

PROG 8870 - Cloud Architectures and Infrastructure as Code

Deploying AWS Infrastructure with Terraform and CloudFormation

Project Report

Prepared by : Nikhil Shankar Chirakkal Sivasankaran **Student ID:** 9026254

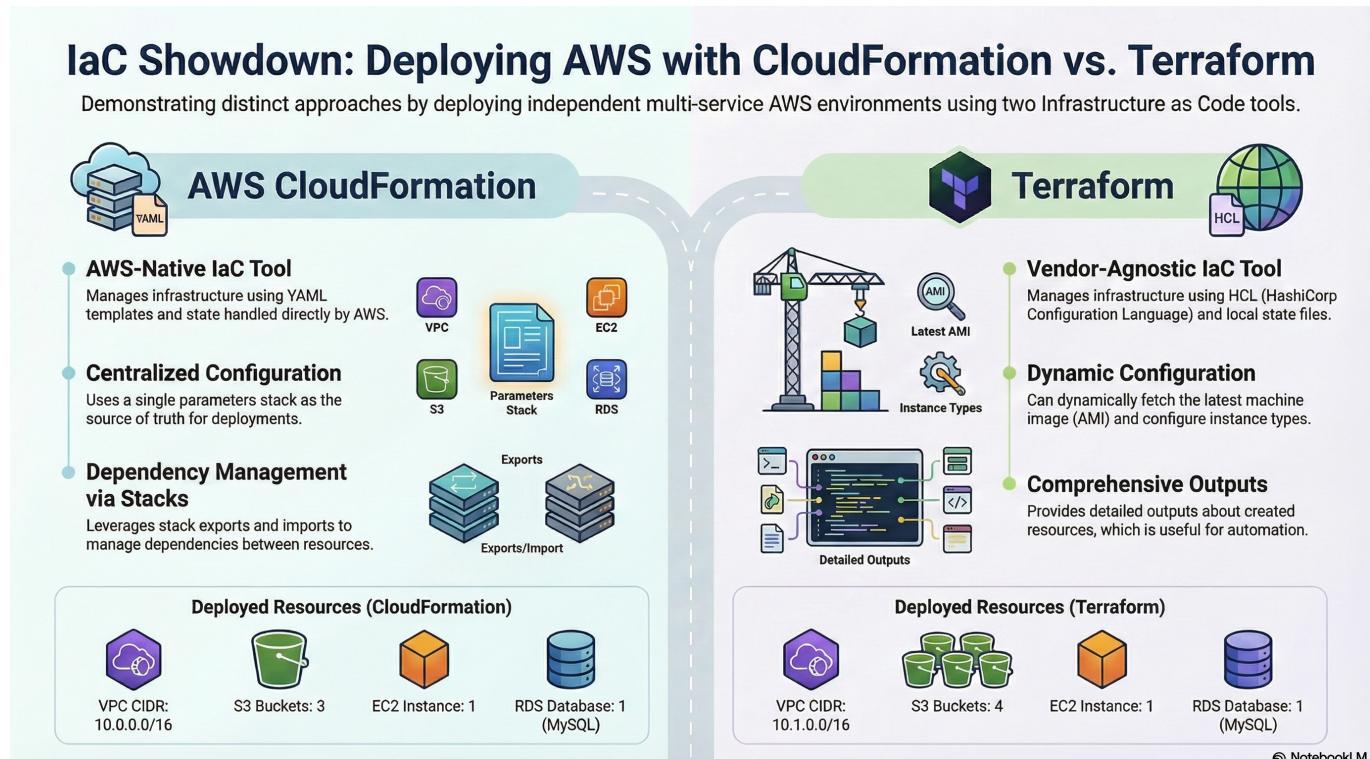
Overview

Deployed identical AWS infrastructure using CloudFormation and Terraform:

- **CloudFormation:** [cloudinfra-finalproject-cf-9026254-](#) (VPC: 10.0.0.0/16)
- **Terraform:** [cloudinfra-finalproject-terra-9026254-](#) (VPC: 10.1.0.0/16)

Resources: S3 buckets, VPC + subnets, EC2 instance, RDS MySQL database

Architecture



Network Layout:

- 1 Public subnet (EC2)
- 2 Private subnets in different AZs (RDS)
- Internet Gateway for public access
- Security groups for EC2 (SSH) and RDS (MySQL)

