

CLOUD COMPUTING LAB: 14

Submitted By:

Seerat Fatima

Registration. No:

2023-BSE-060

Submitted To:

Engr. Shoaib

Section:

5B

Task 0 – Lab Setup (Codespace & GH CLI)

The image shows a GitHub repository named `terraform_machine` forked from `WaqasSaleem97/terraform_machine`. The repository is public and has 0 stars, 0 forks, and 0 watchers. It is currently on the `main` branch, which is up to date with the upstream. The repository contains a `modules` directory and a `README.md` file. The `README.md` file is titled `Ansible` and contains the text `Ansible`.

The screenshot also shows a VS Code Codespace environment. The Explorer view on the left shows the file structure of the `terraform_machine` workspace, including `modules`, `.gitignore`, `locals.tf`, `main.tf`, `outputs.tf`, `README.md`, and `variables.tf`. The main editor area shows the `README.md` file. The terminal at the bottom shows the following commands and output:

```
@SeratFatima00 →/workspaces/terraform_machine (main) $ aws --version
aws-cli/2.33.2 Python/3.13.11 Linux/6.8.0-1030-azure exe/x86_64.ubuntu.24
@SeratFatima00 →/workspaces/terraform_machine (main) $ terraform --version
Terraform v1.14.3
on linux_amd64
@SeratFatima00 →/workspaces/terraform_machine (main) $ ansible --version || echo "ansible not yet installed"
bash: ansible: command not found
ansible not yet installed
```

The terminal output shows that the `aws` CLI is installed, but `terraform` and `ansible` are not. The `aws sts get-caller-identity` command is also shown, returning the following JSON output:

```
{
  "UserId": "AIDA3QUYFZVYDKQOBZV7R",
  "Account": "791666871664",
  "Arn": "arn:aws:iam::791666871664:user/Admin"
}
```

Task 1 – Generate ssh key and Initial Terraform apply

```
• @SeratFatima00 →/workspaces/terraform_machine (main) $ ls ~/.ssh
• @SeratFatima00 →/workspaces/terraform_machine (main) $

• @SeratFatima00 →/workspaces/terraform_machine (main) $ ssh-keygen -t ed25519 -f ~/.ssh/id_ed25519 -N ""
Generating public/private ed25519 key pair.
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:a7Ex2QnP3R3dfaec1GKoKa0GAu2uFHCPC8fwII/eTU codespace@codespaces-647e69
The key's randomart image is:
+--[ED25519 256]--+
|
| .          . .+ |
| o o      . . o o.* |
| .* o    . O + = ++ |
| ..B + E S = . = . |
| o.O o + *         |
| .B + o +         |
| .. = o .         |
| .               |
+----[SHA256]-----+

• @SeratFatima00 →/workspaces/terraform_machine (main) $ ls -la ~/.ssh
total 20
drwxr-xr-x 2 codespace codespace 4096 Jan 17 08:44 .
drwxr-x--- 1 codespace codespace 4096 Jan 17 08:43 ..
-rw----- 1 codespace codespace  419 Jan 17 08:44 id_ed25519
-rw-r--r-- 1 codespace codespace  109 Jan 17 08:44 id_ed25519.pub

• @SeratFatima00 →/workspaces/terraform_machine (main) $ touch terraform.tfvars
• @SeratFatima00 →/workspaces/terraform_machine (main) $ ls -la terraform.tfvars
-rw-rw-rw- 1 codespace codespace 0 Jan 17 08:45 terraform.tfvars

• @SeratFatima00 →/workspaces/terraform_machine (main) $ nano terraform.tfvars
• @SeratFatima00 →/workspaces/terraform_machine (main) $ cat terraform.tfvars
vpc_cidr_block      = "10.0.0.0/16"
subnet_cidr_block   = "10.0.10.0/24"
availability_zone    = "eu-north-1a"
env_prefix           = "dev"
instance_type        = "t3.micro"
public_key            = "~/.ssh/id_ed25519.pub"
private_key           = "~/.ssh/id_ed25519"
```

```

● @SeratFatima00 → /workspaces/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/aws...
- Finding latest version of hashicorp/http...
- Installing hashicorp/aws v6.28.0...
- Installed hashicorp/aws v6.28.0 (signed by HashiCorp)
- Installing hashicorp/http v3.5.0...
- Installed hashicorp/http v3.5.0 (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.

```

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

Changes to Outputs:

```

+ webserver_public_ips = [
  + (known after apply),
]

```

module.myapp-webserver.aws_instance.myapp-server[0]: Creating...

module.myapp-webserver.aws_instance.myapp-server[0]: Still creating... [00m10s elapsed]

module.myapp-webserver.aws_instance.myapp-server[0]: Creation complete after 14s [id=i-0f1a6566a3f2b30b4]

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

Outputs:

```

webserver_public_ips = [
  "13.60.244.14",
]

```

```

● @SeratFatima00 → /workspaces/terraform_machine (main) $ terraform output
webserver_public_ips = [
  "13.60.244.14",
]

```

Task 2 – Static Ansible inventory with two EC2 instances

```
@SeratFatima00 →/workspaces/terraform_machine (main) $ pipx install ansible-core
```

```
⚠ Note: ansible-vault was already on your PATH at /usr/bin/ansible-vault
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! 🌟🌟🌟
```

```
• @SeratFatima00 →/workspaces/terraform_machine (main) $ ansible --version
```

```
ansible [core 2.20.1]
  config file = None
  configured module search path = ['/home/codespace/.ansible/plugins/modules', '/usr/share/ansible/plugins/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
  pyyaml version = 6.0.3 (with libyaml v0.2.5)
```

```
• @SeratFatima00 →/workspaces/terraform_machine (main) $ cd /workspaces/terraform_machine
```

```
• @SeratFatima00 →/workspaces/terraform_machine (main) $ touch hosts
```

```
• @SeratFatima00 →/workspaces/terraform_machine (main) $ ls -la hosts
-rw-rw-rw- 1 codespace codespace 0 Jan 18 03:47 hosts
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

nano + ▾ ... |

```
GNU nano 7.2 hosts
13.60.244.14 ansible_user=ec2-user ansible_ssh_private_key_file=~/.ssh/id_ed25519 ansible_ssh_common_args='-o
51.21.253.216 ansible_user=ec2-user ansible_ssh_private_key_file=~/.ssh/id_ed25519 ansible_ssh_common_args='-o
|
```

```

13.60.244.14 | SUCCESS => {
  "changed": false,
  "invocation": {
    "module_args": {
      "data": "pong"
    }
  },
  "ping": "pong"
}
<51.21.253.216> (0, b'', b'')
51.21.253.216 | SUCCESS => {
  "changed": false,
  "invocation": {
    "module_args": {
      "data": "pong"
    }
  },
  "ping": "pong"
}

```

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

```

vpc_cidr_block      = "10.0.0.0/16"
subnet_cidr_block   = "10.0.10.0/24"
env_prefix           = "dev"
instance_type        = "t3.micro"          # Free Tier eligible
availability_zone     = "eu-north-1a"      # keep original region
instance_count       = 3
public_key            = "~/.ssh/id_ed25519.pub"
private_key           = "~/.ssh/id_ed25519"
~

```

```
@SeratFatima00 →/workspaces/terraform_machine (main) $ terraform apply -auto-approve
```

```
]
module.myapp-webserver.aws_security_group.web_sg[2]: Creating...
module.myapp-webserver.aws_security_group.web_sg[1]: Modifying... [id=sg-0e8b36481765f07c6]
module.myapp-webserver.aws_security_group.web_sg[0]: Modifying... [id=sg-02903d2340f3a08ef]
module.myapp-webserver.aws_security_group.web_sg[1]: Modifications complete after 2s [id=sg-0e8b36481765f07c6]
module.myapp-webserver.aws_security_group.web_sg[0]: Modifications complete after 2s [id=sg-02903d2340f3a08ef]
module.myapp-webserver.aws_instance.myapp-server[2]: Creating...
module.myapp-webserver.aws_security_group.web_sg[2]: Creation complete after 5s [id=sg-05ab5ef421e1494dd]
module.myapp-webserver.aws_instance.myapp-server[2]: Still creating... [00m10s elapsed]
module.myapp-webserver.aws_instance.myapp-server[2]: Creation complete after 14s [id=i-035feacf300b2cc1d]
```

Apply complete! Resources: 2 added, 2 changed, 0 destroyed.

Outputs:

```
webserver_public_ips = [
  "13.60.244.14",
  "51.21.253.216",
  "13.51.241.6",
]
```

```
● @SeratFatima00 →/workspaces/terraform_machine (main) $ terraform output
webserver_public_ips = [
  "13.60.244.14",
  "51.21.253.216",
  "13.51.241.6",
]
```

```
GNU nano 7.2                                     hosts *
[ec2]
13.60.244.14
51.21.253.216

