

EKS Provisioning using Terraform

Prerequisites

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

Steps

1. Create the **eks-terraform** directory.
2. Folders structure for the above-created directory:

```
eks-terraform
├── .terraform.lock.hcl
├── locals.tf
├── main.tf
├── outputs.tf
├── providers.tf
├── terraform.tfstate
├── terraform.tfstate.backup
└── .terraform
```

We need to only create *providers.tf*, *main.tf*, *outputs.tf*, & *locals.tf* file. Other files are generated while initiating terraform.

3. Create a *providers.tf* file inside the above-created directory.
4. Inside the *provider.tf* file, define the following:
 - terraform
 - required_providers
 - provider
 - aws
5. Click [code](#) for reference.
6. The definition of *providers.tf* file is complete.
7. Now, create the *main.tf* file.
8. Inside *main.tf* file, we will use the following predefined modules:
 - vpc
 - rds
 - eks
9. Click [code](#) for reference.
10. The definition of *main.tf* file is complete.
11. Now we will create *outputs.tf* file.
12. Inside it, define the following outputs.
 - DB_HOST
 - bastion-host-ip

13. Click [code](#) for reference.
14. The definition of *outputs.tf* file is complete.
15. Now we will create *locals.tf* file.
16. Inside it, define the following variables:
 - vpc-properties
 - database-properties
 - bastion-properties
 - eks-properties
17. Click [code](#) for reference.
18. The definition of *locals.tf* file is complete.

Make sure you give the appropriate values to the variables defined in *locals.tf* file.

Provisioning the Infrastructure

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

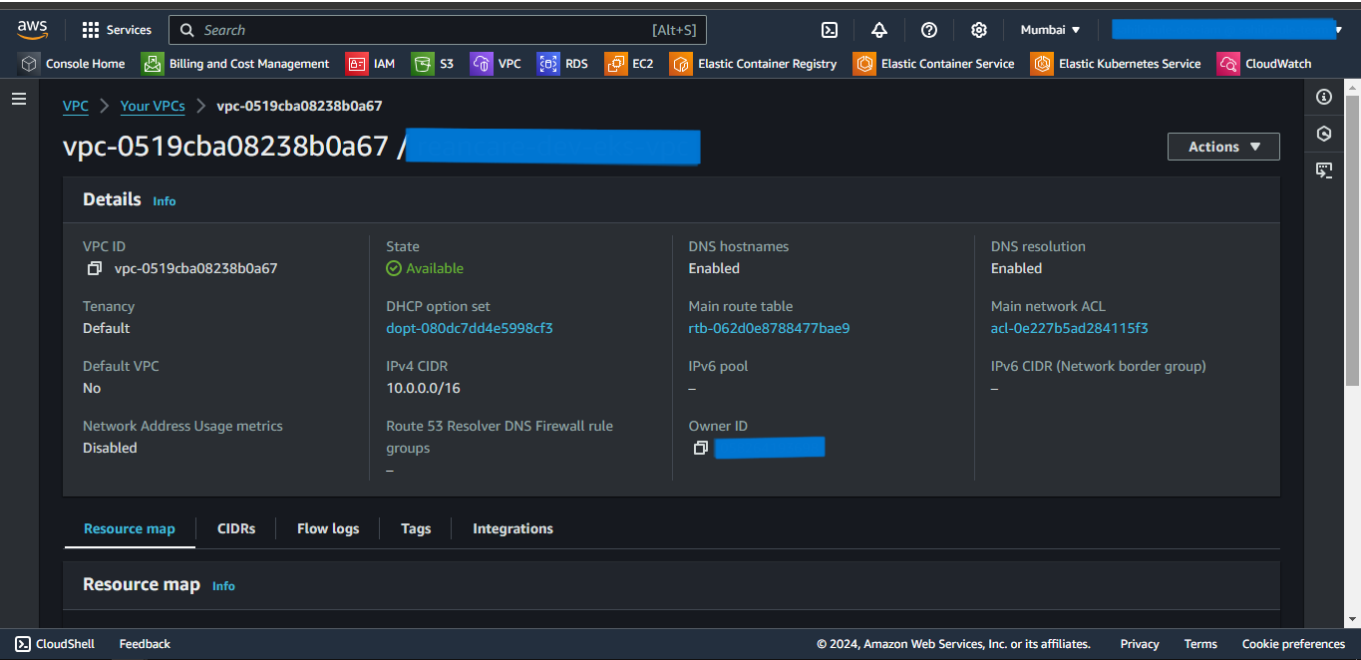
Ensure AWS CLI is configured with appropriate AWS user credentials and enough permissions.

