

Fatima Jinnah Women University



PROJECT SUBMISSION

Automated Server Provisioning and Configuration System

Course Title : Cloud Computing

Submitted By :

Saira Ejaz
(2023-BSE-057)
Shumail Zahra
(2023-BSE-061)

Submitted To:

Prof. Waqas Saleem

Dated:

January 23, 2026

Department Of Software Engineering

Fatima Jinnah Women University

Project 2

Automated Server Provisioning and Configuration System

Repo Link:

<https://github.com/2023-BSE-057-Saira/Project-2-Automated-Servers>

1. Executive Summary

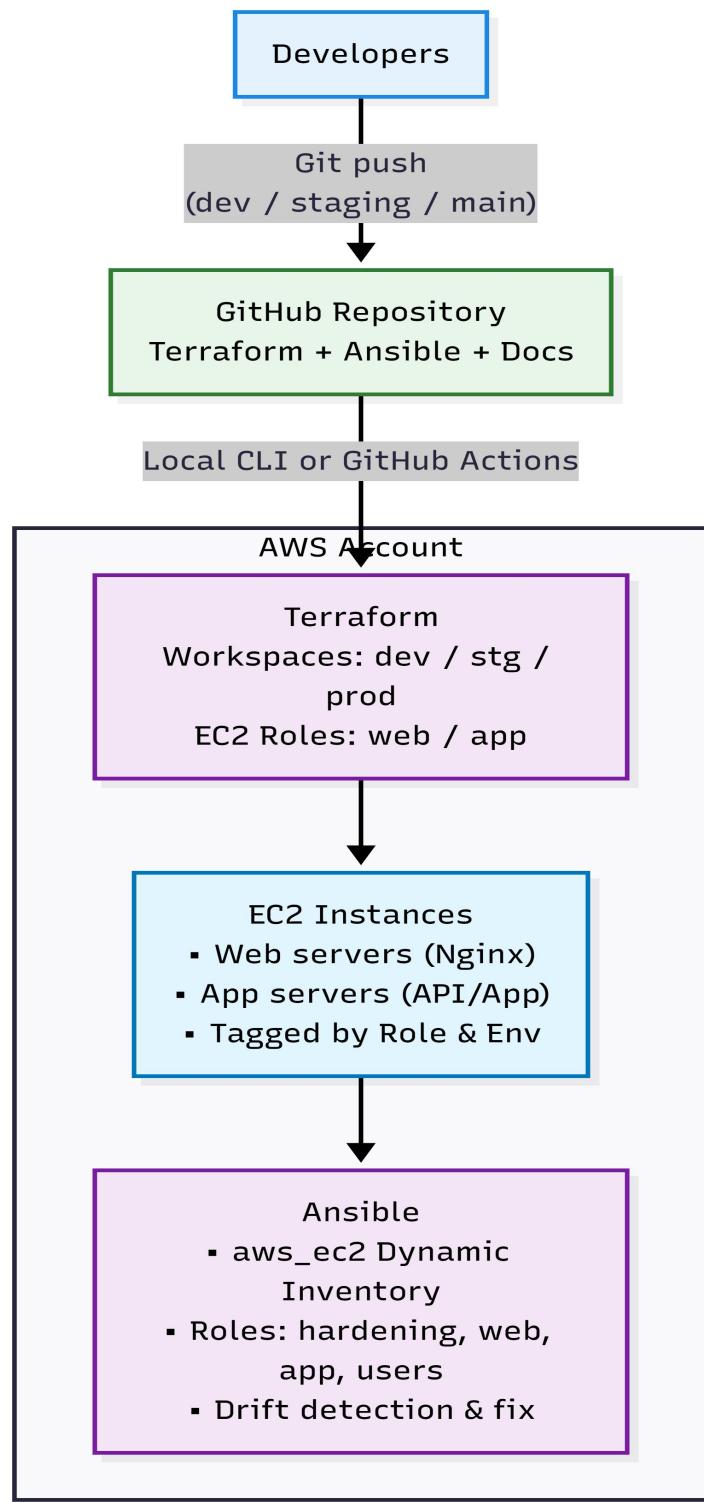
This project demonstrates the automation of server provisioning and configuration management using Terraform and Ansible. It provisions EC2 instances for web and application servers in AWS and applies configurations using Ansible roles. Configuration drift detection and remediation are implemented to maintain consistency and security. Terraform workspaces manage multiple environments such as dev, staging, and production. End-to-end validation ensures web and app services are running correctly and SSH security is enforced.

2. Architecture & Design

Terraform Workspaces & Infrastructure

Terraform is used to create AWS EC2 instances for web and application servers. Multiple workspaces (dev, staging, production) are used to separate environments, allowing independent deployments. Outputs from Terraform include public and private IP addresses for all servers.

The main Terraform files used are main.tf, variables.tf, outputs.tf, and backend.tf.



3. Implementation Details

Terraform initializes and applies infrastructure, generating outputs such as instance IDs and IP addresses. These outputs are used to configure Ansible inventory files. Ansible then connects to the servers, verifies connectivity, and configures web and application services. Both servers were successfully deployed and accessed.

Part 1: Git Repository Setup (15 marks)

1.1 Repository Structure (5 marks)

Tasks:

- . Create all directories and placeholder files.
- . Implement .gitignore for Terraform/Ansible/credentials (see 1.3).
- . Initialize Git repository and create an initial commit.

Deliverables:

Screenshot: project2_part1_repository_structure.png

```
● @2023-BSE-057-Saira → /workspaces/Project-2-Automated-Servers (main) $ tree -a -I ".git"
.
└── .github
    └── workflows
        ├── provision-dev.yml
        ├── provision-production.yml
        └── provision-staging.yml
.
└── .gitignore
.
└── README.md
.
└── ansible
    ├── ansible.cfg
    ├── group_vars
    │   └── all.yml
    └── inventory
        ├── dev_aws_ec2.yml
        ├── production_aws_ec2.yml
        └── staging_aws_ec2.yml
.
└── playbooks
    ├── configure-all.yml
    ├── detect-drift.yml
    └── remediate-drift.yml
.
└── roles
    ├── app
    │   ├── tasks
    │   │   └── main.yml
    │   └── templates
    │       └── app-config.j2
    ├── hardening
    │   ├── tasks
    │   │   └── main.yml
    │   └── templates
    │       └── sshd_config.j2
    └── vars
        └── main.yml
.
└── users
    └── tasks
```

```
@2023-BSE-057-Saira → /workspaces/Project-2-Automated-Servers (main) $ tree -a -I ".git"
.
├── main.yml
├── templates
│   └── sshd_config.j2
└── vars
    └── main.yml
.
├── users
│   └── tasks
│       └── main.yml
└── web
    ├── tasks
    │   └── main.yml
    └── templates
        └── nginx.conf.j2
.
└── app
    ├── api
    └── web
.
└── docs
    ├── architecture.md
    ├── drift-management.md
    ├── provisioning-guide.md
    └── troubleshooting.md
.
└── terraform
    ├── backend.tf
    ├── main.tf
    └── modules
        └── ec2-servers
            ├── main.tf
            ├── outputs.tf
            └── variables.tf
    ├── outputs.tf
    └── terraform.tfvars.example
    └── variables.tf

27 directories, 33 files
@2023-BSE-057-Saira → /workspaces/Project-2-Automated-Servers (main) $
```

Screenshot: project2_part1_gitignore.png

```
● @2023-BSE-057-Saira → /workspaces/Project-2-Automated-Servers (main) $ cat .gitignore
○ @2023-BSE-057-Saira → /workspaces/Project-2-Automated-Servers (main) $
```

Screenshot: project2_part1_initial_commit.png

```
● @2023-BSE-057-Saira → /workspaces/Project-2-Automated-Servers (main) $ git commit -m "Initial project structure for automated server provisioning"
[main 0c2480f] Initial project structure for automated server provisioning
33 files changed, 1 deletion(-)
create mode 100644 .github/workflows/provision-dev.yml
create mode 100644 .github/workflows/provision-production.yml
create mode 100644 .github/workflows/provision-staging.yml
create mode 100644 .gitignore
create mode 100644 ansible/ansible.cfg
create mode 100644 ansible/group_vars/all.yml
create mode 100644 ansible/inventory/dev_aws_ec2.yml
create mode 100644 ansible/inventory/production_aws_ec2.yml
create mode 100644 ansible/inventory/staging_aws_ec2.yml
create mode 100644 ansible/playbooks/configure-all.yml
create mode 100644 ansible/playbooks/detect-drift.yml
create mode 100644 ansible/playbooks/remediate-drift.yml
create mode 100644 ansible/roles/app/tasks/main.yml
create mode 100644 ansible/roles/app/templates/app-config.j2
create mode 100644 ansible/roles/hardening/tasks/main.yml
create mode 100644 ansible/roles/hardening/templates/sshd_config.j2
create mode 100644 ansible/roles/hardening/vars/main.yml
create mode 100644 ansible/roles/users/tasks/main.yml
create mode 100644 ansible/roles/web/tasks/main.yml
create mode 100644 ansible/roles/web/templates/nginx.conf.j2
create mode 100644 docs/architecture.md
create mode 100644 docs/drift-management.md
create mode 100644 docs/provisioning-guide.md
create mode 100644 docs/troubleshooting.md
create mode 100644 terraform/backend.tf
create mode 100644 terraform/main.tf
create mode 100644 terraform/modules/ec2-servers/main.tf
create mode 100644 terraform/modules/ec2-servers/outputs.tf
create mode 100644 terraform/modules/ec2-servers/variables.tf
create mode 100644 terraform/outputs.tf
create mode 100644 terraform/terraform.tfvars.example
create mode 100644 terraform/variables.tf
● @2023-BSE-057-Saira → /workspaces/Project-2-Automated-Servers (main) $ git push -u origin main
```

1.2 Git Branching Strategy (5 marks)

Tasks:

- . Create dev and staging branches from main.
- . Configure branch protection rules on main and staging.
- . Document branching strategy in README.md.
- . Create a sample feature branch (e.g., feature/add-ssh-hardening).