[ec2:vars]
ansible_user=ec2-user
ansible_ssh_private_key_file=~/.ssh/id_ed25519
ansible_ssh_common_args='-o StrictHostKeyChecking=no'
ansible_python_interpreter=/usr/local/bin/python3.9

[droplet]
13.51.241.6

[droplet:vars]
ansible_user=ec2-user
ansible_ssh_private_key_file=~/.ssh/id_ed25519
ansible_ssh_common_args='-o StrictHostKeyChecking=no'
ansible_python_interpreter=/usr/local/bin/python3.9
```

```
● @SeratFatima00 →/workspaces/terraform_machine (main) $ ansible ec2 -i hosts -m ping
13.60.244.14 | SUCCESS => {
  "changed": false,
  "ping": "pong"
}
51.21.253.216 | SUCCESS => {
  "changed": false,
  "ping": "pong"
}
```

```
● @SeratFatima00 →/workspaces/terraform_machine (main) $ ansible 13.60.244.14 -i hosts -m ping
13.60.244.14 | SUCCESS => {
  "changed": false,
  "ping": "pong"
}
```

```

@SeratFatima00 → /workspaces/terraform_machine (main) $ ansible all -i hosts -m ping
13.60.244.14 | SUCCESS => {
  "changed": false,
  "ping": "pong"
}
51.21.253.216 | SUCCESS => {
  "changed": false,
  "ping": "pong"
}
13.51.241.6 | SUCCESS => {
  "changed": false,
  "ping": "pong"
}

```

Task 4 – Global ansible.cfg & first nginx playbook

```

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

```

```

@SeratFatima00 → /workspaces/terraform_machine (main) $ cat hosts
[ec2]
13.60.244.14
51.21.253.216

[ec2:vars]
ansible_user=ec2-user
ansible_ssh_private_key_file=~/.ssh/id_ed25519
ansible_python_interpreter=/usr/local/bin/python3.9

[droplet]
13.51.241.6

[droplet:vars]
ansible_user=ec2-user
ansible_ssh_private_key_file=~/.ssh/id_ed25519
ansible_python_interpreter=/usr/local/bin/python3.9

```



```

@SeratFatima00 →/workspaces/terraform_machine (main) $ ansible all -i hosts -m ping
13.60.244.14 | SUCCESS => {
  "changed": false,
  "ping": "pong"
}
13.51.241.6 | SUCCESS => {
  "changed": false,
  "ping": "pong"
}
51.21.253.216 | SUCCESS => {
  "changed": false,
  "ping": "pong"
}

```

```

---
- 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

```

```

@SeratFatima00 →/workspaces/terraform_machine (main) $ ansible-playbook -i hosts my-playbook.yaml

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

TASK [Enable and install nginx from amazon-linux-extras] *****
changed: [13.60.244.14]
changed: [51.21.253.216]

TASK [start nginx server] *****
changed: [51.21.253.216]
changed: [13.60.244.14]

PLAY RECAP *****
13.60.244.14      : ok=2    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
51.21.253.216   : ok=2    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

```

Welcome to nginx!

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

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

Thank you for using nginx.

```
@SeratFatima00 →/workspaces/terraform_machine (main) $ ansible-playbook -i hosts my-playbook.yaml

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

TASK [Enable and install nginx from amazon-linux-extras] *****
changed: [13.51.241.6]

TASK [start nginx server] *****
changed: [13.51.241.6]

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

Task 5 – Single nginx target group & HTTPS prerequisites

```
@SeratFatima00 →/workspaces/terraform_machine (main) $ touch ansible.cfg
@SeratFatima00 →/workspaces/terraform_machine (main) $ ls -la ansible.cfg
-rw-rw-rw- 1 codespace codespace 0 Jan 18 22:23 ansible.cfg
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
[defaults]
host_key_checking = False
interpreter_python = /usr/local/bin/python3.9
```

```
~
~
~
~
```

```

vpc_cidr_block      = "10.0.0.0/16"
subnet_cidr_block   = "10.0.10.0/24"
env_prefix          = "dev"
instance_type       = "t3.micro"          # Free Tier eligible
availability_zone    = "eu-north-1a"     # keep original region
instance_count      = 1
public_key           = "~/ssh/id_ed25519.pub"
private_key          = "~/ssh/id_ed25519"
~
~
~
~
~
~

```

@SeratFatima00 → /workspaces/terraform_machine (main) \$ terraform apply -auto-approve

```

module.myapp-webserver.aws_instance.myapp-server[1]: Still destroying... [id=i-0cf5bc25b1e90d49a, 00m20s elapsed]
module.myapp-webserver.aws_instance.myapp-server[1]: Still destroying... [id=i-0cf5bc25b1e90d49a, 00m30s elapsed]
module.myapp-webserver.aws_instance.myapp-server[2]: Still destroying... [id=i-035feacf300b2cc1d, 00m30s elapsed]
module.myapp-webserver.aws_instance.myapp-server[2]: Still destroying... [id=i-035feacf300b2cc1d, 00m40s elapsed]
module.myapp-webserver.aws_instance.myapp-server[1]: Still destroying... [id=i-0cf5bc25b1e90d49a, 00m40s elapsed]
module.myapp-webserver.aws_instance.myapp-server[1]: Destruction complete after 41s
module.myapp-webserver.aws_instance.myapp-server[2]: Still destroying... [id=i-035feacf300b2cc1d, 00m50s elapsed]
module.myapp-webserver.aws_instance.myapp-server[2]: Destruction complete after 51s
module.myapp-webserver.aws_security_group.web_sg[1]: Destroying... [id=sg-0e8b36481765f07c6]
module.myapp-webserver.aws_security_group.web_sg[2]: Destroying... [id=sg-05ab5ef421e1494dd]
module.myapp-webserver.aws_security_group.web_sg[2]: Destruction complete after 1s
module.myapp-webserver.aws_security_group.web_sg[1]: Destruction complete after 1s

```

Apply complete! Resources: 0 added, 0 changed, 4 destroyed.

Outputs:

```

webserver_public_ips = [
  "13.60.244.14",
]

```

```

• @SeratFatima00 → /workspaces/terraform_machine (main) $ terraform output
  webserver_public_ips = [
    "13.60.244.14",
  ]

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```

GNU nano 7.2                                                                    hosts
[nginx]
13.60.244.14

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

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
---
- name: Configure nginx web server
  hosts: nginx
  become: true
  gather_facts: false

tasks:
  - name: Install nginx via amazon-linux-extras
    command: amazon-linux-extras install -y nginx1

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

  - name: Start nginx server
    service:
      name: nginx
      state: started
      enabled: true
```

~
~

● @SeratFatima00 → /workspaces/terraform_machine (main) \$ 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 [Install nginx via amazon-linux-extras] *****
changed: [13.60.244.14]

TASK [Install openssl via yum (command fallback)] *****
changed: [13.60.244.14]

TASK [Start nginx server] *****
changed: [13.60.244.14]

TASK [Enable nginx on boot] *****
changed: [13.60.244.14]

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

⚠ Not secure 13.60.244.14

Welcome to nginx!

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

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

Thank you for using nginx.

Task 6 - Ansible-managed SSL certificates

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
- name: Configure SSL certificates
  hosts: nginx
  become: true
  gather_facts: false

  tasks:
    - name: Create SSL private directory
      file:
        path: /etc/ssl/private
        state: directory
        mode: '0700'

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

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

```
[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-wor
ld-writable-dir
```

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

```
TASK [Gathering Facts] *****
[WARNING]: Host '158.252.85.109' is using the discovered Python interpreter at '/usr/bin/python3.9', but future installa
tion of another Python interpreter could cause a different interpreter to be discovered. See https://docs.ansible.com/an
sible-core/2.20/reference_appendices/interpreter_discovery.html for more information.
ok: [158.252.85.109]
```

```
TASK [install nginx and update cache] *****
ok: [158.252.85.109]
```

```
TASK [install openssl] *****
ok: [158.252.85.109]
```

```
TASK [start nginx server] *****
ok: [158.252.85.109]
```

