Terraform EKS Deployment

Prerequisites

1. AWS Account with an IAM User with administrative permissions.
2. Terraform installed.
3. Kubectl & Kubens installed.

Using Terraform Modules to Provision AWS Infrastructure

1. Create the Terraform project.
2. Download the modules folder and copy it inside the above-created terraform project.
3. Create a *provider.tf* file inside the created terraform project.
4. Inside the *provider.tf* file, define the following:
   - terraform
     - required_providers
   - provider
     - aws
5. The reference code is attached below.

```
terraform {
  required_providers {
    aws = {
      source  = "hashicorp/aws"
      version = "~> 5.64"
    }
  }
}

provider "aws" {
  region = local.aws_region
  // shared_config_files = ["~/.aws/config"]
  shared_credentials_files = ["~/.aws/credentials"]
}
```

6. The definition of *provider.tf* file is complete.
7. Create the *main.tf* file.
8. Inside *main.tf* file, call the following modules:
   - vpc
   - rds
   - eks
9. The reference code is attached below.

```
module "vpc" {
  source = "./modules/aws/vpc"

  vpc-properties = local.vpc-properties
}

module "rds" {
  source = "./modules/aws/rds"

  vpc-id               = local.vpc-id
  vpc-public-subnets   = local.vpc-public-subnets
  vpc-private-subnets  = local.vpc-private-subnets
  database-properties  = local.database-properties
  bastion-properties   = local.bastion-properties

  depends_on = [
    module.vpc
  ]
}

module "eks" {
  source = "./modules/aws/eks"

  vpc-public-subnets   = local.vpc-public-subnets
  vpc-private-subnets  = local.vpc-private-subnets

  eks-properties = local.eks-properties

  depends_on = [
    module.rds
  ]
}
```

10. *main.tf* file definition is completed.
11. Now we will create *locals.tf* file.
12. Define the following variables:
    - aws_region
    - vpc-properties
        - availability-zones
        - vpc-cidr-block
        - vpc-subnet-count
        - vpc-public-subnet-cidr-blocks
        - vpc-private-subnet-cidr-blocks
        - vpc-tag-value
        - vpc-public-subnet-tag-value
        - vpc-private-subnet-tag-value
        - vpc-igw-tag-value
    - vpc-id
    - vpc-public-subnets

- vpc-private-subnets
- database-properties
  - identifier
  - allocated-storage
  - engine
  - engine-version
  - instance-class
  - skip-final-snapshot
  - publicly-accessible
  - db-username
  - db-password
  - db-sg-tag-value
  - db-tag-value
- bastion-properties
  - count
  - instance-type
  - bastion-host-public-key
  - bastion-host-sg-tag-value
  - bastion-host-tag-value
- eks-properties
  - eks-cluster-role-name
  - eks-cluster-name
  - eks-node-role-name
  - eks-node-group-name
  - eks-instance-types
  - eks-service-port

13. The reference code is attached below.

```
locals {

  aws_region = "ap-south-1"

  // vpc variables
  vpc-properties = {
    availability-zones = [
      "ap-south-1a",
      "ap-south-1b",
      "ap-south-1c"
    ]
    vpc-cidr-block = "10.0.0.0/16"
    vpc-subnet-count = {
      "public"  = 2,
      "private" = 2
    }
    vpc-public-subnet-cidr-blocks = [
      "10.0.1.0/24",
      "10.0.2.0/24",
      "10.0.3.0/24",
      "10.0.4.0/24"
    ]
    vpc-private-subnet-cidr-blocks = [
      "10.0.101.0/24",
      "10.0.102.0/24",
      "10.0.103.0/24",
      "10.0.104.0/24"
    ]

    vpc-tag-value                  = ""
    vpc-public-subnet-tag-value    = ""
    vpc-private-subnet-tag-value   = ""
    vpc-igw-tag-value              = ""
  }

  vpc-id              = module.vpc.vpc-id
  vpc-public-subnets  = module.vpc.vpc-public-subnets
  vpc-private-subnets = module.vpc.vpc-private-subnets

  // rds variables
  database-properties = {
    identifier           = ""
    allocated-storage    = 20
    engine               = "mysql"
    engine-version       = "8.0.35"
    instance-class       = "db.t3.micro"
    skip-final-snapshot  = true
    publicly-accessible  = false

    db-username = ""
    db-password = ""

    db-sg-tag-value = ""
    db-tag-value    = ""
  }

  bastion-properties = {
    count                   = 1
    instance-type           = "t2.micro"
    bastion-host-public-key = "~/.ssh/bastion-key.pub"

    bastion-host-sg-tag-value = ""
    bastion-host-tag-value    = ""
  }

  // eks variables
  eks-properties = {
    eks-cluster-role-name = ""
    eks-cluster-name      = ""
    eks-node-role-name    = ""
    eks-node-group-name   = ""
    eks-instance-types = [
      "t2.medium"
    ]
    eks-service-port =
  }
}
```

14. The definition of *locals.tf* file is complete.
15. Now we will create *outputs.tf* file.
16. Define the following outputs:
    - DB_HOST
    - bastion-host-ip
17. The reference code is attached below.

```
output "DB_HOST" {
  description = "db host address"
  value       = module.rds.DB_HOST
}

output "bastion-host-ip" {
  description = "bastion host ip address"
  value       = module.rds.bastion-host-ip
}
```

18. The definition of *outputs.tf* file is complete.

Provisioning the Infrastructure

Now we will provision the infrastructure by applying the above-created configuration files.