Code Structure

CloudFormation

```
cloudformation/
└── parameters.yaml      # Global config (single source of truth)
└── s3-buckets.yaml      # Storage
└── network.yaml          # VPC/subnets
└── ec2-instance.yaml     # Compute
└── rds-instance.yaml     # Database
```

Key Feature: Cross-Stack References

```
# parameters.yaml exports prefix
Outputs:
  ProjectPrefix:
    Export:
      Name: 'GlobalProjectPrefix'

# Other stacks import it
Parameters:
  ProjectPrefix:
    Default: !ImportValue GlobalProjectPrefix
```

The screenshot shows the AWS CloudFormation console interface. On the left, there's a sidebar with navigation links for CloudFormation, Stacks, Infrastructure Composer, Hooks, Registry, and Feedback. The main area displays a list of stacks under the heading 'Stacks (5)'. A modal at the top right informs users about new features. The table lists the following stacks:

| Stack name | Status | Created time | Description |
|---|------------------------------|------------------------------|--|
| cloudinfra-finalproject-cf-9026254-parameters | CREATE_COMPLETE | 2025-12-04 16:48:45 UTC-0500 | Global parameters stack - source of truth for project configuration |
| assignment3-networking | CREATE_COMPLETE | 2025-12-03 20:15:29 UTC-0500 | Networking Infrastructure Subnet, IGW, Route Table |
| CdkPipelineLabStack | CREATE_COMPLETE | 2025-11-30 20:46:45 UTC-0500 | - |
| PipelineStack | UPDATE_COMPLETE | 2025-11-30 20:18:14 UTC-0500 | - |
| CDKToolkit | CREATE_COMPLETE | 2025-11-30 20:16:51 UTC-0500 | This stack includes resources needed to deploy AWS CDK into this environment |

Terraform

```
terraform/
├── variables.tf      # Variable definitions
├── terraform.tfvars # Values (not in Git)
└── s3.tf              # Storage
└── vpc.tf             # VPC/subnets
└── ec2.tf             # Compute
└── rds.tf             # Database
└── outputs.tf         # Connection details
```

Key Feature: Dynamic AMI

```
data "aws_ami" "amazon_linux_2" {
  most_recent = true
  owners      = ["amazon"]
  filter {
    name    = "name"
    values  = ["amzn2-ami-hvm-*-*x86_64-gp2"]
  }
}
```



```
        }
    }

Plan: 24 to add, 0 to change, 0 to destroy.

Changes to Outputs:
+ aws_region           = "us-east-1"
+ ec2_instance_id      = (known after apply)
+ ec2_public_dns        = (known after apply)
+ ec2_public_ip         = (known after apply)
+ ec2_ssh_command       = (known after apply)
+ private_subnet_ids   = [
    + (known after apply),
    + (known after apply),
]
+ project_prefix        = "cloudinfra-finalproject-terra-9026254"
+ public_subnet_id      = (known after apply)
+ rds_address           = (known after apply)
+ rds_connection_string = (sensitive value)
+ rds_database_name     = "myappdatabase"
+ rds_endpoint          = (known after apply)
+ rds_port               = 3306
+ s3_bucket_names       = [
    + (known after apply),
    + (known after apply),
    + (known after apply),
    + (known after apply),
]
+ vpc_id                = (known after apply)

Ln 13, Col 29  Spaces: 4  UTF-8  LF  { } terraform-vars  ⚙️  ⚡ Prettier  □
```

```
aws_db_instance.mysql: still creating... [04m20s elapsed]
aws_db_instance.mysql: Still creating... [04m30s elapsed]
aws_db_instance.mysql: Still creating... [04m40s elapsed]
aws_db_instance.mysql: Still creating... [04m50s elapsed]
aws_db_instance.mysql: Still creating... [05m00s elapsed]
aws_db_instance.mysql: Still creating... [05m10s elapsed]
aws_db_instance.mysql: Still creating... [05m20s elapsed]
aws_db_instance.mysql: Still creating... [05m30s elapsed]
aws_db_instance.mysql: Still creating... [05m40s elapsed]
aws_db_instance.mysql: Still creating... [05m50s elapsed]
aws_db_instance.mysql: Still creating... [06m00s elapsed]
aws_db_instance.mysql: Still creating... [06m10s elapsed]
aws_db_instance.mysql: Still creating... [06m20s elapsed]
aws_db_instance.mysql: Still creating... [06m30s elapsed]
aws_db_instance.mysql: Still creating... [06m40s elapsed]
aws_db_instance.mysql: Still creating... [06m50s elapsed]
aws_db_instance.mysql: Still creating... [07m00s elapsed]
```

```
aws_db_instance.mysql: Still creating... [07m00s elapsed]
aws_db_instance.mysql: Still creating... [07m10s elapsed]
aws_db_instance.mysql: Still creating... [07m20s elapsed]
```

