

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

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

**Name:** Suman Kumari Jakhar

**Prof:** Vikas Vattikonda

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).

```
PS C:\Assignments\Finals\Final_Project_Deploying_AWS_Infrastructure\terraform\s3> terraform plan
aws_s3_bucket.buckets["student-proj-bucket-b"]: Refreshing state... [id=student-proj-bucket-b]
aws_s3_bucket.buckets["student-proj-bucket-d"]: Refreshing state... [id=student-proj-bucket-d]
aws_s3_bucket.buckets["student-proj-bucket-c"]: Refreshing state... [id=student-proj-bucket-c]
aws_s3_bucket.buckets["student-proj-bucket-a"]: Refreshing state... [id=student-proj-bucket-a]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
+ create

Terraform will perform the following actions:

# aws_s3_bucket.buckets["student-proj-bucket-a"] will be created
+ resource "aws_s3_bucket" "buckets" {
  + acceleration_status      = (known after apply)
  + acl                      = (known after apply)
  + arn                     = (known after apply)
  + bucket                  = "student-proj-bucket-a"
  + bucket_domain_name      = (known after apply)
  + bucket_prefix           = (known after apply)
  + bucket_region           = (known after apply)
  + bucket_regional_domain_name = (known after apply)
  + force_destroy           = false
  + hosted_zone_id          = (known after apply)
  + id                      = (known after apply)
  + object_lock_enabled      = false
  + policy                  = (known after apply)
  + region                  = "us-east-1"
  + request_payer            = (known after apply)
  + tags_all                = (known after apply)
  + website_domain           = (known after apply)
  + website_endpoint         = (known after apply)

  + cors_rule (known after apply)

  + grant (known after apply)

  + lifecycle_rule (known after apply)

  + logging (known after apply)

  + object_lock_configuration (known after apply)

  + replication_configuration (known after apply)
```

```
+ mfa_delete = (known after apply)
+ status      = "Enabled"
}
}

# aws_s3_bucket_versioning.versioning["student-proj-bucket-b"] will be created
+ resource "aws_s3_bucket_versioning" "versioning" {
+   bucket = (known after apply)
+   id      = (known after apply)
+   region  = "us-east-1"

+   versioning_configuration {
+     mfa_delete = (known after apply)
+     status      = "Enabled"
+   }
}

# aws_s3_bucket_versioning.versioning["student-proj-bucket-c"] will be created
+ resource "aws_s3_bucket_versioning" "versioning" {
+   bucket = (known after apply)
+   id      = (known after apply)
+   region  = "us-east-1"

+   versioning_configuration {
+     mfa_delete = (known after apply)
+     status      = "Enabled"
+   }
}

# aws_s3_bucket_versioning.versioning["student-proj-bucket-d"] will be created
+ resource "aws_s3_bucket_versioning" "versioning" {
+   bucket = (known after apply)
+   id      = (known after apply)
+   region  = "us-east-1"

+   versioning_configuration {
+     mfa_delete = (known after apply)
+     status      = "Enabled"
+   }
}

}

Plan: 16 to add, 0 to change, 0 to destroy.
```

aws

Search

[Alt+S]

United States (N. Virginia)

Account ID: 7950-9629-6861

voclabs/user3807391-Suman\_Jakhar

Amazon S3

Buckets

General purpose buckets

Directory buckets

General purpose buckets (4)

Info

Copy ARN

Empty

Delete

Create bucket

Buckets are containers for data stored in S3.

Find buckets by name

1

Name	AWS Region	Creation date
student-proj-bucket-a	US East (N. Virginia) us-east-1	August 12, 2025, 11:50:59 (UTC-04:00)
student-proj-bucket-b	US East (N. Virginia) us-east-1	August 12, 2025, 11:50:59 (UTC-04:00)
student-proj-bucket-c	US East (N. Virginia) us-east-1	August 12, 2025, 11:50:59 (UTC-04:00)
student-proj-bucket-d	US East (N. Virginia) us-east-1	August 12, 2025, 11:50:59 (UTC-04:00)

Account snapshot

Updated daily

View dashboard

Storage Lens provides visibility into storage usage and activity trends.

