

```
touch project-vars.yaml
```

Add:

```
normal_user: ec2-user
```

```
docker_compose_file_location: <location-of-file>
```

```
@Zuha-Irfan ➔ /workspaces/CC-Zuhairfan-073/terraform_machine (main) $ touch project-vars.yaml
@Zuha-Irfan ➔ /workspaces/CC-Zuhairfan-073/terraform_machine (main) $ vim project-vars.yaml
@Zuha-Irfan ➔ /workspaces/CC-Zuhairfan-073/terraform_machine (main) $ cat ./project-vars.yaml
normal_user: ec2-user
docker_compose_file_location: /workspaces/terraform_machine
```

## 3. Add the deploy containers play:

```
@Zuha-Irfan ➔ /workspaces/CC-Zuhairfan-073/terraform_machine (main) $ cat my-playbook.yaml
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.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.stderr"
      when: compose_file.exists
```

## 4. Create compose.yaml in repo root:

```
@Zuha-Irfan ➔ /workspaces/CC-Zuhairfan-073/terraform_machine (main) $ touch compose.yaml
@Zuha-Irfan ➔ /workspaces/CC-Zuhairfan-073/terraform_machine (main) $ vim compose.yaml
@Zuha-Irfan ➔ /workspaces/CC-Zuhairfan-073/terraform_machine (main) $ cat compose.yaml
services:
  gitea:
    image: gitea/gitea:latest
    container_name: gitea
    environment:
      - DB_TYPE=postgres
      - DB_HOST=db:5432
      - DB_NAME=gitea
      - DB_USER=gitea
      - DB_PASSWORD=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
```

5. Run playbook and visit the public-ip on your browser but webpage will not be shown due to permission issue:

ansible-playbook -i hosts my-playbook.yaml

```
@Zuhai-Irfan ~ /workspaces/CC-ZuhaiIrfan-073/terraform_machine (main) $ ansible-playbook -i hosts my-playbook.yaml
TASK [Gathering Facts] *****
OK: [40.172.187.29]
TASK [create docker cli-plugins directory] *****
OK: [40.172.187.29]
TASK [Install docker-compose] *****
OK: [40.172.187.29]
TASK [View architecture of the system] *****
OK: [40.172.187.29] => {
    "msg": "System architecture of 40.172.187.29 is x86_64"
}
TASK [Alternative method to view architecture of the system] *****
OK: [40.172.187.29] => {
    "msg": "System architecture of 40.172.187.29 is x86_64"
}
TASK [restart docker service] *****
changed: [40.172.187.29]
PLAY [Adding user to docker group] *****
TASK [Gathering Facts] *****
OK: [40.172.187.29]
TASK [Add user to docker group] *****
changed: [40.172.187.29]
TASK [Reconnect to apply group changes] *****
TASK [Verify docker access] *****
OK: [40.172.187.29]
TASK [Display docker ps output] *****
OK: [40.172.187.29] => {
    "docker_ps.stdout": "CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES"
}
TASK [Fail if docker is not accessible] *****
skipping: [40.172.187.29]
PLAY [Deploy Docker containers] *****
TASK [Gathering Facts] *****
OK: [40.172.187.29]
TASK [Check if docker-compose file exists] *****
OK: [40.172.187.29]
TASK [Copy docker-compose file] *****
changed: [40.172.187.29]
TASK [Deploy containers using docker-compose] *****
OK: [40.172.187.29]
PLAY RECAP
40.172.187.29 : ok=16 changed=3 unreachable=0 failed=0 skipped=1 rescued=0 ignored=0
```

6. Update security group in modules/webserver/main.tf to add port 3000:

```
● @Zuhai-Irfan ~ /workspaces/CC-ZuhaiIrfan-073/terraform_machine (main) $ vim modules/webserver/main.tf
● @Zuhai-Irfan ~ /workspaces/CC-ZuhaiIrfan-073/terraform_machine (main) $ cat modules/webserver/main.tf
resource "aws_security_group" "web_sg" {
  vpc_id      = var.vpc_id
  name        = "${var.env_prefix}-web-sg-${var.instance_suffix}"
  description = "Security group for web server allowing HTTP, HTTPS and SSH"
  ingress {
    from_port   = 3000
    to_port     = 3000
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }
  ingress {
    from_port   = 22
    to_port     = 22
    protocol    = "tcp"
    cidr_blocks = [var.my_ip]
}
```

7. Apply:

terraform apply -auto-approve

```
@Zuhai-Irfan ~ /workspaces/CC-ZuhaiIrfan-073/terraform_machine (main) $ terraform apply -auto-approve
[...]
+ 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-0f742f005673765ad]
module.myapp_webserver[0].aws_security_group.web_sg: Modifications complete after 0s [id=sg-0f742f005673765ad]

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

Outputs:

webserver_public_ips = [
  "40.172.187.29",
]
```

## 8. Access Gitea:

Visit <http://<public-ip>:3000>.

The left screenshot shows the 'Initial Configuration' page. It displays a form for setting up a database. The 'Database Type' is set to 'PostgreSQL'. The 'Host' field contains 'db:5432', 'Username' is 'gitea', 'Password' is '.....', 'Database Name' is 'gitea', 'SSL' is set to 'Disable', and the 'Schema' field is empty with a note 'Leave blank for database default ("public")'. The right screenshot shows the Gitea dashboard at 'http://40.172.187.29:3000'. The top navigation bar includes 'Issues', 'Pull Requests', 'Milestones', and 'Explore'. A sidebar on the right has tabs for 'Repository' and 'Organiz'. The main content area shows a message 'No Activity' and a note: 'You are currently not following any repositories or users, so there is no content to display. You can explore repositories or users of interest from the links below.' Below this are links to 'Explore repositories' and 'Explore users'. A message at the bottom right says 'There are no repositories yet'.

## Task 10 – Automating Ansible with Terraform (null\_resource)

Have Terraform trigger Ansible automatically after EC2 creation.