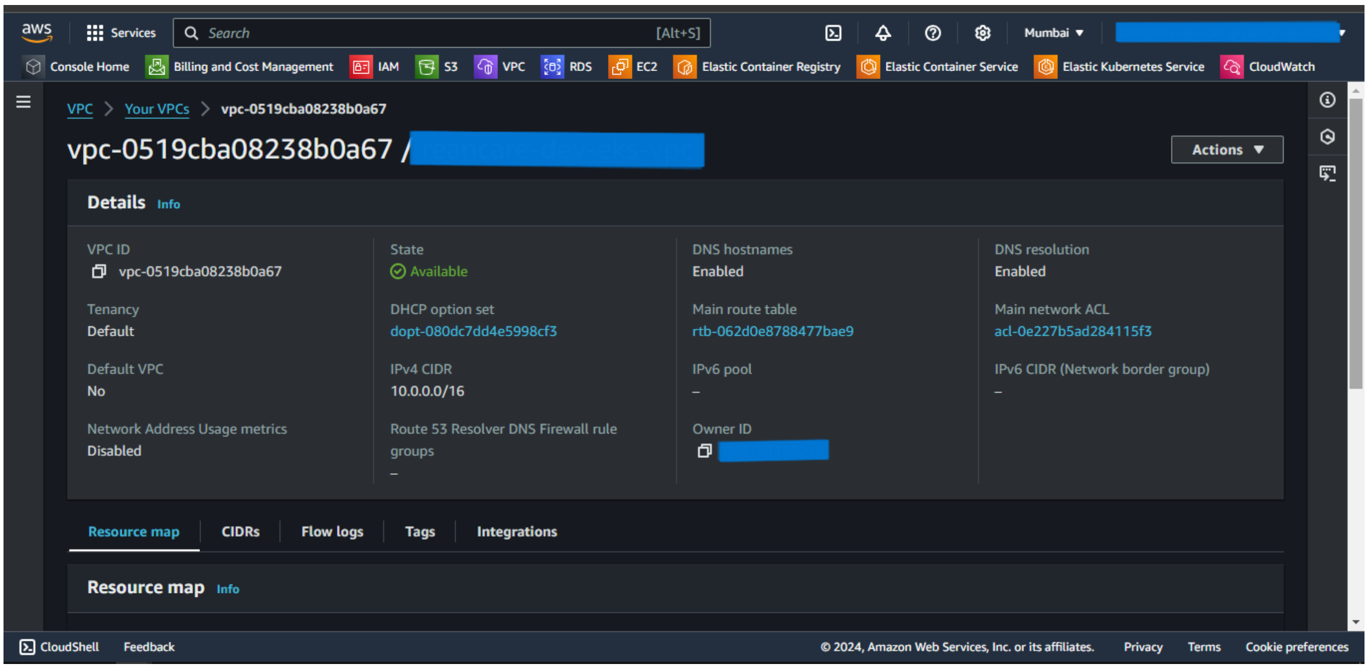
> Ensure AWS CLI is configured with appropriate AWS user credentials and enough permissions.

Steps:

1. Open the PowerShell.
2. Change the directory to the above-created Terraform Project.
3. Run the `terraform init` command to initialize the *terraform*.
4. Run the `terraform fmt --recursive` command to format the syntax of the files.
5. Run the `terraform validate` command to validate the configuration files.
6. Run the `terraform plan` command to plan the resources to be created.
7. Run the `terraform apply` command and if prompted, type `yes` to provision the infrastructure.
8. Run the `terraform output` command to get the values of defined variables in *outputs.tf* file.
9. Head to the AWS Console, and verify the created resources.

Screenshots of Provisioned Infrastructure

VPC Image


aws
Services
Search [Alt+S]
Mumbai
[redacted]
Console Home
Billing and Cost Management
IAM
S3
VPC
RDS
EC2
Elastic Container Registry
Elastic Container Service
Elastic Kubernetes Service
CloudWatch
VPC > Your VPCs > vpc-0519cba08238b0a67
Actions
Resource map
CIDRs
Flow logs
Tags
Integrations
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

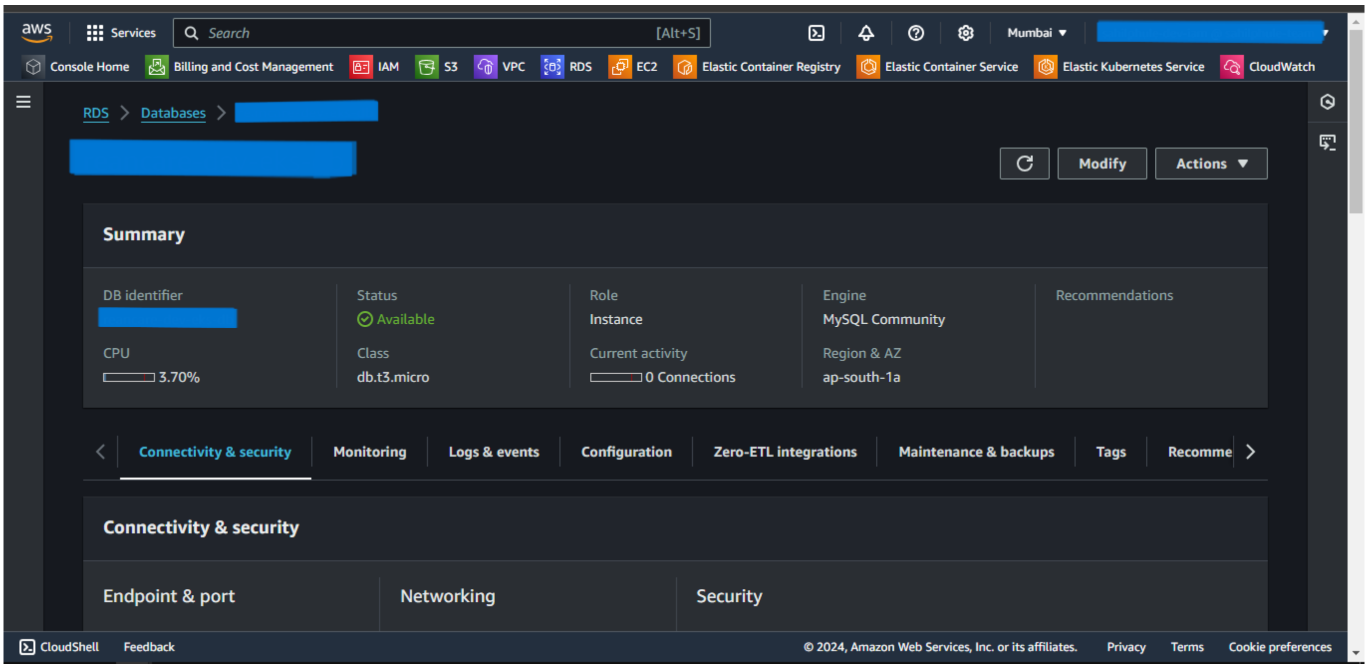
vpc-0519cba08238b0a67 / [redacted]

