Use Terraform to deploy AWS services, including EKS, EC2, S3, RDS, Load Balancer, Route 53, Private Container Registry, and others

step-by-step guide to deploying AWS services (EKS, S3, RDS, Load Balancer, Route 53, Private Container Registry, EC2, and others) using Terraform. This includes scripting and installation instructions for each step.

Step 1: Set Up Terraform Environment

Prepare your local environment for Terraform scripting in VS Code.

Configure AWS CLI:
Install and configure AWS CLI for authentication.

- sudo apt install awscli -y
- aws configure

Set Up IAM Permissions: Ensure your IAM user/role has appropriate permissions to manage EKS, S3, RDS, EC2, ALB, Route 53, and Elastic Container Registry (ECR).

Step 2: Initialize a Terraform Project

Create and configure a directory for your Terraform project.

1. Create a directory:

- mkdir terraform-aws-project
- cd terraform-aws-project

2. Initialize Terraform:

- terraform init

Step 3: Define Provider Configuration

Specify AWS as the provider and region for deployment.

Terraform Script:

<u>Provider.tf:</u>

```
terraform {
  required_providers {
    aws = {
      source  = "hashicorp/aws"
      version = ">= 4.0.0"
    }
  }
}

provider "aws" {
  region = "ap-south-1"  # Change to your desired region
}

provider "kubernetes" {
  host                   = data.aws_eks_cluster.eks.endpoint
  cluster_ca_certificate = base64decode(data.aws_eks_cluster.eks.certificate_authority[0].data)
  token                  = data.aws_eks_cluster_auth.eks.token
}
```

<u>Vpc.tf:</u>

```
variable "region" {
  default = "ap-south-1"
}
data "aws_availability_zones" "available" {}
locals {
  cluster_name = "clusters"
}
module vpc {
  source = "terraform-aws-modules/vpc/aws"
  name = "Bhargav-EKS-VPC"
  cidr = "10.0.0.0/16"

  azs = data.aws_availability_zones.available.names
  private_subnets = ["10.0.1.0/24", "10.0.2.0/24", "10.0.3.0/24"]
  public_subnets =  ["10.0.4.0/24", "10.0.5.0/24", "10.0.6.0/24"]

  enable_nat_gateway = true
  single_nat_gateway = true

 enable_dns_hostnames= true
tags = {
  "Name" = "Bhargav-EKS-VPC"
}
public_subnet_tags = {
  "Name" = "EKS-Public-Subnet"
}
private_subnet_tags = {
  "Name" = "EKS-Private-Subnet"
}
}
```

Ec2.tf:

```
resource "aws_vpc" "main" {
  cidr_block = "10.0.0.0/16"

  tags = {
    Name = "MainVPC"
  }
}

resource "aws_subnet" "public" {
  vpc_id                  = aws_vpc.main.id
  cidr_block              = "10.0.1.0/24"
  availability_zone       = "ap-south-1a"
  map_public_ip_on_launch = true

  tags = {
    Name = "PublicSubnet"
  }
}

resource "aws_instance" "AWS-Services" {
  ami           = "ami-053b12d3152c0cc71" # Replace with a region-specific AMI
  instance_type = "t2.micro"
  subnet_id     = aws_subnet.public.id

  tags = {
    Name = "AWS-Services"
  }
}
```

eks.tf:

```
module "eks" {
  source  = "terraform-aws-modules/eks/aws"
  version = ">= 18.0.0"

  cluster_name    = "Bhargav-EKS-Cluster"
  cluster_version = "1.26"
  vpc_id          = "vpc-01d65159f5c79b8ff"
  subnet_ids      = ["subnet-0ca0ad3708d01ddb6", "subnet-015c73cfce1f416cb"]
}

resource "aws_launch_template" "worker" {
  name_prefix   = "Bhargav-EKS-Template"
  image_id      = "ami-053b12d3152c0cc71"  # Replace with a valid AMI ID
  instance_type = "t3.medium"
  key_name      = "Devops"

  tag_specifications {
    resource_type = "instance"
    tags = {
      Name = "Bhargav-EKS-Worker"
    }
  }
}
```

Sg.tf:

```
resource "aws_security_group" "worker_sg" {
  name        = "eks-worker-sg"
  description = "Security group for EKS worker nodes"
  vpc_id      = aws_vpc.example.id

  ingress {
    from_port   = 0
    to_port     = 65535
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"] # Update this to a more restrictive rule as needed
  }

  egress {
    from_port   = 0
    to_port     = 0
    protocol    = "-1"
    cidr_blocks = ["0.0.0.0/0"]
  }
}

resource "aws_security_group" "worker_group_one" {
  name        = "worker-group-one-sg"
  description = "Security group for Worker Group 1"
  vpc_id      = module.vpc.vpc_id

  ingress {
    from_port   = 0
    to_port     = 65535
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }

  egress {
    from_port   = 0
    to_port     = 65535
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }
}

resource "aws_security_group" "worker_group_two" {
  name        = "worker-group-two-sg"
  description = "Security group for Worker Group 2"
  vpc_id      = module.vpc.vpc_id

  ingress {
    from_port   = 0
    to_port     = 65535
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }

  egress {
    from_port   = 0
    to_port     = 65535
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }
}
```

```
resource "aws_lb" "app_lb" {
  name               = "app-lb"
  internal           = false
  load_balancer_type = "application"
  security_groups    = [aws_security_group.alb_sg.id] # Security group for ALB
  subnets            = module.vpc.public_subnets      # Replace with your public subnets

  tags = {
    Name = "app-lb"
  }
}
resource "aws_security_group" "alb_sg" {
  name        = "alb-sg"
  description = "Security group for ALB"
  vpc_id      = module.vpc.vpc_id

  ingress {
    from_port   = 80
    to_port     = 80
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }

  ingress {
    from_port   = 443
    to_port     = 443
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }

  egress {
    from_port   = 0
    to_port     = 0
    protocol    = "-1"
    cidr_blocks = ["0.0.0.0/0"]
  }
}
```

S3.tf:

```
resource "aws_s3_bucket" "example" {
  bucket = "bhargav-eks-bucket-2024"
}
```