### 1. Add null\_resource to main.tf:

```
@zuha-Irfan ~/workspaces/cc-Zuhairfan-073/terraform_machine (main) $ cat main.tf
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
    }
}
```

### 2. Destroy and recreate infrastructure:

terraform init

terraform destroy -auto-approve

terraform apply -auto-approve

```
@Zuhairfan ~ /workspaces/CC-Zuhairfan-073/terraform_machine (main) $ terraform destroy -auto-approve
- webserver_public_ips = [
  "40.172.187.29",
]
] -> null
module.myapp_subnet.aws_default_route_table.main_rt: Destroying... [id=rtb-0de8e7c2f6e2a21ed]
module.myapp_subnet.aws_default_route_table.main_rt: Destruction complete after 0s
module.myapp_webserver[0].aws_instance.myapp_server: Destroying... [id=i-08faa252eebf7bde9]
module.myapp_webserver[0].aws_instance.myapp_server: Still destroying... [id=i-08faa252eebf7bde9, 0m0s elapsed]
module.myapp_subnet.aws_internet_gateway.myapp_igw: Still destroying... [id=igw-07a66f23c5b03f7a8, 0m1s elapsed]
module.myapp_webserver[0].aws_instance.myapp_server: Still destroying... [id=i-08faa252eebf7bde9, 0m2s elapsed]
module.myapp_subnet.aws_internet_gateway.myapp_igw: Still destroying... [id=igw-07a66f23c5b03f7a8, 0m2s elapsed]
module.myapp_subnet.aws_internet_gateway.myapp_igw: Destruction complete after 27s
module.myapp_webserver[0].aws_instance.myapp_server: Still destroying... [id=i-08faa252eebf7bde9, 0m30s elapsed]
module.myapp_webserver[0].aws_instance.myapp_server: Destruction complete after 30s
module.myapp_webserver[0].aws_key_pair.ssh_key: Destroying... [id=dev-serverkey-0]
module.myapp_subnet.myapp_subnet_1: Destroying... [idsubnet-05e79b89e70ec6e]
module.myapp_webserver[0].aws_security_group.web_sg: Destroying... [idssg-0f742f005673765ad]
module.myapp_webserver[0].aws_key_pair.ssh_key: Destruction complete after 0s
module.myapp_subnet.aws_internet_gateway.myapp_igw: Destruction complete after 0s
module.myapp_webserver[0].aws_security_group.web_sg: Destruction complete after 1s
aws_vpc.myapp_vpc: Destroying... [id=vpc-051f2f19b079b12a8]
aws_vpc.myapp_vpc: Destruction complete after 0s

Destroy complete! Resources: 7 destroyed.

@Zuhairfan ~ /workspaces/CC-Zuhairfan-073/terraform_machine (main) $ terraform apply -auto-approve

Error: local-exec provisioner error

with null_resource.configure_server,
on main.tf line 44, in resource "null_resource" "configure_server":
44:   provisioner "local-exec" {

Error running command 'ansible-playbook -i 51.112.47.121, \
--private-key "./.ssh/id_ed25519" --user ec2-user \
my-playbook.yaml
': exit status 127. Output: [WARNING]: Ansible is being run in a world writable directory
(/workspaces/terraform_machine), ignoring it as an ansible.cfg source. For more information see
https://docs.ansible.com/ansible-devel/reference_appendices/config.html#cfg-in-world-writable-dir
could not be foundbook:
/bin/sh: 2: --private-key: not found
: not found3: my-playbook.yaml

@Zuhairfan ~ /workspaces/CC-Zuhairfan-073/terraform_machine (main) $ vim main.tf
```

3. If Ansible fails due to readiness, add a wait play at the top of my-playbook.yaml:

```
@Zuhairfan ~ /workspaces/CC-Zuhairfan-073/terraform_machine (main) $ cat ./my-playbook.yaml
- name: Wait for some time to ensure system readiness
  hosts: all
  tasks:
    - name: Wait 300 seconds for port 22 to become open and contain "OpenSSH"
      wait_for:
        port: 22
        host: "{{ inventory_hostname }}"
        delay: 10
        timeout: 300
      delegate_to: localhost

resource "null_resource" "configure_server" {
  depends_on = [module.myapp_webserver]

  triggers = {
    webserver_public_ips_for_ansible = join("", [
      for i in module.myapp_webserver : i.aws_instance.public_ip
    ])
  }

  provisioner "local-exec" {
    command = "ANSIBLE_HOST_KEY_CHECKING=False ansible-playbook -i ${self.triggers.webserver_public_ips_for_ansible}, --private-key ${var.private_key} --user ec2-user ${path.module}/my-playbook.yaml"
  }
}
```

4. Destroy and apply again:

terraform destroy -auto-approve

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

```
module.myapp-subnet.aws_subnet.myapp_subnet_1: Destruction complete after 1s
module.myapp-webserver[0].aws_security_group.web_sg: Destruction complete after 1s
aws_vpc.myapp_vpc: Destroying... [id=vpc-0b616feb867897322]
aws_vpc.myapp_vpc: Destruction complete after 1s

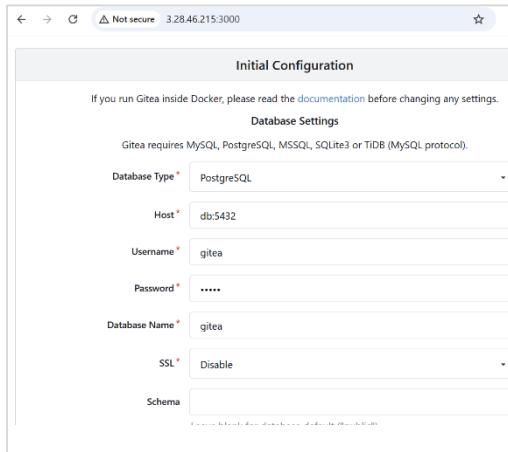
Destroy complete! Resources: 8 destroyed.
@Zuha-Irfan ➔ /workspaces/CC-ZuhaIrfan-073/terraform_machine (main) $ terraform apply -auto-approve
```

```
Apply complete! Resources: 8 added, 0 changed, 0 destroyed.
```

**Outputs:**

```
webserver_public_ips = [
    "3.28.46.215",
]
```

5. Verify Gitea / Nginx application is reachable at the appropriate URL.



## Task 11 – Dynamic inventory with aws\_ec2 plugin

Let Ansible discover EC2 instances dynamically via the aws\_ec2 inventory plugin.

