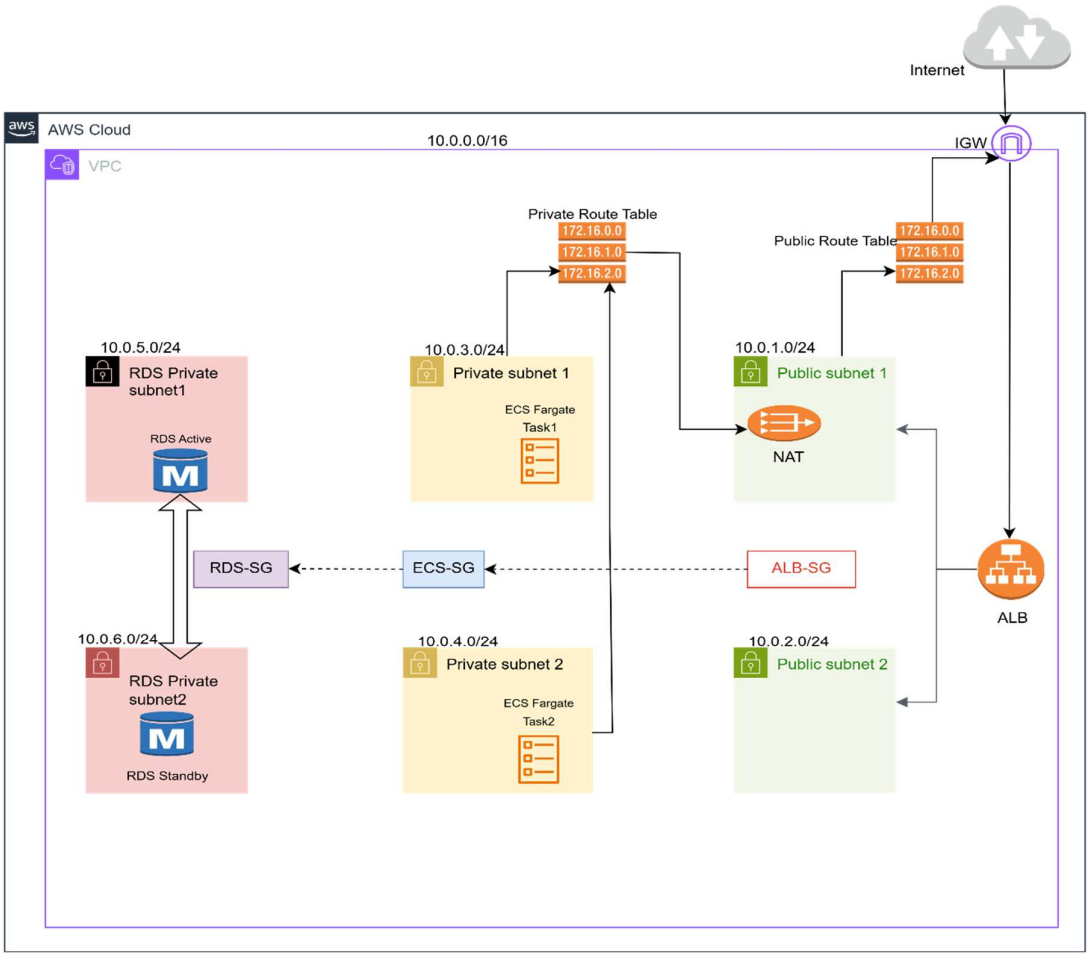
Deploy Student Portal App in AWS ECS using Terraform

Scenario: Student Portal App on AWS ECS using Terraform

This Terraform project provisions a complete AWS infrastructure for hosting a containerized student portal application using ECS Fargate, with RDS PostgreSQL database backend.

Architectural Diagram



Infrastructure Components

Network (network.tf)

- **VPC**: Custom VPC with CIDR 10.0.0.0/16
- **Subnets**:
  - 2 Public Subnets (10.0.1.0/24, 10.0.2.0/24) across AZ a & b - for ALB
  - 2 Private Subnets (10.0.3.0/24, 10.0.4.0/24) across AZ a & b - for ECS tasks
  - 2 RDS Private Subnets (10.0.5.0/24, 10.0.6.0/24) across AZ a & b - for database
- **Internet Gateway**: For public subnet internet access
- **NAT Gateway**: With Elastic IP for private subnet outbound traffic
- **Route Tables**: Separate routing for public and private subnets

Application Layer (ecs.tf)

