## Terraform ECS Deployment

#### Prerequisites

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

## Steps

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

```
ecs-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* files. Other files are generated while initiating terraform.

- 3. Create a *providers.tf* file inside the above-created directory.
- 4. Inside it, define the following:
  - o terraform
    - required\_providers
  - provider
    - docker
    - 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:
  - o vpc
  - o s3
  - o rds
  - o ecr
  - o load-balancer
  - o ecs
- 9. Also define the following s3 resource for uploading local .env file:
  - aws\_s3\_object
- 10. Click code for reference.

- 11. The definition of *main.tf* file is complete.
- 12. Now we will create outputs.tf file.
- 13. Inside it, define the following outputs.
  - DB HOST
  - bastion-host-ip
- 14. Click code for reference.
- 15. The definition of *outputs.tf* file is complete.
- 16. Now we will create locals.tf file.
- 17. Inside it, define the following variables:
  - vpc-properties
  - o s3-properties
  - database-properties
  - o bastion-properties
  - load-balancer-properties
  - ecs-properties
- 18. Click code for reference.
- 19. The definition of *locals.tf* file is complete.

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

## Provisioning the Infrastructure

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

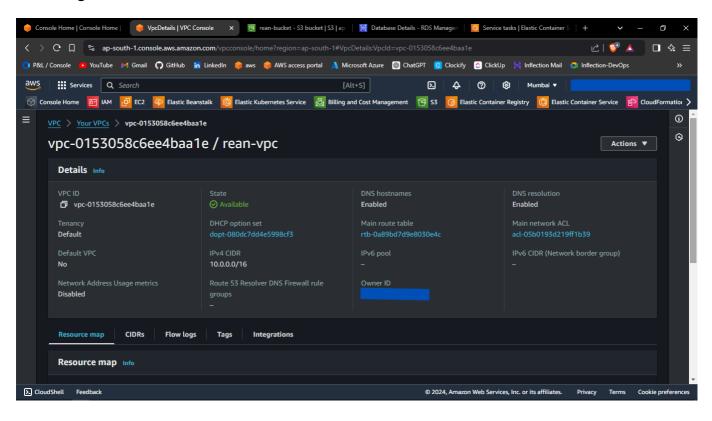
Ensure AWS CLI is configured with appropriate AWS user credentials and enough permissions.

#### Steps:

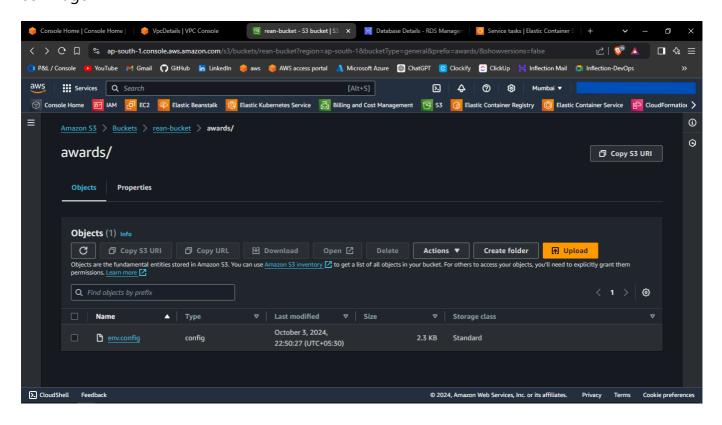
- 1. Open the PowerShell.