1. Update ansible.cfg:

```
● @Zuha-Irfan ➔ /workspaces/CC-ZuhaIrfan-073/terraform_machine (main) $ vim ansible.cfg
● @Zuha-Irfan ➔ /workspaces/CC-ZuhaIrfan-073/terraform_machine (main) $ cat ./ansible.cfg
[defaults]
host_key_checking=False
interpreter_python = /usr/bin/python3

[defaults]
host_key_checking=False
interpreter_python = /usr/bin/python3
deprecation_warnings = False

enable_plugins = aws_ec2
private_key_file = ~/.ssh/id_ed25519
```

2. Create inventory\_aws\_ec2.yaml:

```
touch inventory_aws_ec2.yaml
```

```
ls -la inventory_aws_ec2.yaml
```

```
@Zuha-Irfan ➔ /workspaces/CC-ZuhaIrfan-073/terraform_machine (main) $ touch inventory_aws_ec2.yaml
@Zuha-Irfan ➔ /workspaces/CC-ZuhaIrfan-073/terraform_machine (main) $ ls -la inventory_aws_ec2.yaml
-rw-rw-rw- 1 codespace codespace 0 Jan  8 22:59 inventory_aws_ec2.yaml
@Zuha-Irfan ➔ /workspaces/CC-ZuhaIrfan-073/terraform_machine (main) $ vim inventory aws ec2.vam
```

Add initial content:

```
@Zuha-Irfan ➔ /workspaces/CC-ZuhaIrfan-073/terraform_machine (main) $ vim inventory_aws_ec2.yaml
@Zuha-Irfan ➔ /workspaces/CC-ZuhaIrfan-073/terraform_machine (main) $ cat ./inventory_aws_ec2.yaml
---
plugin: aws_ec2
regions:
- me-central-1
```

3. Ensure Terraform code includes both dev and prod webservers in main.tf:

```
@Zuha-Irfan ➔ /workspaces/CC-ZuhaIrfan-073/terraform_machine (main) $ cat ./main.tf
}

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
}

resource "null_resource" "configure_server" {
  depends_on = [module.myapp-webserver]

  triggers = {
```

4. Add outputs in outputs.tf:

```
@Zuha-Irfan ➔ /workspaces/CC-ZuhaIrfan-073/terraform_machine (main) $ vim outputs.tf
@Zuha-Irfan ➔ /workspaces/CC-ZuhaIrfan-073/terraform_machine (main) $ cat ./outputs.tf
output "webserver_public_ips" {
  # value = [for i in module.myapp-webserver : i.aws_instance.public_ip]
}

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]
}
```

5. Rebuild infra:

```
terraform init
```

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

## terraform output

```
@Zuha-Irfan ~/workspaces/CC-ZuhaIrfan-073/terraform_machine (main) $ terraform apply -auto-approve
aws_vpc.myapp_vpc: Modifying... [id=vpc-0fcce9ae3aefac3a1]
module.myapp-webserver-prod[0].aws_key_pair.ssh-key: Creation complete after 1s [id=prod-serverkey-0]
aws_vpc.myapp_vpc: Still modifying... [id=vpc-0fcce9ae3aefac3a1, 00m10s elapsed]
aws_vpc.myapp_vpc: Modifications complete after 12s [id=vpc-0fcce9ae3aefac3a1]
module.myapp-webserver-prod[0].aws_security_group.web_sg: Creating...
module.myapp-webserver-prod[0].aws_security_group.web_sg: Creation complete after 2s [id=sg-0dd75b6dd32bd7be5]
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 13s [id=i-07b57e378b0548659]

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

Outputs:

prod-webserver_public_ips = [
  "3.28.40.241",
]
webserver_public_ips = [
  "3.28.46.215",
]
@Zuha-Irfan ~/workspaces/CC-ZuhaIrfan-073/terraform_machine (main) $ terraform output
prod-webserver_public_ips = [
  "3.28.40.241",
]
webserver_public_ips = [
  "3.28.46.215",
]
```

## 6. Install boto3 and botocore:

\$(which python) -m pip install boto3 botocore

- Verify the version

\$(which python) -c "import boto3, botocore; print(boto3.\_\_version\_\_)"

```
● @Zuha-Irfan ~/workspaces/CC-ZuhaIrfan-073/terraform_machine (main) $ $(which python) -m pip install boto3 botocore
Collecting boto3
  Downloading boto3-1.42.24-py3-none-any.whl.metadata (6.8 kB)
Collecting botocore
  Downloading botocore-1.42.24-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>botocore) (1.17.0)
Downloading boto3-1.42.24-py3-none-any.whl (140 kB)
Downloading botocore-1.42.24-py3-none-any.whl (14.6 MB)
  14.6/14.6 MB 28.3 MB/s  0:00:00
Downloaded 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 boto3-1.42.24 botocore-1.42.24 jmespath-1.0.1 s3transfer-0.16.0
● @Zuha-Irfan ~/workspaces/CC-ZuhaIrfan-073/terraform_machine (main) $ $(which python) -c "import boto3, botocore; print(boto3.__version__)"
1.42.24
```

## 7. Check inventory graph:

ansible-inventory -i inventory\_aws\_ec2.yaml --graph

```
@Zuha-Irfan ~/workspaces/CC-ZuhaIrfan-073/terraform_machine (main) $ /usr/local/py-utils/venvs/ansible-core/bin/python -m pip install boto3 botocore
@Zuha-Irfan ~/workspaces/CC-ZuhaIrfan-073/terraform_machine (main) $ ansible-inventory -i inventory_aws_ec2.yaml --graph
[WARNING]: Ansible is being run in a world writable directory (/workspaces/terraform_machine), ignoring it as an ansible.cfg source. For more information see https://docs.ansible.com/ansible-devel/reference_appendices/config.html#cfg-in-world-writable-dir
[WARNING]: Deprecation warnings can be disabled by setting 'deprecation_warnings=False' in ansible.cfg.
[DEPRECATION WARNING]: Importing 'to_text' from 'ansible.module_utils._text' is deprecated. This feature will be removed from ansible-core version 2.24. Use ansible.module_utils.common.text.converters instead.
[DEPRECATION WARNING]: Importing 'to_native' from 'ansible.module_utils._text' is deprecated. This feature will be removed from ansible-core version 2.24. Use ansible.module_utils.common.text.converters instead.
[DEPRECATION WARNING]: Passing 'disable_lookups' to 'template' is deprecated. This feature will be removed from ansible-core version 2.23.
@all:
  |--@ungrouped:
  |  |--@aws_ec2:
  |     |--ec2-3-28-46-215.me-central-1.compute.amazonaws.com
  |     |--ec2-3-28-40-241.me-central-1.compute.amazonaws.com
```

## Task 12 – Filtering EC2 instances by tags & instance type

Augment the inventory plugin to group by tags and instance type, then limit plays to specific groups.