External access summary - new

Updated daily

Info

External access findings help you identify bucket permissions that allow public access or access from other AWS accounts.

Amazon S3

Buckets

student-proj-bucket-a

General purpose buckets

Directory buckets

student-proj-bucket-a

Info

Objects

Metadata

Properties

Permissions

Metrics

Management

Access Points

Bucket overview

AWS Region

US East (N. Virginia) us-east-1

Amazon Resource Name (ARN)

arn:aws:s3:::student-proj-bucket-a

Creation date

August 12, 2025, 11:50:59 (UTC-04:00)

Bucket Versioning

Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures. Learn more

Bucket Versioning

Enabled

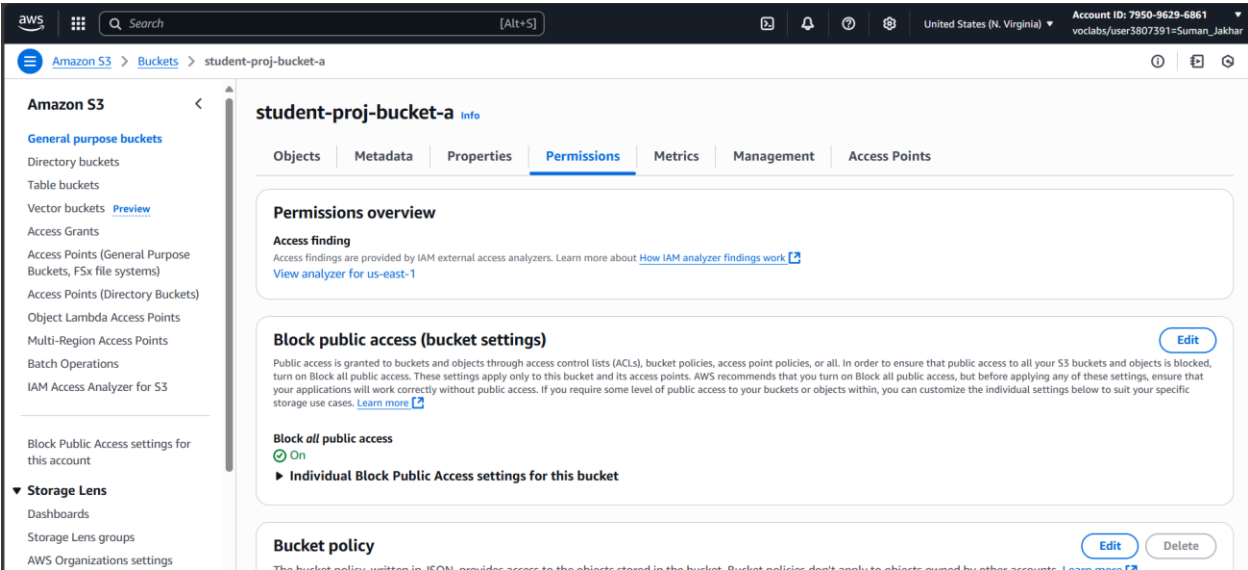
Multi-factor authentication (MFA) delete

An additional layer of security that requires multi-factor authentication for changing Bucket Versioning settings and permanently deleting object versions. To modify MFA delete settings, use the AWS CLI, AWS SDK, or the Amazon S3 REST API. Learn more