Modularity & Best Practices

CloudFormation

- **5 separate stacks:** Parameters → S3 → Network → EC2 → RDS
- **Exports/Imports:** Stack outputs used by other stacks
- **Benefits:** Update one stack independently, clear dependencies

Terraform

- **File separation:** Each service in separate `.tf` file
- **Centralized variables:** Single source in `variables.tf`
- **Benefits:** Easy to find/modify, single deployment command

Both

- Consistent naming with configurable prefix No hardcoded values Security: RDS in private subnets, S3 public access blocked Sensitive data excluded from Git
-

Deployed Resources

S3 Buckets

CloudFormation: 3 buckets | **Terraform:** 4 buckets

- Versioning enabled
- Public access blocked

The screenshot shows the AWS CloudFormation console interface. On the left, there's a sidebar with sections like 'CloudFormation', 'Stacks', 'Infrastructure Composer', 'Hooks', 'Registry', 'Spotlight', and 'Feedback'. The main area is titled 'Stacks (6)' and lists six stacks. One stack, 'cloudinfra-finalproject-cf-9026254-s3-stack', is highlighted with a red border. The table columns are 'Stack name', 'Status', 'Created time', and 'Description'. The 'Status' column shows 'CREATE_COMPLETE' for most stacks. The 'Description' column provides details about each stack, such as 'CloudFormation template creating 3 private S3 buckets with versioning enabled' for the highlighted stack.

| Stack name | Status | Created time | Description |
|---|------------------------------|------------------------------|---|
| cloudinfra-finalproject-cf-9026254-s3-stack | CREATE_COMPLETE | 2025-12-04 17:00:10 UTC-0500 | CloudFormation template creating 3 private S3 buckets with versioning enabled |
| cloudinfra-finalproject-cf-9026254-parameters | CREATE_COMPLETE | 2025-12-04 16:48:45 UTC-0500 | Global parameters stack - source of truth for project configuration |
| assignment3-networking | CREATE_COMPLETE | 2025-12-03 20:15:29 UTC-0500 | Networking Infrastructure Subnet, IGW, Route Table |
| CdkPipelineLabStack | CREATE_COMPLETE | 2025-11-30 20:46:45 UTC-0500 | - |
| PipelineStack | UPDATE_COMPLETE | 2025-11-30 20:18:14 UTC-0500 | - |
| CDKToolkit | CREATE_COMPLETE | 2025-11-30 20:16:51 UTC-0500 | This stack includes resources needed to deploy AWS CDK into this environment |

The screenshot shows the AWS Amazon S3 console interface. On the left, there's a sidebar with sections like 'Amazon S3', 'Buckets', 'Access management and security', 'Storage management and insights', 'Account and organization settings', and 'AWS Marketplace for S3'. The main area is titled 'General purpose buckets' and lists ten buckets. One bucket, 'cloudinfra-finalproject-terra-9026254-bucket-1', is highlighted with a red border. The table columns are 'Name', 'AWS Region', and 'Creation date'. The 'AWS Region' column shows 'US East (N. Virginia)' and 'us-east-1' for all buckets. The 'Creation date' column shows dates ranging from December 9, 2025, to January 3, 2025.

| Name | AWS Region | Creation date |
|--|------------------------------------|--|
| cdk-hnb659fds-assets-86734462708-us-east-1 | US East (N. Virginia) us-east-1 | November 30, 2025, 20:17:04 (UTC-05:00) |
| cloudinfra-finalproject-terra-9026254-bucket-1 | US East (N. Virginia) us-east-1 | December 9, 2025, 01:55:31 (UTC-05:00) |
| cloudinfra-finalproject-terra-9026254-bucket-2 | US East (N. Virginia) us-east-1 | December 9, 2025, 01:55:31 (UTC-05:00) |
| cloudinfra-finalproject-terra-9026254-bucket-3 | US East (N. Virginia) us-east-1 | December 9, 2025, 01:55:31 (UTC-05:00) |
| cloudinfra-finalproject-terra-9026254-bucket-4 | US East (N. Virginia) us-east-1 | December 9, 2025, 01:55:31 (UTC-05:00) |
| himy-react | US East (Ohio) us-east-2 | January 3, 2025, 05:09:32 (UTC-05:00) |
| mindfulminutes-terraform-state | US East (N. Virginia) us-east-1 | December 6, 2025, 18:50:11 (UTC-05:00) |