Details Info

VPC ID vpc-0519cba08238b0a67	State Available	DNS hostnames Enabled	DNS resolution Enabled
Tenancy Default	DHCP option set dopt-080dc7dd4e5998cf3	Main route table rtb-062d0e8788477bae9	Main network ACL acl-0e227b5ad284115f3
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 [redacted]	

Resource map Info

RDS Image



[redacted]

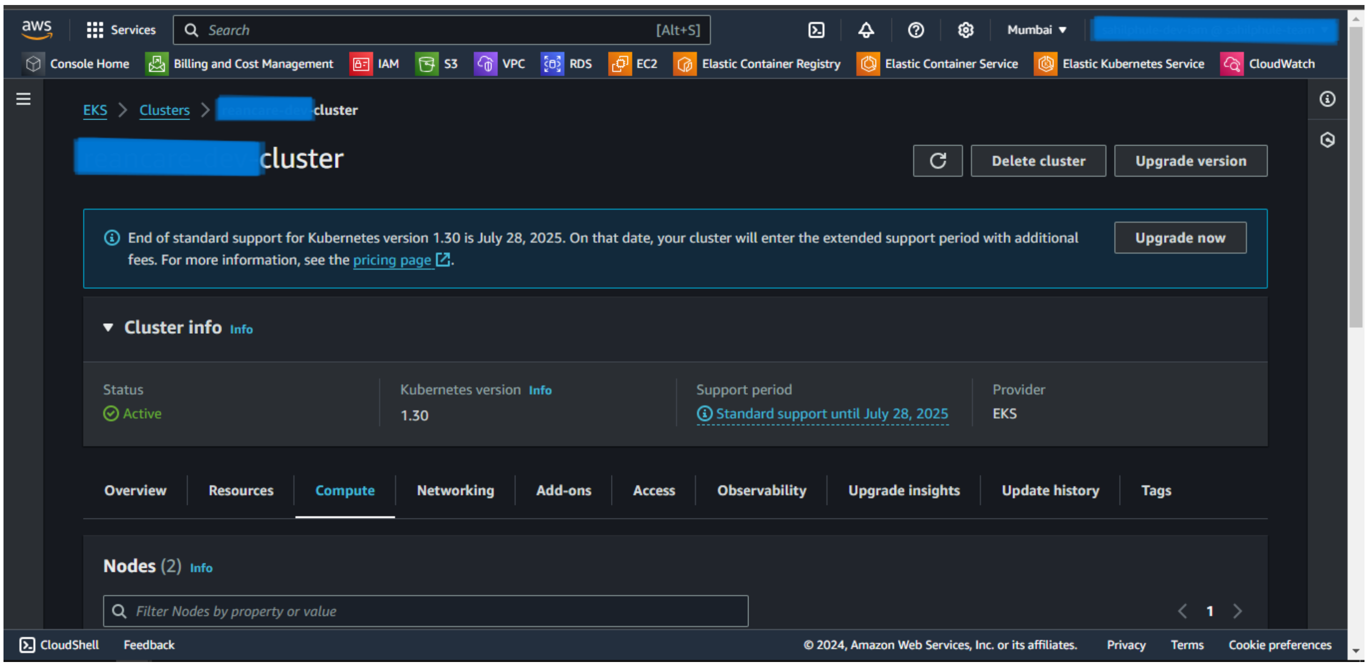
Summary

DB identifier [redacted]	Status Available	Role Instance	Engine MySQL Community	Recommendations
CPU 3.70%	Class db.t3.micro	Current activity 0 Connections	Region & AZ ap-south-1a	

Connectivity & security

Endpoint & port | Networking | Security

EKS Cluster Image



EKS > Clusters > [redacted]cluster

[redacted]cluster

End of standard support for Kubernetes version 1.30 is July 28, 2025. On that date, your cluster will enter the extended support period with additional fees. For more information, see the pricing page.