- **ECS Cluster**: Fargate-based cluster for running containers
- **ECS Task Definition**: ○ Container: Student Portal application (ECR image)
  - Port: 8000
  - Resources: 256 CPU units, 512 MB memory
  - Environment: Database connection string injected via env vars
- **ECS Service**:
  - Desired count: 2 tasks
  - Launch type: Fargate
  - Deployed in private subnets
  - Integrated with ALB
- **Security Group**: Allows inbound on port 8000 from ALB only

Database Layer (rds.tf)

- **RDS PostgreSQL:**
  - Engine: PostgreSQL 14.15
  - Instance: db.t3.micro
  - Storage: 30 GB (auto-scaling up to 50 GB), encrypted with KMS
  - Backup retention: 7 days
  - Multi-AZ deployment via subnet group
  - Not publicly accessible
- **DB Subnet Group:** Spans both RDS subnets
- **Security Group:** Allows inbound on port 5432 from ECS tasks only
- **Secrets Manager:** Stores database connection string securely
- **Random Password:** Generated for RDS master user

Load Balancer (alb.tf)

- **Application Load Balancer**:
  Deployed in public subnets
  Deletion protection: disabled
- **Target Group**: Routes traffic to ECS tasks on port 8000
- **Listeners**:
  HTTP (port 80): Forwards to target group
  HTTPS (port 443): SSL termination with ACM certificate
- **Health Check**: Endpoint /login, 90s interval
- **Security Group**: Allows inbound HTTP/HTTPS from internet

DNS & SSL (route53.tf)

- **Route53 Hosted Zone**: hemantdesale.tech
- **DNS Record**: dev.hemantdesale.tech pointing to ALB
- **ACM Certificate**: SSL certificate for dev.hemantdesale.tech
- **DNS Validation**: Automated via Route53

Monitoring (clowdwatch.tf)

- **CloudWatch Log Group**: /aws/ecs/aug-ecs (30 day retention)
- **Log Query Definition**: Pre-configured query to filter ECS logs

IAM (iam.tf)

- **ECS Task Execution Role**: Allows ECS to pull ECR images and write CloudWatch logs
- **Policy Attachment**: AmazonECSTaskExecutionRolePolicy

Data Sources (data.tf)

- Existing Hosted Zone details
- Current AWS region and account identity

Key management (kms.tf)

- KMS key for RDS and Secrets Manager

Variables file (Variables.tf)

- To define various variables used in different *.tf files

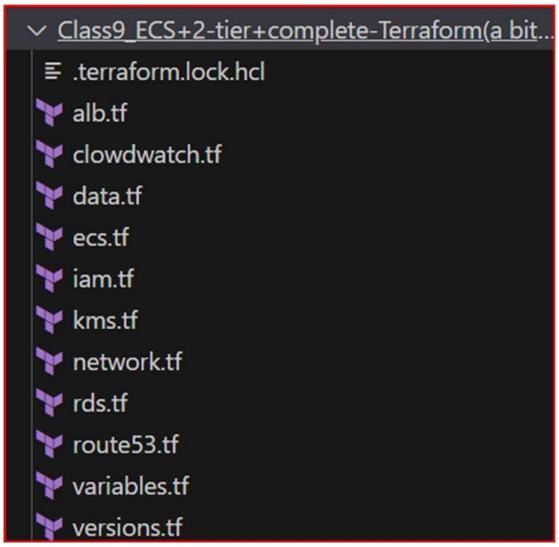
Version Uses (Version.tf)

- To use specific, terraform version for the project

State Management

- **Backend**: S3 bucket state-bucket-655113258765
- **State file**: aug/terraform.tfstate
- **Region**: ap-south-1
- **Encryption**: Enabled

Terraform Code Directory Structure



Terraform Execution

terraform init

- Initializes the working directory, downloads provider plugins, and sets up the backend.

terraform plan

- Shows the execution plan, previewing what resources will be created, changed, or destroyed.

terraform apply –auto-approve

- Applies the changes without asking for interactive approval.

Pre-requisites to run this terraform code

- Create S3 bucket named e.g. "state-bucket-655113258765"
- Use image pushed to ecr from class 6
- Create public hosted zone named "hemantdesale.tech"
  (It should match with your GoDaddy domain)
- Make sure the NS entries are correct in GoDaddy domain settings
  (Copy NS entries from AWS route 53 and add as custom ns entries in your GoDaddy domain.
  NS entries can be added after route 53 records are created in AWS i.e. after route53.tf execution)
- During RDS creation, do skip_final_snapshot = true (For Non-Prod)

GitHub Repository →

https://github.com/desalehemant87/Bootcamp/tree/c422b2dd14730b97f3908ff15fbfe6356bc495d7/Class9_ECS%2B2-tier%2Bcomplete-Terraform(a%20bit%20static)/Assignment_Class9

Application Output