- VPC with Internet Gateway
- Public subnet for EC2
- 2 private subnets (different AZs) for RDS

The screenshot shows the AWS CloudFormation console with the URL us-east-1.console.aws.amazon.com/cloudformation/home?region=us-east-1#/stacks?filteringText=&filteringStatus=active&viewNested=true. The left sidebar includes sections for CloudFormation, Stacks, Infrastructure Composer, Hooks, Registry, Spotlight, and Feedback. The main content area displays a table of stacks with columns for Stack name, Status, Created time, and Description. The first stack, 'cloudinfra-finalproject-cf-9026254-network-stack', is highlighted with a red box. Its status is 'CREATE_COMPLETE' and it was created on 2025-12-04 at 17:02:53 UTC-0500. The description indicates it's a CloudFormation template for VPC, Subnets, Internet Gateways, Route Tables, and Security Groups.

| Stack name | Status | Created time | Description |
|--|--------------------------------|------------------------------|--|
| cloudinfra-finalproject-cf-9026254-network-stack | ✓ CREATE_COMPLETE | 2025-12-04 17:02:53 UTC-0500 | CloudFormation template VPC, Subnets, Internet Ga Route Tables, and Securit Groups |
| cloudinfra-finalproject-cf-9026254-s3-stack | ✓ CREATE_COMPLETE | 2025-12-04 17:00:10 UTC-0500 | CloudFormation template creating 3 private S3 buck with versioning enabled |
| cloudinfra-finalproject-cf-9026254-parameters | ✓ CREATE_COMPLETE | 2025-12-04 16:48:45 UTC-0500 | Global parameters stack - source of truth for project configuration |
| assignment3-networking | ✓ CREATE_COMPLETE | 2025-12-03 20:15:29 UTC-0500 | Networking Infrastructure Subnet, IGW, Route Table |
| CdkPipelineLabStack | ✓ CREATE_COMPLETE | 2025-11-30 20:46:45 UTC-0500 | - |
| PipelineStack | ✓ UPDATE_COMPLETE | 2025-11-30 20:18:14 UTC-0500 | - |
| CDKToolkit | ✓ CREATE_COMPLETE | 2025-11-30 20:16:51 UTC-0500 | This stack includes resources needed to deploy AWS CDK into this environment |

EC2 Instance

- Type: t2.micro (Free Tier)
- AMI: Amazon Linux 2
- Public subnet with public IP
- SSH access (port 22)

The screenshot shows the AWS CloudFormation console interface. On the left, there's a sidebar with sections like Stacks, Infrastructure Composer, Hooks, Registry, and Feedback. The main area displays a list of stacks with the following details:

| Stack name | Status | Created time |
|--|--------------------------------|---------------------|
| cloudinfra-finalproject-cf-9026254-ec2-stack | ✓ CREATE_COMPLETE | 2025-12-04 22:30:11 |
| cloudinfra-finalproject-cf-9026254-network-stack | ✓ CREATE_COMPLETE | 2025-12-04 17:02:11 |
| cloudinfra-finalproject-cf-9026254-s3-stack | ✓ CREATE_COMPLETE | 2025-12-04 17:00:11 |
| cloudinfra-finalproject-cf-9026254-parameters | ✓ CREATE_COMPLETE | 2025-12-04 16:48:11 |
| assignment3-networking | ✓ CREATE_COMPLETE | 2025-12-03 20:15:11 |
| CdkPipelineLabStack | ✓ CREATE_COMPLETE | 2025-11-30 20:46:11 |

At the bottom, there are links for CloudShell, Feedback, and Console Mobile App, along with standard AWS footer links for Privacy, Terms, and Cookie preferences.