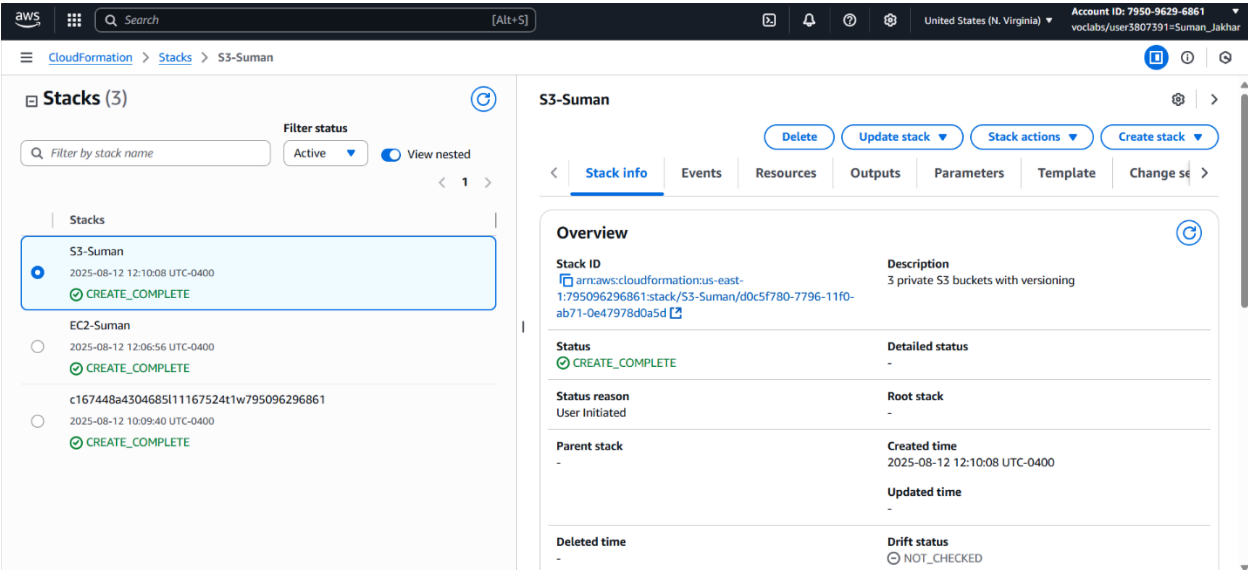
Disabled

Tags (0)

Edit



- 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).

```
PS C:\Assignments\Finals\Final_Project_Deploying_AWS_Infrastructure\terraform\ec2> terraform apply

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
+ create

Terraform will perform the following actions:

# aws_instance.ec2 will be created
+ resource "aws_instance" "ec2" {
  + ami              = "ami-0de716d6197524dd9"
  + arn              = (known after apply)
  + associate_public_ip_address = true
  + availability_zone = (known after apply)
  + disable_api_stop  = (known after apply)
  + disable_api_termination = (known after apply)
  + ebs_optimized     = (known after apply)
  + enable_primary_ipv6 = (known after apply)
  + force_destroy     = false
  + get_password_data = (known after apply)
  + host_id           = (known after apply)
  + host_resource_group_arn = (known after apply)
  + iam_instance_profile = (known after apply)
  + id               = (known after apply)
  + instance_initiated_shutdown_behavior = (known after apply)
  + instance_lifecycle = (known after apply)
  + instance_state    = (known after apply)
  + instance_type     = "t2.micro"
  + ipv6_address_count = (known after apply)
  + ipv6_addresses    = (known after apply)
  + key_name          = (known after apply)
  + monitoring        = (known after apply)
```

```
aws_vpc.vpc: Creating...
aws_vpc.vpc: Creation complete after 2s [id=vpc-0a7859ae9287c8287]
aws_internet_gateway.igw: Creating...
aws_subnet.subnet: Creating...
aws_security_group.sg: Creating...
aws_internet_gateway.igw: Creation complete after 0s [id=igw-09d813de7a08a589d]
aws_route_table.rt: Creating...
aws_route_table.rt: Creation complete after 1s [id=rtb-0f0167e3b4bf8ef40]
aws_security_group.sg: Creation complete after 3s [id=sg-0b5c93f99be30f89e]
aws_subnet.subnet: Still creating... [10s elapsed]
aws_subnet.subnet: Creation complete after 11s [id=subnet-07f8d93e36c13a4c4]
aws_route_table_association.assoc: Creating...
aws_instance.ec2: Creating...
aws_route_table_association.assoc: Creation complete after 0s [id=rtbassoc-0a9002f52126b7d07]
aws_instance.ec2: Still creating... [10s elapsed]
aws_instance.ec2: Creation complete after 13s [id=i-0abd782f35d85ca39]

Apply complete! Resources: 7 added, 0 changed, 0 destroyed.

Outputs:

public_ip = "18.212.202.95"
```