Steps:

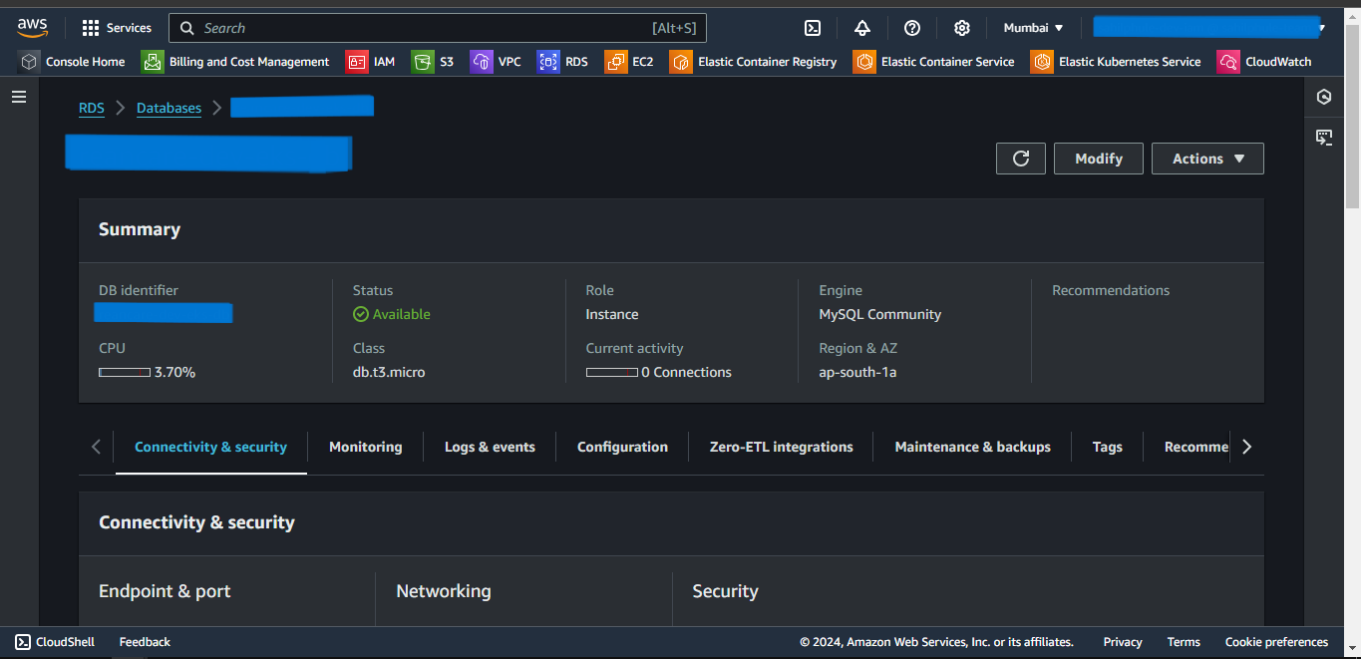
1. Open the PowerShell.
 2. Change the directory to the above-created **eks-terraform** directory using `cd` command.
 3. Run the `terraform init` command to initialize the *terraform*.
 4. Run the `terraform fmt --recursive` command to format the syntax of the files.
 5. Run the `terraform validate` 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