1. Modify inventory\_aws\_ec2.yaml to add tag-based grouping:

```
@Zuha-Irfan ➔ /workspaces/CC-ZuhaIrfan-073/terraform_machine (main) $ vim inventory_aws_ec2.yaml
@Zuha-Irfan ➔ /workspaces/CC-ZuhaIrfan-073/terraform_machine (main) $ cat ./inventory_aws_ec2.yaml

---
plugin: aws_ec2
regions:
- me-central-1

keyed_groups:
- key: tags
  prefix: tag
  separator: "_"
```

Check graph:

```
ansible-inventory -i inventory_aws_ec2.yaml --graph
```

```
separator: "_"
@Zuha-Irfan ➔ /workspaces/CC-ZuhaIrfan-073/terraform_machine (main) $ ansible-inventory -i inventory_aws_ec2.yaml --graph
[WARNING]: Ansible is being run in a world writable directory ('/workspaces/terraform_machine'), ignoring it as an ansible.cfg source. For more information see https://docs.ansible.com/ansible-devel/reference_appendices/config.html#cfg-in-world-writable-dir
[WARNING]: Deprecation warnings can be disabled by setting `deprecation_warnings=False` in ansible.cfg.
[DEPRECATION WARNING]: Importing 'to_text' from 'ansible.module_utils._text' is deprecated. This feature will be removed from ansible-core version 2.24. Use ansible.module_utils.common.text.converters instead.
[DEPRECATION WARNING]: Importing 'to_native' from 'ansible.module_utils._text' is deprecated. This feature will be removed from ansible-core version 2.24. Use ansible.module_utils.common.text.converters instead.
[DEPRECATION WARNING]: Passing "disable_lookups" to "template" is deprecated. This feature will be removed from ansible-core version 2.23.
@all:
|-@ungrouped:
|-@aws_ec2:
| |--ec2-3-28-46-215.me-central-1.compute.amazonaws.com
| |--ec2-3-28-40-241.me-central-1.compute.amazonaws.com
|-@tag_Name_dev_ec2_instance_0:
| |--ec2-3-28-46-215.me-central-1.compute.amazonaws.com
|-@tag_Name_prod_ec2_instance_0:
| |--ec2-3-28-40-241.me-central-1.compute.amazonaws.com
```

2. Extend to group by instance type as well:

```
@Zuha-Irfan ➔ /workspaces/CC-ZuhaIrfan-073/terraform_machine (main) $ vim inventory_aws_ec2.yaml
@Zuha-Irfan ➔ /workspaces/CC-ZuhaIrfan-073/terraform_machine (main) $ cat ./inventory_aws_ec2.yaml
---
plugin: aws_ec2
regions:
- me-central-1

keyed_groups:
- key: tags
  prefix: tag
  separator: "_"

- key: instance_type
  prefix: instance_type
  separator: "_"

○ @Zuha-Irfan ➔ /workspaces/CC-ZuhaIrfan-073/terraform_machine (main) $
```

Check graph again:

```
ansible-inventory -i inventory_aws_ec2.yaml --graph
```

```
[Zuhair-Irfan:~/workspaces/CC-ZuhairIrfan-073/terraform_machine (main)]$ ansible-inventory -i inventory_aws_ec2.yaml --graph
[WARNING]: Ansible is being run in a world writable directory (/workspaces/terraform_machine), ignoring it as an ansible.cfg so
For more information see https://docs.ansible.com/ansible/devel/reference_appendices/config.html#cfg-in-world-writable-dir
[WARNING]: Deprecation warnings can be disabled by setting 'deprecation_warnings=False' in ansible.cfg.
[DEPRECATION WARNING]: Importing 'to_text' from 'ansible.module_utils.common.text.converters' is deprecated. This feature will be removed from a
-core version 2.24. Use ansible.module_utils.common.text.converters instead.
[DEPRECATION WARNING]: Importing 'to_native' from 'ansible.module_utils.common.text' is deprecated. This feature will be removed from
the core version 2.24. Use ansible.module_utils.common.text.converters instead.
[DEPRECATION WARNING]: Passing 'disable_lookups' to 'template' is deprecated. This feature will be removed from ansible-core ve
2.23.
@all:
  |--@ungrouped:
  |   |--@aws_ec2:
  |       |   |--ec2-3-28-46-215.me-central-1.compute.amazonaws.com
  |       |   |--ec2-3-28-40-241.me-central-1.compute.amazonaws.com
  |       |--@tag_Name_dev_ec2_instance_0:
  |           |   |--ec2-3-28-46-215.me-central-1.compute.amazonaws.com
  |       |--@instance_type_t3_micro:
  |           |   |--ec2-3-28-46-215.me-central-1.compute.amazonaws.com
  |       |--@tag_Name_prod_ec2_instance_0:
  |           |   |--ec2-3-28-40-241.me-central-1.compute.amazonaws.com
  |       |--@instance_type_t3_nano:
  |           |   |--ec2-3-28-40-241.me-central-1.compute.amazonaws.com
```

### 3. Prepare my-playbook.yaml for nginx+SSL+PHP on hosts: all:

#### 4. Run on all instances:

```
ansible-playbook -i inventory_aws_ec2.yaml my-playbook.yaml
```

```
[Zuhair-Irfan:~/workspaces/CC-ZuhairIrfan-073/terraform_machine (main)]$ ansible-playbook -i inventory_aws_ec2.yaml my-playbook.yaml
PLAY [Deploy Nginx website and configuration files] ****
  TASK [Gathering Facts] ****
    ok: [ec2-3-28-46-215.me-central-1.compute.amazonaws.com]
    ok: [ec2-3-28-40-241.me-central-1.compute.amazonaws.com]

  TASK [Generate self-signed SSL certificate] ****
    ok: [ec2-3-28-40-241.me-central-1.compute.amazonaws.com]
    ok: [ec2-3-28-46-215.me-central-1.compute.amazonaws.com]

  PLAY RECAP ****
  ec2-3-28-40-241.me-central-1.compute.amazonaws.com : ok=35  changed=2  unreachable=0  failed=0  skipped=2  rescued=0  ignore=0
  ec2-3-28-46-215.me-central-1.compute.amazonaws.com : ok=35  changed=2  unreachable=0  failed=0  skipped=2  rescued=0  ignore=0
```

