kubernetes

Task-6

Deploy the Wordpress application on Kubernetes and AWS using terraform including the following steps;

1. Write an Infrastructure as code using terraform, which automatically deploy the Wordpress application
2. On AWS, use RDS service for the relational database for Wordpress application.
3. Deploy the Wordpress as a container either on top of Minikube or EKS or Fargate service on AWS
4. The Wordpress application should be accessible from the public world if deployed on AWS or through workstation if deployed on Minikube.

Prerequisite Required to do this task :

1. Creation of a AWS profile and its configuration.
2. Terraform installed(Terraform.exe and path variable set)
3. Minikube installed and running

For Wordpress Application:-

Step 1: Create a deployment using terraform and expose the port of the wordpress using service

We have terraform code as follows:-

```
resource "null_resource" "minikubestart" {
 provisioner "local-exec" {
      command = "minikube start"
   }
}
provider "kubernetes" {
 config_context_cluster = "minikube"
}
resource "kubernetes_deployment" "wordpress" {
 metadata {
  name = "wp"
 }
spec {
 replicas = 3
 selector {
  match_labels = {
   env = "production"
   region = "IN"
```

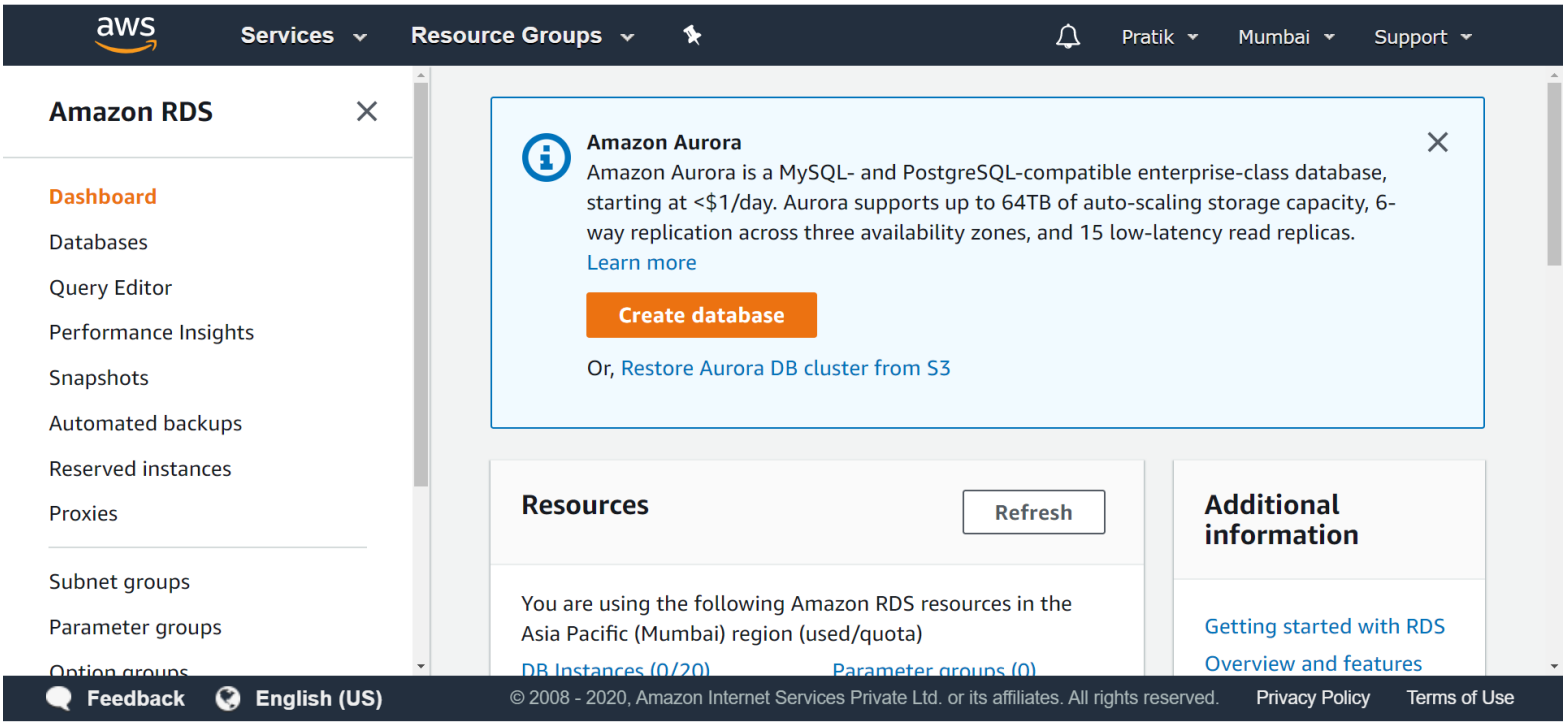
```
    App = "wordpress"
   }
   match_expressions {
    key = "env"
    operator = "In"
    values = ["production" , "webserver"]
   }
  }
  template {
   metadata {
    labels = {
     env = "production"
     region = "IN"
     App = "wordpress"
    }
   }
   spec {
    container {
     image = "wordpress:4.8-apache"
     name = "wp"
     }
    }
   }
  }
}
resource "kubernetes_service" "wordpresslb" {
 metadata {
  name = "wplb"
 }
 spec {
  selector = {
   app = "wordpress"
  }
  port {
   protocol = "TCP"
   port = 80
   target_port = 80
  }
  type = "NodePort"
 }
}
```