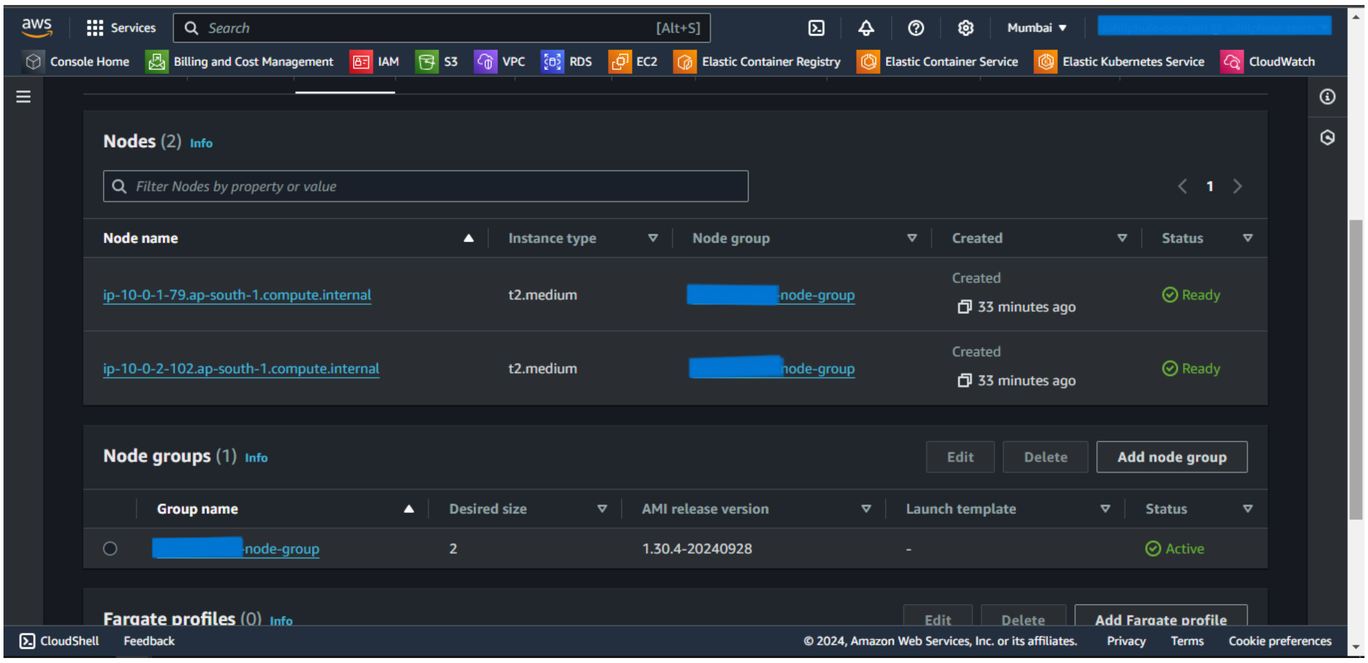
Cluster info Info

Status	Kubernetes version Info	Support period	Provider
Active	1.30	Standard support until July 28, 2025	EKS

Nodes (2) Info

© 2024, Amazon Web Services, Inc. or its affiliates.

EKS Node Group Image



Nodes (2) Info

Node name	Instance type	Node group	Created	Status
ip-10-0-1-79.ap-south-1.compute.internal	t2.medium	[redacted]node-group	Created 33 minutes ago	Ready
ip-10-0-2-102.ap-south-1.compute.internal	t2.medium	[redacted]node-group	Created 33 minutes ago	Ready

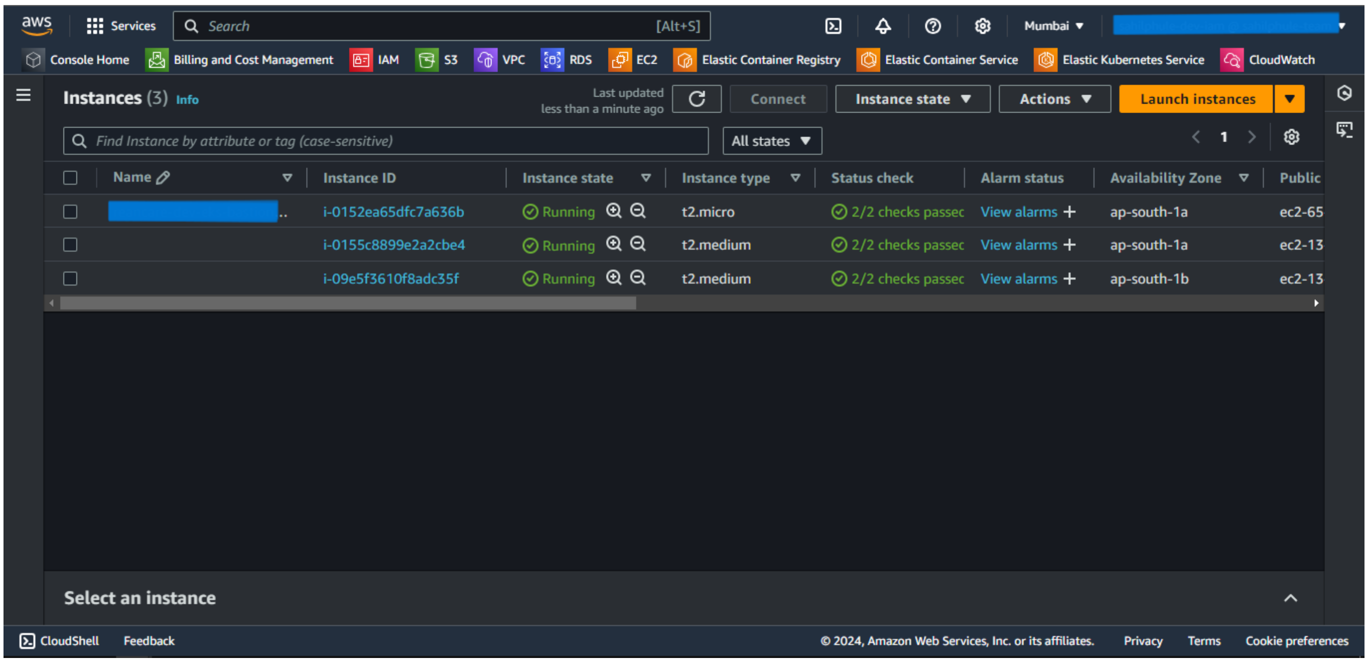
Node groups (1) Info

Group name	Desired size	AMI release version	Launch template	Status
[redacted]node-group	2	1.30.4-20240928	-	Active

Fargate profiles (0) Info

© 2024, Amazon Web Services, Inc. or its affiliates.