Deliverables:

Screenshot: project2_part1_git_branches.png

The screenshot shows the GitHub interface for the 'Project-2-Automated-Servers' repository. The 'Code' tab is selected. In the 'Branches' section, there are two tables: 'Default' and 'Your branches'. The 'Default' table contains one row for 'main'. The 'Your branches' table contains four rows: 'production', 'feature/add-ssh-hardening', 'dev', and 'staging'. Below these tables is a terminal window showing the following command history:

- @2023-BSE-057-Saira →/workspaces/Project-2-Automated-Servers (main) \$ git checkout main
Already on 'main'
Your branch is up to date with 'origin/main'.
- @2023-BSE-057-Saira →/workspaces/Project-2-Automated-Servers (main) \$ git checkout -b staging
Switched to a new branch 'staging'
- @2023-BSE-057-Saira →/workspaces/Project-2-Automated-Servers (staging) \$ git checkout -b dev
Switched to a new branch 'dev'
- @2023-BSE-057-Saira →/workspaces/Project-2-Automated-Servers (dev) \$ git checkout -b feature/add-ssh-hardening
Switched to a new branch 'feature/add-ssh-hardening'
- @2023-BSE-057-Saira →/workspaces/Project-2-Automated-Servers (feature/add-ssh-hardening) \$ git branch
 dev
* feature/add-ssh-hardening
 main
 staging
- @2023-BSE-057-Saira →/workspaces/Project-2-Automated-Servers (feature/add-ssh-hardening) \$ █

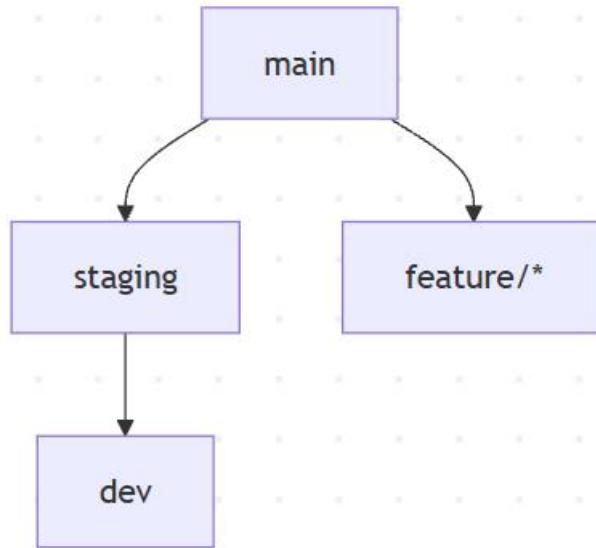
Screenshot: project2_part1_branch_protection.png

The screenshot shows the GitHub repository settings for 'Project-2-Automated-Servers'. In the 'Branch protection rules' section, there are two rules defined:

- main**: Currently applies to 1 branch. Buttons for Edit and Delete.
- staging**: Currently applies to 1 branch. Buttons for Edit and Delete.

The left sidebar shows navigation options like General, Access, Collaborators, Moderation options, Branches, Tags, Rules, Actions, Models, Webhooks, Copilot, Environments, Codespaces, and Pages.

Screenshot: project2_part1_branching_diagram.png



```

● @23-22411-057-sudo →/workspaces/Project-2-Automated-Servers (production) $ git checkout dev
Switched to branch 'dev'
Your branch is up to date with 'origin/dev'.
● @23-22411-057-sudo →/workspaces/Project-2-Automated-Servers (dev) $ git checkout -b feature/add-ssh-hardening
Switched to a new branch 'feature/add-ssh-hardening'
● @23-22411-057-sudo →/workspaces/Project-2-Automated-Servers (feature/add-ssh-hardening) $ git push -u origin feature/add-ssh-hardening
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'feature/add-ssh-hardening' on GitHub by visiting:
remote:   https://github.com/23-22411-057-sudo/Project-2-Automated-Servers/pull/new/feature/add-ssh-hardening
remote:
To https://github.com/23-22411-057-sudo/Project-2-Automated-Servers
 * [new branch]      feature/add-ssh-hardening -> feature/add-ssh-hardening
branch 'feature/add-ssh-hardening' set up to track 'origin/feature/add-ssh-hardening'.
● @23-22411-057-sudo →/workspaces/Project-2-Automated-Servers (feature/add-ssh-hardening) $ 
  
```

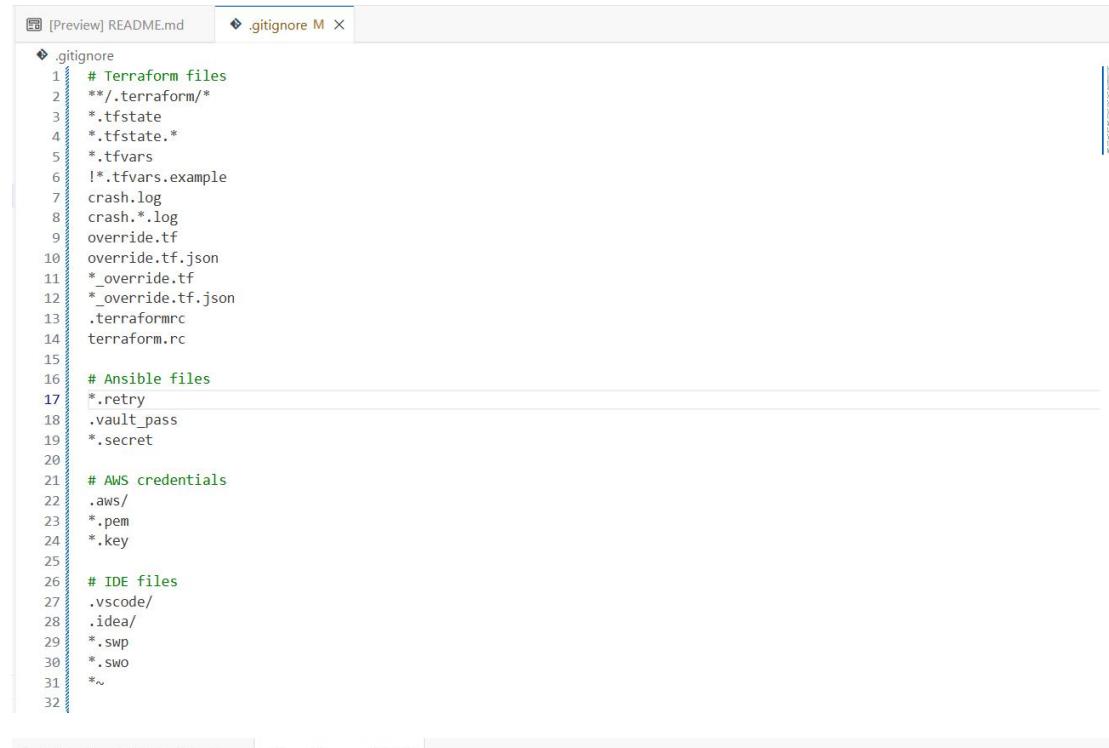
1.3 .gitignore Configuration (5 marks)

Tasks:

- . Add .gitignore with the rules above.
- . Verify no sensitive files or state are committed.
- . Document key ignore rules in README.md.

Deliverables:

Screenshot: project2_part1_gitignore_content.png



```

. [Preview] README.md .gitignore M X
◆ .gitignore
1 # Terraform files
2 **/.terraform/*
3 *.tfstate
4 *.tfstate.*
5 *.tfvars
6 !*.tfvars.example
7 crash.log
8 crash.*.log
9 override.tf
10 override.tf.json
11 *_override.tf
12 *_override.tf.json
13 .terraformrc
14 terraform.rc
15
16 # Ansible files
17 *.retry
18 .vault_pass
19 *.secret
20
21 # AWS credentials
22 .aws/
23 *.pem
24 *.key
25
26 # IDE files
27 .vscode/
28 .idea/
29 *.swp
30 *.swo
31 */
32

. [Preview] README.md .gitignore M X
◆ .gitignore
26 # IDE files
27 .vscode/
28 .idea/
29 *.swp
30 *.swo
31 */
32
33 # OS files
34 .DS_Store
35 Thumbs.db
36
37 # Logs
38 *.log
39 logs/
40
41 # Environment files
42 .env
43 .env.local
44
45 # Backup files
46 *.backup
47 *.bak
48
49 # Temporary files
50 tmp/
51 temp/
52

```

Screenshot: project2_part1_git_status_clean.png

```
● @2023-BSE-057-Saira →/workspaces/Project-2-Automated-Servers (production) $ git status
On branch production
Your branch is up to date with 'origin/production'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
    (use "git restore <file>..." to discard changes in working directory)
      modified:   .gitignore
      modified:   README.md

no changes added to commit (use "git add" and/or "git commit -a")
○ @2023-BSE-057-Saira →/workspaces/Project-2-Automated-Servers (production) $
```

Part 2: Terraform Infrastructure (30 marks)

2.1 EC2 Servers Module (15 marks)

Tasks:

- . Implement web and app EC2 resources.
- . Tag instances with Project, Environment, and Role.
- . Output instance IDs and IPs.

Deliverables:

Screenshot: project2_part2_ec2_module_variables.png

```
GNU nano 7.2                                     terraform/modules/ec2-servers/variables.tf
variable "environment" {
  description = "Environment name (dev, staging, production)"
  type        = string
}

variable "project_name" {
  description = "Project name for tagging"
  type        = string
  default     = "project2-server-provisioning"
}

variable "web_instance_count" {
  description = "Number of web servers"
  type        = number
  default     = 1
}

variable "app_instance_count" {
  description = "Number of application servers"
  type        = number
  default     = 1
}

variable "instance_type" {
  description = "EC2 instance type"
  type        = string
  default     = "t3.micro"
}

variable "ami_id" {
  description = "Base AMI ID for instances"
  type        = string
}
```

```

GNU nano 7.2                                     terraform/modules/ec2-servers/variables.tf
description = "EC2 instance type"
type        = string
default     = "t3.micro"
}

variable "ami_id" {
  description = "Base AMI ID for instances"
  type        = string
}

variable "key_name" {
  description = "SSH key pair name"
  type        = string
}

variable "subnet_id" {
  description = "Subnet ID for instances"
  type        = string
}

variable "security_group_ids" {
  description = "List of security group IDs"
  type        = list(string)
  default     = []
}

variable "tags" {
  description = "Additional tags"
  type        = map(string)
  default     = {}
}

```

Screenshot: project2_part2_ec2_module_main.png

```

GNU nano 7.2                                     terraform/modules/ec2-servers/main.tf *
# Web servers
resource "aws_instance" "web" {
  count      = var.web_instance_count
  ami        = var.ami_id
  instance_type = var.instance_type
  subnet_id   = var.subnet_id
  key_name    = var.key_name

  vpc_security_group_ids = var.security_group_ids

  tags = merge(
    {
      Name      = "${var.project_name}-${var.environment}-web-${count.index}"
      Project   = var.project_name
      Environment = var.environment
      Role      = "web"
      ManagedBy = "Terraform"
    },
    var.tags
  )
}

# Application servers
resource "aws_instance" "app" {
  count      = var.app_instance_count
  ami        = var.ami_id
  instance_type = var.instance_type
  subnet_id   = var.subnet_id
  key_name    = var.key_name

  vpc_security_group_ids = var.security_group_ids

```

```

tags = merge(
  {
    Name      = "${var.project_name}-${var.environment}-app-${count.index}"
    Project   = var.project_name
    Environment = var.environment
    Role      = "app"
    ManagedBy = "Terraform"
  },
  var.tags
)

```

Screenshot: project2_part2_ec2_module_outputs.png

```
GNU nano 7.2                                         terraform/modules/ec2-servers/outputs.tf *
```

```
output "web_instance_ids" {
  description = "IDs of web instances"
  value       = aws_instance.web["*"].id
}

output "web_private_ips" {
  description = "Private IPs of web instances"
  value       = aws_instance.web["*"].private_ip
}

output "app_instance_ids" {
  description = "IDs of application instances"
  value       = aws_instance.app["*"].id
}

output "app_private_ips" {
  description = "Private IPs of application instances"
  value       = aws_instance.app["*"].private_ip
}
```

2.2 Root Terraform & Workspaces (15 marks)

Tasks:

- . Configure Terraform with AWS provider and default tags.
- . Implement workspace-dependent configuration via local.env_config.
- . Output summary information (counts, IPs).