The screenshot shows the AWS EC2 Instances page. The left sidebar is collapsed, and the main content area displays a table of instances. A red box highlights the breadcrumb navigation 'EC2 > Instances' and the table header. Another red box highlights the first instance in the list, which has its name 'cloudinfra-finalproject-t-terra...' partially obscured by a red box.

| Name | Instance ID | Instance state | Instance type | Status check | Alarm sta |
|---|---------------------|----------------|---------------|--------------|---------------------------|
| cloudinfra-finalproject-t-terra-026254-ec2... | i-0a7a5810b7fe5bab1 | Running | t2.micro | Initializing | View alar |

RDS Database

- Engine: MySQL 8.0
- Instance: db.t3.micro (Free Tier)
- Private subnets only (not publicly accessible)

The screenshot shows the AWS CloudFormation console interface. On the left, there's a sidebar with sections like 'Stacks', 'Infrastructure Composer', 'Hooks', 'Registry', and 'Feedback'. The main area displays a list of 'Stacks (9)'. A specific stack, 'cloudinfra-finalproject-cf-9026254-rds-stack', is highlighted with a red border. The table columns are 'Stack name', 'Status', and 'Created time'. The status for this stack is 'CREATE_COMPLETE'. Other visible stacks include 'cloudinfra-finalproject-cf-9026254-ec2-stack', 'cloudinfra-finalproject-cf-9026254-network-stack', 'cloudinfra-finalproject-cf-9026254-s3-stack', and 'cloudinfra-finalproject-cf-9026254-parameters'.

| Stack name | Status | Created time |
|--|-----------------|--------------------|
| cloudinfra-finalproject-cf-9026254-rds-stack | CREATE_COMPLETE | 2025-12-04 22:45:3 |
| cloudinfra-finalproject-cf-9026254-ec2-stack | CREATE_COMPLETE | 2025-12-04 22:30:5 |
| cloudinfra-finalproject-cf-9026254-network-stack | CREATE_COMPLETE | 2025-12-04 17:02:5 |
| cloudinfra-finalproject-cf-9026254-s3-stack | CREATE_COMPLETE | 2025-12-04 17:00:1 |
| cloudinfra-finalproject-cf-9026254-parameters | CREATE_COMPLETE | 2025-12-04 16:48:4 |

The screenshot shows the AWS RDS console under the 'Aurora and RDS' section. In the 'Databases' tab, there is one database entry:

| DB identifier | Status | Role |
|---|----------|----------|
| cloudinfra-finalproject-terra-026254-rds- | Creating | Instance |

A red box highlights the 'DB identifier' column. To the right, the Amazon Q AI assistant is displayed with various conversational options like 'Chatting about your resources' and 'Understanding and optimizing your costs'.

Deployment

CloudFormation (Sequential):

```
aws cloudformation create-stack --stack-name ...-params-stack --template-body
file://parameters.yaml
aws cloudformation create-stack --stack-name ...-s3-stack --template-body
file://s3-buckets.yaml
aws cloudformation create-stack --stack-name ...-network-stack --template-body
file://network.yaml
aws cloudformation create-stack --stack-name ...-ec2-stack --template-body
file://ec2-instance.yaml
aws cloudformation create-stack --stack-name ...-rds-stack --template-body
file://rds-instance.yaml
```

Terraform (Single Command):

```
terraform init
terraform plan
terraform apply
```

Challenges & Solutions

| Challenge | Solution |
|-------------------------|---|
| Stack dependency order | Created parameters stack first, used Exports/Imports |
| RDS needs 2+ AZs | Created 2 private subnets in different AZs |
| Key pair must pre-exist | Created in AWS Console before deployment |
| Sensitive data security | Used .gitignore for tfvars, NoEcho for CloudFormation |

Key Learnings

Technical:

- IaC modularity patterns
- AWS service dependencies
- State management (CloudFormation: AWS-managed, Terraform: local file)

Best Practices:

- Separate concerns (one service per file/stack)
- Centralized configuration
- Security-first design
- Version control with sensitive data protection

Tool Comparison:

- **CloudFormation:** AWS-native, auto state management, sequential stacks
- **Terraform:** Multi-cloud, HCL syntax, single deployment

Conclusion

Successfully deployed identical infrastructure with two IaC tools, demonstrating:

- Modular, maintainable code
- Security best practices
- Production-ready templates

Both tools achieve the same result with different approaches. Choice depends on use case: CloudFormation for AWS-only, Terraform for multi-cloud flexibility.

Project Repository: [\[GitHub URL\]](#) **Student ID:** 9026254