

Final Project - PROG8870

**Terraform & CloudFormation
Implementation**

Nidhun Murali
8981611

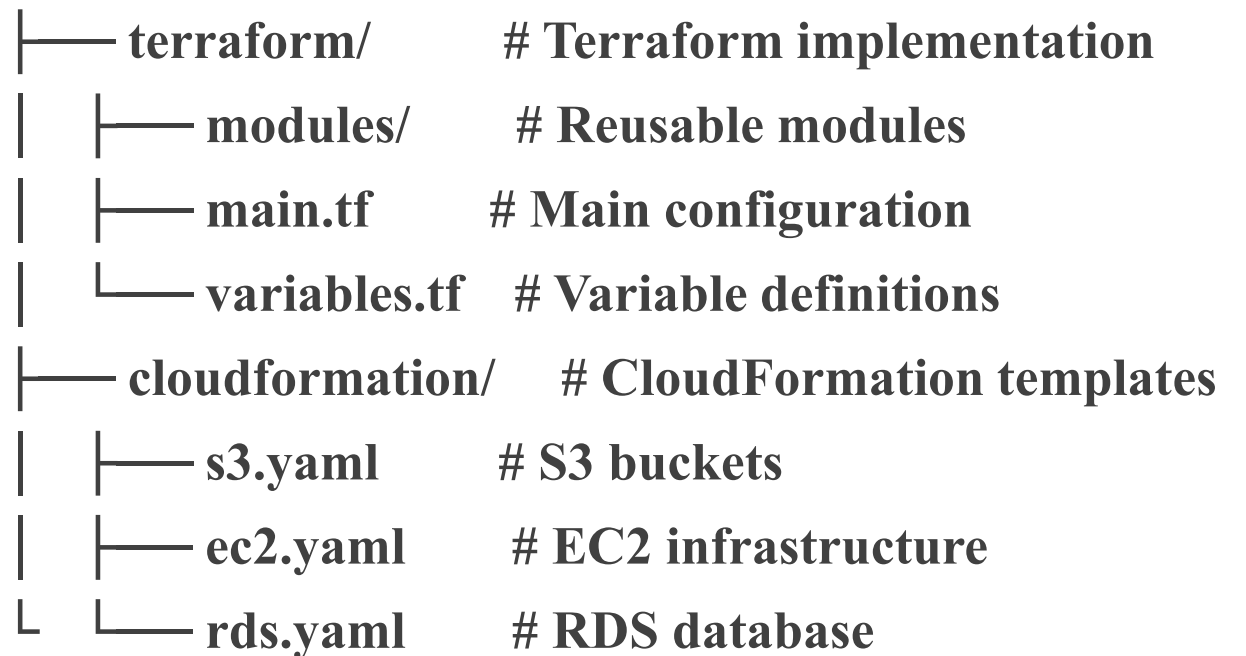
Presentation Overview

- ▶ **1. Code Structure & Implementation**
- ▶ **2. AWS Infrastructure Overview**
- ▶ **3. Key Features & Challenges**
- ▶ **4. Modularity & Best Practices**
- ▶ **5. Live Demo**

Project Structure

► Code Structure Overview

► Repository Structure:



Terraform Implementation

► Modular Design:

- `modules/vpc/` - VPC and networking
- `modules/ec2/` - EC2 instances
- `modules/rds/` - RDS database
- `modules/s3/` - S3 buckets

► Key Files:

- `main.tf` - Calls all modules
- `variables.tf` - Input parameters
- `terraform.tfvars` - Variable values
- `backend.tf` - State management

CloudFormation Implementation

► **Template-Based Approach:**

- s3.yaml - 3 private S3 buckets
- ec2.yaml - VPC + EC2 instance
- rds.yaml - MySQL database

► **Each Template Contains:**

- Parameters - Input variables
- Resources - AWS resources
- Outputs - Return values

► **Deployment: AWS CLI commands**

Infrastructure Components

- ▶ **Storage Layer:**
 - 3 private S3 buckets with versioning
 - Public access completely blocked
- ▶ **Compute Layer:**
 - VPC with public subnet (10.0.0.0/16)
 - EC2 instance (t2.micro) with public IP
 - Internet Gateway and Route Tables
- ▶ **Database Layer:**
 - MySQL RDS (db.t3.micro)
 - Public access enabled for demo
 - Multi-AZ subnet configuration

Network Architecture

- ▶ **VPC Configuration:**
 - CIDR: 10.0.0.0/16
 - Public subnet: 10.0.1.0/24
 - Internet Gateway attached
- ▶ **Security Groups:**
 - EC2: SSH access (port 22)
 - RDS: MySQL access (port 3306)
- ▶ **Connectivity:**
 - EC2 has public IP for SSH access

Key Features Implemented

Security Features:

- Private S3 buckets with versioning
- Security groups with specific ports
- No hardcoded credentials

Configuration Features:

- Parameterized templates/modules
- Dynamic resource naming
- Environment-specific deployments

Management Features:

- Infrastructure as Code
- Version control integration

Challenges & Solutions

- ▶ Challenge 1: S3 bucket naming conflicts
- ▶ Challenge 2: RDS public access requirements
- ▶ Challenge 3: MySQL version compatibility

Terraform Modularity

- ▶ **Modularity:**
 - **Separate modules for each service**
 - **Reusable across environments**
 - **Clear input/output interfaces**

CloudFormation Modularity

► **Modularity:**

- **Separate templates per service**
- **Independent stack deployment**
- **Parameter-driven configuration**

Live Demo



Thank You