Deliverables:

Screenshot: project2_part2_main_tf.png

```
GNU nano 7.2                                         terraform/main.tf *
```

```
terraform {
  required_version = ">= 1.0"

  required_providers {
    aws = {
      source  = "hashicorp/aws"
      version = "~> 5.0"
    }
  }
}

provider "aws" {
  region = var.aws_region

  default_tags {
    tags = {
      Project      = var.project_name
      Environment = terraform.workspace
      ManagedBy   = "Terraform"
    }
  }
}

locals {
  workspace = terraform.workspace

  env_config = lookup(
    {
      dev = {
        environment      = "dev"
        web_instance_count = 1
        app_instance_count = 1
      }
    }
  )
}
```

```

GNU nano 7.2                                         terraform/main.tf *
dev = {
    environment      = "dev"
    web_instance_count = 1
    app_instance_count = 1
    instance_type     = "t3.micro"
}
staging = {
    environment      = "staging"
    web_instance_count = 2
    app_instance_count = 2
    instance_type     = "t3.micro"
}
production = {
    environment      = "production"
    web_instance_count = 3
    app_instance_count = 3
    instance_type     = "t3.small"
}
},
local.workspace
}

module "ec2_servers" {
source = "./modules/ec2-servers"

environment      = local.env_config.environment
project_name     = var.project_name
web_instance_count = local.env_config.web_instance_count
app_instance_count = local.env_config.app_instance_count
instance_type     = local.env_config.instance_type

```

```

GNU nano 7.2                                         terraform/main.tf *
module "ec2_servers" {
source = "./modules/ec2-servers"

environment      = local.env_config.environment
project_name     = var.project_name
web_instance_count = local.env_config.web_instance_count
app_instance_count = local.env_config.app_instance_count
instance_type     = local.env_config.instance_type

ami_id           = var.ami_id
key_name         = var.key_name
subnet_id        = var.subnet_id
security_group_ids = var.security_group_ids
}

# Output instance IDs & IPs
output "web_instance_ids" {
    value = module.ec2_servers.web_instance_ids
}

output "app_instance_ids" {
    value = module.ec2_servers.app_instance_ids
}

output "web_private_ips" {
    value = module.ec2_servers.web_private_ips
}

output "app_private_ips" {
    value = module.ec2_servers.app_private_ips
}

```

Screenshot: project2_part2_variables_tf.png

```

GNU nano 7.2
variable "aws_region" {
  description = "AWS region to deploy resources"
  type        = string
  default     = "us-east-1"
}

variable "project_name" {
  description = "Project name for tagging"
  type        = string
  default     = "project2-server-provisioning"
}

variable "ami_id" {
  description = "Base AMI ID for EC2 instances"
  type        = string
}

variable "key_name" {
  description = "SSH key name to use for EC2"
  type        = string
}

variable "subnet_id" {
  description = "Subnet ID for EC2 instances"
  type        = string
}

variable "security_group_ids" {
  description = "List of security group IDs for EC2 instances"
  type        = list(string)
  default     = []
}

```

Screenshot: project2_part2_terraform_output.png

```

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

GNU nano 7.2                                         outputs.tf *
output "web_instance_ids" {
  description = "IDs of web instances"
  value       = module.ec2_servers.web_instance_ids
}

output "web_private_ips" {
  description = "Private IPs of web instances"
  value       = module.ec2_servers.web_private_ips
}

output "app_instance_ids" {
  description = "IDs of application instances"
  value       = module.ec2_servers.app_instance_ids
}

output "app_private_ips" {
  description = "Private IPs of application instances"
  value       = module.ec2_servers.app_private_ips
}

```

1. Dev Workspace

Heading: Dev Workspace - Terraform Output

Description:

This workspace provisions a single web and a single application EC2 instance for development.

The instances use t3.nano type to fit under our vCPU quota.

Private IPs and instance IDs are shown below.

```

@2023-BSE-057-Saira →/workspaces/Project-2-Automated-Servers/terraform (production) $ terraform plan -var-file="terraform.tfvars"
+ ebs_block_device (known after apply)
+ enclave_options (known after apply)
+ ephemeral_block_device (known after apply)
+ instance_market_options (known after apply)
+ maintenance_options (known after apply)
+ metadata_options (known after apply)
+ network_interface (known after apply)
+ private_dns_name_options (known after apply)
+ root_block_device (known after apply)
}

Plan: 2 to add, 0 to change, 0 to destroy.

Changes to Outputs:
+ app_instance_ids = [
  + (known after apply),
]
+ app_private_ips = [
  + (known after apply),
]
+ web_instance_ids = [
  + (known after apply),
]
+ web_private_ips = [
  + (known after apply),
]

@2023-BSE-057-Saira →/workspaces/Project-2-Automated-Servers/terraform (production) $ terraform apply -auto-approve -var-file="terraform.tfvars"
+ (known after apply),
]
+ app_private_ips = [
  + (known after apply),
]
+ web_instance_ids = [
  + (known after apply),
]
+ web_private_ips = [
  + (known after apply),
]
module.ec2_servers.aws_instance.app[0]: Creating...
module.ec2_servers.aws_instance.web[0]: Creating...
module.ec2_servers.aws_instance.app[0]: Still creating... [0m10s elapsed]
module.ec2_servers.aws_instance.web[0]: Still creating... [0m10s elapsed]
module.ec2_servers.aws_instance.web[0]: Creation complete after 12s [id=i-0f0a7f51fb7bcdca]
module.ec2_servers.aws_instance.app[0]: Creation complete after 12s [id=i-0e3fb7295a2258126]

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

Outputs:

app_instance_ids = [
  "i-0e3fb7295a2258126",
]
app_private_ips = [
  "172.31.1.183",
]
web_instance_ids = [
  "i-0f0a7f51fb7bcdca",
]
web_private_ips = [
  "172.31.1.44",
]

```

2. Staging Workspace

Heading: Staging Workspace - Terraform Output

Description:

Staging workspace provisions 2 web and 2 app instance for pre-production validation.
Used to test combined configuration and ensure automation works as expected.

IPs and instance IDs are displayed below.

- @2023-BSE-057-Saira → /workspaces/Project-2-Automated-Servers/terraform (production) \$ terraform workspace select staging
Switched to workspace "staging".
- @2023-BSE-057-Saira → /workspaces/Project-2-Automated-Servers/terraform (production) \$ terraform plan -var-file="terraform.tfvars"

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:

```
# module.ec2_servers.aws_instance.app[0] will be created
+ resource "aws_instance" "app" {
  + ami = "ami-00c08fcaceeb2de00b"
  + arn = (known after apply)
  + associate_public_ip_address = (known after apply)
  + availability_zone = (known after apply)
  + cpu_core_count = (known after apply)
  + cpu_threads_per_core = (known after apply)
  + disable_api_stop = (known after apply)
  + disable_api_termination = (known after apply)
  + ebs_optimized = (known after apply)
  + enable_primary_ipv6 = (known after apply)
  + get_password_data = false
  + host_id = (known after apply)
  + host_resource_group_arn = (known after apply)
  + iam_instance_profile = (known after apply)
  + id = (known after apply)
  + instance_initiated_shutdown_behavior = (known after apply)
  + instance_lifecycle = (known after apply)
  + instance_state = (known after apply)
  + instance_type = "t3.micro"
  + ipv6_address_count = (known after apply)
  + ipv6_addresses = (known after apply)
  + key_name = "myproject-ssh-key"
  + monitoring = (known after apply)
  + outpost_arn = (known after apply)
  + password_data = (known after apply)
}
```

induction* ⌂ ⊗ 0 △ 0 ⚡ 0

```
@2023-BSE-057-Saira → /workspaces/Project-2-Automated-Servers/terraform (production) $ terraform apply -auto-approve -var-file="terraform.tfvars"
]
module.ec2_servers.aws_instance.app[0]: Creating...
module.ec2_servers.aws_instance.web[0]: Creating...
module.ec2_servers.aws_instance.web[1]: Creating...
module.ec2_servers.aws_instance.app[1]: Creating...
module.ec2_servers.aws_instance.app[0]: Still creating... [00m10s elapsed]
module.ec2_servers.aws_instance.web[0]: Still creating... [00m10s elapsed]
module.ec2_servers.aws_instance.web[1]: Still creating... [00m10s elapsed]
module.ec2_servers.aws_instance.app[1]: Still creating... [00m10s elapsed]
module.ec2_servers.aws_instance.web[0]: Creation complete after 12s [id=i-0c990b6794ba80cb0]
module.ec2_servers.aws_instance.app[0]: Creation complete after 13s [id=i-04cb9e7a5651e0ba2]
module.ec2_servers.aws_instance.app[1]: Creation complete after 13s [id=i-079d383b0ce100ac6]
module.ec2_servers.aws_instance.web[1]: Creation complete after 13s [id=i-06b7f185f8fd98c02]
```

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

Outputs:

```
app_instance_ids = [
  "i-04cb9e7a5651e0ba2",
  "i-079d383b0ce100ac6"
]
app_private_ips = [
  "172.31.12.182",
  "172.31.3.104"
]
web_instance_ids = [
  "i-0c990b6794ba80cb0",
  "i-06b7f185f8fd98c02"
]
web_private_ips = [
  "172.31.14.96",
  "172.31.10.51"
]
```

o @2023-BSE-057-Saira → /workspaces/Project-2-Automated-Servers/terraform (production) \$ []

3. Production Workspace

Heading: Production Workspace - Terraform Output

Description:

Production workspace provisions 1 web and 1 app instance for live environment.