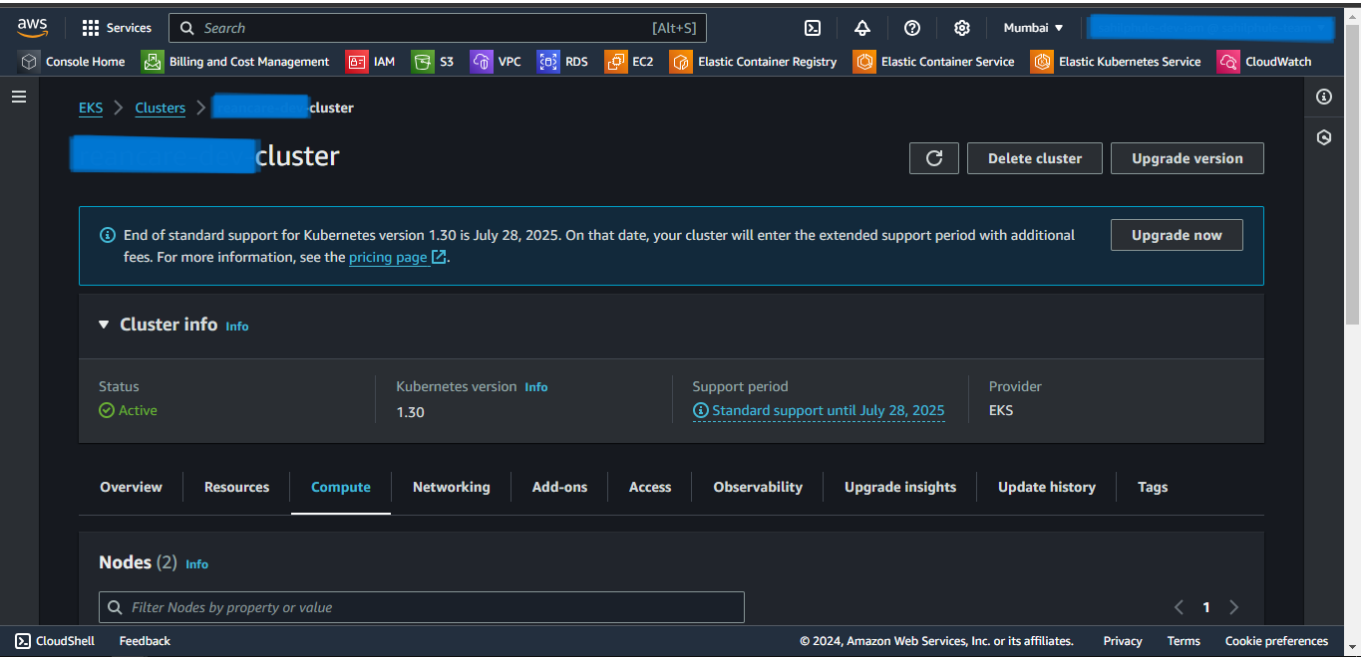
VPC Image



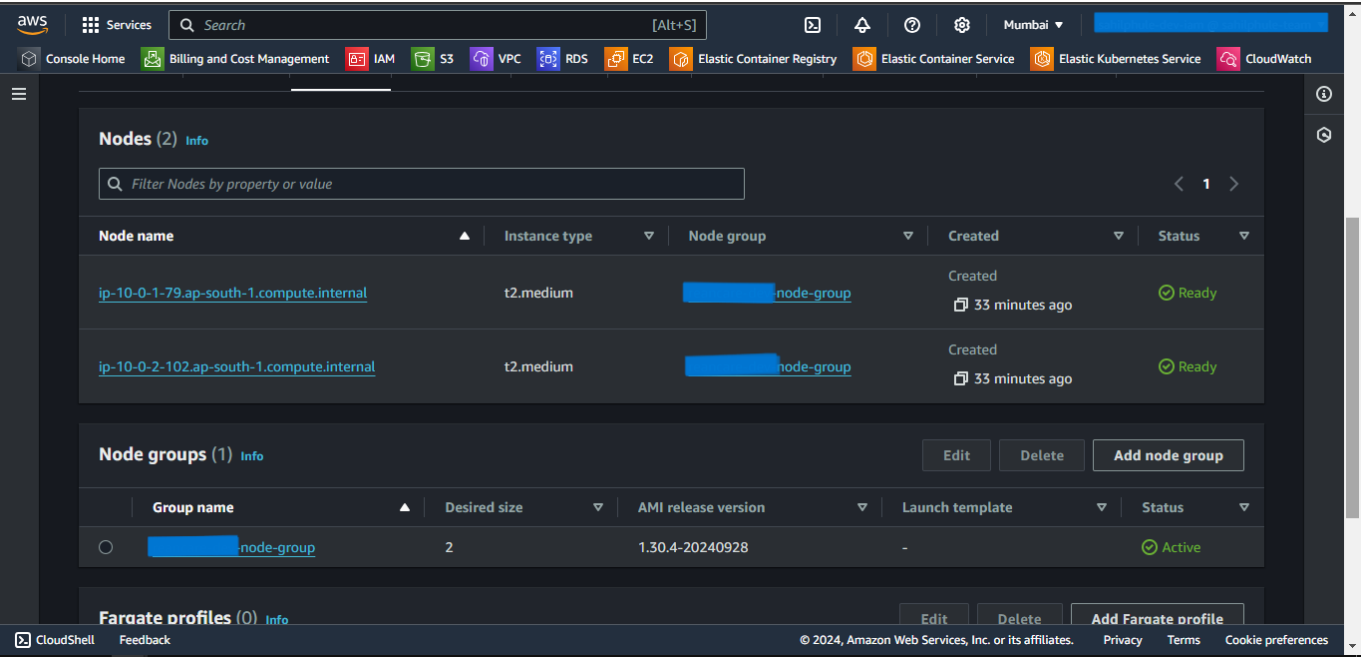
RDS Image



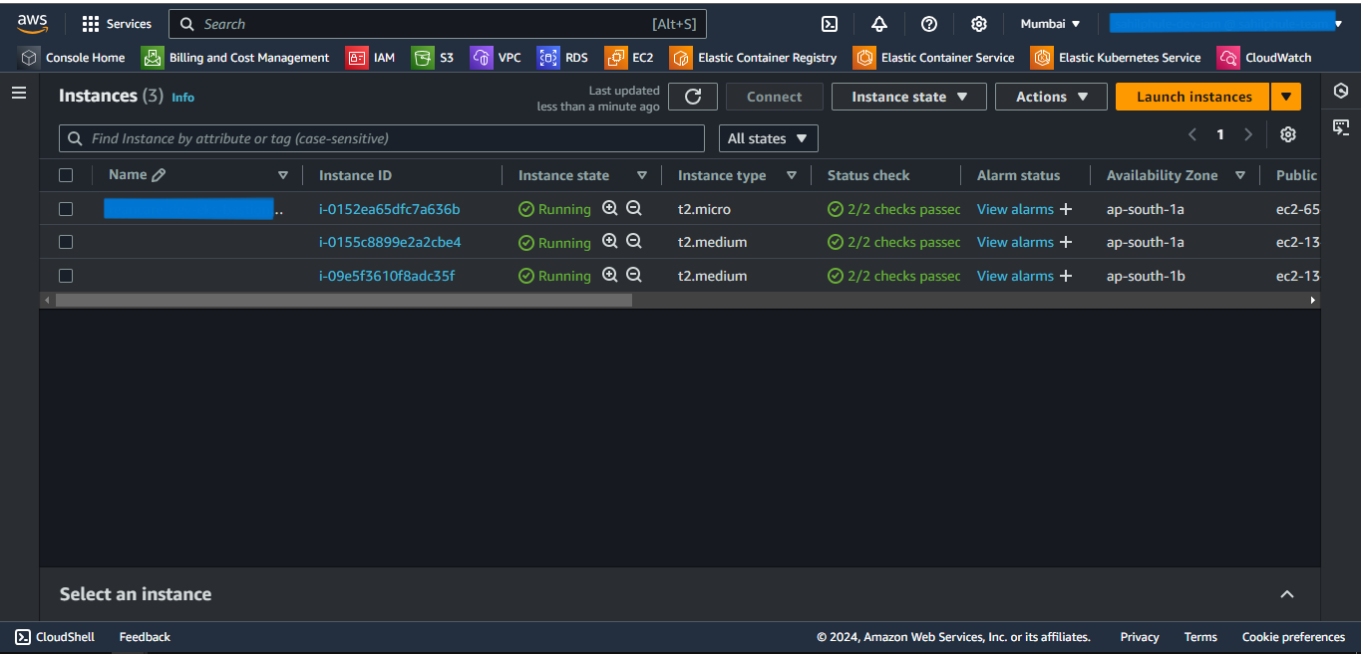
EKS Cluster Image



EKS Node Group Image



EKS Nodes Image



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

```
Select Windows PowerShell
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: 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
default              Active   21m
kube-node-lease      Active   21m