#### 5. Run only on dev instances:

```
ansible-playbook -i inventory_aws_ec2.yaml -l tag_Name_dev_* my-playbook.yaml
```

```
[zuha-irfan ~] [workspaces/CC-zuhazfan-073/terraform_machine (main)] $ ansible-playbook -i inventory_aws_ec2.yaml -l tag_Name_dev_* my-playbook.yaml
1
TASK [Show current public IP] ****
ok: [ec2-3-28-46-215.me-central-1.compute.amazonaws.com] => {
    "msg": "Public IP: 3.28.46.215"
}

TASK [Save public IP as fact] ****
ok: [ec2-3-28-46-215.me-central-1.compute.amazonaws.com]

TASK [Generate self-signed SSL certificate] ****
ok: [ec2-3-28-46-215.me-central-1.compute.amazonaws.com]

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

TASK [Gathering Facts] ****
ok: [ec2-3-28-46-215.me-central-1.compute.amazonaws.com]

TASK [Install php-fpm and php-curl] ****
ok: [ec2-3-28-46-215.me-central-1.compute.amazonaws.com]

TASK [Copy website files] ****
ok: [ec2-3-28-46-215.me-central-1.compute.amazonaws.com]

TASK [Copy nginx.conf template] ****
ok: [ec2-3-28-46-215.me-central-1.compute.amazonaws.com]

TASK [Restart nginx] ****
changed: [ec2-3-28-46-215.me-central-1.compute.amazonaws.com]

TASK [Start and enable php-fpm] ****
ok: [ec2-3-28-46-215.me-central-1.compute.amazonaws.com]

PLAY RECAP ****
ec2-3-28-46-215.me-central-1.compute.amazonaws.com : ok=35  changed=2  unreachable=0  failed=0  skipped=2  rescued=0  ignored=0
```

## 6. Run only on prod instances:

```
ansible-playbook -i inventory_aws_ec2.yaml -l tag_Name_prod_* my-playbook.yaml
```

```
****
ok: [ec2-3-28-46-215.me-central-1.compute.amazonaws.com]
****
ok: [ec2-3-28-40-241.me-central-1.compute.amazonaws.com]

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

TASK [Gathering Facts] ****
ok: [ec2-3-28-40-241.me-central-1.compute.amazonaws.com]

TASK [Install php-fpm and php-curl] ****
ok: [ec2-3-28-40-241.me-central-1.compute.amazonaws.com]

TASK [Copy website files] ****
ok: [ec2-3-28-40-241.me-central-1.compute.amazonaws.com]

TASK [Copy nginx.conf template] ****
ok: [ec2-3-28-40-241.me-central-1.compute.amazonaws.com]

TASK [Restart nginx] ****
changed: [ec2-3-28-40-241.me-central-1.compute.amazonaws.com]

TASK [Start and enable php-fpm] ****
ok: [ec2-3-28-40-241.me-central-1.compute.amazonaws.com]

PLAY RECAP ****
****
ec2-3-28-40-241.me-central-1.compute.amazonaws.com : ok=35  changed=2  unreachable=0  failed=0  skipped=2  rescued=0  ignored=0
```

## 7. Run only on t3.micro instances:

```
ansible-playbook -i inventory_aws_ec2.yaml -l instance_type_t3_micro my-playbook.yaml
```

```
[zuha-irfan ~] [workspaces/CC-zuhazfan-073/terraform_machine (main)] $ ansible-playbook -i inventory_aws_ec2.yaml -l instance_type_t3_micro my-
playbook.yaml
TASK [Show current public IP] ****
ok: [ec2-3-28-46-215.me-central-1.compute.amazonaws.com] => {
    "msg": "Public IP: 3.28.46.215"
}

TASK [Save public IP as fact] ****
ok: [ec2-3-28-46-215.me-central-1.compute.amazonaws.com]

TASK [Generate self-signed SSL certificate] ****
ok: [ec2-3-28-46-215.me-central-1.compute.amazonaws.com]

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

TASK [Gathering Facts] ****
ok: [ec2-3-28-46-215.me-central-1.compute.amazonaws.com]

TASK [Install php-fpm and php-curl] ****
ok: [ec2-3-28-46-215.me-central-1.compute.amazonaws.com]

TASK [Copy website files] ****
ok: [ec2-3-28-46-215.me-central-1.compute.amazonaws.com]

TASK [Copy nginx.conf template] ****
ok: [ec2-3-28-46-215.me-central-1.compute.amazonaws.com]

TASK [Restart nginx] ****
changed: [ec2-3-28-46-215.me-central-1.compute.amazonaws.com]

TASK [Start and enable php-fpm] ****
ok: [ec2-3-28-46-215.me-central-1.compute.amazonaws.com]

PLAY RECAP ****
ec2-3-28-46-215.me-central-1.compute.amazonaws.com : ok=35  changed=2  unreachable=0  failed=0  skipped=2  rescued=0  ignored=0
```

## 8. Run only on t3.nano instances:

```
ansible-playbook -i inventory_aws_ec2.yaml -l instance_type_t3_nano my-playbook.yaml
```

```
@ZuhairFan ~>/workspaces/CC-ZuhairFan-073/terraform_machine (main) $ ansible-playbook -i inventory_aws_ec2.yaml -l instance_type_t3_nano my-playbook.yaml
TASK [Show current public IP] ****
ok: [ec2-3-28-40-241.me-central-1.compute.amazonaws.com] => {
    "msg": "Public IP: 3.28.40.241"
}

TASK [Save public IP as fact] ****
ok: [ec2-3-28-40-241.me-central-1.compute.amazonaws.com]

TASK [Generate self-signed SSL certificate] ****
ok: [ec2-3-28-40-241.me-central-1.compute.amazonaws.com]

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

TASK [Gathering Facts] ****
ok: [ec2-3-28-40-241.me-central-1.compute.amazonaws.com]

TASK [install php-fpm and php-curl] ****
ok: [ec2-3-28-40-241.me-central-1.compute.amazonaws.com]

TASK [Copy website files] ****
ok: [ec2-3-28-40-241.me-central-1.compute.amazonaws.com]

TASK [Copy nginx.conf template] ****
ok: [ec2-3-28-40-241.me-central-1.compute.amazonaws.com]

TASK [Restart nginx] ****
changed: [ec2-3-28-40-241.me-central-1.compute.amazonaws.com]

TASK [Start and enable php-fpm] ****
ok: [ec2-3-28-40-241.me-central-1.compute.amazonaws.com]

PLAY RECAP ****
ec2-3-28-40-241.me-central-1.compute.amazonaws.com : ok=35  changed=2  unreachable=0  failed=0  skipped=2  rescued=0
ignored=0
```