EKS Nodes Image

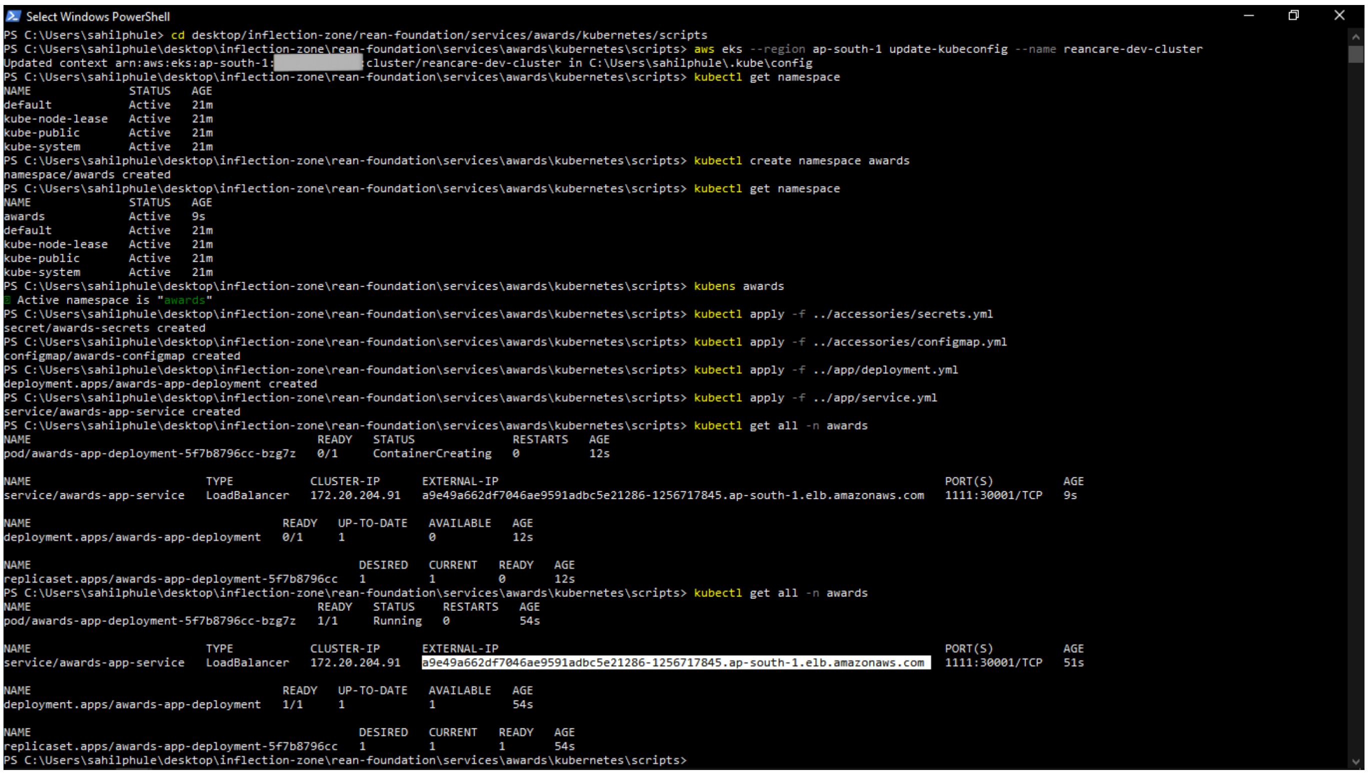

aws
Services
Search [Alt+S]
Mumbai
[redacted]
Console Home
Billing and Cost Management
IAM
S3
VPC
RDS
EC2
Elastic Container Registry
Elastic Container Service
Elastic Kubernetes Service
CloudWatch
Instances (3) Info
Last updated less than a minute ago
Connect
Instance state
Actions
Launch instances
Find Instance by attribute or tag (case-sensitive)
All states
1
Select an instance
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public
[redacted] ..	i-0152ea65dfc7a636b	Running	t2.micro	2/2 checks passec	View alarms +	ap-south-1a	ec2-65
	i-0155c8899e2a2cbe4	Running	t2.medium	2/2 checks passec	View alarms +	ap-south-1a	ec2-13
	i-09e5f3610f8adc35f	Running	t2.medium	2/2 checks passec	View alarms +	ap-south-1b	ec2-13

Connect to EKS Cluster from Powershell

1. Open a new Powershell window.
2. Run the following command to configure local kubectl with eks cluster
   `aws eks --region "region-name" update-kubeconfig --name "cluster-name"`
   Substitute *region-name* and *cluster-name* with the values defined in the above-created locals.tf file.
3. Now apply the Kubernetes manifest files of the application.
4. To list them all, run `kubectl get all`.