kube-public           Active   21m
kube-system           Active   21m
awards               Active   9s
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    a9e49a662df7046ae9591adb5e21286-1256717845.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    a9e49a662df7046ae9591adb5e21286-1256717845.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

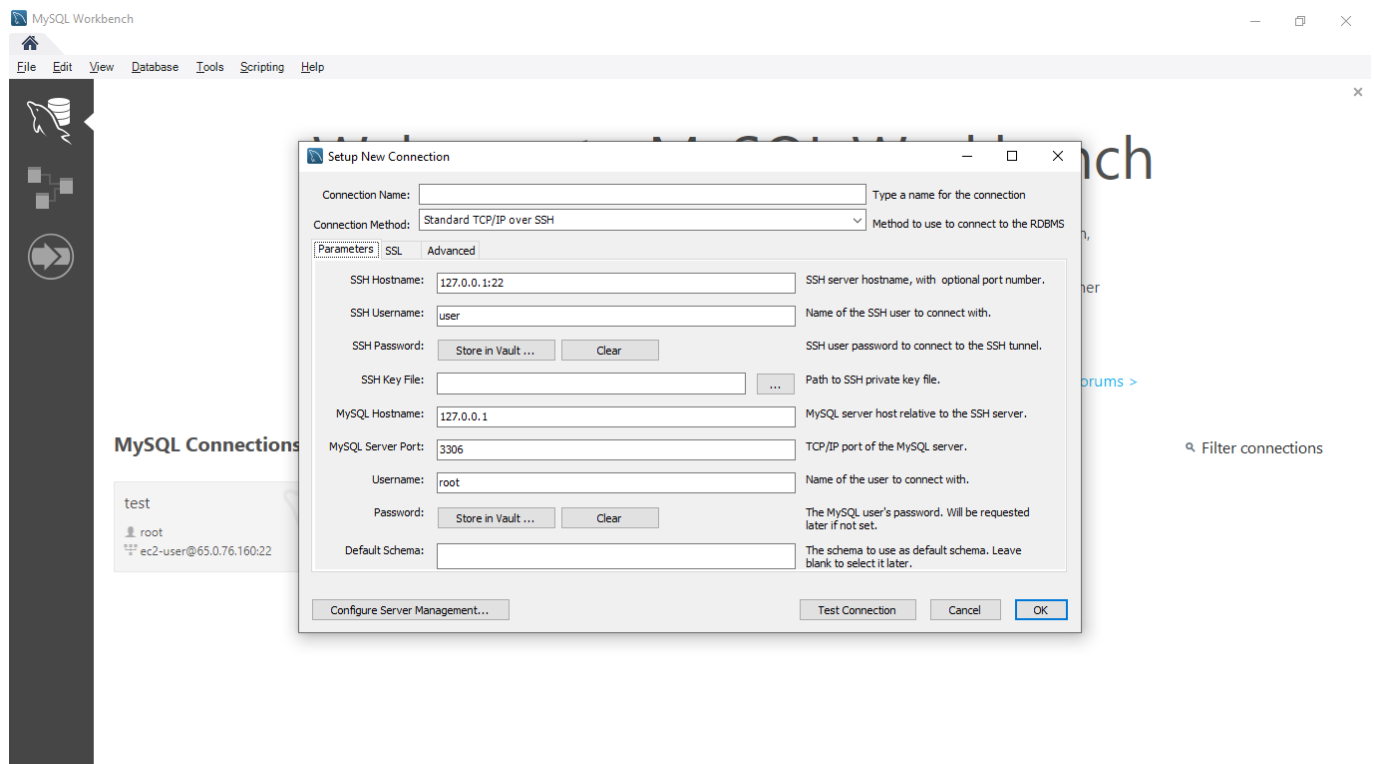
```
a9e49a662df7046ae9591adb5e21286-1256717845.ap-south-1.elb.amazonaws.com:1111/api/v1