Asg.tf:

```
data "aws_ami" "eks_optimized" {
  most_recent = true

  owners = ["602401143452"]  # Amazon EKS AMI owner ID

  filter {
    name   = "name"
    values = ["amazon-eks-node-*-v*"]
  }
}

resource "aws_launch_template" "eks_workers" {
  name_prefix   = "eks-workers-"
  instance_type = "t3.micro"
```

```
 key_name = "devops"  # Replace with your key pair name

 iam_instance_profile {
  name = "eks-node-instance-profile"  # Replace with the appropriate IAM role
 }

 network_interfaces {
  security_groups          = [aws_security_group.worker_sg.id]
  associate_public_ip_address = true
 }

 block_device_mappings {
  device_name = "/dev/xvda"
  ebs {
   volume_size = 20
   volume_type = "gp2"
  }
 }

 # Reference the user_data template
 user_data = base64encode(data.template_file.user_data.rendered)

 # Use the dynamic AMI ID
 image_id = data.aws_ami.eks_optimized.id
}
```

rds.tf:

```
resource "aws_db_instance" "db" {
  allocated_storage    = 20
  engine               = "mysql"
  engine_version       = "8.0"
  instance_class       = "db.t3.micro"
  db_name              = "appdb"  # Correct attribute for database name
  username             = "admin"
  password             = "strongpassword"
  parameter_group_name = "default.mysql8.0"
  skip_final_snapshot  = true
}
```

alb.tf:

```
# IAM Role for EKS Cluster
resource "aws_iam_role" "eks_role" {
  name = "eks-cluster-role"

  assume_role_policy = jsonencode({
    Version = "2012-10-17"
    Statement = [
      {
        Action    = "sts:AssumeRole"
        Principal = {
          Service = "eks.amazonaws.com"
        }
        Effect    = "Allow"
        Sid       = ""
      },
    ]
  })
}

# Attach policies to the IAM role for EKS Cluster
resource "aws_iam_role_policy_attachment" "eks_policy_attachment" {
  role       = aws_iam_role.eks_role.name
  policy_arn = "arn:aws:iam::aws:policy/AmazonEKSClusterPolicy"
}

# Example VPC and Subnets
resource "aws_vpc" "example" {
  cidr_block = "10.0.0.0/16"
}

# Subnet 1 in Availability Zone ap-south-1a
resource "aws_subnet" "example_subnet_1" {
  vpc_id            = aws_vpc.example.id
  cidr_block        = "10.0.1.0/24"
  availability_zone = "ap-south-1a"
}

# Subnet 2 in Availability Zone ap-south-1b
resource "aws_subnet" "example_subnet_2" {
  vpc_id            = aws_vpc.example.id
  cidr_block        = "10.0.2.0/24"
  availability_zone = "ap-south-1b"
}

variable "cluster_name" {
  description = "Bhargav-EKS-Cluster"
  type        = string
  default     = "my-cluster-name" # Optional: Provide a default value
}

# EKS Cluster
resource "aws_eks_cluster" "eks_cluster" {
  name     = var.cluster_name
  role_arn = aws_iam_role.eks_role.arn  # Reference the IAM role created above

  vpc_config {
    subnet_ids = [
      aws_subnet.example_subnet_1.id,
      aws_subnet.example_subnet_2.id
    ]
  }
}

# New subnet for Worker Nodes
resource "aws_subnet" "worker_subnet" {
  vpc_id            = aws_vpc.example.id
  cidr_block        = "10.0.3.0/24"
  availability_zone = "ap-south-1c"
}

variable "subnet_ids" {
  description = "List of subnet IDs to associate with the Auto Scaling group"
  type        = list(string)
default     = ["subnet-0050faad53dcb5fc4", "subnet-0207799ffc86f3680"]
}
```

```
# IAM Role for Worker Nodes
resource "aws_iam_role" "eks_workers_role" {
  name = "eks-workers-role"

  assume_role_policy = jsonencode({
    Version = "2012-10-17"
    Statement = [
      {
        Action    = "sts:AssumeRole"
        Principal = {
          Service = "ec2.amazonaws.com"
        }
        Effect    = "Allow"
        Sid       = ""
      },
    ]
  })
}

# Attach policies to the IAM role for Worker Nodes
resource "aws_iam_role_policy_attachment" "eks_workers_policy_attachment" {
  role       = aws_iam_role.eks_workers_role.name
  policy_arn = "arn:aws:iam::aws:policy/AmazonEKSWorkerNodePolicy"
}

resource "aws_iam_role_policy_attachment" "eks_cni_policy_attachment" {
  role       = aws_iam_role.eks_workers_role.name
  policy_arn = "arn:aws:iam::aws:policy/AmazonEKS_CNI_Policy"
}

resource "aws_iam_role_policy_attachment" "eks_registry_policy_attachment" {
  role       = aws_iam_role.eks_workers_role.name
  policy_arn = "arn:aws:iam::aws:policy/AmazonEC2ContainerRegistryReadOnly"
}

# IAM Instance Profile for Worker Nodes
resource "aws_iam_instance_profile" "eks_workers_profile" {
  name = "eks-workers-profile"
  role = aws_iam_role.eks_workers_role.name
}
```

<u>Route53.tf:</u>

```
resource "aws_route53_zone" "main" {
  name = "your-custom-domain.com"  # Replace with your actual domain name
}

resource "aws_route53_record" "app" {
  zone_id = aws_route53_zone.main.zone_id
  name    = "app"
  type    = "A"

  alias {
    name                   = aws_lb.app_lb.dns_name
    zone_id                = aws_lb.app_lb.zone_id
    evaluate_target_health = true
  }
}
```

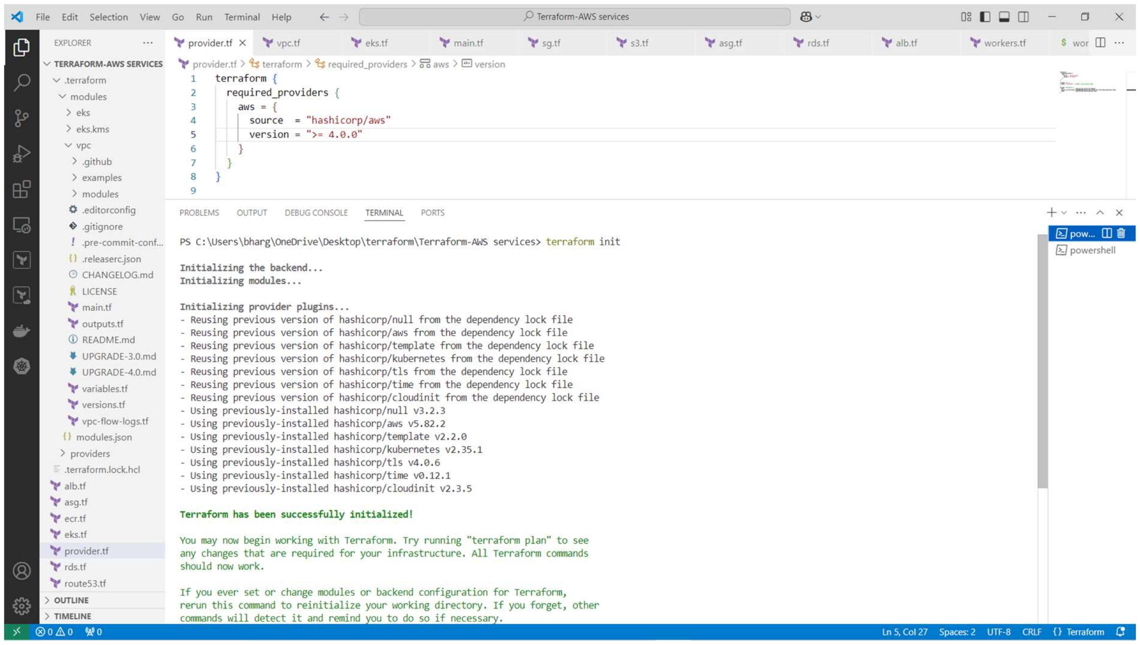
ecr.tf:

```
resource "aws_ecr_repository" "app_repo" {
  name = "app-repo"
}
```