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

```
TASK [Gathering Facts] *****
ok: [158.252.85.109]
```

```

#_
~\_##### Amazon Linux 2023
~\_#####
~\_###|
~\_#/ https://aws.amazon.com/linux/amazon-linux-2023
~\_V~' ->
~\_
~\_
~\_/m/'

Last login: Fri Jan 9 19:16:56 2026 from 4.240.18.226
[ec2-user@ip-10-0-10-74 ~]$ sudo cat /etc/ssl/certs/selfsigned.crt
-----BEGIN CERTIFICATE-----
MIIDQTCCAimgAwIBAgIUf7C1kQDYrSyFckDPmma8PGn50kcwDQYJKoZIhvcNAQEL
BQAwGTEXMBUGA1UEAwOMTU4LjI1M144NS4xMDkwHhcNMjYwMTA5MTkxNjU2WhcN
MjcwMTA5MTkxNjU2WjAQMRCwFQYDVQQDDA4xNTguMjYwMTA5MTkxNjU2WjEwOTCC
ASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBAlfrch90mc8I1duRRbJ0pZz2DpkFkuK
Py6JmyQDRRtfcwonW1Nqk5E3d8Pcc9uEfztTN3DuCpSrexw02AylfhtUuc0xXf
0JCy+mW47+a4Uoy1yImun4LFv+IZNuQgiQ17U13ITrfxiOFYwwlWnWqD26NSB0S
YLereb0Wz9KZa7GbmMFkHG/o36eLc6RgVJ5rv3sMppzWmeQDBztK+AeU5NrZyoi5
o2tTGZiBhzKB9t1VTYAmWQZ8yLS5VST+RK7MMxpWZj607inAbun1nv+CjRi/eOCp
mINqOxlegyqUnvJtPHM+1gQ+p/uXVjFVxQFfabQ0dYa8hspUyiXuzH4cAwEAAaOB
gDB+MB0GA1UdDgQWBBT64nL3eNFNxUnnUAKJ7PxKmiTNNzAFBgNVHSMEGDAWgBT6
4nL3eNFNxUnnUAKJ7PxKmiTNNzAPBgNVHREEDAGhwSe/FVtMAKGA1UdEwQCMAAw
CwYDVRR0PBAQDAgWgMBMGA1UdJQQMMAoGCCsGAQUFBwMBMA0GCSCqGSIb3DQEBCwUA
A4IBAQA4ogdZvkrFyhKZUAsp1eh2MMWIwqQJ7zKBPGqqEhWUW4Y7SEiuAkuKaE7n
mkNqYmwRvbIzbgcsizI4fewSZ3CnZbIZuVkcj4fc4VqbQ1kyXwS/LO/nXT0ShLyn
Ez1IZuKAi2H07ONGKpWiseBkpVo9KZlBTWBpnp95dqmpcpYa8LhMZ0s96018LEQW
zZYXwOpEVwhVyp1xbPnbgEvVefN7iDvhIwKZu+eLtCVN5MvyThs0Kxu85EDfWdx
lHcW6DzF3QeLtIKhBVpwK34LvahJhS/H5IuMukcMFbHiTANLPp9DrQ97wYYA10Qn
FwZkL1lrLhkyJ0636YmIEVPm8zo6
-----END CERTIFICATE-----
```

```
[ec2-user@ip-10-0-10-74 ~]$ sudo cat /etc/ssl/private/selfsigned.key
-----BEGIN PRIVATE KEY-----
MIIEvgIBADANBgkqhkiG9w0BAQEFAASCBAgEAAoIBAQCCHn63IfdJnPCNX
bkUwydKWc9g6ZBZLij8uiZska0UbX3MKJ1tTapORN3fDwnPbhH87Uzdw7gqUq3sW
1ntNgMpX4bVLnNMV39CQsvplu0/muFKMtcoCJrp+Cxb/iGTbkIIkNe1NdyE638Yj
hWMMJVp1qg9ujUgdEmCxK3m9Fs/Smlwuxm5jBZBxv6N+ni30kYFSea797DKac1pnk
Awc7Svghl0Ta2cqIuaNrUxmYgYcygfbdVU2AJlkGfMi0uVUk/kSuzDMacGY+t04p
wG7p9Z7/go0Yv3jgqZiDajsZXoMqlJ7ybTxzPtYEPqf7l1YxVcUBWm0DnWgvIbKV
Mol7sx+HAgMBAAECggEAP2lHZcB+icoydASYxQjkl3hGlzaIvhULaYTpRrNIsn20
WgEA0EW6bWFT2/GlHfQbU0+dsm16AP0Udh8eU6Rt9x0PGSVgBJA7AgzMypbnhCaW
Yjf1FYBqHditjhiFn6s3ThrwsxGKNe6+w4jwQJ00cglrYbMpDn6z0uMq7mAmFz76
PJ+ehuDgdwefuYZmpffCsnczESGAN7b6YLnzfuWzhhnCsbPr9Pa295ZDV7c4mAV
vZVxN20VUCz4Iet8QgwXcqzpkU/Dp+kbZbbGgkByV75wUaNLlTrssdC/EiU1UrOH
v+zQW9uvA+VGh/sYeRJDSa0qgtf2xbsrFdoMXIFboQKBgQC8YT+fba7Y0ks32TKt
g87zI0waj2xqDRqpjQW4a1WoDHeKPi2kX6tuoC76mmJ0AjVHNmBt9o8UKwUbza0
+BxUTNT8wSA7wAB4wuYx1yU08rQTM0g/32SefSB0y/ARd0jyioXE3knaCucMgBsJ
Q9R0o0wn6gYbwke3Q0mPxPEv3wKBgQC4ToXA9osLuK8S2gI4Keek7viw0T7xCvvV
aD9SMD6TTsz0KV0+fUZEG9mgb2sqXzTw6fqld3zya+iYehSxcXhxJzrupfadJ8tn
7WjMdlwHTR0CJELIXeujDCz09C+SgEa0hJJ/vzsJx3rX4A+Jh72Y6zXaKWTMve96
wG5lUNnlWQKBgQCKi5mES81HT0Ew5kJy98gj7rgCFzd0pyWb5jeR8AeA42rjUQ6T
ShdJ7reV7NrE6vdcF3gR66Lvvl0hAce50BWeuQHtyU+7gSmoUi26aT/Qf8U1ZDz+
tcXw/kDjMxX+W6BXYxIXdUcy8wUAKUGMZThFrYH094bgC4rLFDZEI3W9JwKBgQCK
kud4FSaocnyXxPikN8WmOdMcRsd3PpXod9AcQI/b/AS0jgNAP7o8HWB8gyllG15X
wOSmVEXSk5W3hXql67XuBdwGcRxCPxhtlJAWIuvAxZcI9DCOLCeUwMob9HcMRixI
PZ9Uk3iV0rm/NAUal8fTAQ5lcHgMhMZZWfPnE2oxSQKBgAsQuPV0x0BhapfR0lnb
OgAIQi8dvFyAbrnBwe6ak0In/XER1Wc700wV2u0BsR0zSh/WM8Pq8u35Znb0JlRm
Fb0uD0CSm0CcCS7on0mB/uht9dNpBzYDpvHXvlpnNiz1f9lYcQRURax2N0mIstZL
3hUhBP6g37Ym/aq7eu27jUe4
-----END PRIVATE KEY-----
[ec2-user@ip-10-0-10-74 ~]$
```

Task 7 - PHP front-end deployment with templates

```
GNU nano 7.2
<?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 IMDSv2 token
function getImdsV2Token() {
    $ch = curl_init("http://169.254.169.254/latest/api/token");
    curl_setopt_array($ch, [
        CURLOPT_RETURNTRANSFER => true,
        CURLOPT_CUSTOMREQUEST  => "PUT",
        CURLOPT_HTTPHEADER     => [
            "X-aws-ec2-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-ec2-metadata-token: $token"
        ],
        CURLOPT_TIMEOUT         => 2
    ]);
}
```



```

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 158.252.85.109:80;
    }

    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;

            location ~ \.php$ {
                include fastcgi_params;

```

```
- name: Deploy Nginx website and configuration files
hosts: nginx
become: true
vars:
  server_public_ip: "{{ ansible_host }}"
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
```

```
[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-wor
ld-writable-dir

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

TASK [Gathering Facts] *****
[WARNING]: Host '158.252.85.109' is using the discovered Python interpreter at '/usr/bin/python3.9', but future installa
tion of another Python interpreter could cause a different interpreter to be discovered. See https://docs.ansible.com/an
sible-core/2.20/reference_appendices/interpreter_discovery.html for more information.
ok: [158.252.85.109]

TASK [install nginx and update cache] *****
ok: [158.252.85.109]

TASK [install openssl] *****
ok: [158.252.85.109]

TASK [start nginx server] *****
ok: [158.252.85.109]

PLAY [Configure SSL certificates] *****

TASK [Gathering Facts] *****
ok: [158.252.85.109]

TASK [Create SSL private directory] *****
ok: [158.252.85.109]
```

Not secure https://158.252.85.109

Nginx Front End Web Server

Hostname: ip-10-0-10-74.me-central-1.compute.internal


Instance ID: i-03205fae6da402e84

Private IP: 10.0.10.74

Public IP: 158.252.85.109

Public DNS: https://N/A

Deployed: 2026-01-10 11:38:55

Status:  Active and Running

Managed By: Terraform + Ansible

Task 8 – Docker & Docker Compose provisioning via Ansible

```

module.myapp-webserver[0].aws_key_pair.ssh-key: Refreshing state... [id=dev-serverkey-0]
aws_vpc.myapp_vpc: Refreshing state... [id=vpc-0e89fb9c7b6c7527b]
module.myapp-subnet.aws_internet_gateway.myapp_igw: Refreshing state... [id=igw-0232fd16d032b7629]
module.myapp-subnet.aws_subnet.myapp_subnet_1: Refreshing state... [id=subnet-07ff526267785d880]
module.myapp-webserver[0].aws_security_group.web_sg: Refreshing state... [id=sg-00a549d943c197cb4]
module.myapp-subnet.aws_default_route_table.main_rt: Refreshing state... [id=rtb-0d413420cf62f26af]
module.myapp-webserver[0].aws_instance.myapp-server: Refreshing state... [id=i-03205fae6da402e84]