Not secure a9e49a662df7046ae9591adb5e21286-1256717845.ap-south-1.elb.amazonaws.com:1111/api/v1
P&L / Console YouTube Gmail GitHub LinkedIn aws AWS access portal Microsoft Azure ChatGPT Clockify Inflection Mail Inflection-DevOps Inflection-General
Pretty-print
{"message": "Awards Service API [Version 0.1.000]"}
```

Connection to the RDS database through Bastion Host using MySQL Workbench

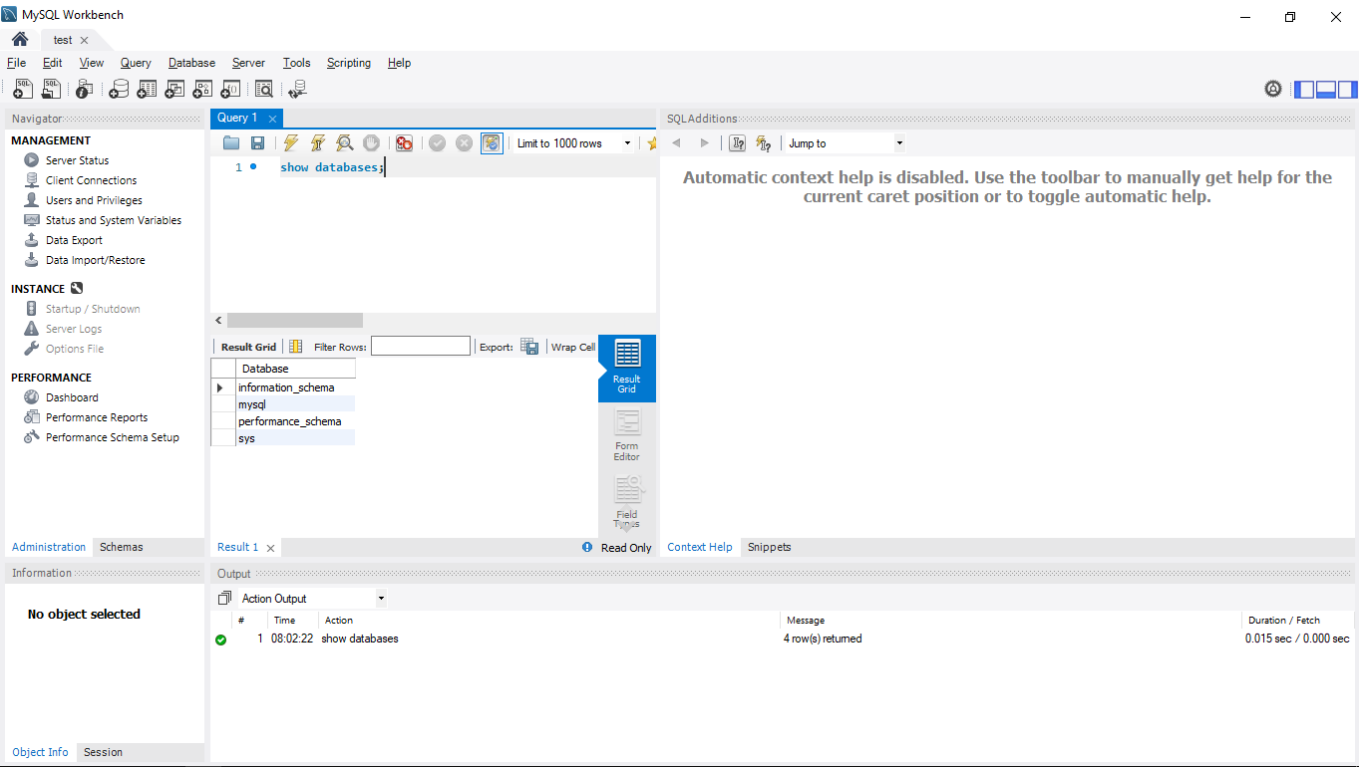
1. Open MySQL Workbench.
2. Click Add Connection.
3. Select connection method as **Standard TCP/IP over SSH**.
4. In SSH Hostname, enter *bastion-host-ip:22* where bastion-host-ip is received from *terraform output*.
5. In SSH Username, enter *ec2-user*.
6. In SSH Key File, select *bastion-key.pem* file 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*.

Screenshots of MySQL Workbench

Connection Page



Commands Page



Destroy the provisioned infrastructure

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