Minimal instance counts are used due to vCPU quota limits.

Private IPs and instance IDs are listed

below.

```
● @23-22411-057-sudo →/workspaces/Project-2-Automated-Servers/terraform (main) $ terraform workspace select production
Switched to workspace "production".
● @23-22411-057-sudo →/workspaces/Project-2-Automated-Servers/terraform (main) $ terraform plan -var-file=terraform.tfvars
```

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:

```
# module.ec2_servers.aws_instance.app[0] will be created
+ resource "aws_instance" "app" {
  + ami                               = "ami-0e45d11b390730530"
  + arn                             = (known after apply)
  + associate_public_ip_address      = (known after apply)
  + availability_zone                = (known after apply)
  + cpu_core_count                  = (known after apply)
  + cpu_threads_per_core            = (known after apply)
  + disable_api_stop                = (known after apply)
  + disable_api_termination         = (known after apply)
  + ebs_optimized                   = (known after apply)
  + enable_primary_ipv6              = (known after apply)
  + get_password_data               = false
  + host_id                          = (known after apply)
  + host_resource_group_arn          = (known after apply)
  + iam_instance_profile             = (known after apply)
  + id                                = (known after apply)
  + instance_initiated_shutdown_behavior = (known after apply)
  + instance_lifecycle               = (known after apply)
  + instance_state                  = (known after apply)
  + instance_type                   = "t3.small"
  + ipv6_address_count              = (known after apply)
  + ipv6_addresses                  = (known after apply)
  + key_name                         = "project2-key"
  + monitoring                       = (known after apply)
  + outpost_arn                      = (known after apply)
  + password_data                   = (known after apply)
```

```
@2023-BSE-057-Saira →/workspaces/Project-2-Automated-Servers/terraform (production) $ terraform apply -auto-approve -var-file="terraform.tfvars"
module.ec2_servers.aws_instance.app[0]: Refreshing state... [id=i-04e4ce6e23235004c]
```

Changes to Outputs:
+ app_instance_ids = [
 + "i-04e4ce6e23235004c",
]
+ app_private_ips = [
 + "172.31.11.210",
]
+ web_instance_ids = [
 + "i-03612104105a12321",
]
+ web_private_ips = [
 + "172.31.13.100",
]

You can apply this plan to save these new output values to the Terraform state, without changing any real infrastructure.

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

Outputs:

```
app_instance_ids = [  
  "i-04e4ce6e23235004c",  
]  
app_private_ips = [  
  "172.31.11.210",  
]  
web_instance_ids = [  
  "i-03612104105a12321",  
]  
web_private_ips = [  
  "172.31.13.100",  
]
```

```
○ @2023-BSE-057-Saira →/workspaces/Project-2-Automated-Servers/terraform (production) $
```

4. Ansible Roles & Playbooks

Ansible automates configuration management with the following roles:

hardening: Applies SSH hardening, installs required packages.

users: Creates the deployer user with SSH key access.

web: Installs Nginx and deploys a test web page.

app: Deploys application code and creates a systemd service.

Key playbooks:

configure-all.yml: Applies all roles.

detect-drift.yml: Checks for configuration drift, including service status and SSH configuration.

remediate-drift.yml: Restores the system to the desired configuration if drift is detected.

Part 3: Ansible Configuration & Dynamic Inventory (25 marks)

3.1 Ansible Configuration and Dynamic Inventory (10 marks)

Tasks:

- . Create dynamic inventory files for dev, staging, production.
- . Configure ansible.cfg.
- . Create group_vars/all.yml with reasonable defaults.

Deliverables:

Screenshot: project2_part3_ansible_inventory_dev.png

```
GNU nano 7.2                                         ansible/inventory/dev aws_ec2.yml *
plugin: aws_ec2
regions:
  - me-central-1
filters:
  tag:Project: "project2-server-provisioning"
  tag:Environment: "dev"
keyed_groups:
  - key: tags.Role
    prefix: role
hostnames:
  - private-ip-address
compose:
  ansible_host: private_ip_address
```

```
GNU nano 7.2                                         ansible/inventory/staging aws_ec2.yml *
plugin: aws_ec2
regions:
  - me-central-1
filters:
  tag:Project: "project2-server-provisioning"
  tag:Environment: "staging"
keyed_groups:
  - key: tags.Role
    prefix: role
hostnames:
  - private-ip-address
compose:
  ansible_host: private_ip_address
```

```
GNU nano 7.2                                         ansible/inventory/production aws_ec2.yml
plugin: aws_ec2
regions:
  - me-central-1
filters:
  tag:Project: "project2-server-provisioning"
  tag:Environment: "production"
keyed_groups:
  - key: tags.Role
    prefix: role
hostnames:
  - private-ip-address
compose:
  ansible_host: private_ip_address
```

Screenshot: project2_part3_ansible_cfg.png

```
GNU nano 7.2                                         ansible/ansible.cfg *
```

```
[defaults]
inventory = inventory/dev_aws_ec2.yml
host_key_checking = False
retry_files_enabled = False
gathering = smart
stdout_callback = yaml
roles_path = roles

[privilegeEscalation]
become = True
become_method = sudo

[sshConnection]
pipelining = True
```

Screenshot: project2_part3_group_vars_all.png

```
GNU nano 7.2                                         ansible/group_vars/all.yml *
```

```
---
common_packages:
  - vim
  - git
  - curl

default_users:
  - name: deployer
    groups: sudo
    shell: /bin/bash

ssh_hardening:
  permit_root_login: "no"
  password_auth: "no"

nginx_listen_port: 80
app_service_name: "sample-app"
```

Ansible Dynamic Inventory for All Environments

Description:

This dynamic inventory uses the aws_ec2 plugin to automatically discover EC2 instances for dev, staging, and production environments based on tags.

Hosts are grouped by their Role tags (web or app), and connectivity is verified using the Ansible ping module.

The inventory ensures that playbooks can run seamlessly across all environments without manually listing hosts.

```
(.venv) @2023-BSE-057-Saira →/workspaces/Project-2-Automated-Servers (main) $ ANSIBLE_CONFIG=/workspaces/Project-2-Automated-Servers/ansible/ansible.cfg ansible all -i /workspaces/Project-2-Automated-Servers/ansible/inventory/dev_aws_ec2.yml -m ping
[WARNING]: Host '172.31.1.183' 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.
172.31.1.183 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3.9"
  },
  "changed": false,
  "ping": "pong"
}
[WARNING]: Host '172.31.1.44' 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.
172.31.1.44 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3.9"
  },
  "changed": false,
  "ping": "pong"
}
(.venv) @2023-BSE-057-Saira →/workspaces/Project-2-Automated-Servers (main) $
```

```

● (.venv) @2023-BSE-057-Saira →/workspaces/Project-2-Automated-Servers (main) $ ANSIBLE_CONFIG=/workspaces/Project-2-Automated-Servers/ansible/ansible.cfg \
ansible all -i /workspaces/Project-2-Automated-Servers/ansible/inventory/staging_aws_ec2.yml -m ping
[WARNING]: Host '172.31.2.20' 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.
172.31.2.20 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3.9"
    },
    "changed": false,
    "ping": "pong"
}
[WARNING]: Host '172.31.2.10' 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.
172.31.2.10 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3.9"
    },
    "changed": false,
    "ping": "pong"
}
○ (.venv) @2023-BSE-057-Saira →/workspaces/Project-2-Automated-Servers (main) $

● (.venv) @2023-BSE-057-Saira →/workspaces/Project-2-Automated-Servers (main) $ ANSIBLE_CONFIG=/workspaces/Project-2-Automated-Servers/ansible/ansible.cfg \
ansible all -i /workspaces/Project-2-Automated-Servers/ansible/inventory/production_aws_ec2.yml -m ping
[WARNING]: Host '172.31.3.20' 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.
172.31.3.20 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3.9"
    },
    "changed": false,
    "ping": "pong"
}
[WARNING]: Host '172.31.3.10' 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.
172.31.3.10 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3.9"
    },
    "changed": false,
    "ping": "pong"
}
○ (.venv) @2023-BSE-057-Saira →/workspaces/Project-2-Automated-Servers (main) $

```

3.2 Ansible Roles for Hardening, Web, App, Users (15 marks)

Tasks:

.Implement roles: hardening, users, web, app.

.Make roles idempotent (re-runs should not fail or re-change unnecessarily).

.Verify that web servers and app servers are correctly configured by role and hardened.

Deliverables:

Screenshot: project2_part3_roles_structure.png

```

● (.venv) @2023-BSE-057-Saira →/workspaces/Project-2-Automated-Servers (main) $ cd /workspaces/Project-2-Automated-Servers/ansible
● (.venv) @2023-BSE-057-Saira →/workspaces/Project-2-Automated-Servers/ansible (main) $ cd /workspaces/Project-2-Automated-Servers/ansible
● (.venv) @2023-BSE-057-Saira →/workspaces/Project-2-Automated-Servers/ansible (main) $ tes,vars
● bash: tes,vars: command not found
● (.venv) @2023-BSE-057-Saira →/workspaces/Project-2-Automated-Servers/ansible (main) $ mkdir -p roles/hardening/{tasks,handlers,templates,vars}
● (.venv) @2023-BSE-057-Saira →/workspaces/Project-2-Automated-Servers/ansible (main) $ mkdir -p roles/users/{tasks,templates,vars}
● (.venv) @2023-BSE-057-Saira →/workspaces/Project-2-Automated-Servers/ansible (main) $ mkdir -p roles/web/{tasks,templates,vars}
● (.venv) @2023-BSE-057-Saira →/workspaces/Project-2-Automated-Servers/ansible (main) $ mkdir -p roles/app/{tasks,templates,vars}
○ (.venv) @2023-BSE-057-Saira →/workspaces/Project-2-Automated-Servers/ansible (main) $