```

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:

- destroy

Terraform will perform the following actions:

```

# aws_vpc.myapp_vpc will be destroyed
- resource "aws_vpc" "myapp_vpc" {
  - arn                                = "arn:aws:ec2:me-central-1:535942231766:vpc/vpc-0e89fb9c7b6c7527b" -> null
  - assign_generated_ipv6_cidr_block  = false -> null
  - cidr_block                        = "10.0.0.0/16" -> null
  - default_network_acl_id           = "acl-0cdccb62e169e82ec" -> null
  - default_route_table_id           = "rtb-0d413420cf62f26af" -> null
  - default_security_group_id        = "sg-06469fcae02526253" -> null
  - dhcp_options_id                  = "dopt-02c0a1e8f3dd284d6" -> null
  - enable_dns_hostnames              = false -> null
  - enable_dns_support                = true -> null
  - enable_network_address_usage_metrics = false -> null
  - id                               = "vpc-0e89fb9c7b6c7527b" -> null
  - instance_tenancy                 = "default" -> null
  - ipv6_netmask_length               = 0 -> null
  - main_route_table_id               = "rtb-0d413420cf62f26af" -> null
  - owner_id                         = "535942231766" -> null
  - region                           = "me-central-1" -> null
  - tags                             = {
    - "Name" = "dev-vpc"
  } -> null
- tags_all                          = {}

```

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:

- + create

Terraform will perform the following actions:

```

# aws_vpc.myapp_vpc will be created
+ resource "aws_vpc" "myapp_vpc" {
  + arn                                = (known after apply)
  + cidr_block                        = "10.0.0.0/16"
  + default_network_acl_id           = (known after apply)
  + default_route_table_id           = (known after apply)
  + default_security_group_id        = (known after apply)
  + dhcp_options_id                  = (known after apply)
  + enable_dns_hostnames              = (known after apply)
  + enable_dns_support                = true
  + enable_network_address_usage_metrics = (known after apply)
  + id                               = (known after apply)
  + instance_tenancy                 = "default"
  + ipv6_association_id              = (known after apply)
  + ipv6_cidr_block                   = (known after apply)
  + ipv6_cidr_block_network_border_group = (known after apply)
  + main_route_table_id               = (known after apply)
  + owner_id                         = (known after apply)
  + region                           = "me-central-1"
  + tags                             = {
    + "Name" = "dev-vpc"
  }
+ tags_all                          = {
  + "Name" = "dev-vpc"
}
}

```

Outputs:

```

webserver_public_ips = [
  "158.252.78.82",
]

```

02eeb8b4-f19e-4655-8171-406000000000

```
Command Prompt - gti - codespace ssh - C miniature-adventure-pj0r554jix4jixgp
GNU nano 7.2
[docker_servers]
158.252.78.82

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

```
Command Prompt - gti - codespace ssh - C miniature-adventure-pj0r554jix4jixgp
GNU nano 7.2 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: +x

    - 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
```

```
#  
~_##### Amazon Linux 2023  
nnn\_\_\_\_\_\_  
nnn\\\____|\\  
nnn\\#/  
V~' ->  
  
Last login: Sat Jan 10 12:39:55 2026 from 20.192.21.50  
[ec2-user@ip-10-0-10-90 ~]$ sudo docker ps
```

	CONTAINER ID	IMAGE	CMD	STATUS	PORTS	NAMES
[ec2-user@ip-10-0-10-90 ~]	\$ exit					

```
logout  
Connection to 158.252.78.82 closed.  
toobashafique065 @ /workspaces/terraform_machine (main) $
```

```
- 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
```

GNU nano 7.2

normal_user: ec2-user

docker_compose_file_location: .

```
- name: Deploy Docker Containers
  hosts: all
  become: true
  user: "{{ normal_user }}"
  vars_files:
    - project-vars.yaml
  tasks:
    - name: check if docker-compose file exists
      stat:
        path: /home/{{ normal_user }}/compose.yaml
      register: compose_file

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

    - name: deploy containers using docker-compose
      command: docker compose up -d
      args:
        chdir: /home/{{ normal_user }}
      register: compose_result
      changed_when: "'Creating' in compose_result.stdout or 'Recreating' in compose_result.stdout"
```


GNU nano 7.2

```
services:
  gitea:
    image: gitea/gitea:latest
    container_name: gitea
    environment:
      - DB_TYPE=postgres
      - DB_HOST=db:5432
      - DB_NAME=gitea
      - DB_USER=gitea
      - DB_PASSWD=gitea
    restart: always
    volumes:
      - gitea:/data
    ports:
      - 3000:3000
    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:
```

```
@toobashafique065 @ /workspaces/terraform_machine (main) $ 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 informat
cs.ansible.com/ansible/devel/reference_appendices/config.html#cfg-in-world-writable-dir

PLAY [Configure Docker] *****

TASK [Gathering Facts] *****
[WARNING]: Host '158.252.78.82' is using the discovered Python interpreter at '/usr/bin/python3.9', but future installation of another Python interpre
different interpreter to be discovered. See https://docs.ansible.com/ansible-core/2.20/reference_appendices/interpreter_discovery.html for more inform
ok: [158.252.78.82]

TASK [install docker and update cache] *****
ok: [158.252.78.82]

PLAY [Install Docker Compose] *****

TASK [Gathering Facts] *****
ok: [158.252.78.82]

TASK [create docker cli-plugins directory] *****
ok: [158.252.78.82]

TASK [install docker-compose] *****
ok: [158.252.78.82]

TASK [View architecture of the system] *****
ok: [158.252.78.82] => {
  "msg": "System architecture of 158.252.78.82 is x86_64"
}

TASK [Alternate method to view architecture of the system] *****
ok: [158.252.78.82] => {
  "msg": "System architecture of 158.252.78.82 is x86_64"
}
```

```
ingress {
  from_port    = 3000
  to_port      = 3000
  protocol     = "tcp"
  cidr_blocks  = ["0.0.0.0/0"]
}
```