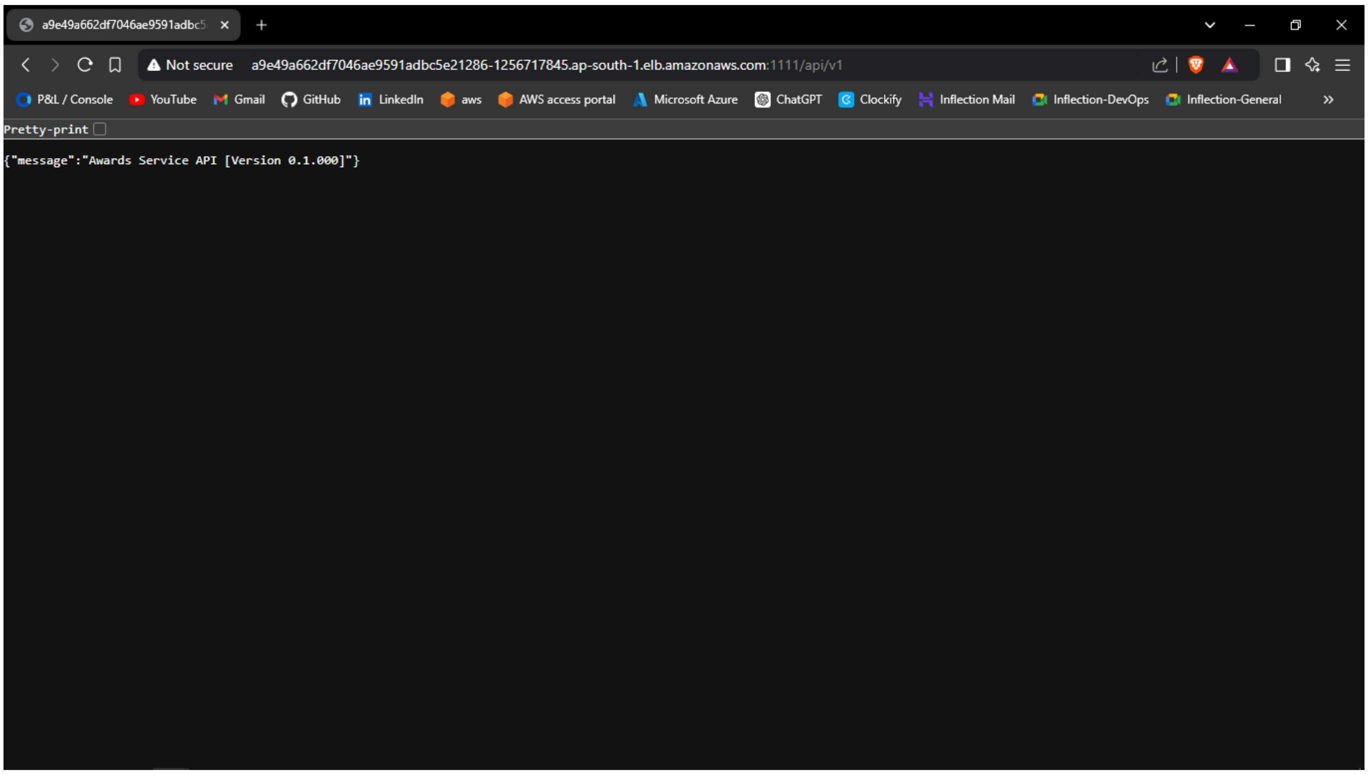
Powershell Image



```
PS C:\Users\sahilphule> cd desktop/inflection-zone/rean-foundation/services/awards/kubernetes/scripts
PS C:\Users\sahilphule\desktop\inflection-zone\rean-foundation\services\awards\kubernetes\scripts> aws eks --region ap-south-1 update-kubeconfig --name reancare-dev-cluster
Updated context arn:aws:eks:ap-south-1:[redacted]:cluster/reancare-dev-cluster in C:\Users\sahilphule\.kube\config
PS C:\Users\sahilphule\desktop\inflection-zone\rean-foundation\services\awards\kubernetes\scripts> kubectl get namespace
NAME              STATUS   AGE
default           Active   21m
kube-node-lease   Active   21m
kube-public       Active   21m
kube-system       Active   21m
PS C:\Users\sahilphule\desktop\inflection-zone\rean-foundation\services\awards\kubernetes\scripts> kubectl create namespace awards
namespace/awards created
PS C:\Users\sahilphule\desktop\inflection-zone\rean-foundation\services\awards\kubernetes\scripts> kubectl get namespace
NAME              STATUS   AGE
awards            Active   9s
default           Active   21m
kube-node-lease   Active   21m
kube-public       Active   21m
kube-system       Active   21m
PS C:\Users\sahilphule\desktop\inflection-zone\rean-foundation\services\awards\kubernetes\scripts> kubens awards
 Active namespace is "awards"
PS C:\Users\sahilphule\desktop\inflection-zone\rean-foundation\services\awards\kubernetes\scripts> kubectl apply -f ../accessories/secrets.yml
secret/awards-secrets created
PS C:\Users\sahilphule\desktop\inflection-zone\rean-foundation\services\awards\kubernetes\scripts> kubectl apply -f ../accessories/configmap.yml
configmap/awards-configmap created
PS C:\Users\sahilphule\desktop\inflection-zone\rean-foundation\services\awards\kubernetes\scripts> kubectl apply -f ../app/deployment.yml
deployment.apps/awards-app-deployment created
PS C:\Users\sahilphule\desktop\inflection-zone\rean-foundation\services\awards\kubernetes\scripts> kubectl apply -f ../app/service.yml
service/awards-app-service created
PS C:\Users\sahilphule\desktop\inflection-zone\rean-foundation\services\awards\kubernetes\scripts> kubectl get all -n awards
NAME                                     READY   STATUS              RESTARTS   AGE
pod/awards-app-deployment-5f7b8796cc-bzg7z   0/1     ContainerCreating   0          12s

NAME                         TYPE           CLUSTER-IP      EXTERNAL-IP                                                                PORT(S)          AGE
service/awards-app-service   LoadBalancer   172.20.204.91   a9e49a662df7046ae9591adbc5e21286-1256717845.ap-south-1.elb.amazonaws.com   1111:30001/TCP   9s

NAME                                    READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/awards-app-deployment   0/1     1            0           12s

NAME                                               DESIRED   CURRENT   READY   AGE
replicaset.apps/awards-app-deployment-5f7b8796cc   1         1         0       12s
PS C:\Users\sahilphule\desktop\inflection-zone\rean-foundation\services\awards\kubernetes\scripts> kubectl get all -n awards
NAME                                     READY   STATUS    RESTARTS   AGE
pod/awards-app-deployment-5f7b8796cc-bzg7z   1/1     Running   0          54s

NAME                         TYPE           CLUSTER-IP      EXTERNAL-IP                                                                PORT(S)          AGE
service/awards-app-service   LoadBalancer   172.20.204.91   a9e49a662df7046ae9591adbc5e21286-1256717845.ap-south-1.elb.amazonaws.com   1111:30001/TCP   51s

NAME                                    READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/awards-app-deployment   1/1     1            1           54s

NAME                                               DESIRED   CURRENT   READY   AGE
replicaset.apps/awards-app-deployment-5f7b8796cc   1         1         1       54s
PS C:\Users\sahilphule\desktop\inflection-zone\rean-foundation\services\awards\kubernetes\scripts>
```