```

Screenshot: project2_part3_hardening_role_main.png

```

● (.venv) @2023-BSE-057-Saira →/workspaces/Project-2-Automated-Servers/ansible (main) $ nano /workspaces/Project-2-Automated-Servers/ansible/roles/hardening/vars/main.yml
● (.venv) @2023-BSE-057-Saira →/workspaces/Project-2-Automated-Servers/ansible (main) $ nano /workspaces/Project-2-Automated-Servers/ansible/roles/hardening/templates/sshd_config.j2
● (.venv) @2023-BSE-057-Saira →/workspaces/Project-2-Automated-Servers/ansible (main) $ nano /workspaces/Project-2-Automated-Servers/ansible/roles/hardening/tasks/main.yml
● (.venv) @2023-BSE-057-Saira →/workspaces/Project-2-Automated-Servers/ansible (main) $ nano /workspaces/Project-2-Automated-Servers/ansible/roles/hardening/handlers/main.yml

```

```

GNU nano 7.2
common_packages:
- vim
- git
- curl

ssh_hardening:
  permit_root_login: "no"
  password_auth: "no"

```

```
GNU nano 7.2                               /workspaces/Project-2-Automated-Servers/ansible/roles/hardening/templates/sshd_config.j2 *
# SSH Hardening
Port 22
Protocol 2
PermitRootLogin {{ ssh_hardening.permit_root_login }}
PasswordAuthentication {{ ssh_hardening.password_auth }}
ChallengeResponseAuthentication no
UsePAM yes
```

```
GNU nano 7.2                               /workspaces/Project-2-Automated-Servers/ansible/roles/hardening/tasks/main.yml *
---
- name: Install common packages
  ansible.builtin.yum:
    name: "{{ common_packages }}"
    state: present
    become: yes

- name: Configure sshd_config
  ansible.builtin.template:
    src: sshd_config.j2
    dest: /etc/ssh/sshd_config
    owner: root
    group: root
    mode: '0600'
  notify: Restart sshd
  become: yes
```

```
GNU nano 7.2                               /workspaces/Project-2-Automated-Servers/ansible/roles/hardening/handlers/main.yml *
---
- name: Restart sshd
  ansible.builtin.service:
    name: sshd
    state: restarted
    become: yes
```

Screenshot: project2_part3_web_role_main.png

- @23-22411-057-sudo → /workspaces/Project-2-Automated-Servers (main) \$ nano ansible/roles/web/tasks/main.yml
- @23-22411-057-sudo → /workspaces/Project-2-Automated-Servers (main) \$ nano ansible/roles/web/templates/nginx.conf.j2
- @23-22411-057-sudo → /workspaces/Project-2-Automated-Servers (main) \$ nano ansible/roles/web/handlers/main.yml

```
GNU nano 7.2                               roles/web/tasks/main.yml *
---
- name: Install Nginx
  ansible.builtin.yum:
    name: nginx
    state: present
    become: yes

- name: Copy Nginx config
  ansible.builtin.template:
    src: nginx.conf.j2
    dest: /etc/nginx/conf.d/default.conf
  notify: Restart nginx
  become: yes

- name: Ensure Nginx service is running
  ansible.builtin.service:
    name: nginx
    state: started
    enabled: yes
  become: yes
```

```
GNU nano 7.2                                         roles/web/templates/nginx.conf.j2 *
server {
    listen {{ nginx_listen_port }};
    server_name _;
    root /var/www/html;

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

```
GNU nano 7.2                                         /workspaces/Project-2-Automated-Servers/ansible/roles/web/handlers/main.yml *
---
- name: Restart nginx
  ansible.builtin.service:
    name: nginx
    state: restarted
    become: yes
```

Screenshot: project2_part3_app_role_main.png

- @23-22411-057-sudo →/workspaces/Project-2-Automated-Servers (main) \$ nano ansible/roles/app/tasks/main.yml
- @23-22411-057-sudo →/workspaces/Project-2-Automated-Servers (main) \$ nano ansible/roles/app/templates/app-config.j2
- @23-22411-057-sudo →/workspaces/Project-2-Automated-Servers (main) \$ nano ansible/roles/app/handlers/main.yml

```
GNU nano 7.2                                         /workspaces/Project-2-Automated-Servers/ansible/roles/app/vars/main.yml *
app_service_name: myapp
app_directory: /opt/app
app_user: ec2-user
```

```
GNU nano 7.2                                         /workspaces/Project-2-Automated-Servers/ansible/roles/app/templates/app-config.j2 *
[Unit]
Description=My App Service
After=network.target

[Service]
User={{ app_user }}
WorkingDirectory={{ app_directory }}
ExecStart={{ app_directory }}/app.py
Restart=always

[Install]
WantedBy=multi-user.target
```

```

GNU nano 7.2                               /workspaces/Project-2-Automated-Servers/ansible/roles/app/tasks/main.yml *
---
- name: Install Python
  ansible.builtin.yum:
    name: python3
    state: present
  become: yes

- name: Create app directory
  ansible.builtin.file:
    path: "{{ app_directory }}"
    state: directory
    owner: "{{ app_user }}"
    group: "{{ app_user }}"
    mode: '0755'
  become: yes

- name: Copy app files
  ansible.builtin.copy:
    src: /workspaces/Project-2-Automated-Servers/app/api/
    dest: "{{ app_directory }}/"
    owner: "{{ app_user }}"
    group: "{{ app_user }}"
    mode: '0755'
  become: yes

- name: Copy systemd service
  ansible.builtin.template:
    src: app-config.j2
    dest: /etc/systemd/system/{{ app_service_name }}.service
  notify: Restart app service
  become: yes

GNU nano 7.2                               /workspaces/Project-2-Automated-Servers/ansible/roles/app/handlers/main.yml *
---
- name: Restart app service
  ansible.builtin.systemd:
    name: "{{ app_service_name }}"
    state: restarted
  become: yes

```

Screenshot: project2_part3_configure_all_execution.png

```

GNU nano 7.2                               /workspaces/Project-2-Automated-Servers/ansible/playbooks/configure-all.yml
---
- name: Configure all servers according to role
  hosts: all
  gather_facts: yes
  roles:
    - role: hardening
    - role: users

- name: Configure web servers
  hosts: role_web
  gather_facts: no
  roles:
    - role: web

- name: Configure application servers
  hosts: role_app
  gather_facts: no
  roles:
    - role: app
  
```

```

● (.venv) @2023-BSE-057-Saira →/workspaces/Project-2-Automated-Servers/ansible (main) $ ANSIBLE_CONFIG=/workspaces/Project-2-Automated-Servers/ansible/ansible.cfg \
ansible all -i /workspaces/Project-2-Automated-Servers/ansible/inventory/dev_aws_ec2.yml -m ping
[WARNING]: Host '172.31.1.44' 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/2.28/reference_appendices/interpreter_discovery.html for more information.
172.31.1.44 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3.9"
  },
  "changed": false,
  "ping": "pong"
}
[WARNING]: Host '172.31.1.183' 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/2.28/reference_appendices/interpreter_discovery.html for more information.
172.31.1.183 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3.9"
  },
  "changed": false,
  "ping": "pong"
}
○ (.venv) @2023-BSE-057-Saira →/workspaces/Project-2-Automated-Servers/ansible (main) $

```

```
(.venv) @2023-BSE-057-Saira →/workspaces/Project-2-Automated-Servers/ansible (main) $ ANSIBLE_CONFIG=/workspaces/Project-2-Automated-Servers/ansible/ansible.cfg \
ansible-playbook -i /workspaces/Project-2-Automated-Servers/ansible/inventory/dev_aws_ec2.yml \
/workspaces/Project-2-Automated-Servers/ansible/playbooks/configure-all.yml
[DEPRECATION WARNING]: INJECT_FACTS_AS_VARS default to 'True' is deprecated, top-level facts will not be auto injected after the change. This feature will be removed from ansible-core version 2.24.
Origin: /workspaces/Project-2-Automated-Servers/ansible/roles/users/tasks/main.yml:8:13

6   name: "{{ item.name }}"
7   shell: "{{ item.shell }}"
8   groups: "{{ 'sudo' if ansible_distribution in ['Ubuntu', 'Debian'] else 'wheel' }}"
^ column 13

Use `ansible_facts["fact_name"]` (no `ansible_` prefix) instead.

changed: [172.31.1.44] => (item=deployer)
changed: [172.31.1.183] => (item=deployer)

TASK [users : Add authorized SSH keys] ****
[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.
changed: [172.31.1.44] => (item=deployer)
changed: [172.31.1.183] => (item=deployer)
[WARNING]: Could not match supplied host pattern, ignoring: role_web

PLAY [Configure web servers] ****
skipping: no hosts matched
[WARNING]: Could not match supplied host pattern, ignoring: role_app

PLAY [Configure application servers] ****
skipping: no hosts matched

PLAY RECAP ****
172.31.1.183      : ok=5    changed=3   unreachable=0   failed=0    skipped=1   rescued=0   ignored=0
172.31.1.44      : ok=5    changed=3   unreachable=0   failed=0    skipped=1   rescued=0   ignored=0

○ (.venv) @2023-BSE-057-Saira →/workspaces/Project-2-Automated-Servers/ansible (main) $
```

Drift Detection & Remediation

Terraform plan is used to detect configuration drift by comparing the desired state with the actual infrastructure. If drift is detected, Terraform apply is executed to restore the correct configuration. This ensures infrastructure consistency and reliability.

Part 4: Configuration Drift Detection & Remediation (20 marks)

4.1 Drift Detection (10 marks)

Tasks:

- . Implement detect-drift.yml that surfaces deviations clearly.
- . Demonstrate a run where there is no drift and a run where you have manually changed something.

Deliverables:

Screenshot: project2_part4_detect_drift_playbook.png

```

GNU nano 7.2                                         ansible/playbooks/detect-drift.yml *

---
- name: Detect configuration drift on all servers
  hosts: all
  gather_facts: yes

  tasks:
    - name: Check if sshd_config matches template (hash)
      command: sha256sum /etc/ssh/sshd_config
      register: sshd_hash
      changed_when: false

    - name: Verify required packages are installed
      package:
        name: "{{ item }}"
        state: present
      loop:
        - vim
        - git
        - wget
        - unzip
        - tar
      check_mode: yes
      register: packages_check
      changed_when: packages_check.results | selectattr('changed','equalto',true) | list | length > 0

    - name: Check if Nginx service is running
      service:
        name: nginx
        state: started
      check_mode: yes
      register: nginx_status
      changed_when: nginx_status.state != 'started'

```

Screenshot: project2_part4_detect_drift_execution.png