## 9. Update ansible.cfg to use inventory by default:

```
inventory = ./inventory_aws_ec2.yaml
```

```
@ZuhairFan ~>/workspaces/CC-ZuhairFan-073/terraform_machine (main) $ vim ansible.cfg
@ZuhairFan ~>/workspaces/CC-ZuhairFan-073/terraform_machine (main) $ cat ansible.cfg
[defaults]
remote_user = ec2-user
host_key_checking=False
interpreter_python = /usr/bin/python3
deprecation_warnings = False

enable_plugins = aws_ec2
private_key_file = ~/.ssh/id_ed25519

inventory = ./inventory_aws_ec2.yaml
@ZuhairFan ~>/workspaces/CC-ZuhairFan-073/terraform_machine (main) $
```

## 10. Now you can simply run:

```
ansible-playbook -l instance_type_t3_nano my-playbook.yaml
```

```
inventory = ./inventory_aws_ec2.yaml
TASK [Show current public IP] ****
ok: [ec2-3-28-40-241.me-central-1.compute.amazonaws.com] => {
    "msg": "Public IP: 3.28.40.241"
}

TASK [Save public IP as fact] ****
ok: [ec2-3-28-40-241.me-central-1.compute.amazonaws.com]

TASK [Generate self-signed SSL certificate] ****
ok: [ec2-3-28-40-241.me-central-1.compute.amazonaws.com]

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

TASK [Gathering Facts] ****
ok: [ec2-3-28-40-241.me-central-1.compute.amazonaws.com]

TASK [install php-fpm and php-curl] ****
ok: [ec2-3-28-40-241.me-central-1.compute.amazonaws.com]

TASK [Copy website files] ****
ok: [ec2-3-28-40-241.me-central-1.compute.amazonaws.com]

TASK [Copy nginx.conf template] ****
ok: [ec2-3-28-40-241.me-central-1.compute.amazonaws.com]

TASK [Restart nginx] ****
changed: [ec2-3-28-40-241.me-central-1.compute.amazonaws.com]

TASK [Start and enable php-fpm] ****
ok: [ec2-3-28-40-241.me-central-1.compute.amazonaws.com]

PLAY RECAP ****
ec2-3-28-40-241.me-central-1.compute.amazonaws.com : ok=35  changed=2  unreachable=0  failed=0  skipped=2  rescued=0
ignored=0
```

## Task 13 – Ansible roles: nginx, ssl, webapp

Reorganize your configuration into roles.

1. Update main.tf for a simple dev environment with 1 instance (as shown previously).

```
@Zuha-Irfan →/workspaces/CC-ZuhaIrfan-073/terraform_machine (main) $ cat main.tf
  default_route_table_id = aws_vpc.myapp_vpc.default_route_table_id
}

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
}
```

2. Create /ansible structure: mkdir -p ansible

```
cd ansible
```

```
mkdir inventory roles
```

```
touch ansible.cfg my-playbook.yaml
```

```
● @Zuha-Irfan →/workspaces/CC-ZuhaIrfan-073/terraform_machine (main) $ mkdir -p ansible
● @Zuha-Irfan →/workspaces/CC-ZuhaIrfan-073/terraform_machine (main) $ cd ansible
● @Zuha-Irfan →/workspaces/CC-ZuhaIrfan-073/terraform_machine/ansible (main) $ mkdir inventory roles
● @Zuha-Irfan →/workspaces/CC-ZuhaIrfan-073/terraform_machine/ansible (main) $ touch ansible.cfg my-playbook.yaml
● @Zuha-Irfan →/workspaces/CC-ZuhaIrfan-073/terraform_machine/ansible (main) $ ls -R
..
ansible.cfg  inventory  my-playbook.yaml  roles

./inventory:
./roles:
```

3. ansible/ansible.cfg:

```
● @Zuha-Irfan →/workspaces/CC-ZuhaIrfan-073/terraform_machine/ansible (main) $ vim ansible.cfg
● @Zuha-Irfan →/workspaces/CC-ZuhaIrfan-073/terraform_machine/ansible (main) $ cat ansible.cfg
[defaults]
host_key_checking=False
interpreter_python = /usr/bin/python3
```

4. ansible/inventory/hosts:

```
● @Zuha-Irfan →/workspaces/CC-ZuhaIrfan-073/terraform_machine/ansible (main) $ vim inventory/hosts
● @Zuha-Irfan →/workspaces/CC-ZuhaIrfan-073/terraform_machine/ansible (main) $ cat inventory/hosts
[nginx]
51.112.49.120

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

## 5. Create roles:

```
● @Zuha-Irfan →.../cc-ZuhaIrfan-073/terraform_machine/ansible/roles (main) $ ansible-galaxy role init nginx
- Role nginx was created successfully
● @Zuha-Irfan →.../cc-ZuhaIrfan-073/terraform_machine/ansible/roles (main) $ ansible-galaxy role init ssl
- Role ssl was created successfully
● @Zuha-Irfan →.../cc-ZuhaIrfan-073/terraform_machine/ansible/roles (main) $ ansible-galaxy role init webapp
- Role webapp was created successfully
● @Zuha-Irfan →.../cc-ZuhaIrfan-073/terraform_machine/ansible/roles (main) $ cd ..
● @Zuha-Irfan →/workspaces/CC-ZuhaIrfan-073/terraform_machine/ansible (main) $ ls -R
:
ansible.cfg inventory my-playbook.yaml roles

./inventory:
hosts

./roles:
nginx ssl webapp

./roles/nginx:
README.md defaults files handlers meta tasks templates tests vars
```

## 6. Role: nginx

- ansible/roles/nginx/handlers/main.yml:

```
● @Zuha-Irfan →/workspaces/CC-ZuhaIrfan-073/terraform_machine/ansible (main) $ vim roles/nginx/handlers/main.yml
● @Zuha-Irfan →/workspaces/CC-ZuhaIrfan-073/terraform_machine/ansible (main) $ cat roles/nginx/handlers/main.yml
#SPDX-License-Identifier: MIT-0
---
# handlers file for nginx
- name: Restart nginx
  service:
    name: nginx
    state: restarted
```

- ansible/roles/nginx/tasks/main.yml:

```
● @Zuha-Irfan →/workspaces/CC-ZuhaIrfan-073/terraform_machine/ansible (main) $ vim roles/nginx/tasks/main.yml
● @Zuha-Irfan →/workspaces/CC-ZuhaIrfan-073/terraform_machine/ansible (main) $ cat roles/nginx/tasks/main.yml
#SPDX-License-Identifier: MIT-0
---
# tasks file for nginx
- name: Install nginx
  yum:
    name: nginx
    state: present
    update_cache: yes
    notify: Restart nginx

- name: Install openssl
  yum:
    name: openssl
    state: present

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

## 7. First role-based playbook ansible/my-playbook.yaml:

```
● @Zuha-Irfan →/workspaces/CC-ZuhaIrfan-073/terraform_machine/ansible (main) $ vim my-playbook.yaml
● @Zuha-Irfan →/workspaces/CC-ZuhaIrfan-073/terraform_machine/ansible (main) $ cat my-playbook.yaml
---
- name: Deploy NGINX Web Stack with SSL and PHP
  hosts: nginx
  become: true
  roles:
    - nginx
```

Run:

chmod 755 \$(pwd)

```
ansible-playbook -i inventory/hosts my-playbook.yaml
```

```
● @Zuhairfan → /workspaces/cc-zuhairfan-073/terraform_machine/ansible (main) $ : chmod 755 $(pwd)
● @Zuhairfan → /workspaces/cc-zuhairfan-073/terraform_machine/ansible (main) $ ansible-playbook -i inventory/hosts my-playbook.yaml

PLAY [Deploy NGINX Web Stack with SSL and PHP] ****
TASK [Gathering Facts] ****
ok: [51.112.49.120]

TASK [nginx : Install nginx] ****
ok: [51.112.49.120]

TASK [nginx : Install openssl] ****
ok: [51.112.49.120]

TASK [nginx : Start and enable nginx] ****
ok: [51.112.49.120]

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

## 8. Role: ssl

- ansible/roles/ssl/defaults/main.yml:

```
● @Zuhairfan → /workspaces/cc-zuhairfan-073/terraform_machine/ansible (main) $ vim roles/ssl/defaults/main.yml
● @Zuhairfan → /workspaces/cc-zuhairfan-073/terraform_machine/ansible (main) $ cat roles/ssl/defaults/main.yml
#SPDX-License-Identifier: MIT-0
---
# defaults file for ssl
imdsv2_token_ttl: "3600"
ssl_days_valid: 365
```

## 9. Role: webapp

- ansible/roles/webapp/defaults/main.yml:

```
● @Zuhairfan → /workspaces/CC-Zuhairfan-073/terraform_machine/ansible (main) $ vim roles/webapp/handlers/main.yml
● @Zuhairfan → /workspaces/CC-Zuhairfan-073/terraform_machine/ansible (main) $ cat roles/webapp/handlers/main.yml
#SPDX-License-Identifier: MIT-0
---
# handlers file for webapp
- name: Restart nginx
  service:
    name: nginx
    state: restarted

- name: Restart php-fpm
  service:
    name: php-fpm
    state: restarted
```

```
● @Zuhairfan → /workspaces/CC-Zuhairfan-073/terraform_machine/ansible (main) $ vim roles/ssl/tasks/main.yml
● @Zuhairfan → /workspaces/CC-Zuhairfan-073/terraform_machine/ansible (main) $ cat roles/ssl/tasks/main.yml
#SPDX-License-Identifier: MIT-0
---
# tasks file for ssl
- 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: GET
    headers:
      X-amz-ec2-metadata-token-ttl-seconds: "{{ imdsv2_token_ttl }}"
    return_content: yes
  register: imds_token

- name: Get public IP
  uri:
    url: http://169.254.169.254/latest/meta-data/public-ipv4
    method: GET
    headers:
      X-amz-ec2-metadata-token: "{{ imds_token.content }}"
    return_content: yes
  register: public_ip

- name: Save public IP as fact
  set_fact:
    server_public_ip: "{{ public_ip.content }}"

- name: Generate self-signed certificate
  command: >
    openssl req -x509 -nodes -days {{ ssl_days_valid }}
    -newkey rsa:2048
    -keyout /etc/ssl/certs/selfSigned.key
    -out /etc/ssl/certs/selfSigned.crt
    -subj "/CN={{ server_public_ip }}"
    -addext "subjectAltName:DNS:{{ server_public_ip }}"
  args:
    creates: /etc/ssl/certs/selfSigned.crt
```

- ansible/roles/webapp/files/index.php – PHP metadata page.

```
@Zuha-Irfan → /workspaces/CC-ZuhaIrfan-073/terraform_machine/ansible (main) $ vim roles/webapp/defaults/main.yml
@Zuha-Irfan → /workspaces/CC-ZuhaIrfan-073/terraform_machine/ansible (main) $ cat roles/webapp/defaults/main.yml
#SPDX-License-Identifier: MIT-0
---
# defaults file for webapp
nginx_user: nginx
nginx_worker_processes: auto
nginx_worker_connections: 1024
nginx_error_log_level: notice

# Webapp settings
web_root: /usr/share/nginx/html
web_index_file: index.php
```

- ansible/roles/webapp/tasks/main.yml:

```
@Zuha-Irfan → /workspaces/CC-ZuhaIrfan-073/terraform_machine/ansible (main) $ cat roles/webapp/files/index.php
font-family: Arial, sans-serif;
margin: 50px;
background: linear-gradient(135deg, #667eea 0%, #768a92 100%);
color: white;
}
.container {
background: rgba(255, 255, 255, 0.1);
padding: 30px;
border-radius: 10px;
box-shadow: 0 8px 32px rgba(31, 38, 135, 0.37);
}
h1 {
color: #fff;
text-shadow: 2px 2px 4px rgba(0,0,0,0.3);
}
.info {
margin: 15px 0;
padding: 10px;
background: rgba(255,255,255,0.2);
border-radius: 8px;
}
.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 */
}

```

```
</head>
<body>
<div class="container">
<h2>nginx Front End Web Server </h2>
<div class="info"><span class="label">External IP:</span><span>$hostname </span></div>
<div class="info"><span class="label">Instance ID:</span><span>$instance_id </span></div>
<div class="info"><span class="label">Private IP:</span><span>$private_ip </span></div>
<div class="info"><span class="label">Public IP:</span><span>$public_ip </span></div>
<div class="info"><span class="label">Label Public DNS:</span><span>$public_dns </span></div>
<a href="https://$public_ip:$public_port" target="_blank">
https://\$public\_ip:\$public\_port

```

## 10. Final role-based playbook ansible/my-playbook.yaml:

- ansible/roles/webapp/tasks/main.yml:

```
@Zuha-Irfan → /workspaces/CC-ZuhaIrfan-073/terraform_machine/ansible (main) $ cat roles/webapp/templates/nginx.conf.j2
server {
listen 443 ssl;
server_name {{ 'server_public_ip '}};
ssl_certificate /etc/ssl/certs/selfsigned.crt;
ssl_certificate_key /etc/ssl/private/selfsigned.key;

location / {
root {{ 'web_root '}};
index {{ 'web_index_file '}} index.html index.htm;
# proxy_pass http://158.252.94.241:80;
# proxy_pass http://backend_servers;

location / {
try_files $uri $uri/ =404;
}

# ● This block is necessary for Php Website
location ~ \.php$ {
include fastcgi_params;
fastcgi_pass unix:/run/php-fpm/www.sock;
fastcgi_index index.php;
fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
}
}
server {
listen 80;
server_name _;
return 301 https://$host$request_uri;
}
```

## 10. Final role-based playbook ansible/my-playbook.yaml:

```

@Zuha-Irfan →/workspaces/CC-Zuhairfan-073/terraform_machine/ansible (main) $ ansible-playbook -i inventory/hosts my-playbook.yaml
TASK [ssl : Create SSL certs directory] ****
ok: [51.112.49.120]

TASK [ssl : Get IMDSv2 token] ****
ok: [51.112.49.120]

TASK [ssl : Get public IP] ****
ok: [51.112.49.120]

TASK [ssl : Save public IP as fact] ****
ok: [51.112.49.120]

TASK [ssl : Generate self-signed certificate] ****
ok: [51.112.49.120]

TASK [webapp : Install PHP packages] ****
ok: [51.112.49.120]

TASK [webapp : Copy PHP website] ****
changed: [51.112.49.120]

TASK [webapp : Deploy nginx config] ****
changed: [51.112.49.120]

TASK [webapp : Start and enable php-fpm] ****
ok: [51.112.49.120]

RUNNING HANDLER [webapp : Restart nginx] ****
changed: [51.112.49.120]

PLAY RECAP ****
51.112.49.120          : ok=15   changed=3   unreachable=0   failed=0    skipped=0   rescued=0   ignored=0

```

## 11. Run:

```

● @Zuha-Irfan →/workspaces/CC-Zuhairfan-073/terraform_machine/ansible (main) $ vim roles/webapp/tasks/main.yml
● @Zuha-Irfan →/workspaces/CC-Zuhairfan-073/terraform_machine/ansible (main) $ cat roles/webapp/tasks/main.yml
#SPDX-License-Identifier: MIT-0
---
# tasks file for webapp
- name: Install PHP packages
  yum:
    name:
      - php-fpm
      - php-curl
    state: present
  notify: Restart php-fpm

- name: Copy PHP website
  copy:
    src: index.php
    dest: "{{ web_root }}/{{ web_index_file }}"
    owner: nginx
    group: nginx
    mode: '0644'
  notify: Restart nginx

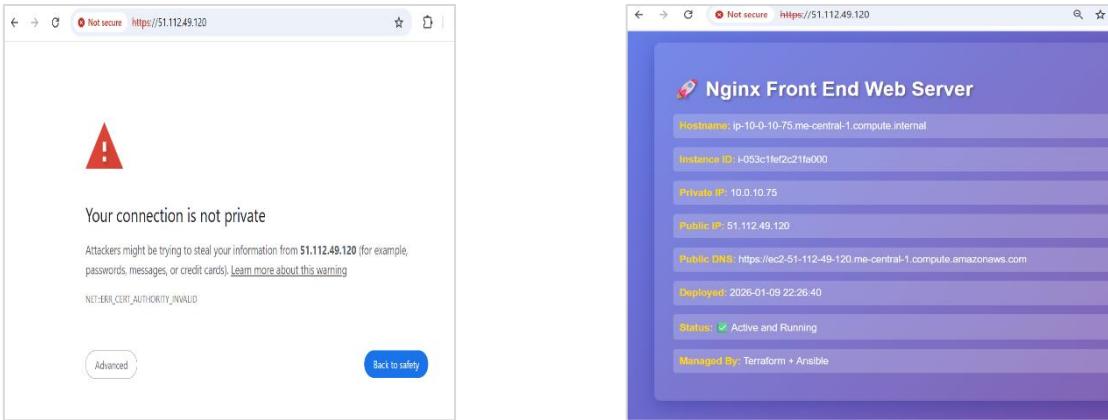
- name: Deploy nginx config
  template:
    src: nginx.conf.j2
    dest: /etc/nginx/nginx.conf
  notify: Restart nginx

- name: Start and enable php-fpm
  service:
    name: php-fpm
    state: started
    enabled: true
  notify: Restart nginx

● @Zuha-Irfan →/workspaces/CC-Zuhairfan-073/terraform_machine/ansible (main) $ vim my-playbook.yaml
● @Zuha-Irfan →/workspaces/CC-Zuhairfan-073/terraform_machine/ansible (main) $ cat my-playbook.yaml
---
- name: Deploy NGINX Web Stack with SSL and PHP
  hosts: nginx
  become: true
  roles:
    - nginx
    - ssl
    - webapp

```

Visit <https://<public-ip>> and verify the PHP page with metadata.



## Cleanup

### 1. From the Terraform root:

terraform destroy -auto-approve

```
Destroy complete! Resources: 11 destroyed.
● @Zuhai-Irfan →/workspaces/CC-ZuhaiIrfan-073/terraform_machine/ansible (main) $ terraform destroy -auto-approve
No changes. No objects need to be destroyed.

Either you have not created any objects yet or the existing objects were already deleted outside of Terraform.

Destroy complete! Resources: 0 destroyed.
● @Zuhai-Irfan →/workspaces/CC-ZuhaiIrfan-073/terraform_machine/ansible (main) $ cat terraform.tfstate
```

### 2. Verify state:

cat terraform.tfstate

```
● @Zuhai-Irfan →/workspaces/CC-ZuhaiIrfan-073/terraform_machine/ansible (main) $ cat terraform.tfstate
{
  "version": 4,
  "terraform_version": "1.14.3",
  "serial": 209,
  "lineage": "34978037-947f-49bd-9fd6-644456e744b7",
  "outputs": {},
  "resources": [],
  "check_results": null
}
```

### 3. Confirm that no EC2 instances remain in AWS console.