EC2

Dashboard

EC2 Global View

Events

Instances

Instances

Instance Types

Launch Templates

Spot Requests

Savings Plans

Reserved Instances

Dedicated Hosts

Capacity Reservations

Images

AMIs

AMI Catalog

Elastic Block Store

Volumes

Instance summary for i-0abd782f35d85ca39

Updated 1 minute ago

Instance ID

i-0abd782f35d85ca39

IPv6 address

-

Hostname type

IP name: ip-10-0-1-76.ec2.internal

Answer private resource DNS name

-

Auto-assigned IP address

18.212.202.95 [Public IP]

IAM Role

-

IMDSv2

Required

Public IPv4 address

18.212.202.95 | open address

Instance state

Running

Private IP DNS name (IPv4 only)

ip-10-0-1-76.ec2.internal

Instance type

t2.micro

VPC ID

vpc-0a7859ae9287c8287 (proj-vpc)

Subnet ID

subnet-07f8d93e36c13a4c4 (proj-subnet)

Instance ARN

arn:aws:ec2:us-east-1:795096296861:instance/i-0abd782f35d85ca39

Private IPv4 addresses

10.0.1.76

Public DNS

-

Elastic IP addresses

-

AWS Compute Optimizer finding

Opt-in to AWS Compute Optimizer for recommendation

Learn more

Auto Scaling Group name

-

Managed

false



VPC dashboard

EC2 Global View

Filter by VPC

Virtual private cloud

Your VPCs

Subnets

Route tables

Internet gateways

Egress-only internet gateways

Carrier gateways

DHCP option sets

Elastic IPs

Managed prefix lists

NAT gateways

Peering connections

Route servers

subnet-07f8d93e36c13a4c4 / proj-subnet

Details

Subnet ID

subnet-07f8d93e36c13a4c4

IPv4 CIDR

10.0.1.0/24

Availability Zone

us-east-1a

Network ACL

acl-08cb11e4d287146a8

Auto-assign customer-owned IPv4 address

No

IPv6 CIDR reservations

-

Resource name DNS AAAA record

Disabled

Subnet ARN

arn:aws:ec2:us-east-1:795096296861:subnet/subnet-07f8d93e36c13a4c4

Available IPv4 addresses

250

Network border group

us-east-1

Default subnet

No

Customer-owned IPv4 pool

-

IPv6-only

No

DNS64

Disabled

State

Available

IPv6 CIDR

-

VPC

vpc-0a7859ae9287c8287 | proj-vpc

Auto-assign public IPv4 address

Yes

Outpost ID

-

Hostname type

IP name

Owner

795096296861

Block Public Access

Off

IPv6 CIDR association ID

-

Route table

rtb-0f0167e3b4bf8ef40

Auto-assign IPv6 address

No

IPv4 CIDR reservations

-

Resource name DNS A record

Disabled



VPC dashboard

EC2 Global View

Filter by VPC

Virtual private cloud

Your VPCs

Subnets

Route tables

Internet gateways

Egress-only internet gateways

Carrier gateways

DHCP option sets

Elastic IPs

Managed prefix lists

NAT gateways