Step 4: Apply Configuration

1. Initialize Terraform:

- terraform init

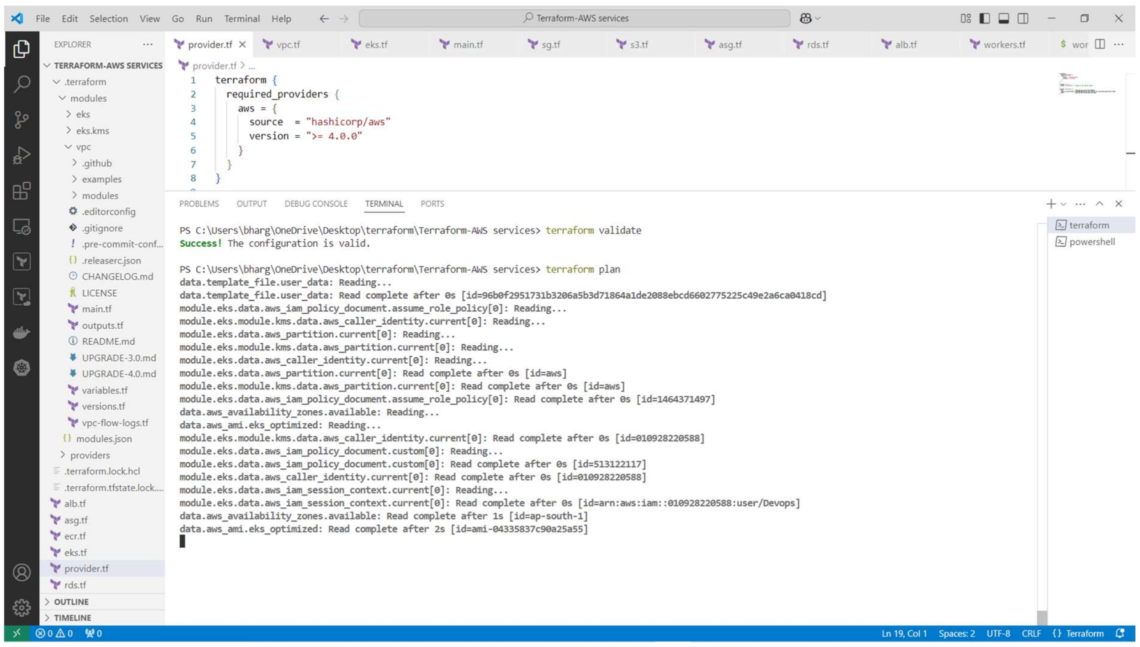


```
terraform {
  required_providers {
    aws = {
      source  = "hashicorp/aws"
      version = ">= 4.0.0"
    }
  }
}
```

```
PS C:\Users\bharg\OneDrive\Desktop\terraform\Terraform-AWS services> terraform init

Initializing the backend...
Initializing modules...

Initializing provider plugins...
- Reusing previous version of hashicorp/null from the dependency lock file
- Reusing previous version of hashicorp/aws from the dependency lock file
- Reusing previous version of hashicorp/template from the dependency lock file
- Reusing previous version of hashicorp/kubernetes from the dependency lock file
- Reusing previous version of hashicorp/tls from the dependency lock file
- Reusing previous version of hashicorp/time from the dependency lock file
- Reusing previous version of hashicorp/cloudinit from the dependency lock file
- Using previously-installed hashicorp/null v3.2.3
- Using previously-installed hashicorp/aws v5.82.2
- Using previously-installed hashicorp/template v2.2.0
- Using previously-installed hashicorp/kubernetes v2.35.1
- Using previously-installed hashicorp/tls v4.0.6
- Using previously-installed hashicorp/time v0.12.1
- Using previously-installed hashicorp/cloudinit v2.3.5

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
```

2. Validate configuration:

- terraform validate
- terraform plan



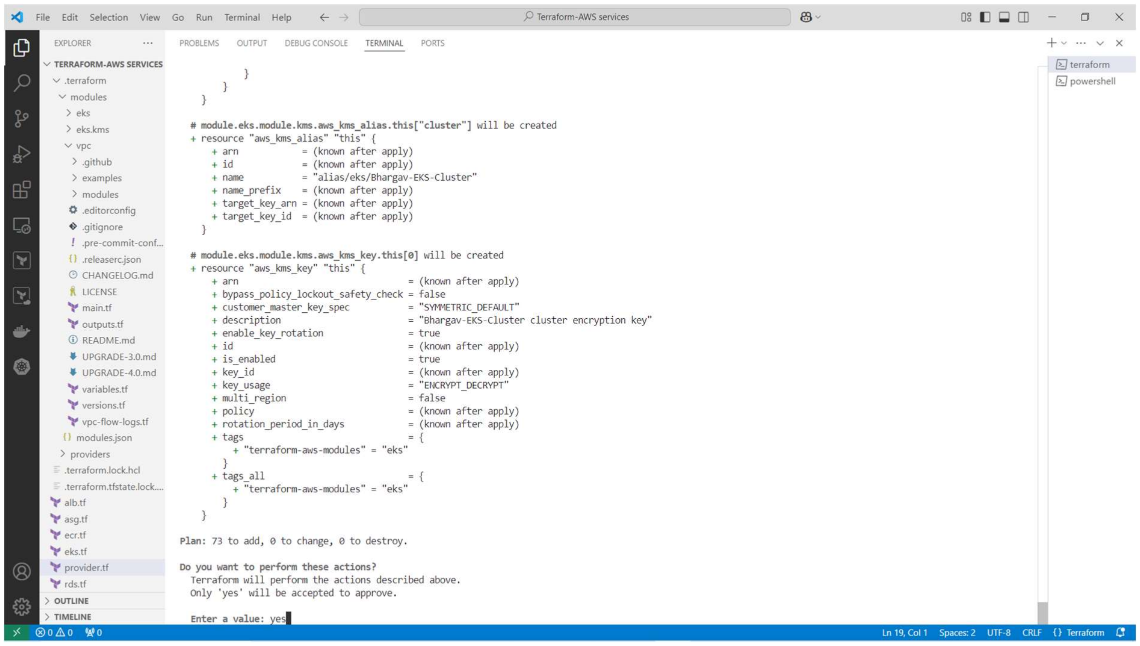
```
terraform {
  required_providers {
    aws = {
      source  = "hashicorp/aws"
      version = ">= 4.0.0"
    }
  }
}
```

```
PS C:\Users\bharg\OneDrive\Desktop\terraform\Terraform-AWS services> terraform validate
Success! The configuration is valid.

PS C:\Users\bharg\OneDrive\Desktop\terraform\Terraform-AWS services> terraform plan
data.template_file.user_data: Reading...
data.template_file.user_data: Read complete after 0s [id=96b0f2951731b3206a5b3d71864a1de2088ebcd6602775225c49e2a6ca0418cd]
module.eks.data.aws_iam_policy_document.assume_role_policy[0]: Reading...
module.eks.module.kms.data.aws_caller_identity.current[0]: Reading...
module.eks.data.aws_partition.current[0]: Reading...
module.eks.module.kms.data.aws_partition.current[0]: Reading...
module.eks.data.aws_caller_identity.current[0]: Reading...
module.eks.data.aws_partition.current[0]: Read complete after 0s [id=aws]
module.eks.module.kms.data.aws_partition.current[0]: Read complete after 0s [id=aws]
module.eks.data.aws_iam_policy_document.assume_role_policy[0]: Read complete after 0s [id=1464371497]
data.aws_availability_zones.available: Reading...
data.aws_ami.eks_optimized: Reading...
module.eks.module.kms.data.aws_caller_identity.current[0]: Read complete after 0s [id=010928220588]
module.eks.data.aws_iam_policy_document.custom[0]: Reading...
module.eks.data.aws_iam_policy_document.custom[0]: Read complete after 0s [id=513122117]
module.eks.data.aws_caller_identity.current[0]: Read complete after 0s [id=010928220588]
module.eks.data.aws_iam_session_context.current[0]: Reading...
module.eks.data.aws_iam_session_context.current[0]: Read complete after 0s [id=arn:aws:iam::010928220588:user/Devops]
data.aws_availability_zones.available: Read complete after 1s [id=ap-south-1]
data.aws_ami.eks_optimized: Read complete after 2s [id=ami-04335837c90a25a55]
```