- 2. Change the directory to the above-created **ecs-terraform** directory using cd command.
- 3. Run the terraform init command to initialize the terraform.
- 4. Run the terraform fmt --recursive command to format the syntax of the files.
- 5. Run the terraform validate command 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.
- 10. Then,
  - Head towards EC2 dashboard.
  - Select Load Balancers, and select the created load balancer.
  - Copy the DNS address.
  - Paste the address in the browser to access the application.

#### Screenshots of Provisioned Infrastructure

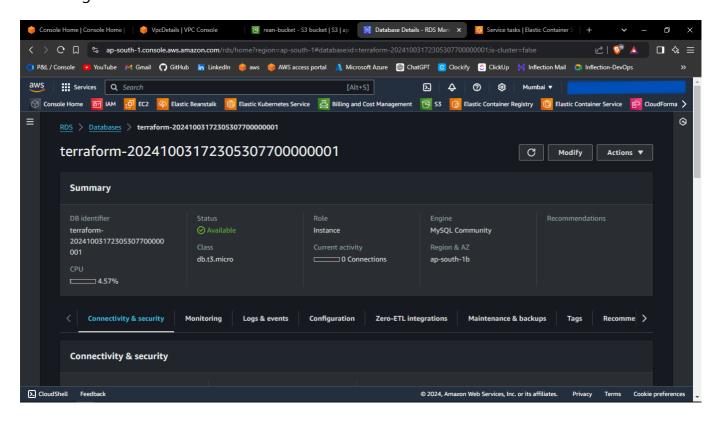
#### **VPC** Image



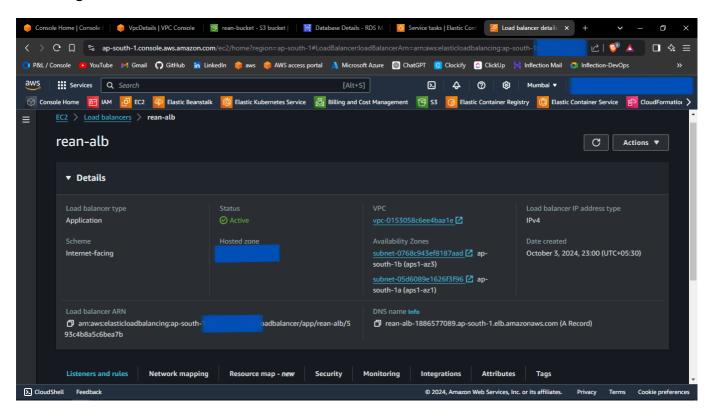
#### S3 Image



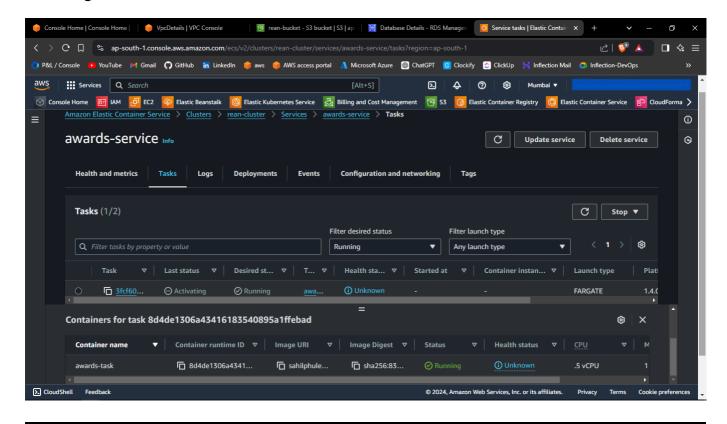
#### **RDS Image**



#### **ALB** Image



#### **ECS Image**

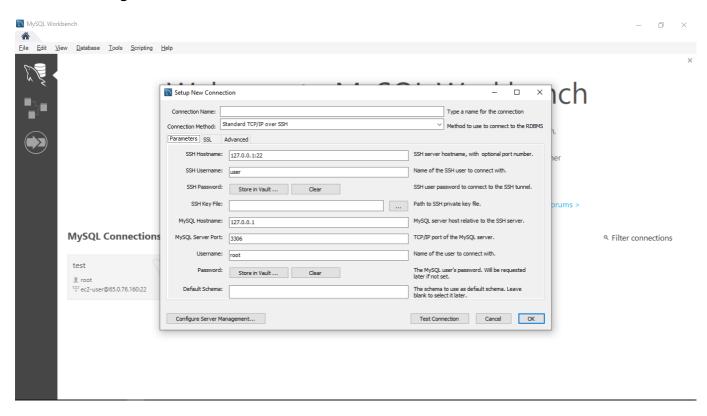


# Connection to the RDS database through Bastion Host using MySQL Workbench

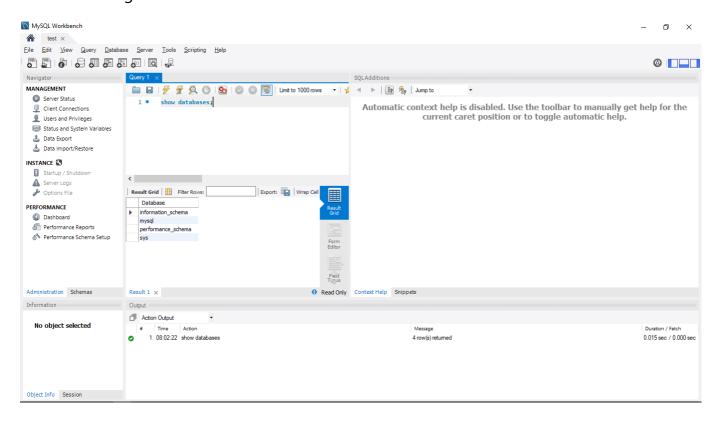
- 1. Open MySQL Workbench.
- 2. Click Add Connection.
- 3. Select connection method as Standard TCP/IP over SSH.
- 4. In SSH Hostname, enter bastion-host-ip:22 where bastion-host-ip is received from terraform output.
- 5. In SSH Username, enter ec2-user.
- 6. In SSH Key File, select bastion-key.pem file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

## Screenshots of MySQL Workbench

#### **Connection Page**



#### **Commands Page**



# Destroy the provisioned infrastructure

1. To destroy infrastructure, change directory to the above-created **ecs-terraform** directory using **cd** command.

- 2. Run terraform destroy & if prompted, type yes.
- 3. Infrastructure will be destroyed.