Peering connections

Route servers

igw-09d813de7a08a589d

Details

Internet gateway ID

igw-09d813de7a08a589d

State

Attached

VPC ID

vpc-0a7859ae9287c8287 | proj-vpc

Owner

795096296861

Tags

Search tags

Key

Value

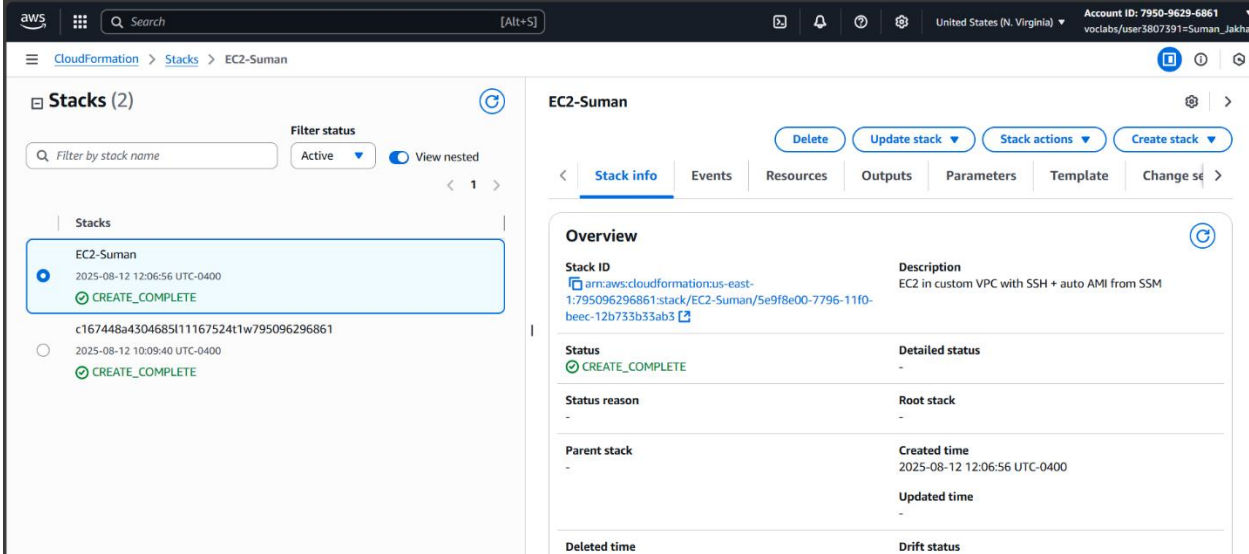
No tags associated with this resource

Manage tags

- 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.

```
PS C:\Assignments\Finals\Final_Project_Deploying_AWS_Infrastructure\terraform\rds> terraform apply

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
+ create

Terraform will perform the following actions:

# aws_db_instance.mysql will be created
+ resource "aws_db_instance" "mysql" {
  + address                               = (known after apply)
  + allocated_storage                     = 20
  + apply_immediately                     = false
  + arn                                   = (known after apply)
  + auto_minor_version_upgrade           = true
  + availability_zone                     = (known after apply)
  + backup_retention_period               = (known after apply)
  + backup_target                         = (known after apply)
  + backup_window                         = (known after apply)
  + ca_cert_identifier                    = (known after apply)
  + character_set_name                    = (known after apply)
  + copy_tags_to_snapshot                 = false
  + database_insights_mode                = (known after apply)
  + db_name                               = "Suman_db"
  + db_subnet_group_name                  = "rds-subnet-group"
  + dedicated_log_volume                  = false
  + delete_automated_backups              = true
  + domain_fqdn                           = (known after apply)
  + endpoint                             = (known after apply)
  + engine                               = "mysql"
  + engine_lifecycle_support              = (known after apply)
  + engine_version                       = (known after apply)
  + engine_version_actual                 = (known after apply)
  + hosted_zone_id                       = (known after apply)
  + id                                    = (known after apply)
  + identifier                            = (known after apply)
  + identifier_prefix                     = (known after apply)
  + instance_class                        = "db.t3.micro"
  + iops                                 = (known after apply)
  + kms_key_id                           = (known after apply)
```