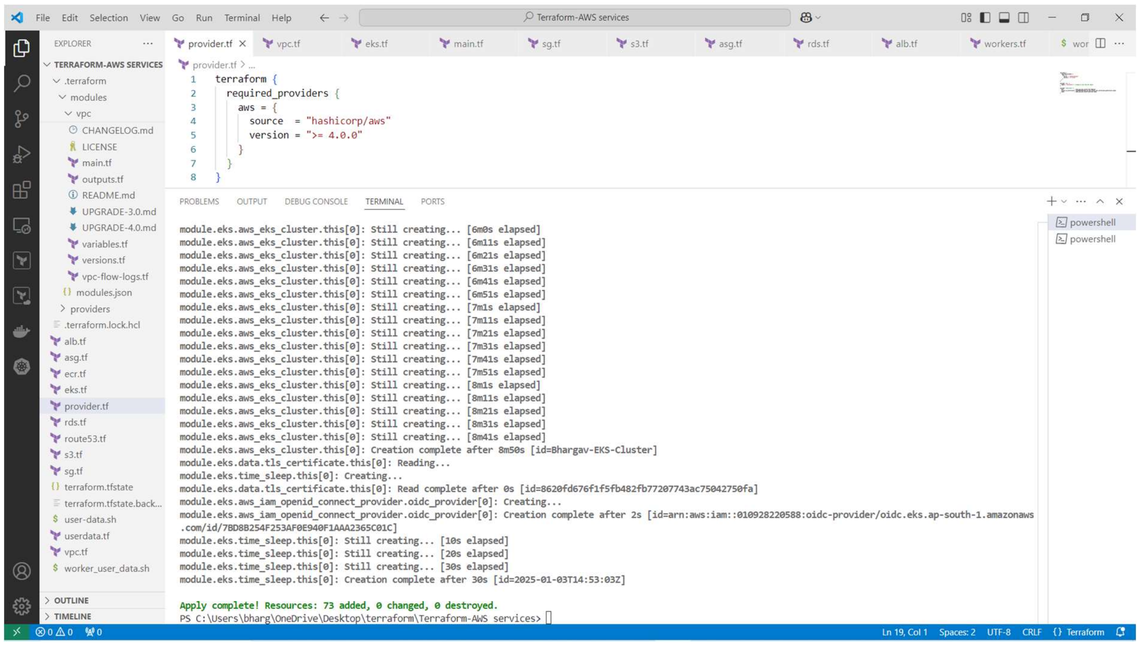
3. Deploy:

- terraform apply

Confirm by typing yes.



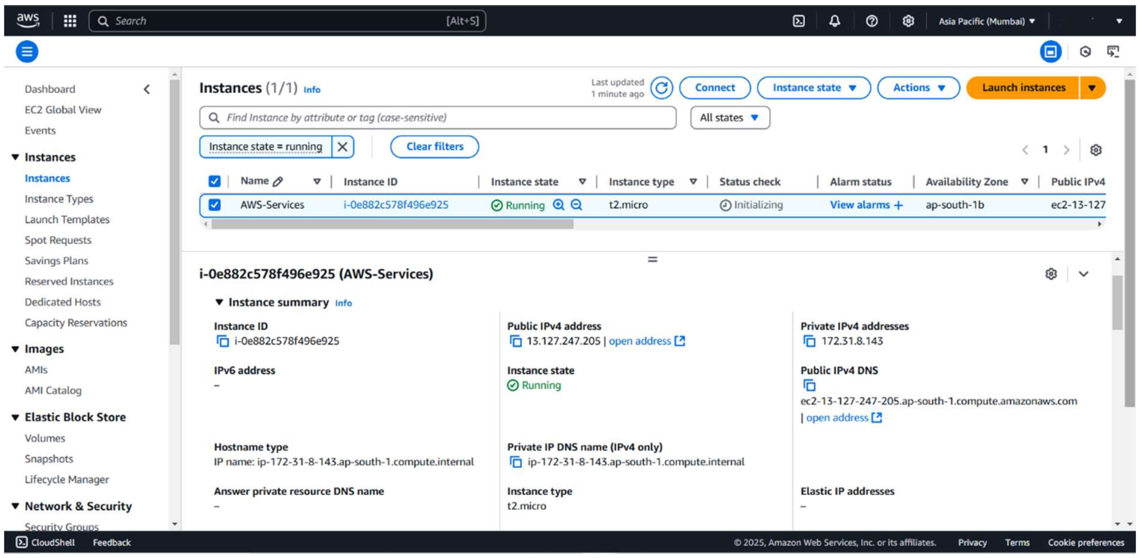
```
          }
      }
    }

  # module.eks.module.kms.aws_kms_alias.this["cluster"] will be created
  + resource "aws_kms_alias" "this" {
      + arn            = (known after apply)
      + id             = (known after apply)
      + name           = "alias/eks/Bhargav-EKS-Cluster"
      + name_prefix    = (known after apply)
      + target_key_arn = (known after apply)
      + target_key_id  = (known after apply)
    }

  # module.eks.module.kms.aws_kms_key.this[0] will be created
  + resource "aws_kms_key" "this" {
      + arn                                = (known after apply)
      + bypass_policy_lockout_safety_check = false
      + customer_master_key_spec           = "SYMMETRIC_DEFAULT"
      + description                        = "Bhargav-EKS-Cluster cluster encryption key"
      + enable_key_rotation                = true
      + id                                 = (known after apply)
      + is_enabled                         = true
      + key_id                             = (known after apply)
      + key_usage                          = "ENCRYPT_DECRYPT"
      + multi_region                       = false
      + policy                             = (known after apply)
      + rotation_period_in_days            = (known after apply)
      + tags                               = {
          + "terraform-aws-modules" = "eks"
        }
      + tags_all                           = {
          + "terraform-aws-modules" = "eks"
        }
    }

Plan: 73 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes
```



```
terraform {
  required_providers {
    aws = {
      source  = "hashicorp/aws"
      version = ">= 4.0.0"
    }
  }
}
```

```
module.eks.aws_eks_cluster.this[0]: Still creating... [6m0s elapsed]
module.eks.aws_eks_cluster.this[0]: Still creating... [6m11s elapsed]
module.eks.aws_eks_cluster.this[0]: Still creating... [6m21s elapsed]
module.eks.aws_eks_cluster.this[0]: Still creating... [6m31s elapsed]
module.eks.aws_eks_cluster.this[0]: Still creating... [6m41s elapsed]
module.eks.aws_eks_cluster.this[0]: Still creating... [6m51s elapsed]
module.eks.aws_eks_cluster.this[0]: Still creating... [7m1s elapsed]
module.eks.aws_eks_cluster.this[0]: Still creating... [7m11s elapsed]
module.eks.aws_eks_cluster.this[0]: Still creating... [7m21s elapsed]
module.eks.aws_eks_cluster.this[0]: Still creating... [7m31s elapsed]
module.eks.aws_eks_cluster.this[0]: Still creating... [7m41s elapsed]
module.eks.aws_eks_cluster.this[0]: Still creating... [7m51s elapsed]
module.eks.aws_eks_cluster.this[0]: Still creating... [8m1s elapsed]
module.eks.aws_eks_cluster.this[0]: Still creating... [8m11s elapsed]
module.eks.aws_eks_cluster.this[0]: Still creating... [8m21s elapsed]
module.eks.aws_eks_cluster.this[0]: Still creating... [8m31s elapsed]
module.eks.aws_eks_cluster.this[0]: Still creating... [8m41s elapsed]
module.eks.aws_eks_cluster.this[0]: Creation complete after 8m50s [id=Bhargav-EKS-Cluster]
module.eks.data.tls_certificate.this[0]: Reading...
module.eks.time_sleep.this[0]: Creating...
module.eks.data.tls_certificate.this[0]: Read complete after 0s [id=8620fd676f1f5fb482fb77207743ac75042750fa]
module.eks.aws_iam_openid_connect_provider.oidc_provider[0]: Creating...
module.eks.aws_iam_openid_connect_provider.oidc_provider[0]: Creation complete after 2s [id=arn:aws:iam::010928220588:oidc-provider/oidc.eks.ap-south-1.amazonaws
.com/id/7BD8B254F253AF0E940F1AAA2365C01C]
module.eks.time_sleep.this[0]: Still creating... [10s elapsed]
module.eks.time_sleep.this[0]: Still creating... [20s elapsed]
module.eks.time_sleep.this[0]: Still creating... [30s elapsed]
module.eks.time_sleep.this[0]: Creation complete after 30s [id=2025-01-03T14:53:03Z]

Apply complete! Resources: 73 added, 0 changed, 0 destroyed.
PS C:\Users\bharg\OneDrive\Desktop\terraform\Terraform-AWS services>
```

