

# PROG 8870 -- Final Project: Deploying AWS Infrastructure with Terraform and CloudFormation

**Title:** AWS Infrastructure Automation with Terraform and CloudFormation

## Overview

In this project, students will utilize Terraform and CloudFormation to create a scalable AWS infrastructure. The project will demonstrate a multi-service environment including S3 Buckets, EC2 Instances, and RDS Database Instances, applying Infrastructure as Code (IaC) best practices across different tools.

## Project Scope

You are required to complete deployment tasks across six modules using Terraform and CloudFormation, ensuring:

- Private and secure AWS storage and compute resources.
- Automated and modular infrastructure.
- Proper documentation, reusability, and dynamic configuration.
- A live demo showcasing resource provisioning.

## Tasks and Deliverables:

### 1. S3 Bucket Setup

#### - Using Terraform:

- o Create 4 Private S3 Buckets without public access.
- o Enable versioning on all buckets (Bonus Challenge).

#### - Using CloudFormation:

- o Create 3 Private S3 Buckets with PublicAccessBlockConfiguration.
- o Enable versioning on all buckets (Bonus Challenge)

### 2. VPC and EC2 Instance

#### - Using Terraform:

- o Set up an EC2 instance inside a custom VPC.
- o Use dynamic variables for AMI ID and instance type.
- o Enable public IP and allow SSH access (port 22).

#### - Using CloudFormation:

- o Deploy EC2 Instance with YAML-based configuration.
- o Attach necessary networking components (IGW, Route Tables).
- o Output EC2 Public IP as part of the CloudFormation outputs.

### 3. RDS Instance Deployment

#### - Using Terraform:

- o Create a MySQL RDS Database with db.t3.micro instance type.
- o Define database name, username, password via input variables.
- o Deploy into a dedicated DB Subnet Group.

#### - Using CloudFormation:

- o Deploy RDS instance using YAML templates.
- o Ensure public access is enabled (for this project only).
- o Configure security groups to allow MySQL traffic on port 3306.

### 4. Dynamic Configuration

- o Avoid hardcoding all values.
- o Use variables files (variables.tf, .tfvars) and CloudFormation Parameters where applicable.

### 5. Backend/State Management

- o Store Terraform state file locally on your laptop.
- o Use AWS CLI or AWS Console for CloudFormation stack deployment.

## 6. GitHub Repository

- Push your Terraform code and CloudFormation YAML files to a GitHub repository.
- Your repository must include:
  - main.tf (Terraform configuration)
  - variables.tf (Variables definition)
  - terraform.tfvars (Variables values; sensitive data should not be pushed to GitHub)
  - backend.tf (Backend configuration)
  - CloudFormation YAML files for S3, EC2, and RDS.
  - README.md (Documentation)
- Share your GitHub repository URL along with the submission document.

## 7. Presentation/Demo

- Prepare a **5-10 minute presentation** explaining:
  - Your code structure and implementation.
  - The AWS infrastructure you created.
  - Key features or challenges you encountered.
  - How your Terraform code and CloudFormation ensures modularity and best practices.
- **Live Demo:**
  - Run your Terraform configuration (terraform init, terraform plan, terraform apply).
  - Create and run CloudFormation Stack using YAML's.
  - Show the resources created in the AWS Management Console (e.g., the S3 bucket, EC2 instance, VPC, RDS, etc.).
  - Demonstrate the use of your tfvars file, YAML and backend configuration.

---

### Submission Details • Submission Items: One document containing below items

1. GitHub repository link.
2. Screenshot(s) showing:
  - S3 Buckets created and versioning enabled.
  - EC2 Instances launched with Public IP.
  - RDS Instances running.
  - Terraform and CloudFormation code snippets.
  - Terraform apply or CloudFormation deployment outputs.
3. **Terraform Files:**
  - main.tf, provider.tf, variables.tf, vars.tfvars
4. **CloudFormation Templates:**
  - YAML files for S3, EC2, and RDS.
5. Clear and concise documentation in the README.md (in GitHub repo)
6. Power Point Presentation(PPT) slides for demo.

---

### Assessment Criteria (Total: 30 points)

#### Weightage in Final grade: 35%

1. **Functionality** (10 points)
  - Are all resources deployed correctly?
2. **Best Practices** (5 points)
  - Use of variables and tfvars files.
  - Proper backend configuration.
  - Dynamic configuration, clean code.
3. **Documentation** (5 points)
  - Is the README.md clear and comprehensive?
  - Can a third party replicate the setup using your documentation?
  - Are comments added in the code?
4. **Presentation/Demo** (10 points)
  - Was the presentation well-structured and informative?
  - Did the demo showcase the infrastructure and code effectively?
  - Were challenges and solutions explained clearly during presentation in the class?
  - Was the presentation well-structured and informative?

---

## Resources

- **Terraform Documentation:** <https://registry.terraform.io/>
- **AWS Free Tier:** <https://aws.amazon.com/free/>
- **GitHub Guides:** <https://guides.github.com/>