```
module.myapp-webserver[0].aws_key_pair.ssh-key: Refreshing state... [id=dev-serverkey-0]
aws_vpc.myapp_vpc: Refreshing state... [id=vpc-0f313f58fd0996cd0]
module.myapp-subnet.aws_internet_gateway.myapp_igw: Refreshing state... [id=igw-0a86095ed97fd33d4]
module.myapp-subnet.aws_subnet.myapp_subnet_1: Refreshing state... [id=subnet-0a454ec6f2a1ce5]
module.myapp-webserver[0].aws_security_group.web_sg: Refreshing state... [id=sg-0bf285c4870ec7dff]
module.myapp-subnet.aws_default_route_table.main_rt: Refreshing state... [id=rtb-0d1f69a2073f9fd17]
module.myapp-webserver[0].aws_instance.myapp-server: Refreshing state... [id=i-01d7b80f722d4f566]
```

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:

- ~ update in-place

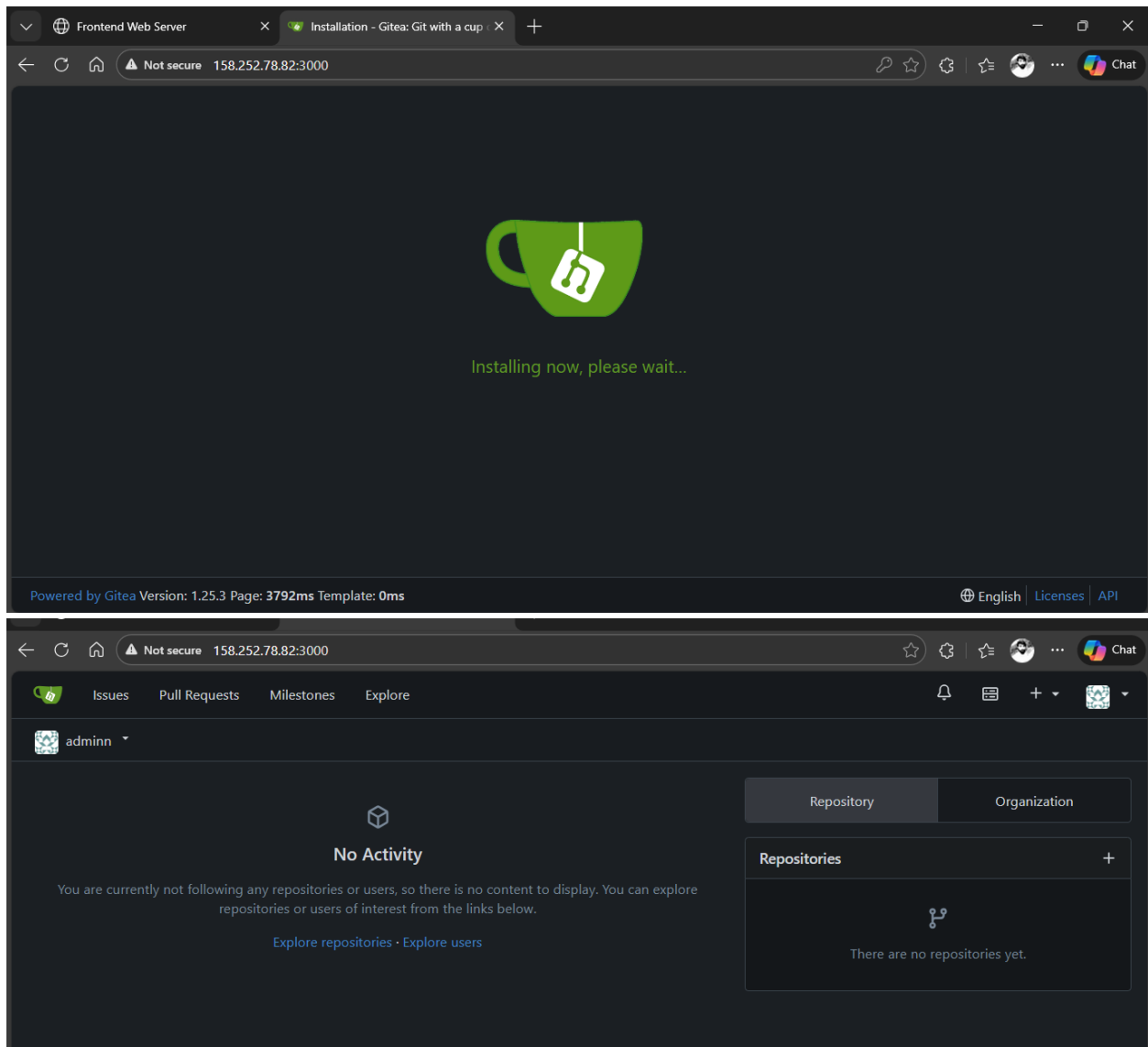
Terraform will perform the following actions:

```
# module.myapp-webserver[0].aws_security_group.web_sg will be updated in-place
~ resource "aws_security_group" "web_sg" {
  id      = "sg-0bf285c4870ec7dff"
  ~ ingress = [
    - {
      - cidr_blocks = [
        - "0.0.0.0/0",
      ]
      - from_port    = 22
      - ipv6_cidr_blocks = []
      - prefix_list_ids = []
      - protocol     = "tcp"
      - security_groups = []
      - self         = false
      - to_port      = 22
      # (1 unchanged attribute hidden)
    },
    - {
      - cidr_blocks = [
        - "0.0.0.0/0",
      ]
      - from_port    = 22
      - ipv6_cidr_blocks = []
      - prefix_list_ids = []
      - protocol     = "tcp"
      - security_groups = []
      - self         = false
      - to_port      = 22
      # (1 unchanged attribute hidden)
    },
  ]
}
```

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

Outputs:

```
webserver_public_ips = [
  "158.252.78.82",
]
```



Task 10 – Automating Ansible with Terraform (null_resource)

```

resource "null_resource" "configure_server" {
  triggers = {
    webserver_public_ips_for_aws = module.myapp-webserver[0].aws_instance_public_ip
  }

  depends_on = [module.myapp-webserver]

  provisioner "local-exec" {
    command = <<-EOT
      ansible-playbook -i ${self.triggers.webserver_public_ips_for_aws}, \
        --private-key ~/.ssh/id_ed25519 --user ec2-user \
        my-playbook.yaml
    EOT
  }
}

```

```

module.myapp-webserver[0].aws_key_pair.ssh-key: Refreshing state... [id=dev-serverkey-0]
aws_vpc.myapp_vpc: Refreshing state... [id=vpc-0f313f58fd0996cd0]
module.myapp-subnet.aws_internet_gateway.myapp_igw: Refreshing state... [id=igw-0a86095ed97fd33d4]
module.myapp-subnet.aws_subnet.myapp_subnet_1: Refreshing state... [id=subnet-0a454ec6f2a1ce5]
module.myapp-webserver[0].aws_security_group.web_sg: Refreshing state... [id=sg-0bf285c4870ec7dff]
module.myapp-subnet.aws_default_route_table.main_rt: Refreshing state... [id=rtb-0d1f69a2073f9fd17]
module.myapp-webserver[0].aws_instance.myapp-server: Refreshing state... [id=i-01d7b80f722d4f566]

```

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:

- destroy

Terraform will perform the following actions:

```

# aws_vpc.myapp_vpc will be destroyed
- resource "aws_vpc" "myapp_vpc" {
  - arn                                = "arn:aws:ec2:me-central-1:535942231766:vpc/vpc-0f313f58fd0996cd0"
  - assign_generated_ipv6_cidr_block  = false -> null
  - cidr_block                        = "10.0.0.0/16" -> null
  - default_network_acl_id           = "acl-0eb7230b55259fd9d" -> null
  - default_route_table_id           = "rtb-0d1f69a2073f9fd17" -> null
  - default_security_group_id        = "sg-062bcb28cec0a999d" -> null
  - dhcp_options_id                  = "dopt-02c0a1e8f3dd284d6" -> null
  - enable_dns_hostnames              = false -> null
  - enable_dns_support               = true -> null
  - enable_network_address_usage_metrics = false -> null
  - id                               = "vpc-0f313f58fd0996cd0" -> null
  - instance_tenancy                 = "default" -> null
  - ipv6_netmask_length              = 0 -> null
  - main_route_table_id              = "rtb-0d1f69a2073f9fd17" -> null
  - owner_id                         = "535942231766" -> null
  - region                           = "me-central-1" -> null
  - tags                             = {
    - "Name" = "dev-vpc"
  } -> null
  - tags_all                         = {
    - "Name" = "dev-vpc"
  } -> null
  # (4 unchanged attributes hidden)
}

```

```

- name: Wait for some time to ensure system readiness
  hosts: all
  tasks:
    - name: Wait 300 seconds for SSH to become ready
      wait_for:
        port: 22
        host: "{{ inventory_hostname }}"
        delay: 10
        timeout: 300
      delegate_to: localhost

```

```
---
```

```
@toobashafique065 [ /workspaces/terraform_machine (main) ] $ terraform apply
module.myapp-webserver[0].aws_key_pair.ssh-key: Refreshing state... [id=dev-serverkey-0]
aws_vpc.myapp_vpc: Refreshing state... [id=vpc-0205617480dd13552]
module.myapp-subnet.aws_internet_gateway.myapp_igw: Refreshing state... [id=igw-0804e2e546734638f]
module.myapp-subnet.aws_subnet.myapp_subnet_1: Refreshing state... [id=subnet-06457d2c9d8e6bde8]
module.myapp-webserver[0].aws_security_group.web_sg: Refreshing state... [id=sg-08f82ea6da5036e6c]
module.myapp-subnet.aws_default_route_table.main_rt: Refreshing state... [id=rtb-0674138a5d65be459]
module.myapp-webserver[0].aws_instance.myapp-server: Refreshing state... [id=i-09d4eabad356e65ec]
null_resource.configure_server: Refreshing state... [id=8789740229408547428]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the fol
-/+ destroy and then create replacement

Terraform will perform the following actions:

  # null_resource.configure_server is tainted, so must be replaced
-/+ resource "null_resource" "configure_server" {
    ~ id      = "8789740229408547428" -> (known after apply)
      # (1 unchanged attribute hidden)
}