EC2:



Instances (1/1)

Instance state = running

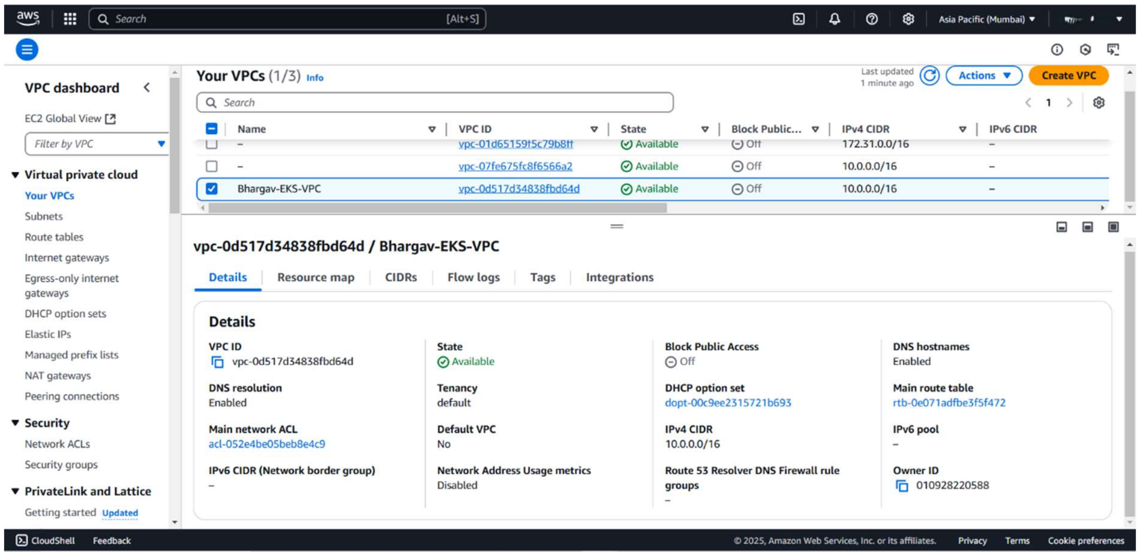
Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4
AWS-Services	i-0e882c578f496e925	Running	t2.micro	Initializing	View alarms +	ap-south-1b	ec2-13-127

i-0e882c578f496e925 (AWS-Services)

Instance summary

Instance ID i-0e882c578f496e925	**Public IPv4 address** 13.127.247.205 | open address	**Private IPv4 addresses** 172.31.8.143
IPv6 address –	**Instance state** Running	**Public IPv4 DNS** ec2-13-127-247-205.ap-south-1.compute.amazonaws.com | open address
Hostname type IP name: ip-172-31-8-143.ap-south-1.compute.internal	**Private IP DNS name (IPv4 only)** ip-172-31-8-143.ap-south-1.compute.internal	
Answer private resource DNS name –	**Instance type** t2.micro	**Elastic IP addresses** –

VPC:



Your VPCs (1/3)

Name	VPC ID	State	Block Public...	IPv4 CIDR	IPv6 CIDR
–	vpc-01d65159f5c79b8ff	Available	Off	172.31.0.0/16	–
–	vpc-07fe675fc8f6566a2	Available	Off	10.0.0.0/16	–
Bhargav-EKS-VPC	vpc-0d517d34838fbd64d	Available	Off	10.0.0.0/16	–