5. If a Load Balancer type Service is present then try accessing the External IP of that service in the browser.

Browser Service Access

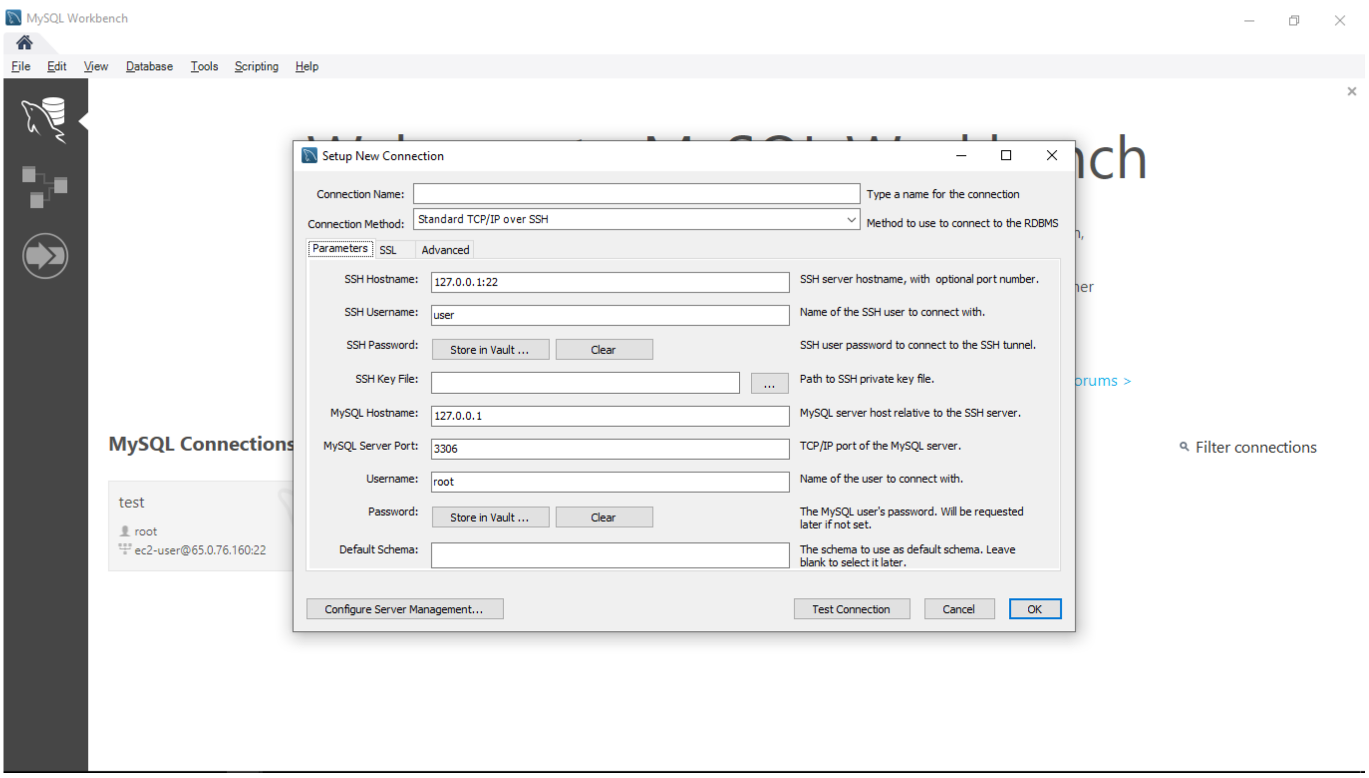


```
{"message":"Awards Service API [Version 0.1.000]"}
```

Connect to the RDS database through Bastion Host

1. Open MySQL Workbench.
2. Click Add Connection.
3. Select connection method as **Standard TCP/IP over SSH**.
4. In SSH Hostname, enter *bastion-host-ip:22* where bastion-host-ip is received from `terraform output`.
5. In SSH Username, enter *ec2-user*.
6. In SSH Key File, select *bastion-key.pem* file passed in above *locals.tf* file from your local computer.
7. In MySQL Hostname, enter *DB_HOST* where DB_HOST is received from `terraform output`.
8. In the Password section, select *Store in Vault*, and enter the password passed in above-created *locals.tf* file.
9. Click *OK* and open the connection.
10. Now you can run mysql commands to access databases, and verify the successful connection of *eks-nodes*.