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

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

null_resource.configure_server: Destroying... [id=8789740229408547428]
null_resource.configure_server: Destruction complete after 0s
null_resource.configure_server: Creating...
null_resource.configure_server: Provisioning with 'local-exec'...
null_resource.configure_server (local-exec): Executing: ["/bin/sh" "-c" "sleep 60\nANSIBLE_HOST_KEY_CHECKING=False ansible-
ate-key ~/.ssh/id_ed25519 \\n-u ec2-user \\nmy-playbook.yaml\n"]
```

40.172.100.31

Installation - Gitea: Git with a cup

Not secure

40.172.100.31:3000

Chat

Initial Configuration

If you run Gitea inside Docker, please read the [documentation](#) before changing any settings.

Database Settings

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

Database Type *

PostgreSQL

Host *

db:5432

Username *

gitea

Password *

.....

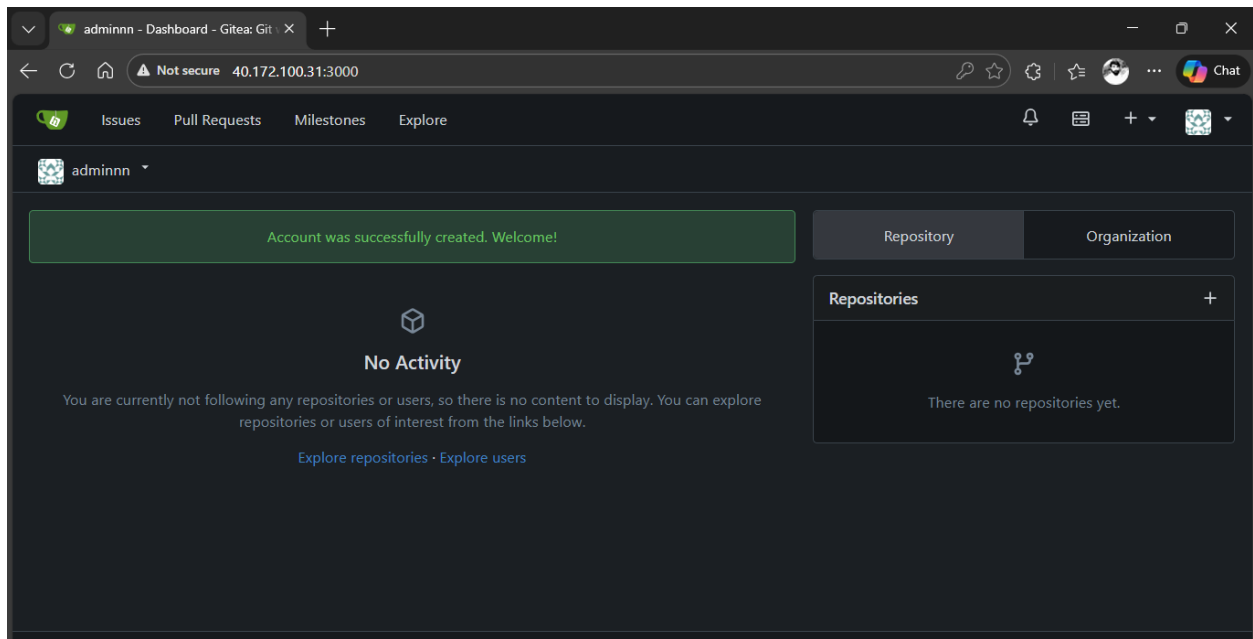
Database Name *

gitea

SSL *

Disable

Schema



Task 11 – Dynamic inventory with aws_ec2 plugin

```
Command Prompt - gh codespace ssh -c miniature-
GNU nano 7.2
[defaults]
host_key_checking = False
interpreter_python = /usr/bin/python3
deprecation_warnings = False

enable_plugins = aws_ec2
private_key_file = ~/.ssh/id_ed25519

Command Prompt - gh codespace ssh -c min
GNU nano 7.2
---
plugin: aws_ec2
regions:
  - me-central-1
```

```

module "myapp-subnet" {
  source = "./modules/subnet"
  vpc_id = aws_vpc.myapp_vpc.id
  subnet_cidr_block = var.subnet_cidr_block
  availability_zone = var.availability_zone
  env_prefix = var.env_prefix
  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
}

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
  count = 1
  instance_suffix = count.index
}

```

GNU nano 7.2

```

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]
}

```

```

null_resource.configure_server: Refreshing state... [id=6944963618505235679]
module.myapp-webserver[0].aws_key_pair.ssh-key: Refreshing state... [id=dev-serverkey-0]
aws_vpc.myapp_vpc: Refreshing state... [id=vpc-0205617480dd13552]
module.myapp-subnet.aws_subnet.myapp_subnet_1: Refreshing state... [id=subnet-06457d2c9d8e6bde8]
module.myapp-subnet.aws_internet_gateway.myapp_igw: Refreshing state... [id=igw-0804e2e546734638f]
module.myapp-webserver[0].aws_security_group.web_sg: Refreshing state... [id=sg-08f82ea6da5036e6c]
module.myapp-subnet.aws_default_route_table.main_rt: Refreshing state... [id=rtb-0674138a5d65be459]
module.myapp-webserver[0].aws_instance.myapp-server: Refreshing state... [id=i-09d4eabad356e65ec]

```

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:

```

+ create
~ update in-place
- destroy

```

Terraform will perform the following actions:

```

# aws_vpc.myapp_vpc will be updated in-place
~ resource "aws_vpc" "myapp_vpc" {
  ~ enable_dns_hostnames = false -> true
    id                   = "vpc-0205617480dd13552"
    tags                 = {
      "Name" = "dev-vpc"
    }
  # (19 unchanged attributes hidden)
}

# null_resource.configure_server will be destroyed
# (because null_resource.configure_server is not in configuration)
- resource "null_resource" "configure_server" {
  - id      = "6944963618505235679" -> null
  - triggers = {
    - "webserver_ip" = "40.172.100.31"
  } -> null
}

# module.myapp-webserver-prod[0].aws_instance.myapp-server will be created
+ resource "aws_instance" "myapp-server" {
  + ami      = "ami-05524d6658fcf35b6"
  + arn      = (known after apply)

```

Outputs:

```

prod-webserver_public_ips = [
  "3.29.113.101",
]
webserver_public_ips = [
  "40.172.100.31",
]

```

```

@terraform: ~ /workspaces/terraform_machine (main) $ $(which python) -m pip install boto3 botocore

```

```

Collecting boto3
  Downloading boto3-1.42.25-py3-none-any.whl.metadata (6.8 kB)
Collecting botocore
  Downloading botocore-1.42.25-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 boto3-1.42.25-py3-none-any.whl (140 kB)
Downloading botocore-1.42.25-py3-none-any.whl (14.6 MB)
14.6/14.6 MB 27.4 MB/s 0:00:00
Downloading jmespath-1.0.1-py3-none-any.whl (20 kB)
Downloading s3transfer-0.16.0-py3-none-any.whl (86 kB)
Installing collected packages: jmespath, botocore, s3transfer, boto3
Successfully installed boto3-1.42.25 botocore-1.42.25 jmespath-1.0.1 s3transfer-0.16.0

```

```

[WARNING]: Ansible is being run in a world writable directory (/workspaces/terraform_machine), ignoring it as an unsafe
cs.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.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.module_utils.common.text.converters instead.

```

```

[DEPRECATION WARNING]: Passing 'disable_lookups' to 'template' is deprecated. This feature will be removed from ansible.

```

```

@all:
|--@ungrouped:
|--@aws_ec2:
| |--ec2-40-172-100-31.me-central-1.compute.amazonaws.com
| |--ec2-3-29-113-101.me-central-1.compute.amazonaws.com

```


Task 12 – Filtering EC2 instances by tags & instance type

```
---
plugin: aws_ec2
regions:
  - me-central-1
keyed_groups:
  - key: tags
    prefix: tag
    separator: "_"
```

```
@toobashafique065 @ /workspaces/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 ans
cs.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 rem
ule_utils.common.text.converters instead.
[DEPRECATION WARNING]: Importing 'to_native' from 'ansible.module_utils._text' is deprecated. This feature will be r
odule_utils.common.text.converters instead.
[DEPRECATION WARNING]: Passing 'disable_lookups' to 'template' is deprecated. This feature will be removed from ansi
@all:
|--@ungrouped:
|--@aws_ec2:
| |--ec2-40-172-100-31.me-central-1.compute.amazonaws.com
| |--ec2-3-29-113-101.me-central-1.compute.amazonaws.com
|--@tag_Name_dev_ec2_instance_0:
| |--ec2-40-172-100-31.me-central-1.compute.amazonaws.com
|--@tag_Name_prod_ec2_instance_0:
| |--ec2-3-29-113-101.me-central-1.compute.amazonaws.com
```

```
---
plugin: aws_ec2
regions:
  - me-central-1
keyed_groups:
  - key: tags
    prefix: tag
    separator: "_"
  - key: instance_type
    prefix: instance_type
    separator: "_"
```