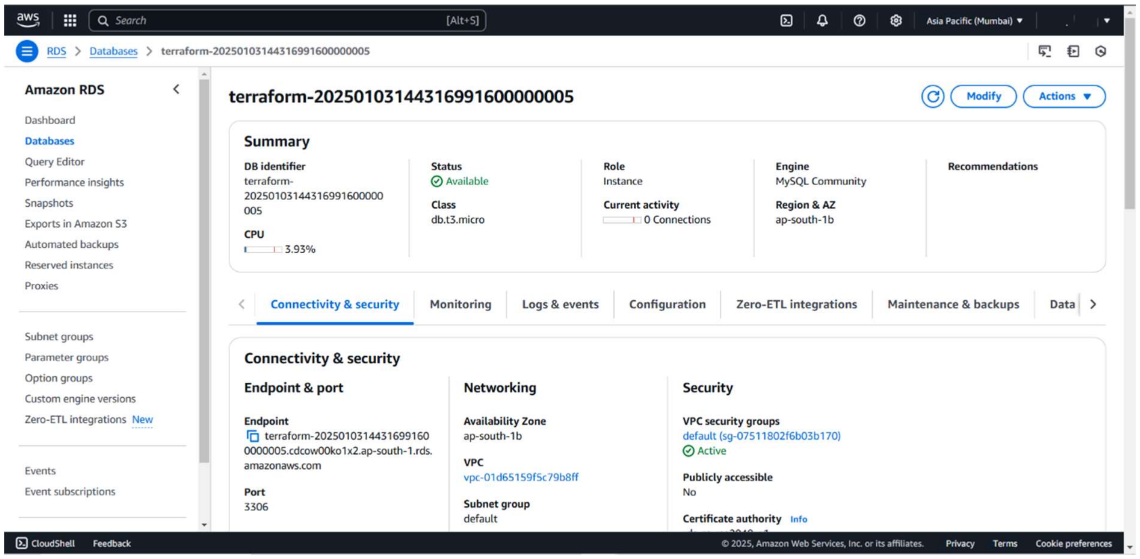
vpc-0d517d34838fbd64d / Bhargav-EKS-VPC

Details

VPC ID vpc-0d517d34838fbd64d	**State** Available	**Block Public Access** Off	**DNS hostnames** Enabled
DNS resolution Enabled	**Tenancy** default	**DHCP option set** dopt-00c9ee2315721b693	**Main route table** rtb-0e071adfbe3f5f472
Main network ACL acl-052e4be05beb8e4c9	**Default VPC** No	**IPv4 CIDR** 10.0.0.0/16	**IPv6 pool** –
IPv6 CIDR (Network border group) –	**Network Address Usage metrics** Disabled	**Route 53 Resolver DNS Firewall rule groups** –	**Owner ID** 010928220588

© 2025, Amazon Web Services, Inc. or its affiliates.

RDS:



terraform-20250103144316991600000005

Summary

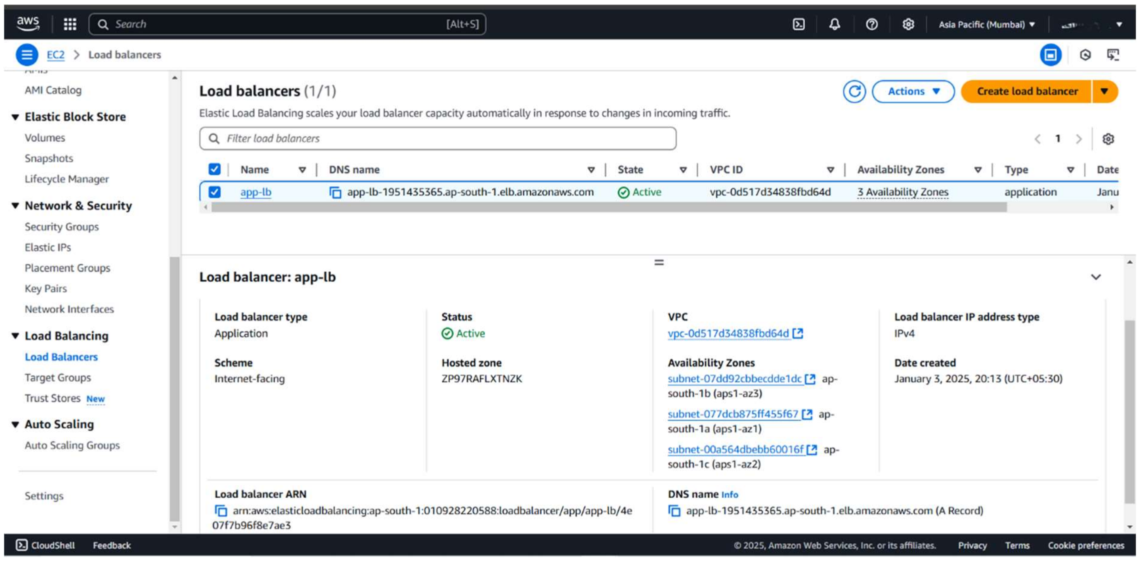
DB identifier terraform-20250103144316991600000005	**Status** Available	**Role** Instance	**Engine** MySQL Community	**Recommendations**
CPU 3.93%	**Class** db.t3.micro	**Current activity** 0 Connections	**Region & AZ** ap-south-1b	

Connectivity & security

Endpoint & port	Networking	Security
Endpoint terraform-202501031443169916000000005.cdcow00ko1x2.ap-south-1.rds.amazonaws.com	**Availability Zone** ap-south-1b	**VPC security groups** default (sg-07511802f6b03b170) Active
Port 3306	**VPC** vpc-01d65159f5c79b8ff	**Publicly accessible** No
	Subnet group default	**Certificate authority**

© 2025, Amazon Web Services, Inc. or its affiliates.

ALB:



Load balancers (1/1)

Elastic Load Balancing scales your load balancer capacity automatically in response to changes in incoming traffic.

Name	DNS name	State	VPC ID	Availability Zones	Type	Date
app-lb	app-lb-1951435365.ap-south-1.elb.amazonaws.com	Active	vpc-0d517d34838fbd64d	3 Availability Zones	application	Janu

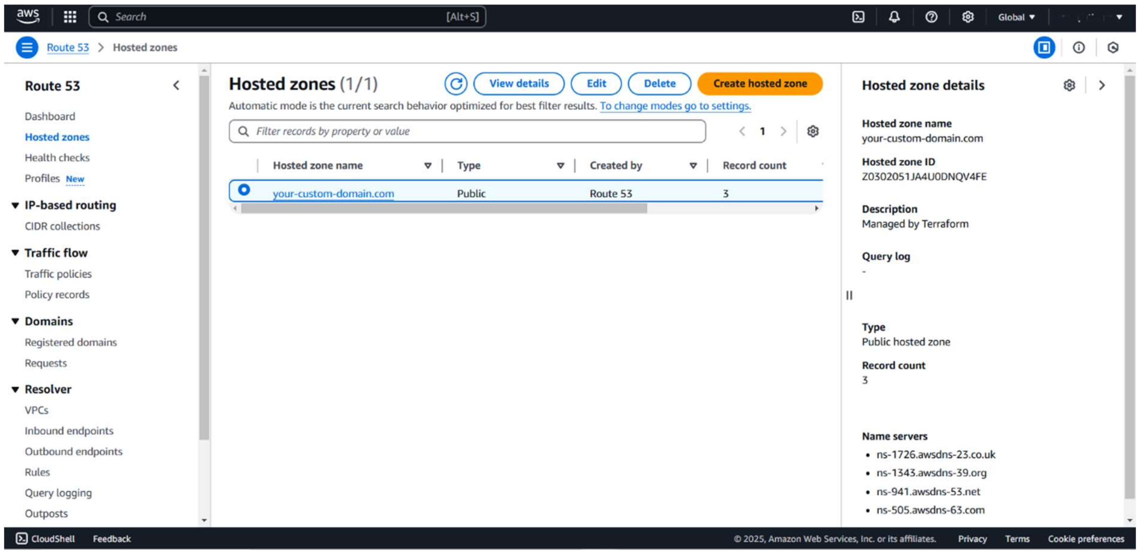
Load balancer: app-lb

Load balancer type Application	**Status** Active	**VPC** vpc-0d517d34838fbd64d	**Load balancer IP address type** IPv4
Scheme Internet-facing	**Hosted zone** ZP97RAFLXTNZK	**Availability Zones** subnet-07dd92cbbecdde1dc ap-south-1b (aps1-az3) subnet-077dcb875ff455f67 ap-south-1a (aps1-az1) subnet-00a564dbebb60016f ap-south-1c (aps1-az2)	**Date created** January 3, 2025, 20:13 (UTC+05:30)

Load balancer ARN
arn:aws:elasticloadbalancing:ap-south-1:010928220588:loadbalancer/app/app-lb/4e07f7b96f8e7ae3

DNS name Info
app-lb-1951435365.ap-south-1.elb.amazonaws.com (A Record)

Route53:



Hosted zones (1/1)

Automatic mode is the current search behavior optimized for best filter results. To change modes go to settings.