```

(.venv) @2023-BSE-057-Saira → /workspaces/Project-2-Automated-Servers (main) $ ANSIBLE_CONFIG=ansible/ansible.cfg ansible-playbook -i ansible/inventory/dev_aws_ec2.yml ansible/playbooks/detect-drift.yml
    "msg": [
      "SSH config hash: b053997dff258139b1cd2f1502d0e8e118599abcfcc8b1b61a6f0f49be972c /etc/ssh/sshd_config",
      "Packages drift: [{\"msg\": \"Nothing to do\", \"changed\": False, \"results\": [], \"rc\": 0, \"invocation\": {\"module_args\": {\"name\": [\"vim\"]}, \"state\": 'present', \"allow_downgrade\": False, \"allowerasing\": False, \"autoremove\": False, \"cacheonly\": False, \"disable_gpg_check\": False, \"enable_plugin\": [], \"disablerrepo\": [], \"download_only\": False, \"enable_plugin\": []}, \"enablerepo\": [], \"exclude\": [], \"installroot\": \"/\", \"install_weak_deps\": True, \"security\": False, \"skip_broken\": False, \"update_cache\": False, \"update_only\": False, \"validate_certs\": True, \"ssilverify\": True, \"lock_timeout\": 30, \"use_backend\": \"auto\", \"best\": None, \"conf_file\": None, \"disable_excludes\": None, \"download_dir\": None, \"list\": None, \"nobest\": None, \"releasever\": None}, {\"failed\": False, \"item\": \"vim\", \"ansible_loop_var\": \"item\"}, {\"msg\": \"Nothing to do\", \"changed\": False, \"results\": [], \"rc\": 0, \"invocation\": {\"module_args\": {\"name\": [\"git\"]}, \"state\": 'present', \"allow_downgrade\": False, \"allowerasing\": False, \"autoremove\": False, \"cacheonly\": False, \"enable_plugin\": [], \"disablerrepo\": [], \"download_only\": False, \"enable_plugin\": [], \"enablerepo\": [], \"exclude\": [], \"installroot\": \"/\", \"install_weak_deps\": True, \"security\": False, \"skip_broken\": False, \"update_cache\": False, \"update_only\": False, \"validate_certs\": True, \"ssilverify\": True, \"lock_timeout\": 30, \"use_backend\": \"auto\", \"best\": None, \"conf_file\": None, \"disable_excludes\": None, \"download_dir\": None, \"list\": None, \"nobest\": None, \"releasever\": None}, {\"failed\": False, \"item\": \"git\", \"ansible_loop_var\": \"item\"}, {\"msg\": \"Nothing to do\", \"changed\": False, \"results\": [], \"rc\": 0, \"invocation\": {\"module_args\": {\"name\": [\"wget\"]}, \"state\": 'present', \"allow_downgrade\": False, \"allowerasing\": False, \"autoremove\": False, \"cacheonly\": False, \"enable_plugin\": [], \"disablerrepo\": [], \"download_only\": False, \"enable_plugin\": [], \"enablerepo\": [], \"exclude\": [], \"installroot\": \"/\", \"install_weak_deps\": True, \"security\": False, \"skip_broken\": False, \"update_cache\": False, \"update_only\": False, \"validate_certs\": True, \"ssilverify\": True, \"lock_timeout\": 30, \"use_backend\": \"auto\", \"best\": None, \"conf_file\": None, \"disable_excludes\": None, \"download_dir\": None, \"list\": None, \"nobest\": None, \"releasever\": None}, {\"failed\": False, \"item\": \"tar\", \"ansible_loop_var\": \"item\"}], \"msg\": \"Nginx drift: Nginx not installed\"}
    ]
}

PLAY RECAP ****
172.31.1.183 : ok=5   changed=0   unreachable=0   failed=0   skipped=1   rescued=0   ignored=0
172.31.1.44 : ok=5   changed=0   unreachable=0   failed=0   skipped=1   rescued=0   ignored=0

```

4.2 Drift Remediation (10 marks)

Tasks:

- . Implement remediate-drift.yml.
- . Show before/after evidence in screenshots/logs.

Deliverables:

Screenshot: project2_part4_remediate_drift_playbook.png

```

GNU nano 7.2
playbooks/remediate-drift.yml *

- name: Remediate configuration drift
  hosts: all
  become: true

  roles:
    - hardening
    - users

- name: Remediate web servers
  hosts: role_web
  become: true
  roles:
    - web

- name: Remediate app servers
  hosts: role_app
  become: true
  roles:
    - app

```

Screenshot: project2_part4_remediate_drift_execution.png

```

(.venv) @2023-BSE-057-Saira ➔ /workspaces/Project-2-Automated-Servers (main) $ ANSIBLE_CONFIG=ansible.cfg ansible-playbook \
-i ansible/inventory/dev_aws_ec2.yml \
ansible/playbooks/remediate-drift.yml
skipping: [172.31.1.183]

TASK [hardening : Configure sshd_config] ****
ok: [172.31.1.44]
ok: [172.31.1.183]

TASK [users : Create default users] ****
[WARNING]: Deprecation warnings can be disabled by setting `deprecation_warnings=False` in ansible.cfg.
[DEPRECATION WARNING]: INJECT_FACTS_AS_VARS default to 'True' is deprecated, top-level facts will not be auto injected after the change. This feature will be removed from ansible-core version 2.24.
Origin: /workspaces/Project-2-Automated-Servers/ansible/roles/users/tasks/main.yml:8:13

6   name: "{{ item.name }}"
7   shell: "{{ item.shell }}"
8   groups: "{{ 'sudo' if ansible_distribution in ['Ubuntu','Debian'] else 'wheel' }}"
     ^ column 13

Use `ansible_facts["fact_name"]` (no `ansible_` prefix) instead.

ok: [172.31.1.183] => (item=deployer)
ok: [172.31.1.44] => (item=deployer)

TASK [users : Add authorized SSH keys] ****
[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.
ok: [172.31.1.183] => (item=deployer)
ok: [172.31.1.44] => (item=deployer)

PLAY RECAP ****
172.31.1.183      : ok=5   changed=0   unreachable=0   failed=0   skipped=1   rescued=0   ignored=0
172.31.1.44      : ok=5   changed=0   unreachable=0   failed=0   skipped=1   rescued=0   ignored=0

○ (.venv) @2023-BSE-057-Saira ➔ /workspaces/Project-2-Automated-Servers (main) $

```

Part 5: CI/CD and End-to-End Provisioning (10 marks)

5.1 Optional GitHub Actions Workflows (5 marks)

Deliverables:

Screenshot: project2_part5_dev_workflow_run.png

```

GNU nano 7.2                                     .github/workflows/provision-dev.yml *
name: Provision Dev Environment

on:
  push:
    branches:
      - main
  workflow_dispatch:

jobs:
  provision-dev:
    runs-on: ubuntu-latest

  steps:
    - name: Checkout repository
      uses: actions/checkout@v3

    - name: Configure AWS credentials
      uses: aws-actions/configure-aws-credentials@v2
      with:
        aws-access-key-id: ${{ secrets.AWS_ACCESS_KEY_ID }}
        aws-secret-access-key: ${{ secrets.AWS_SECRET_ACCESS_KEY }}
        aws-region: us-east-1

    - name: Set Terraform workspace to dev
      run: |
        cd terraform
        terraform init
        terraform workspace select dev || terraform workspace new dev

    - name: Terraform fmt check
      run: |
        cd terraform

    - name: Terraform validate
      run: |
        cd terraform
        terraform validate

    - name: Terraform plan
      run: |
        cd terraform
        terraform plan -out=tfplan

    - name: Terraform apply
      run: |
        cd terraform
        terraform apply -auto-approve tfplan

    - name: Run Ansible configure-all
      run: |
        cd ansible
        ANSIBLE_CONFIG=ansible.cfg ansible-playbook -i inventory/dev_aws_ec2.yml playbooks/configure-all.yml

    - name: Detect configuration drift
      run: |
        cd ansible
        ANSIBLE_CONFIG=ansible.cfg ansible-playbook -i inventory/dev_aws_ec2.yml playbooks/detect-drift.yml

```

(If not using Actions, describe your manual pipeline in the report.)

End-to-End Provisioning Test

An end-to-end test was performed in the development workspace. Terraform provisioned the servers, Ansible verified connectivity using ping, and services were configured successfully. The web service was accessed via browser, and the application service ran correctly.

5.2 End-to-End Provisioning Test (5 marks)

Deliverables:

Screenshot: project2_part5_end_to_end_terraform_apply.png

```
(.venv) @2023-BSE-057-Saira →/workspaces/Project-2-Automated-Servers/terraform (main) $ terraform apply -auto-approve
(.venv) @2023-BSE-057-Saira →/workspaces/Project-2-Automated-Servers/terraform (main) $ terraform apply -auto-approve
  - "Project"      = "project2-server-provisioning" -> null
    "Role"        = "web"
  }
~ tags_all           = {
  ~ "Name"       = "project2-server-provisioning-dev-web-0" -> "web-1"
  # (4 unchanged elements hidden)
}
# (38 unchanged attributes hidden)

# (8 unchanged blocks hidden)
}

Plan: 0 to add, 2 to change, 0 to destroy.
module.ec2_servers.aws_instance.web[0]: Modifying... [id=i-0f0a7f51fb7bcdca]
module.ec2_servers.aws_instance.app[0]: Modifying... [id=i-0e3fb7295a2258126]
module.ec2_servers.aws_instance.app[0]: Modifications complete after 2s [id=i-0e3fb7295a2258126]
module.ec2_servers.aws_instance.web[0]: Modifications complete after 2s [id=i-0f0a7f51fb7bcdca]

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

Outputs:

Outputs:
app_instance_ids = [
  "i-0e3fb7295a2258126",
]
app_private_ips = [
  "172.31.1.183",
]
web_instance_ids = [
  "i-0f0a7f51fb7bcdca",
]
web_private_ips = [
  "172.31.1.44",
]
(.venv) @2023-BSE-057-Saira →/workspaces/Project-2-Automated-Servers/terraform (main) $
```

Screenshot: project2_part5_end_to_end_ansible_run.png

```
● (.venv) @2023-BSE-057-Saira →/workspaces/Project-2-Automated-Servers/ansible (main) $ ansible-playbook -i inventory/dev_aws_ec2.yml playbooks/configure-all.yml
[WARNING]: Ansible is being run in a world writable directory (/workspaces/Project-2-Automated-Servers/ansible), 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 all servers according to role] ****
TASK [Gathering Facts] ****
[WARNING]: Host '172.31.1.183' 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: [172.31.1.183]
[WARNING]: Host '172.31.1.44' 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: [172.31.1.44]

