**Lab 8: Terraform Secured Infra Module**.

Focuses on provisioning secure foundational infrastructure components like VPC, IAM, and S3 using Terraform modules with built-in best practices for security and compliance.

**✅ Objective**

Provision secure AWS infrastructure using Terraform modules that encapsulate best practices:

* VPC with private/public subnets
* S3 buckets with encryption & block public access
* IAM roles with least privilege

**🔧 Tools Required**

* AWS CLI (configured with programmatic access)
* Terraform v1.3+ (v1.5+ recommended)
* Git, Shell
* Prebuilt Terraform modules (from registry or local repo)
* [Optional] VS Code with Terraform plugin

**📁 Folder Structure**

(venv) root@ip-172-31-14-172:~/terraform-secure-infra# tree

.

├── main.tf

├── outputs.tf

├── terraform.tfstate

├── variable.tf

└── versions.tf

**🔹 Step-by-Step Guide**

**🔹 Step 1: Create a New Terraform Project Directory**

bash

CopyEdit

mkdir terraform-secure-infra && cd terraform-secure-infra

(venv) root@ip-172-31-14-172:~/terraform-secure-infra# ls

main.tf outputs.tf terraform.tfstate variable.tf versions.tf

**🔹 Step 2: Define Providers and Versions (s3 backend optional for now)**

**versions.tf**

(venv) root@ip-172-31-14-172:~/terraform-secure-infra# cat versions.tf

terraform {

required\_providers {

aws = {

source = "hashicorp/aws"

version = "~> 5.0"

}

}

required\_version = ">= 1.3.0"

}

provider "aws" {

region = "us-east-1"

}

If the backend doesn’t exist yet, initialize with local backend first and switch later.

**Step3 : VPC Module (Secure Network), Secure S3 Bucket (Block Public Access, Enable Encryption) , IAM Module with Least Privilege Role.**

(venv) root@ip-172-31-14-172:~/terraform-secure-infra# cat main.tf

module "vpc" {

source = "terraform-aws-modules/vpc/aws"

version = "5.1.0"

name = "secure-vpc"

cidr = "10.0.0.0/16"

azs = ["us-east-1a", "us-east-1b"]

private\_subnets = ["10.0.1.0/24", "10.0.2.0/24"]

public\_subnets = ["10.0.101.0/24", "10.0.102.0/24"]

enable\_nat\_gateway = true

single\_nat\_gateway = true

tags = {

Environment = "platform-lab"

Owner = "raman"

}

}

resource "aws\_s3\_bucket" "secure\_bucket" {

bucket = "platform-secure-artifacts-${random\_id.id.hex}"

force\_destroy = true

tags = {

Name = "SecureBucket"

Environment = "platform-lab"

}

}

resource "aws\_s3\_bucket\_public\_access\_block" "block\_public" {

bucket = aws\_s3\_bucket.secure\_bucket.id

block\_public\_acls = true

block\_public\_policy = true

ignore\_public\_acls = true

restrict\_public\_buckets = true

}

resource "aws\_s3\_bucket\_server\_side\_encryption\_configuration" "encrypt" {

bucket = aws\_s3\_bucket.secure\_bucket.id

rule {

apply\_server\_side\_encryption\_by\_default {

sse\_algorithm = "AES256"

}

}

}

resource "random\_id" "id" {

byte\_length = 4

}

data "aws\_iam\_policy\_document" "assume\_role\_policy" {

statement {

effect = "Allow"

principals {

type = "Service"

identifiers = ["ec2.amazonaws.com"]

}

actions = ["sts:AssumeRole"]

}

}

resource "aws\_iam\_role" "custom" {

name = "platform-app-role"

assume\_role\_policy = data.aws\_iam\_policy\_document.assume\_role\_policy.json

}

resource "aws\_iam\_role\_policy\_attachment" "attach" {

role = aws\_iam\_role.custom.name

policy\_arn = "arn:aws:iam::aws:policy/AmazonS3ReadOnlyAccess"

}

**🔹 Step 6: Variables and Outputs**

**variables.tf** – for reusability (optional)

(venv) root@ip-172-31-14-172:~/terraform-secure-infra# cat variable.tf

variable "region" {

type = string

default = "sa-east-1"

}

**outputs.tf**

(venv) root@ip-172-31-14-172:~/terraform-secure-infra# cat outputs.tf

output "vpc\_id" {

value = module.vpc.vpc\_id

}

output "s3\_bucket" {

value = aws\_s3\_bucket.secure\_bucket.bucket

}

**🔹 Step 7: Initialize and Apply**

bash

CopyEdit

terraform init

terraform plan

terraform apply -auto-approve

**🔹 Step 8: Validate Resources**

**✅ VPC**

* 2 public + 2 private subnets
* NAT gateway in public subnet

**✅ S3**

* Encryption: AES256
* Public Access Blocked

**✅ IAM**

* Role visible in AWS Console with AmazonS3ReadOnlyAccess

**✅ Extra Enhancements (Optional)**

* Add logging and versioning to S3
* Add CloudWatch log group outputs
* Use terraform-docs to generate module README
* Add tfsec or checkov scan for policies

**✅ Deliverables**

| **Component** | **Status** |
| --- | --- |
| VPC with NAT | ✅ |
| Encrypted S3 with public access blocked | ✅ |
| IAM role with least privilege | ✅ |
| Terraform module-based infra | ✅ |