Hosted zone name	Type	Created by	Record count
your-custom-domain.com	Public	Route 53	3

Hosted zone details

Hosted zone name
your-custom-domain.com

Hosted zone ID
Z0302051JA4U0DNQV4FE

Description
Managed by Terraform

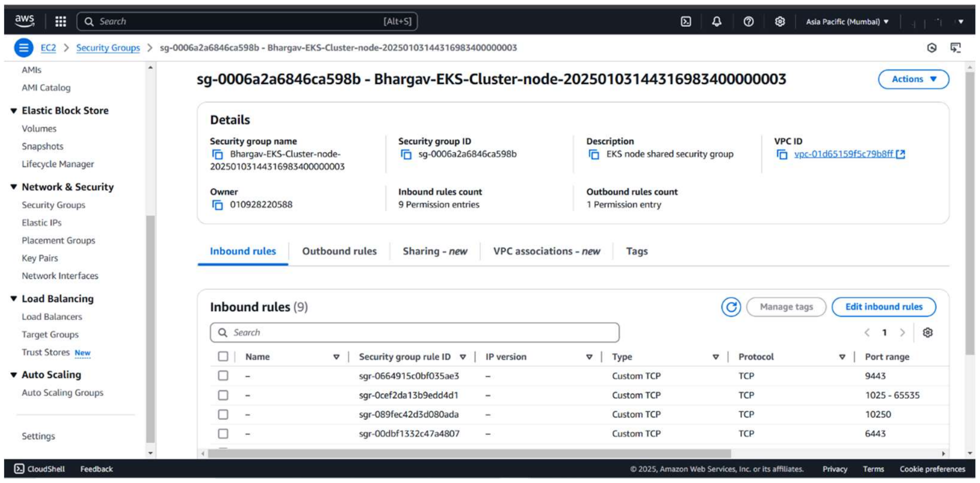
Query log
-

Type
Public hosted zone

Record count
3

Name servers
- ns-1726.awsdns-23.co.uk
- ns-1343.awsdns-39.org
- ns-941.awsdns-53.net
- ns-505.awsdns-63.com

Security Groups:



EC2 > Security Groups > sg-0006a2a6846ca598b - Bhargav-EKS-Cluster-node-20250103144316983400000003

sg-0006a2a6846ca598b - Bhargav-EKS-Cluster-node-20250103144316983400000003

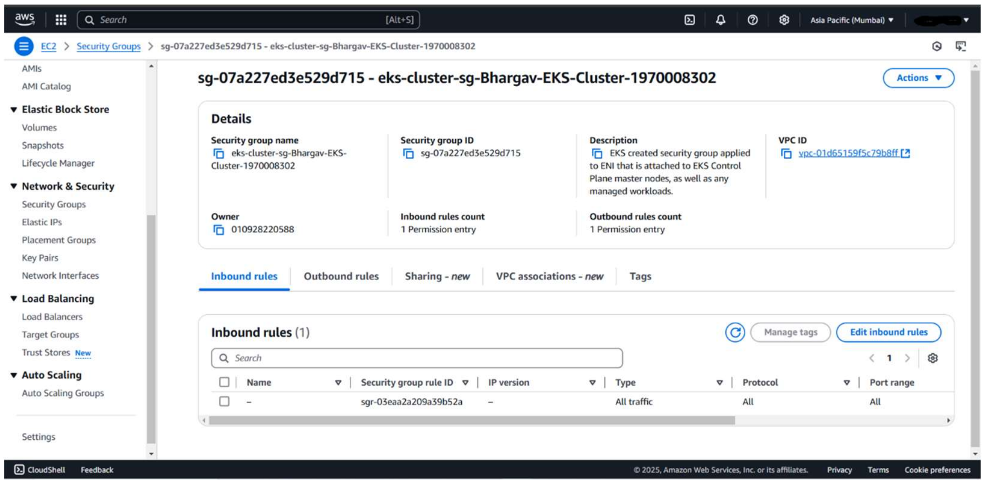
Details

Security group name Bhargav-EKS-Cluster-node-20250103144316983400000003	**Security group ID** sg-0006a2a6846ca598b	**Description** EKS node shared security group	**VPC ID** vpc-01d65159f5c79b8ff
Owner 010928220588	**Inbound rules count** 9 Permission entries	**Outbound rules count** 1 Permission entry	

Inbound rules | Outbound rules | Sharing - *new* | VPC associations - *new* | Tags

Inbound rules (9)

Name	Security group rule ID	IP version	Type	Protocol	Port range
–	sgr-0664915c0bf035ae3	–	Custom TCP	TCP	9443
–	sgr-0cef2da13b9edd4d1	–	Custom TCP	TCP	1025 - 65535
–	sgr-089fec42d3d080ada	–	Custom TCP	TCP	10250
–	sgr-00dbf1332c47a4807	–	Custom TCP	TCP	6443



[redacted]

EC2 > Security Groups > sg-07a227ed3e529d715 - eks-cluster-sg-Bhargav-EKS-Cluster-1970008302

sg-07a227ed3e529d715 - eks-cluster-sg-Bhargav-EKS-Cluster-1970008302

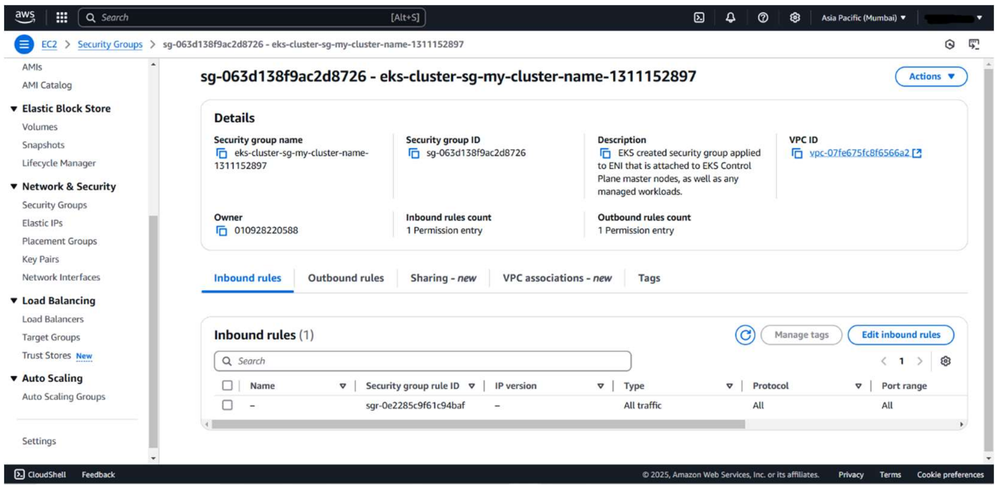
Details

Security group name eks-cluster-sg-Bhargav-EKS-Cluster-1970008302	**Security group ID** sg-07a227ed3e529d715	**Description** EKS created security group applied to ENI that is attached to EKS Control Plane master nodes, as well as any managed workloads.	**VPC ID** vpc-01d65159f5c79b8ff
Owner 010928220588	**Inbound rules count** 1 Permission entry	**Outbound rules count** 1 Permission entry	

