Create the variables file (variables.tf) and add four variables with these default values:

1. container\_name: mysql.

2. mysql\_root\_password: P4sSw0rd0!.

3. mysql\_network\_name: mysql\_internal\_network.

4. mysql\_volume\_name: mysql\_data.

variable "container\_name" {

default = "mysql"

}

variable "mysql\_root\_password"{

default = "P4sSw0rd0!"

}

variable "mysql\_network\_name" {

default = "mysql\_internal\_network"

}

variable "mysql\_volume\_name" {

default = "mysql\_data"

}

---------------------------------------------------------------------------------------------------------

Create the images file (images.tf)

1. Add the docker\_image resource and call it mysql\_image.

2. Set the name to mysql:5.7

resource "docker\_image" "mysql\_image" {

name = "mysql:5.7"

}

----------------------------------------------------------------------------------------------------------

Create the networks file (networks.tf):-

1. Add the docker\_network resource and call it private\_bridge\_network.

2. Set the name to use the mysql\_network\_name variable.

3. Set the driver to bridge.

4. Set internal to true.

resource "docker\_network" "private\_bridge\_network" {

name = var.mysql\_network\_name

driver = "bridge"

internal = true

}

--------------------------------------------------------------------------------------------------------

Create the volumes file (volume.tf):-

1. In volumes.tf add the docker\_volume resource and call it mysql\_data\_volume.

2. Set the name to use the mysql\_volume\_name variable.

resource "docker\_volume" "mysql\_data\_volume" {

name = var.mysql\_volume\_name

}

----------------------------------------------------------------------------------------------------------

Create the main file (main.tf):-

1. In main.tf add the docker\_container resource and call it mysql\_container.

2. Set the name to use the container\_name variable.

3. Set the image to use the name of the image coming from docker\_image.

4. Create an environment variable for MYSQL\_ROOT\_PASSWORD and set it to the mysql\_root\_password variable.

5. Configure the container volume to use the volume created by docker\_volume, and make sure the container\_path is set to /var/lib/mysql.

7. The container needs to use the network created by docker\_network.

terraform {

required\_providers {

docker = {

source = "kreuzwerker/docker"

version = ">= 2.13.0"

}

}

}

provider "docker" {

host = "npipe:////.//pipe//docker\_engine"

}

resource "docker\_container" "mysql\_container" {

name = var.container\_name

image = docker\_image.mysql\_image.name

env = [ "MYSQL\_ROOT\_PASSWORD = var.mysql\_root\_password" ]

volumes {

volume\_name = docker\_volume.mysql\_data\_volume.name

container\_path = "/var/lib/mysql"

}

networks\_advanced {

name = docker\_network.private\_bridge\_network.name

}

}

--------------------------------------------------------------------------------

Deploy the infrastructure

1. Initialize Terraform.

PS C:\Users\user\Desktop\terraform\ass5> terraform init

2. Validate the files.

PS C:\Users\user\Desktop\terraform\ass5> terraform validate

Success! The configuration is valid.

3. Generate a Terraform plan.

PS C:\Users\user\Desktop\terraform\ass5> terraform plan -out="tfass5.tfplan"

4. Deploy the infrastructure using the plan file.

PS C:\Users\user\Desktop\terraform\ass5> terraform apply "tfass5.tfplan"