```
[WARNING]: Ansible is being run in a world writable directory (/workspaces/terraform_machine), ignoring it as an a
cs.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 r
ule_utils.common.text.converters instead.
[DEPRECATION WARNING]: Importing 'to_native' from 'ansible.module_utils._text' is deprecated. This feature will be
odule_utils.common.text.converters instead.
[DEPRECATION WARNING]: Passing `disable_lookups` to `template` is deprecated. This feature will be removed from an
@all:
|--@ungrouped:
|--@aws_ec2:
| |--ec2-40-172-100-31.me-central-1.compute.amazonaws.com
| |--ec2-3-29-113-101.me-central-1.compute.amazonaws.com
|--@tag_Name_dev_ec2_instance_0:
| |--ec2-40-172-100-31.me-central-1.compute.amazonaws.com
|--@instance_type_t3_micro:
| |--ec2-40-172-100-31.me-central-1.compute.amazonaws.com
|--@tag_Name_prod_ec2_instance_0:
| |--ec2-3-29-113-101.me-central-1.compute.amazonaws.com
|--@instance_type_t3_nano:
| |--ec2-3-29-113-101.me-central-1.compute.amazonaws.com
@toobashafique065 █ /workspaces/terraform_machine (main) $
```

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

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

    - name: Get IMDSv2 token
      uri:
```

```
@toobashafique065 [ /workspaces/terraform_machine (main) ] $ ansible-playbook my-playbook.yaml
```

```
PLAY [Configure nginx web server] *****
[WARNING]: Found variable using reserved name 'tags'.
Origin: <unknown>
```

```
tags
```

```
TASK [Gathering Facts] *****
ok: [ec2-3-29-113-101.me-central-1.compute.amazonaws.com]
ok: [ec2-40-172-100-31.me-central-1.compute.amazonaws.com]
```

```
TASK [install nginx and update cache] *****
changed: [ec2-40-172-100-31.me-central-1.compute.amazonaws.com]
changed: [ec2-3-29-113-101.me-central-1.compute.amazonaws.com]
```

```
TASK [install openssl] *****
ok: [ec2-40-172-100-31.me-central-1.compute.amazonaws.com]
ok: [ec2-3-29-113-101.me-central-1.compute.amazonaws.com]
```

```
TASK [start nginx server] *****
changed: [ec2-40-172-100-31.me-central-1.compute.amazonaws.com]
changed: [ec2-3-29-113-101.me-central-1.compute.amazonaws.com]
```

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

```
TASK [Gathering Facts] *****
ok: [ec2-40-172-100-31.me-central-1.compute.amazonaws.com]
ok: [ec2-3-29-113-101.me-central-1.compute.amazonaws.com]
```

```
TASK [Create SSL private directory] *****
changed: [ec2-40-172-100-31.me-central-1.compute.amazonaws.com]
changed: [ec2-3-29-113-101.me-central-1.compute.amazonaws.com]
```

```
TASK [Create SSL certs directory] *****
```

```
@toobashafique065 [ /workspaces/terraform_machine (main) ] $ ansible-playbook -l tag_Name_dev_* my-playbook.yaml
```

```
PLAY [Configure nginx web server] *****
[WARNING]: Found variable using reserved name 'tags'.
Origin: <unknown>
```

```
tags
```

```
TASK [Gathering Facts] *****
ok: [ec2-40-172-100-31.me-central-1.compute.amazonaws.com]
```

```
TASK [install nginx and update cache] *****
ok: [ec2-40-172-100-31.me-central-1.compute.amazonaws.com]
```

```
TASK [install openssl] *****
ok: [ec2-40-172-100-31.me-central-1.compute.amazonaws.com]
```

```
TASK [start nginx server] *****
ok: [ec2-40-172-100-31.me-central-1.compute.amazonaws.com]
```

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

```
TASK [Gathering Facts] *****
ok: [ec2-40-172-100-31.me-central-1.compute.amazonaws.com]
```

```
TASK [Create SSL private directory] *****
ok: [ec2-40-172-100-31.me-central-1.compute.amazonaws.com]
```

```
TASK [Create SSL certs directory] *****
ok: [ec2-40-172-100-31.me-central-1.compute.amazonaws.com]
```

```
TASK [Get IMDSv2 token] *****
ok: [ec2-40-172-100-31.me-central-1.compute.amazonaws.com]
```

```
TASK [Get current public IP] *****
ok: [ec2-40-172-100-31.me-central-1.compute.amazonaws.com]
```

```
@toobashafique065 [ ] /workspaces/terraform_machine (main) $ ansible-playbook -l tag_Name_prod_* my-playbook.yaml
```

```
PLAY [Configure nginx web server] *****
[WARNING]: Found variable using reserved name 'tags'.
Origin: <unknown>

tags

TASK [Gathering Facts] *****
ok: [ec2-3-29-113-101.me-central-1.compute.amazonaws.com]

TASK [install nginx and update cache] *****
ok: [ec2-3-29-113-101.me-central-1.compute.amazonaws.com]

TASK [install openssl] *****
ok: [ec2-3-29-113-101.me-central-1.compute.amazonaws.com]

TASK [start nginx server] *****
ok: [ec2-3-29-113-101.me-central-1.compute.amazonaws.com]

PLAY [Configure SSL certificates] *****

TASK [Gathering Facts] *****
ok: [ec2-3-29-113-101.me-central-1.compute.amazonaws.com]

TASK [Create SSL private directory] *****
ok: [ec2-3-29-113-101.me-central-1.compute.amazonaws.com]

TASK [Create SSL certs directory] *****
ok: [ec2-3-29-113-101.me-central-1.compute.amazonaws.com]

TASK [Get IMDSv2 token] *****
```

```
@toobashafique065 [ ] /workspaces/terraform_machine (main) $ ansible-playbook -l instance_type_t3_micro my-playbook.yaml
```

```
PLAY [Configure nginx web server] *****
[WARNING]: Found variable using reserved name 'tags'.
Origin: <unknown>

tags

TASK [Gathering Facts] *****
ok: [ec2-40-172-100-31.me-central-1.compute.amazonaws.com]

TASK [install nginx and update cache] *****
ok: [ec2-40-172-100-31.me-central-1.compute.amazonaws.com]

TASK [install openssl] *****
ok: [ec2-40-172-100-31.me-central-1.compute.amazonaws.com]

TASK [start nginx server] *****
ok: [ec2-40-172-100-31.me-central-1.compute.amazonaws.com]

PLAY [Configure SSL certificates] *****

TASK [Gathering Facts] *****
ok: [ec2-40-172-100-31.me-central-1.compute.amazonaws.com]

TASK [Create SSL private directory] *****
ok: [ec2-40-172-100-31.me-central-1.compute.amazonaws.com]

TASK [Create SSL certs directory] *****
ok: [ec2-40-172-100-31.me-central-1.compute.amazonaws.com]

TASK [Get IMDSv2 token] *****
ok: [ec2-40-172-100-31.me-central-1.compute.amazonaws.com]

TASK [Get current public IP] *****
ok: [ec2-40-172-100-31.me-central-1.compute.amazonaws.com]

TASK [Show current public IP] *****
```

```
@toobashafique065 @ /workspaces/terraform_machine (main) $ ansible-playbook -l instance_type_t3_nano my-playbook.yaml
```

```
PLAY [Configure nginx web server] *****
[WARNING]: Found variable using reserved name 'tags'.
Origin: <unknown>
```

```
tags
```

```
TASK [Gathering Facts] *****
ok: [ec2-3-29-113-101.me-central-1.compute.amazonaws.com]
```

```
TASK [install nginx and update cache] *****
ok: [ec2-3-29-113-101.me-central-1.compute.amazonaws.com]
```

```
TASK [install openssl] *****
ok: [ec2-3-29-113-101.me-central-1.compute.amazonaws.com]
```

```
TASK [start nginx server] *****
ok: [ec2-3-29-113-101.me-central-1.compute.amazonaws.com]
```

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

```
TASK [Gathering Facts] *****
ok: [ec2-3-29-113-101.me-central-1.compute.amazonaws.com]
```

```
TASK [Create SSL private directory] *****
ok: [ec2-3-29-113-101.me-central-1.compute.amazonaws.com]
```

```
TASK [Create SSL certs directory] *****
ok: [ec2-3-29-113-101.me-central-1.compute.amazonaws.com]
```

```
TASK [Get IMDSv2 token] *****
ok: [ec2-3-29-113-101.me-central-1.compute.amazonaws.com]
```

```
TASK [Get current public IP] *****
```

```
ec2-user@ip-10-0-10-167:~
```

```
GNU nano 7.2
```

```
[defaults]
```

```
inventory = ./inventory_aws_ec2.yaml
host_key_checking = False
deprecation_warnings = False
interpreter_python = /usr/bin/python3
remote_user = ec2-user
private_key_file = ~/.ssh/id_ed25519
```

```
@toobashafique065 @ /workspaces/terraform_machine (main) $ ansible-playbook -l instance_type_t3_nano my-playbook.yaml
```

```
PLAY [Configure nginx web server] *****
[WARNING]: Found variable using reserved name 'tags'.
Origin: <unknown>
```

```
tags
```

```
TASK [Gathering Facts] *****
ok: [ec2-3-29-113-101.me-central-1.compute.amazonaws.com]
```

```
TASK [install nginx and update cache] *****
ok: [ec2-3-29-113-101.me-central-1.compute.amazonaws.com]
```

```
TASK [install openssl] *****
ok: [ec2-3-29-113-101.me-central-1.compute.amazonaws.com]
```

```
TASK [start nginx server] *****
ok: [ec2-3-29-113-101.me-central-1.compute.amazonaws.com]
```