MySQL Workbench Connection Page



Setup New Connection

Connection Name:		Type a name for the connection
Connection Method:	Standard TCP/IP over SSH	Method to use to connect to the RDBMS

Parameters | SSL | Advanced

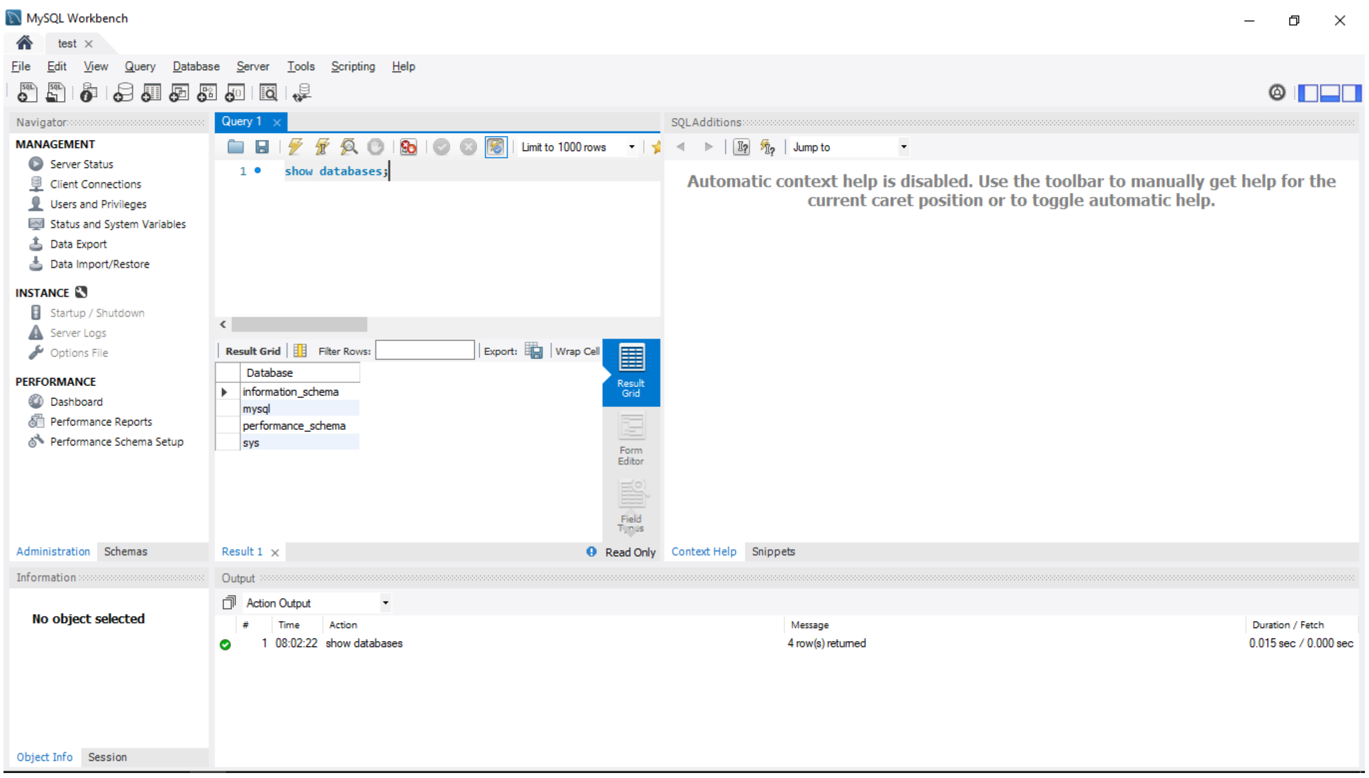
SSH Hostname:	127.0.0.1:22	SSH server hostname, with optional port number.
SSH Username:	user	Name of the SSH user to connect with.
SSH Password:	Store in Vault ... Clear	SSH user password to connect to the SSH tunnel.
SSH Key File:	...	Path to SSH private key file.
MySQL Hostname:	127.0.0.1	MySQL server host relative to the SSH server.
MySQL Server Port:	3306	TCP/IP port of the MySQL server.
Username:	root	Name of the user to connect with.
Password:	Store in Vault ... Clear	The MySQL user's password. Will be requested later if not set.
Default Schema:		The schema to use as default schema. Leave blank to select it later.

Configure Server Management... | Test Connection | Cancel | OK

MySQL Connections

test
root
ec2-user@65.0.76.160:22

MySQL Workbench Commands Page



```
show databases;
```

Database
information_schema
mysql
performance_schema
sys

Action Output

#	Time	Action	Message	Duration / Fetch
1	08:02:22	show databases	4 row(s) returned	0.015 sec / 0.000 sec

Destroy the provisioned infrastructure

1. Firstly, delete all the Kubernetes Deployments.
2. To destroy infrastructure, change directory to the above created Terraform Project.
3. Run `terraform destroy` & if prompted, type `yes`.
4. Infrastructure will be destroyed.