aws\_vpc.rds\_vpc: Creating...  
aws\_vpc.rds\_vpc: Creation complete after 2s [id=vpc-0f1f235f68040e1fd]  
aws\_subnet.db\_subnet\_b: Creating...  
aws\_subnet.db\_subnet\_a: Creating...  
aws\_subnet.db\_subnet\_b: Creation complete after 1s [id=subnet-0d62bd4348dbb2abb]  
aws\_subnet.db\_subnet\_a: Creation complete after 1s [id=subnet-049dc3c77ae8be4b5]  
aws\_db\_subnet\_group.db\_group: Creating...  
aws\_db\_subnet\_group.db\_group: Creation complete after 1s [id=rds-subnet-group]  
aws\_db\_instance.mysql: Creating...  
aws\_db\_instance.mysql: Still creating... [10s elapsed]  
aws\_db\_instance.mysql: Still creating... [20s elapsed]  
aws\_db\_instance.mysql: Still creating... [30s elapsed]  
aws\_db\_instance.mysql: Still creating... [40s elapsed]  
aws\_db\_instance.mysql: Still creating... [50s elapsed]  
aws\_db\_instance.mysql: Still creating... [1m0s elapsed]  
aws\_db\_instance.mysql: Still creating... [1m10s elapsed]  
aws\_db\_instance.mysql: Still creating... [1m20s elapsed]  
aws\_db\_instance.mysql: Still creating... [1m30s elapsed]  
aws\_db\_instance.mysql: Still creating... [1m40s elapsed]  
aws\_db\_instance.mysql: Still creating... [1m50s elapsed]  
aws\_db\_instance.mysql: Still creating... [2m0s elapsed]  
aws\_db\_instance.mysql: Still creating... [2m10s elapsed]  
aws\_db\_instance.mysql: Still creating... [2m20s elapsed]  
aws\_db\_instance.mysql: Still creating... [2m30s elapsed]  
aws\_db\_instance.mysql: Still creating... [2m40s elapsed]  
aws\_db\_instance.mysql: Still creating... [2m50s elapsed]  
aws\_db\_instance.mysql: Still creating... [3m0s elapsed]  
aws\_db\_instance.mysql: Still creating... [3m10s elapsed]  
aws\_db\_instance.mysql: Still creating... [3m20s elapsed]  
aws\_db\_instance.mysql: Still creating... [3m30s elapsed]  
aws\_db\_instance.mysql: Still creating... [3m40s elapsed]  
aws\_db\_instance.mysql: Still creating... [3m50s elapsed]  
aws\_db\_instance.mysql: Still creating... [4m0s elapsed]  
aws\_db\_instance.mysql: Still creating... [4m10s elapsed]  
aws\_db\_instance.mysql: Creation complete after 4m15s [id=db-WA2KIK6WHYTED4AIL7P7S7HTSQ]

Apply complete! Resources: 5 added, 0 changed, 0 destroyed.

aws

Search [Alt+S]

United States (N. Virginia) Account ID: 7950-9629-6861 voclabs/user3807391=Suman\_Jakhar

Aurora and RDS > Databases > terraform-20250812150616053900000001

Aurora and RDS

Dashboard

Databases

Query editor

Performance insights

Snapshots

Exports in Amazon S3

Automated backups

Reserved instances

Proxies

Subnet groups

Parameter groups

Option groups

Custom engine versions

Zero-ETL integrations New

Events

Event subscriptions

terraform-20250812150616053900000001

Summary

DB identifier

terraform-20250812150616053900000001

Status

Available

Role

Instance

Engine

MySQL Community

Recommendations

Class

db.t3.micro

Current activity

0 Connections

Region & AZ

us-east-1a

CPU

3.96%

Connectivity & security

MonitoringLogs & eventsConfigurationZero-ETL integrationsMaintenance & backupsData

Connectivity & security

Endpoint & port

Endpoint

terraform-2025081215061605390000001.c08wqmsvbnbx.us-east-1.rds.amazonaws.com

Port

3306

Networking

Availability Zone

us-east-1a

VPC

vpc-0f1f235f68040e1fd

Subnet group

Security

VPC security groups

default (sg-0533160d2aa39a6dd)

Active

Publicly accessible

No

aws

Search [Alt+S]

United States (N. Virginia) Account ID: 7950-9629-6861 voclabs/user3807391=Suman\_Jakhar

Aurora and RDS > Subnet groups > rds-subnet-group

Aurora and RDS

Dashboard

Databases

Query editor

Performance insights

Snapshots

Exports in Amazon S3

Automated backups

Reserved instances

Proxies

Subnet groups

Parameter groups

Option groups

Custom engine versions

Zero-ETL integrations New

Events

Event subscriptions

rds-subnet-group

Subnet group details

VPC ID

vpc-0f1f235f68040e1fd

ARN

arn:aws:rds:us-east-1:795096296861:subgrp:rds-subnet-group

Supported network types

IPv4

Description

Managed by Terraform

Subnets (2)