Inbound rules | Outbound rules | Sharing - *new* | VPC associations - *new* | Tags

Inbound rules (1)

Name	Security group rule ID	IP version	Type	Protocol	Port range
–	sgr-03eaa2a209a39b52a	–	All traffic	All	All



[redacted]

EC2 > Security Groups > sg-063d138f9ac2d8726 - eks-cluster-sg-my-cluster-name-1311152897

sg-063d138f9ac2d8726 - eks-cluster-sg-my-cluster-name-1311152897

Details

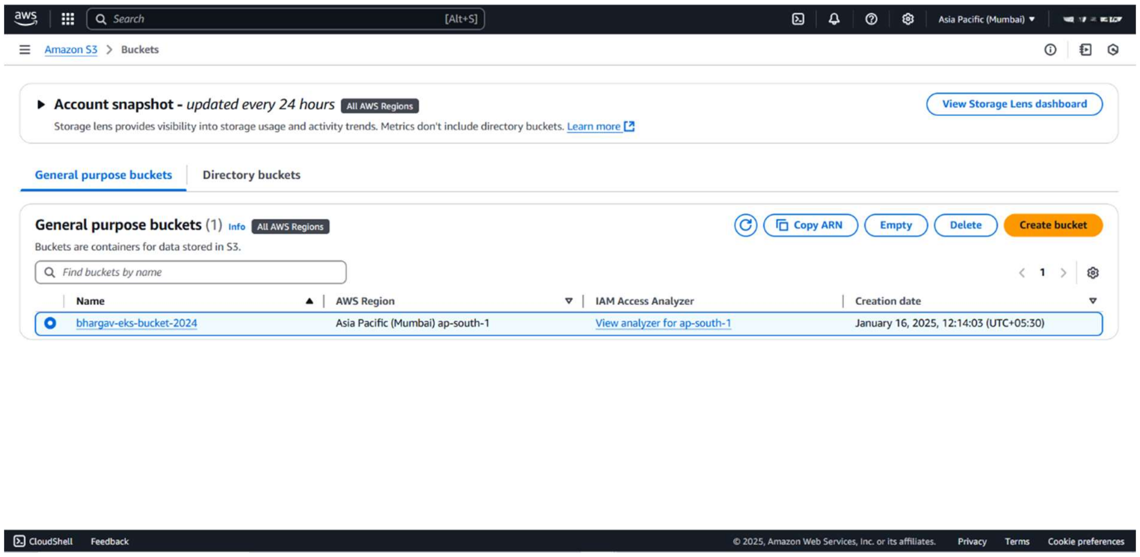
Security group name eks-cluster-sg-my-cluster-name-1311152897	**Security group ID** sg-063d138f9ac2d8726	**Description** EKS created security group applied to ENI that is attached to EKS Control Plane master nodes, as well as any managed workloads.	**VPC ID** vpc-07fe675fc8f6566a2
Owner 010928220588	**Inbound rules count** 1 Permission entry	**Outbound rules count** 1 Permission entry	

Inbound rules | Outbound rules | Sharing - *new* | VPC associations - *new* | Tags

Inbound rules (1)

Name	Security group rule ID	IP version	Type	Protocol	Port range
–	sgr-0e2285c9f61c94baf	–	All traffic	All	All

S3:



Storage lens provides visibility into storage usage and activity trends. Metrics don't include directory buckets. Learn more

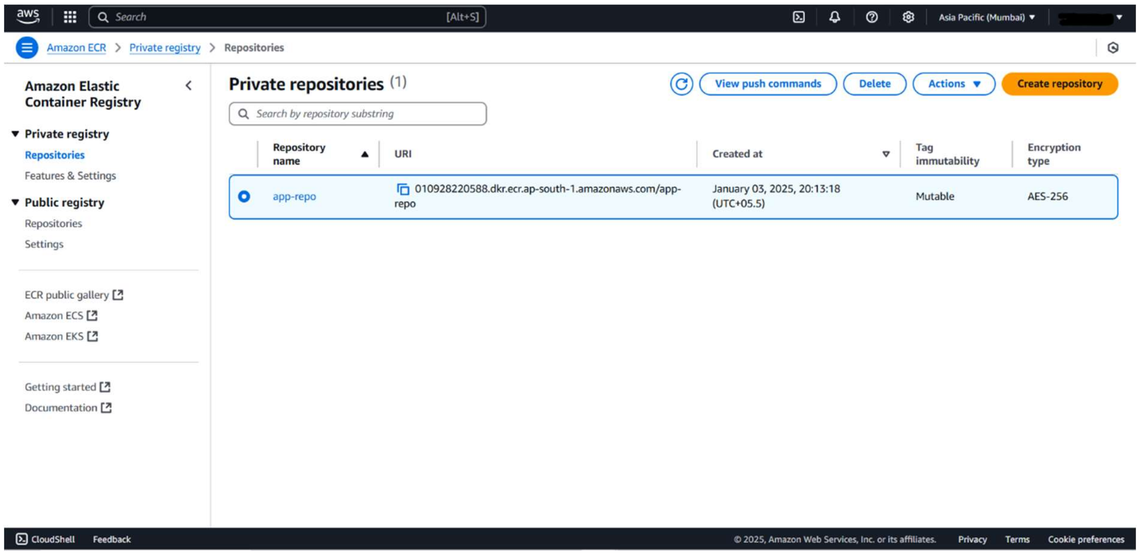
General purpose buckets (1) Info All AWS Regions

Buckets are containers for data stored in S3.

Name	AWS Region	IAM Access Analyzer	Creation date
bhargav-eks-bucket-2024	Asia Pacific (Mumbai) ap-south-1	View analyzer for ap-south-1	January 16, 2025, 12:14:03 (UTC+05:30)

© 2025, Amazon Web Services, Inc. or its affiliates.

ECR:



Private repositories (1)

Repository name	URI	Created at	Tag immutability	Encryption type
app-repo	010928220588.dkr.ecr.ap-south-1.amazonaws.com/app-repo	January 03, 2025, 20:13:18 (UTC+05.5)	Mutable	AES-256

© 2025, Amazon Web Services, Inc. or its affiliates.