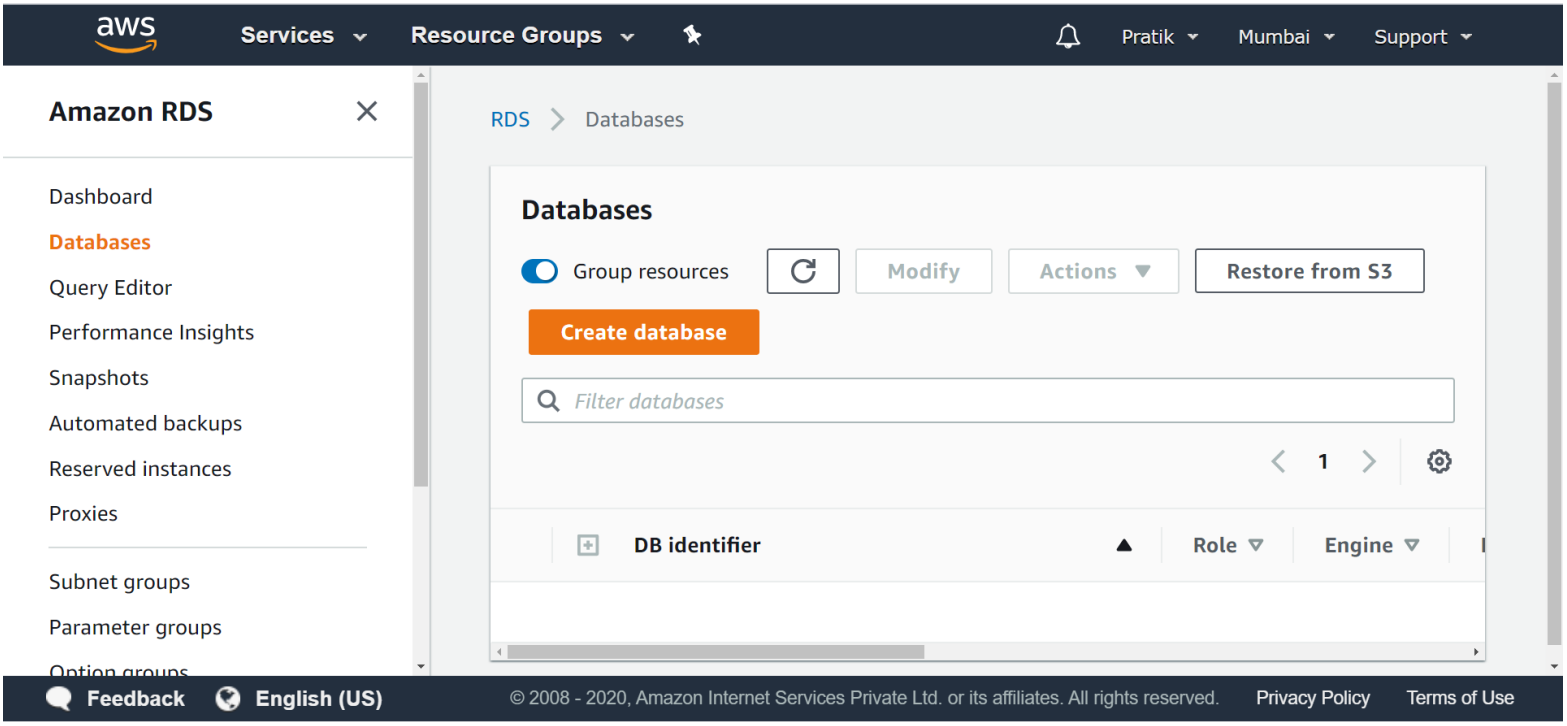
Step 2 :- Creation of RDS on top of AWS.