Availability zone	Subnet name	Subnet ID	CIDR block
us-east-1b	-	subnet-0d62bd4348dbb2abb	10.1.2.0/24
us-east-1a	-	subnet-049dc3c77ae8be4b5	10.1.1.0/24

aws

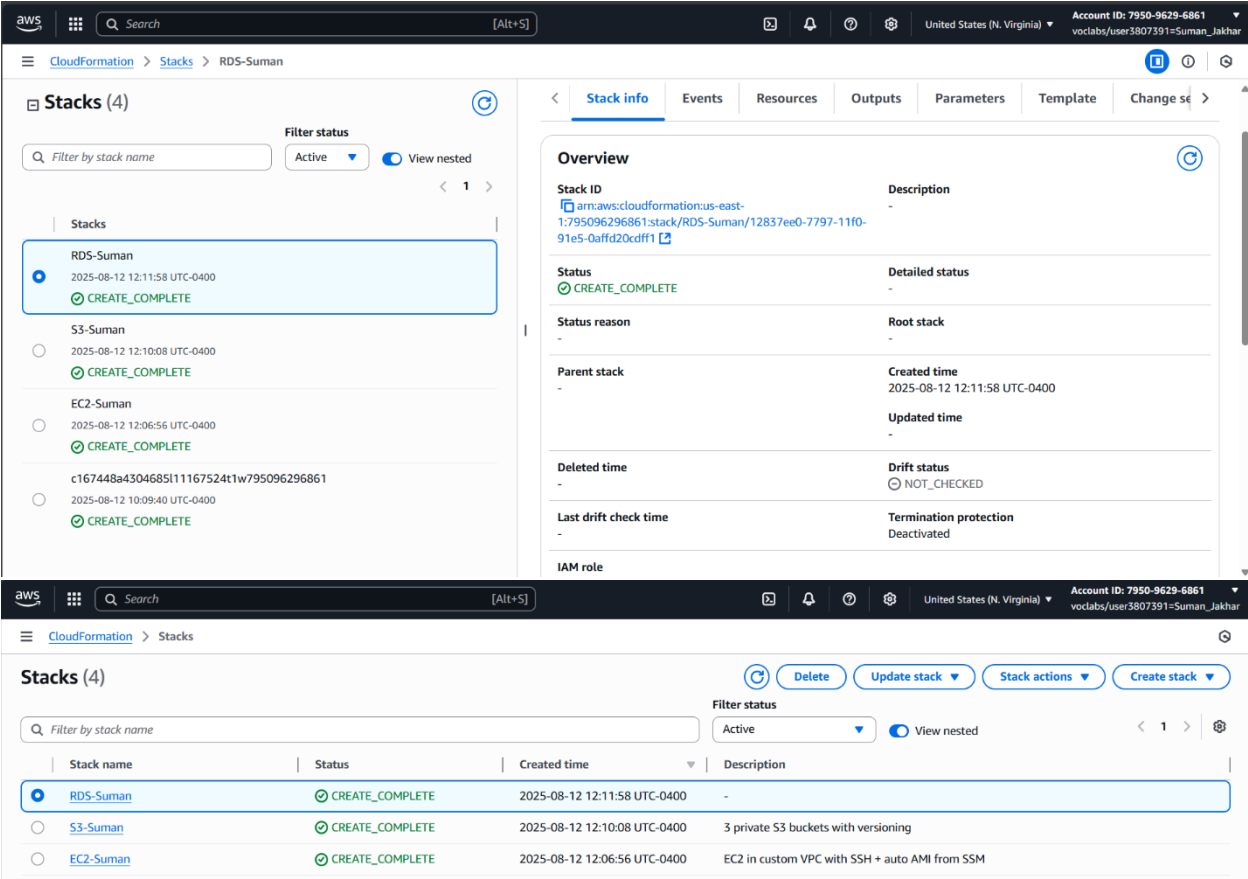
Search [Alt+S]

United States (N. Virginia) Account ID: 7950-9629-6861 voclabs/user3807391=Suman\_Jakhar

- 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 traLic on port 3306.





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

7. **Presentation/Demo**

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