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

```
TASK [Gathering Facts] *****
ok: [ec2-3-29-113-101.me-central-1.compute.amazonaws.com]
```

```
TASK [Create SSL private directory] *****
ok: [ec2-3-29-113-101.me-central-1.compute.amazonaws.com]
```

```
TASK [Create SSL certs directory] *****
ok: [ec2-3-29-113-101.me-central-1.compute.amazonaws.com]
```

```
TASK [Get IMDSv2 token] *****
ok: [ec2-3-29-113-101.me-central-1.compute.amazonaws.com]
```

```
TASK [Get current public IP] *****
ok: [ec2-3-29-113-101.me-central-1.compute.amazonaws.com]
```

Task 13 – Ansible roles: nginx, ssl, webapp

GNU nano 7.2

```
provider "aws" {
  shared_config_files      = ["~/.aws/config"]
  shared_credentials_files = ["~/.aws/credentials"]
}

resource "aws_vpc" "myapp_vpc" {
  cidr_block      = var.vpc_cidr_block
  enable_dns_hostnames = true
  tags = {
    Name = "${var.env_prefix}-vpc"
  }
}

module "myapp-subnet" {
  source                = "./modules/subnet"
  vpc_id                = aws_vpc.myapp_vpc.id
  subnet_cidr_block     = var.subnet_cidr_block
  availability_zone     = var.availability_zone
  env_prefix            = var.env_prefix
  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

  count                = 1
  instance_suffix      = count.index
}
```

GNU nano 7.2

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

GNU nano 7.2

```
[nginx]
40.172.100.31

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



```
GNU nano 7.2
#SPDX-License-Identifier: MIT-0
---
# handlers file for nginx
- name: Restart nginx
  service:
    name: nginx
    state: restarted
```

```
- 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
```

```
---
- name: Deploy NGINX Web Stack
  hosts: nginx
  become: true
  roles:
    - nginx
```

```
@toobashafique065 @ /workspaces/terraform_machine/ansible (main) $ ansible-playbook -i inventory/hosts my-playbook.yaml
```

```
PLAY [Deploy NGINX Web Stack] *****
TASK [Gathering Facts] *****
ok: [40.172.100.31]

TASK [nginx : Install nginx] *****
ok: [40.172.100.31]

TASK [nginx : Install openssl] *****
ok: [40.172.100.31]

TASK [nginx : Start and enable nginx] *****
ok: [40.172.100.31]

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

Not secure https://40.172.100.31

Nginx Front End Web Server

Hostname: ip-10-0-10-239.me-central-1.compute.internal

Instance ID: i-09d4eabad356e65ec

Private IP: 10.0.10.239

Public IP: 40.172.100.31

Public DNS: https://N/A

Deployed: 2026-01-15 18:42:43

Status: ☒ Active and Running

```
Command Prompt - gh - codespace ssh - c miniature-
GNU nano 7.2
#SPDX-License-Identifier: MIT-0
---
# defaults file for ssl
imdsv2_token_ttl: "3600"
ssl_days_valid: 365
```

```

mode: '0700'

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

- name: Get IMDSv2 token
  uri:
    url: http://169.254.169.254/latest/api/token
    method: PUT
    headers:
      X-aws-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
    headers:
      X-aws-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/private/selfsigned.key
    -out /etc/ssl/certs/selfsigned.crt
    -subj "/CN={{ server_public_ip }}"
    -addext "subjectAltName=IP:{{ server_public_ip }}"
  args:
    creates: /etc/ssl/certs/selfsigned.crt

```

Command Prompt - gn codespace ssh -c min

```

GNU nano 7.2
#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

```

```
GNU nano 7.2 roles/webapp/files
<?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 IMDSv2 token
function getImdsV2Token() {
    $ch = curl_init("http://169.254.169.254/latest/api/token");
    curl_setopt_array($ch, [
        CURLOPT_RETURNTRANSFER => true,
        CURLOPT_CUSTOMREQUEST => "PUT",
        CURLOPT_HTTPHEADER => [
            "X-aws-ec2-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-ec2-metadata-token: $token"
        ],
        CURLOPT_TIMEOUT => 2
    ]);
```

```
Command Prompt - gh codespace ssh -c m
GNU nano 7.2
#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
```

```
user {{ nginx_user }};
worker_processes {{ nginx_worker_processes }};
error_log /var/log/nginx/error.log {{ nginx_error_log_level }};
pid /run/nginx.pid;

events {
    worker_connections {{ nginx_worker_connections }};
}

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 158.252.94.241:80;
        server 158.252.94.242:80 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 {{ web_root }};
            index {{ web_index_file }} index.html index.htm;
#            proxy_pass http://158.252.94.241:80;
```

GNU nano 7.2

#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

GNU nano 7.2

- name: Deploy NGINX Web Stack with SSL and PHP

hosts: nginx

become: true

roles:

- nginx
- ssl
- webapp

```
@toobashafique065 ▢ /workspaces/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: [40.172.100.31]

TASK [nginx : Install nginx] *****
ok: [40.172.100.31]

TASK [nginx : Install openssl] *****
ok: [40.172.100.31]

TASK [nginx : Start and enable nginx] *****
ok: [40.172.100.31]

TASK [ssl : Create SSL private directory] *****
ok: [40.172.100.31]

TASK [ssl : Create SSL certs directory] *****
ok: [40.172.100.31]

TASK [ssl : Get IMDSv2 token] *****
ok: [40.172.100.31]

TASK [ssl : Get public IP] *****
ok: [40.172.100.31]

TASK [ssl : Save public IP as fact] *****
ok: [40.172.100.31]

TASK [ssl : Generate self-signed certificate] *****
ok: [40.172.100.31]

TASK [webapp : Install PHP packages] *****
ok: [40.172.100.31]

TASK [webapp : Copy PHP website] *****
changed: [40.172.100.31]

TASK [webapp : Deploy nginx config] *****
changed: [40.172.100.31]
```

Not secure https://40.172.100.31

Nginx Front End Web Server

Hostname: ip-10-0-10-239.me-central-1.compute.internal


Instance ID: i-09d4eabad356e65ec

Private IP: 10.0.10.239

Public IP: 40.172.100.31

Public DNS: https://N/A

Deployed: 2026-01-15 19:04:46

Status:  Active and Running

Cleanup

```

module.myapp-webserver[0].aws_key_pair.ssh-key: Refreshing state... [id=dev-serverkey-0]
module.myapp-webserver-prod[0].aws_key_pair.ssh-key: Refreshing state... [id=prod-serverkey-0]
aws_vpc.myapp_vpc: Refreshing state... [id=vpc-0205617480dd13552]
module.myapp-webserver-prod[0].aws_instance.myapp-server: Refreshing state... [id=i-0b6aaae3206164312]
module.myapp-webserver-prod[0].aws_security_group.web_sg: Refreshing state... [id=sg-09f5e6ef555c4d948]
module.myapp-subnet.aws_internet_gateway.myapp_igw: Refreshing state... [id=igw-0804e2e546734638f]
module.myapp-subnet.aws_subnet.myapp_subnet_1: Refreshing state... [id=subnet-06457d2c9d8e6bde8]
module.myapp-webserver[0].aws_security_group.web_sg: Refreshing state... [id=sg-08f82ea6da5036e6c]
module.myapp-subnet.aws_default_route_table.main_rt: Refreshing state... [id=rtb-0674138a5d65be459]
module.myapp-webserver[0].aws_instance.myapp-server: Refreshing state... [id=i-09d4eabad356e65ec]

```

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:

- destroy

Terraform will perform the following actions:

```

# aws_vpc.myapp_vpc will be destroyed
- resource "aws_vpc" "myapp_vpc" {
  - arn                                = "arn:aws:ec2:me-central-1:535942231766:vpc/vpc-0205617480dd13552" -> null
  - assign_generated_ipv6_cidr_block  = false -> null
  - cidr_block                        = "10.0.0.0/16" -> null
  - default_network_acl_id            = "acl-0a0bb3e84d84e2c1c" -> null
  - default_route_table_id            = "rtb-0674138a5d65be459" -> null
  - default_security_group_id         = "sg-0f7d4fa1a045b46b4" -> null
  - dhcp_options_id                   = "dopt-02c0a1e8f3dd284d6" -> null
  - enable_dns_hostnames               = true -> null
  - enable_dns_support                 = true -> null
  - enable_network_address_usage_metrics = false -> null
  - id                                = "vpc-0205617480dd13552" -> null
  - instance_tenancy                  = "default" -> null
  - ipv6_netmask_length                = 0 -> null
  - main_route_table_id                = "rtb-0674138a5d65be459" -> null
  - owner_id                           = "535942231766" -> null
  - region                             = "me-central-1" -> null
  - tags                               = {
    - "Name" = "dev-vpc"
  } -> null
  - tags_all                             = {

```