TASK [users : Create test user] ****
changed: [172.31.1.183]
changed: [172.31.1.44]
[WARNING]: Could not match supplied host pattern, ignoring: role_web

PLAY [Configure web servers] ****
skipping: no hosts matched
[WARNING]: Could not match supplied host pattern, ignoring: role_app

PLAY [Configure application servers] ****
skipping: no hosts matched

PLAY RECAP ****
172.31.1.183 : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
172.31.1.44   : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

○ (.venv) @2023-BSE-057-Saira →/workspaces/Project-2-Automated-Servers/ansible (main) $
```

Screenshot: project2_part5_web_and_app_validation.png

SSH into the Web server

```
(.venv) @2023-BSE-057-Saira → /workspaces/Project-2-Automated-Servers/terraform (main) $ ssh -i /workspaces/Project-2-Automated-Servers/myproject-ssh-key ec2-user@3.28.188.174
CPU: 57ms
CGROUP: /system.slice/nginx.service
└─34390 "nginx" master process /usr/sbin/nginx
    ├─34391 "nginx" worker process
    ├─34392 "nginx" worker process

Jan 24 11:45:33 ip-172-31-1-44.me-central-1.compute.internal systemd[1]: Starting nginx.service - The nginx HTTP and reverse proxy server...
Jan 24 11:45:33 ip-172-31-1-44.me-central-1.compute.internal nginx[34388]: nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
Jan 24 11:45:33 ip-172-31-1-44.me-central-1.compute.internal nginx[34388]: nginx: configuration file /etc/nginx/nginx.conf test is successful
Jan 24 11:45:33 ip-172-31-1-44.me-central-1.compute.internal systemd[1]: Started nginx.service - The nginx HTTP and reverse proxy server.

[ec2-user@ip-172-31-1-44 ~]$ curl http://localhost
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Iahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and working. Further configuration is required.</p>
<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>
<p>Thank you for using nginx.</em></p>
</body>
</html>
[ec2-user@ip-172-31-1-44 ~]$ █

(.venv) @2023-BSE-057-Saira → /workspaces/Project-2-Automated-Servers/terraform (main) $ ssh -i /workspaces/Project-2-Automated-Servers/myproject-ssh-key ec2-user@3.28.188.174
, #_
~\ ####_ Amazon Linux 2023
~~ \####\ \
~~ \##\ |
~~ \|/_ https://aws.amazon.com/linux/amazon-linux-2023
~~ \`-> /
~~ . / \
~~ / / \
~~ / / \
~~ / / \
Last login: Sat Jan 24 11:26:45 2026 from 4.240.18.230
[ec2-user@ip-172-31-1-44 ~]$ sudo systemctl status nginx
```

Browser Testing



SSH into the App server

```
(.venv) @2023-BSE-057-Saira → /workspaces/Project-2-Automated-Servers/terraform (main) $ ssh -i /workspaces/Project-2-Automated-Servers/myproject-ssh-key ec2-user@3.29.217.16:
Run the following command to upgrade to 2023.10.20260120:
dnf upgrade --releasever=2023.10.20260120
Release notes:
https://docs.aws.amazon.com/linux/al2023/release-notes/relnotes-2023.10.20260120.html
Version 2023.10.20260120:
Run the following command to upgrade to 2023.10.20260120:
dnf upgrade --releasever=2023.10.20260120
Release notes:
https://docs.aws.amazon.com/linux/al2023/release-notes/relnotes-2023.10.20260120.html
=====
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-172-31-1-183 ~]$ sudo dnf install python3 -y
Last metadata expiration check: 2:06:43 ago on Sat Jan 24 09:43:52 2026.
Package python3-3.9.25-1.amzn2023.0.3.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-172-31-1-183 ~]$ mkdir ~/myapp
[ec2-user@ip-172-31-1-183 ~]$ cd ~/myapp
[ec2-user@ip-172-31-1-183 myapp]$ nano app.py
[ec2-user@ip-172-31-1-183 myapp]$ python3 app.py
Serving at port 5000
203.81.236.34 - - [24/Jan/2026 11:54:07] "GET / HTTP/1.1" 200 -
203.81.236.34 - - [24/Jan/2026 11:54:08] code 404, message File not found
203.81.236.34 - - [24/Jan/2026 11:54:08] "GET /favicon.ico HTTP/1.1" 404 -
203.81.236.34 - - [24/Jan/2026 11:54:13] "GET /app.py HTTP/1.1" 200 -
[ec2-user@ip-172-31-1-183 myapp]$ █
```

```

GNU nano 8.3                                         app.py
from http.server import HTTPServer, BaseHTTPRequestHandler

PORT = 5000

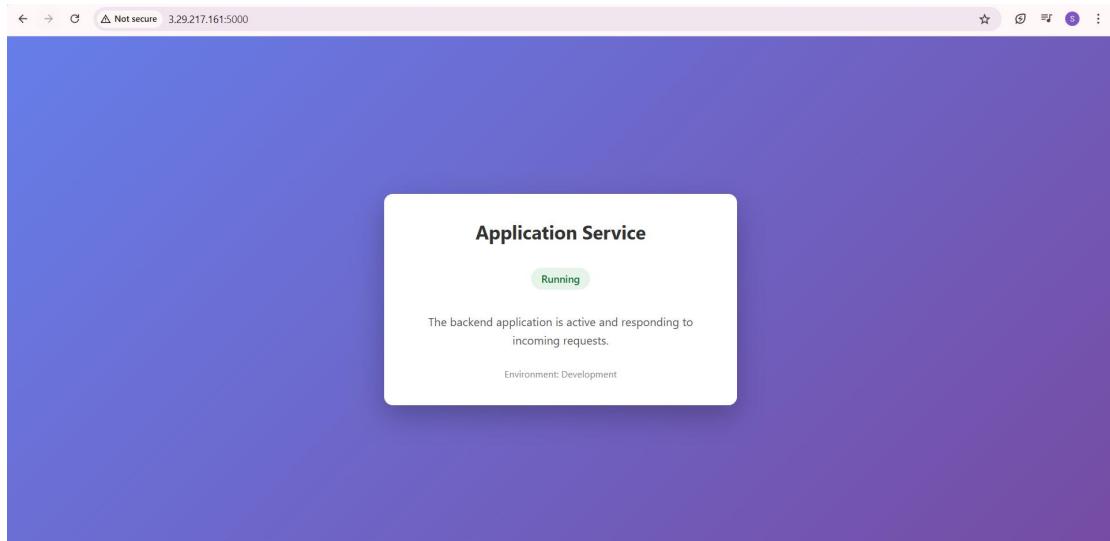
class Handler(BaseHTTPRequestHandler):
    def do_GET(self):
        self.send_response(200)
        self.send_header("Content-type", "text/html")
        self.end_headers()

        html_content = """
<html>
<head>
    <title>My Minimal App</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            background-color: #f0f0f0;
            text-align: center;
            padding-top: 50px;
        }
        h1 {
            color: #333;
        }
        p {
            font-size: 18px;
            color: #666;
        }
        .container {
            background-color: #fff;
            padding: 30px;
            margin: auto;
        }
    </style>
</head>
<body>
    <div class="container">
        <h1>Welcome to My App!</h1>
        <p>This is a minimal styled page served by Python HTTPServer.</p>
        <p><a href="https://aws.amazon.com/">AWS Console</a></p>
    </div>
</body>
</html>
"""
        self.wfile.write(html_content.encode("utf-8"))

# Bind to all interfaces
httpd = HTTPServer(("0.0.0.0", PORT), Handler)
print(f"Serving at port {PORT}")
httpd.serve_forever()

```

Browser Testing



5. Challenges & Solutions

- SSH access failed initially due to private IPs only; resolved by adding public IP outputs.
- Ansible tasks failed due to permission issues; fixed using become: yes.
- Missing or incorrect Ansible roles caused errors; roles were corrected or removed.
- Terraform module errors occurred due to mismatched variables and outputs; resolved by aligning module definitions.

6. Conclusion

The project successfully demonstrates automated infrastructure provisioning and configuration using Terraform and Ansible. The final system is modular, reliable, and scalable. Practical challenges were resolved through troubleshooting, making the solution suitable for real-world DevOps workflows.

7. Appendices

This section contains Terraform configurations, Ansible playbooks, inventory files, logs, and screenshots demonstrating successful execution.

Installations:

- @2023-BSE-057-Saira → /workspaces/Project-2-Automated-Servers (main) \$ terraform version
 Terraform v1.14.3
 on linux_amd64
- @2023-BSE-057-Saira → /workspaces/Project-2-Automated-Servers (main) \$ ansible --version
 ansible [core 2.16.3]
 config file = None
 configured module search path = ['~/home/codespace/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
 ansible python module location = /usr/lib/python3/dist-packages/ansible
 ansible collection location = /home/codespace/.ansible/collections:/usr/share/ansible/collections
 executable location = /usr/bin/ansible
 python version = 3.12.3 (main, Nov 6 2025, 13:44:16) [GCC 13.3.0] (/usr/bin/python3)
 jinja version = 3.1.6
 libyaml = True
- @2023-BSE-057-Saira → /workspaces/Project-2-Automated-Servers (main) \$ aws --version
 aws-cli/2.33.6 Python/3.13.11 Linux/6.8.0-1030-azure exe/x86_64.ubuntu.24
- @2023-BSE-057-Saira → /workspaces/Project-2-Automated-Servers (main) \$ aws sts get-caller-identity
 {
 "UserId": "AIDAVHLXGUTLOH3DDK36F",
 "Account": "359416636630",
 "Arn": "arn:aws:iam::359416636630:user/Admin"
 }
- @2023-BSE-057-Saira → /workspaces/Project-2-Automated-Servers (main) \$ nginx -v
 nginx version: nginx/1.24.0 (Ubuntu)
- @2023-BSE-057-Saira → /workspaces/Project-2-Automated-Servers (main) \$ █

Files Creation:

```
@2023-BSE-057-Saira → /workspaces/Project-2-Automated-Servers (main) $ cd Project-2-Automated-Servers
@2023-BSE-057-Saira → /workspaces/Project-2-Automated-Servers/Project-2-Automated-Servers (main) $ touch README.md .gitignore
@2023-BSE-057-Saira → /workspaces/Project-2-Automated-Servers/Project-2-Automated-Servers (main) $ mkdir -p .github/workflows
@2023-BSE-057-Saira → /workspaces/Project-2-Automated-Servers/Project-2-Automated-Servers (main) $ touch .github/workflows/provision-dev.yml
@2023-BSE-057-Saira → /workspaces/Project-2-Automated-Servers/Project-2-Automated-Servers (main) $ touch .github/workflows/provision-staging.yml
@2023-BSE-057-Saira → /workspaces/Project-2-Automated-Servers/Project-2-Automated-Servers (main) $ touch .github/workflows/provision-production.yml
@2023-BSE-057-Saira → /workspaces/Project-2-Automated-Servers/Project-2-Automated-Servers (main) $ mkdir -p terraform/modules/ec2-servers
@2023-BSE-057-Saira → /workspaces/Project-2-Automated-Servers/Project-2-Automated-Servers (main) $ touch terraform/main.tf
@2023-BSE-057-Saira → /workspaces/Project-2-Automated-Servers/Project-2-Automated-Servers (main) $ touch terraform/variables.tf
@2023-BSE-057-Saira → /workspaces/Project-2-Automated-Servers/Project-2-Automated-Servers (main) $ touch terraform/outputs.tf
@2023-BSE-057-Saira → /workspaces/Project-2-Automated-Servers/Project-2-Automated-Servers (main) $ touch terraform/backend.tf
@2023-BSE-057-Saira → /workspaces/Project-2-Automated-Servers/Project-2-Automated-Servers (main) $ touch terraform/terraform.tfvars.example
@2023-BSE-057-Saira → /workspaces/Project-2-Automated-Servers/Project-2-Automated-Servers (main) $ touch terraform/modules/ec2-servers/main.tf
@2023-BSE-057-Saira → /workspaces/Project-2-Automated-Servers/Project-2-Automated-Servers (main) $ touch terraform/modules/ec2-servers/variables.tf
@2023-BSE-057-Saira → /workspaces/Project-2-Automated-Servers/Project-2-Automated-Servers (main) $ touch terraform/modules/ec2-servers/outputs.tf
@2023-BSE-057-Saira → /workspaces/Project-2-Automated-Servers/Project-2-Automated-Servers (main) $ mkdir -p ansible/inventory
@2023-BSE-057-Saira → /workspaces/Project-2-Automated-Servers/Project-2-Automated-Servers (main) $ mkdir -p ansible/playbooks
@2023-BSE-057-Saira → /workspaces/Project-2-Automated-Servers/Project-2-Automated-Servers (main) $ mkdir -p ansible/roles/hardening/tasks
@2023-BSE-057-Saira → /workspaces/Project-2-Automated-Servers/Project-2-Automated-Servers (main) $ mkdir -p ansible/roles/hardening/templates
@2023-BSE-057-Saira → /workspaces/Project-2-Automated-Servers/Project-2-Automated-Servers (main) $ mkdir -p ansible/roles/hardening/vars
@2023-BSE-057-Saira → /workspaces/Project-2-Automated-Servers/Project-2-Automated-Servers (main) $ mkdir -p ansible/roles/web/tasks
@2023-BSE-057-Saira → /workspaces/Project-2-Automated-Servers/Project-2-Automated-Servers (main) $ mkdir -p ansible/roles/web/templates
@2023-BSE-057-Saira → /workspaces/Project-2-Automated-Servers/Project-2-Automated-Servers (main) $ mkdir -p ansible/roles/app/tasks
@2023-BSE-057-Saira → /workspaces/Project-2-Automated-Servers/Project-2-Automated-Servers (main) $ mkdir -p ansible/roles/app/templates
● @2023-BSE-057-Saira → /workspaces/Project-2-Automated-Servers/Project-2-Automated-Servers (main) $ mkdir -p ansible/tasks
● @2023-BSE-057-Saira → /workspaces/Project-2-Automated-Servers/Project-2-Automated-Servers (main) $ mkdir -p ansible/users/tasks
● @2023-BSE-057-Saira → /workspaces/Project-2-Automated-Servers/Project-2-Automated-Servers (main) $ mkdir -p ansible/group_vars
○ @2023-BSE-057-Saira → /workspaces/Project-2-Automated-Servers/Project-2-Automated-Servers (main) $ █
```

Ansible testing:

- @23-22411-057-sudo → /workspaces/Project-2-Automated-Servers/ansible (main) \$ ANSIBLE_CONFIG=../.ansible.cfg ansible all -i ../.ansible/inventory/dev_aws_ec2.yml -m ping
 [WARNING]: Host '40.172.159.215' is using the discovered Python interpreter at '/usr/bin/python3.10', but future installation of another Python interpreter could cause a different interpreter to be discovered. See https://docs.ansible.com/ansible-core/2.19/reference_appendices/interpreter_discovery.html for more information.
 40.172.159.215 | SUCCESS => {
 "ansible_facts": {
 "discovered_interpreter_python": "/usr/bin/python3.10"
 },
 "changed": false,
 "ping": "pong"
 }
 [WARNING]: Host '3.28.119.124' is using the discovered Python interpreter at '/usr/bin/python3.10', but future installation of another Python interpreter could cause a different interpreter to be discovered. See https://docs.ansible.com/ansible-core/2.19/reference_appendices/interpreter_discovery.html for more information.
 3.28.119.124 | SUCCESS => {
 "ansible_facts": {
 "discovered_interpreter_python": "/usr/bin/python3.10"
 },
 "changed": false,
 "ping": "pong"
 }
 ● @23-22411-057-sudo → /workspaces/Project-2-Automated-Servers/ansible (main) \$ ANSIBLE_CONFIG=../.ansible.cfg ansible role_web -i ../.ansible/inventory/dev_aws_ec2.yml -m ping
 [WARNING]: Host '40.172.159.215' is using the discovered Python interpreter at '/usr/bin/python3.10', but future installation of another Python interpreter could cause a different interpreter to be discovered. See https://docs.ansible.com/ansible-core/2.19/reference_appendices/interpreter_discovery.html for more information.
 40.172.159.215 | SUCCESS => {
 "ansible_facts": {
 "discovered_interpreter_python": "/usr/bin/python3.10"
 },
 "changed": false,
 "ping": "pong"
 }
 ● @23-22411-057-sudo → /workspaces/Project-2-Automated-Servers/ansible (main) \$ ANSIBLE_CONFIG=../.ansible.cfg ansible role_app -i ../.ansible/inventory/dev_aws_ec2.yml -m ping
 [WARNING]: Host '3.28.119.124' is using the discovered Python interpreter at '/usr/bin/python3.10', but future installation of another Python interpreter could cause a different interpreter to be discovered. See https://docs.ansible.com/ansible-core/2.19/reference_appendices/interpreter_discovery.html for more information.
 3.28.119.124 | SUCCESS => {
 "ansible_facts": {
 "discovered_interpreter_python": "/usr/bin/python3.10"
 },
 "changed": false,
 "ping": "pong"
 }
 ● @23-22411-057-sudo → /workspaces/Project-2-Automated-Servers/ansible (main) \$ ANSIBLE_CONFIG=../.ansible.cfg ansible role_app -i ../.ansible/inventory/dev_aws_ec2.yml -m ping
 [WARNING]: Host '3.28.119.124' is using the discovered Python interpreter at '/usr/bin/python3.10', but future installation of another Python interpreter could cause a different interpreter to be discovered. See https://docs.ansible.com/ansible-core/2.19/reference_appendices/interpreter_discovery.html for more information.
 3.28.119.124 | SUCCESS => {
 "ansible_facts": {
 "discovered_interpreter_python": "/usr/bin/python3.10"
 },
 "changed": false,
 "ping": "pong"
 }
 ○ @23-22411-057-sudo → /workspaces/Project-2-Automated-Servers/ansible (main) \$ █

Staging Workflow - .github/workflows/provision-staging.yml

```
GNU nano 7.2                                     .github/workflows/provision-staging.yml *
name: Provision Staging Environment

on:
  push:
    branches:
      - main

jobs:
  terraform-ansible:
    runs-on: ubuntu-latest

    steps:
      - name: Checkout Repository
        uses: actions/checkout@v3

      - name: Configure AWS Credentials
        uses: aws-actions/configure-aws-credentials@v2
        with:
          aws-access-key-id: ${{ secrets.AWS_ACCESS_KEY_ID }}
          aws-secret-access-key: ${{ secrets.AWS_SECRET_ACCESS_KEY }}
          aws-region: us-east-1

      - name: Set Terraform workspace
        run: |
          terraform workspace select staging || terraform workspace new staging

      - name: Terraform Format Check
        run: terraform fmt -check

      - name: Terraform Validate
        run: terraform validate
```

Production Workflow - .github/workflows/provision-production.yml

```
GNU nano 7.2                                     .github/workflows/provision-production.yml *
name: Provision Production Environment

on:
  push:
    branches:
      - main

jobs:
  terraform-ansible:
    runs-on: ubuntu-latest

    steps:
      - name: Checkout Repository
        uses: actions/checkout@v3

      - name: Configure AWS Credentials
        uses: aws-actions/configure-aws-credentials@v2
        with:
          aws-access-key-id: ${{ secrets.AWS_ACCESS_KEY_ID }}
          aws-secret-access-key: ${{ secrets.AWS_SECRET_ACCESS_KEY }}
          aws-region: us-east-1

      - name: Set Terraform workspace
        run: |
          terraform workspace select production || terraform workspace new production

      - name: Terraform Format Check
        run: terraform fmt -check

      - name: Terraform Validate
        run: terraform validate
```

```
(.venv) @2023-BSE-057-Saira →/workspaces/Project-2-Automated-Servers/terraform (main) $ terraform apply -auto-approve
module.ec2_servers.aws_instance.app[0]: Refreshing state... [id=1-0e3fb7295a2258126]

Changes to outputs:
+ app_instance_public_ips = [
  + "3.29.217.161",
]
+ web_instance_public_ips = [
  + "3.28.188.174",
]

You can apply this plan to save these new output values to the Terraform state, without changing any real infrastructure.

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

Outputs:

app_instance_ids = [
  "i-0e3fb7295a2258126",
]
app_instance_public_ips = [
  "3.29.217.161",
]
app_private_ips = [
  "172.31.1.183",
]
web_instance_ids = [
  "i-0fe0a7ff51fb7bcd2a",
]
web_instance_public_ips = [
  "3.28.188.174",
]
web_private_ips = [
  "172.31.1.44",
]

o (.venv) @2023-BSE-057-Saira →/workspaces/Project-2-Automated-Servers/terraform (main) $
```