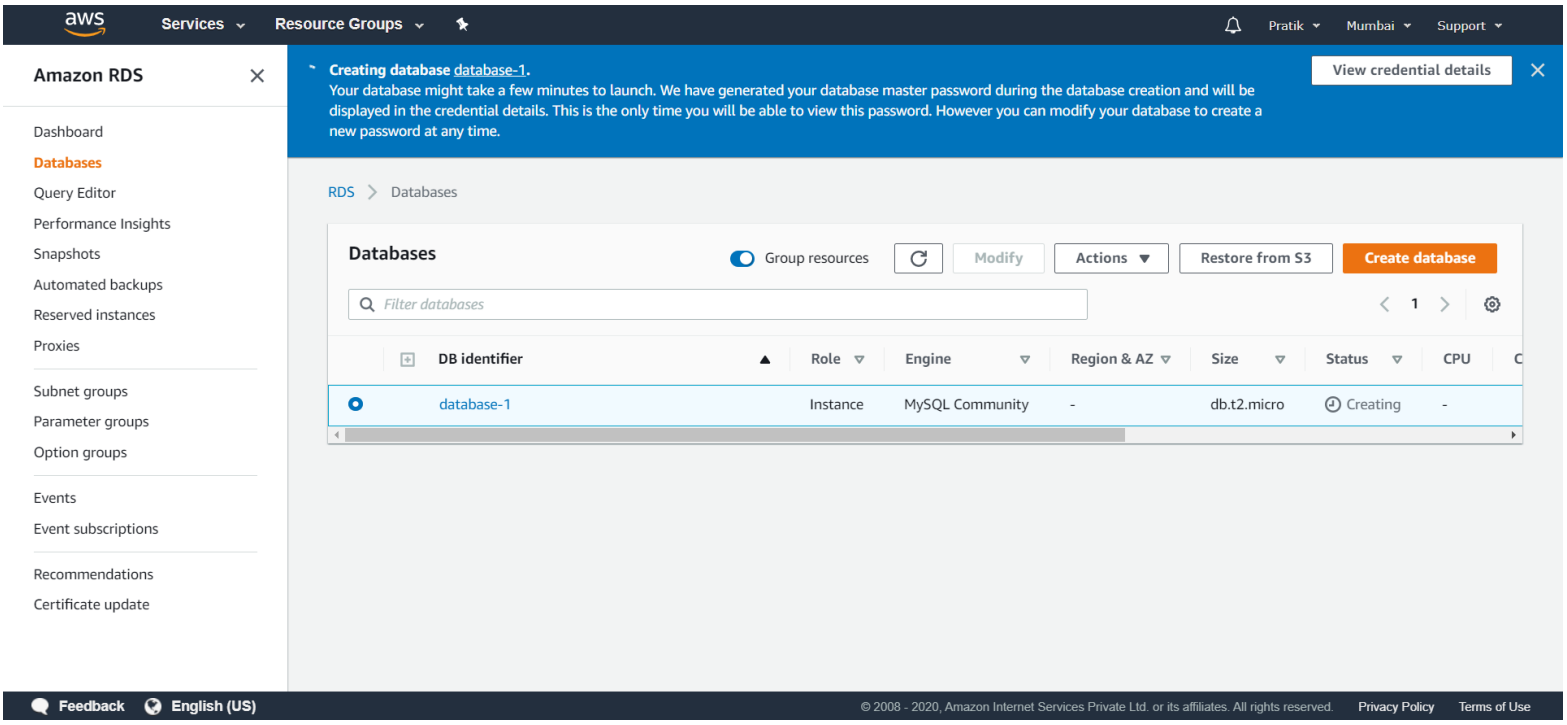
```
provider "aws" {
  region = "ap-south-1"
  profile = "mypratik"
}
resource "aws_db_instance" "mydb" {
  allocated_storage    = 20
  identifier           = "db-instance"
  storage_type         = "gp2"
  engine               = "mysql"
  engine_version       = "5.7.30"
  instance_class       = "db.t2.micro"
  name                 = "mydb"
  username             = "pratik"
  password             = "root123456"
  iam_database_authentication_enabled = true
  parameter_group_name = "default.mysql5.7"
  skip_final_snapshot  = true
  publicly_accessible = true
  tags = {
    Name = "sqldb"
  }
}
```

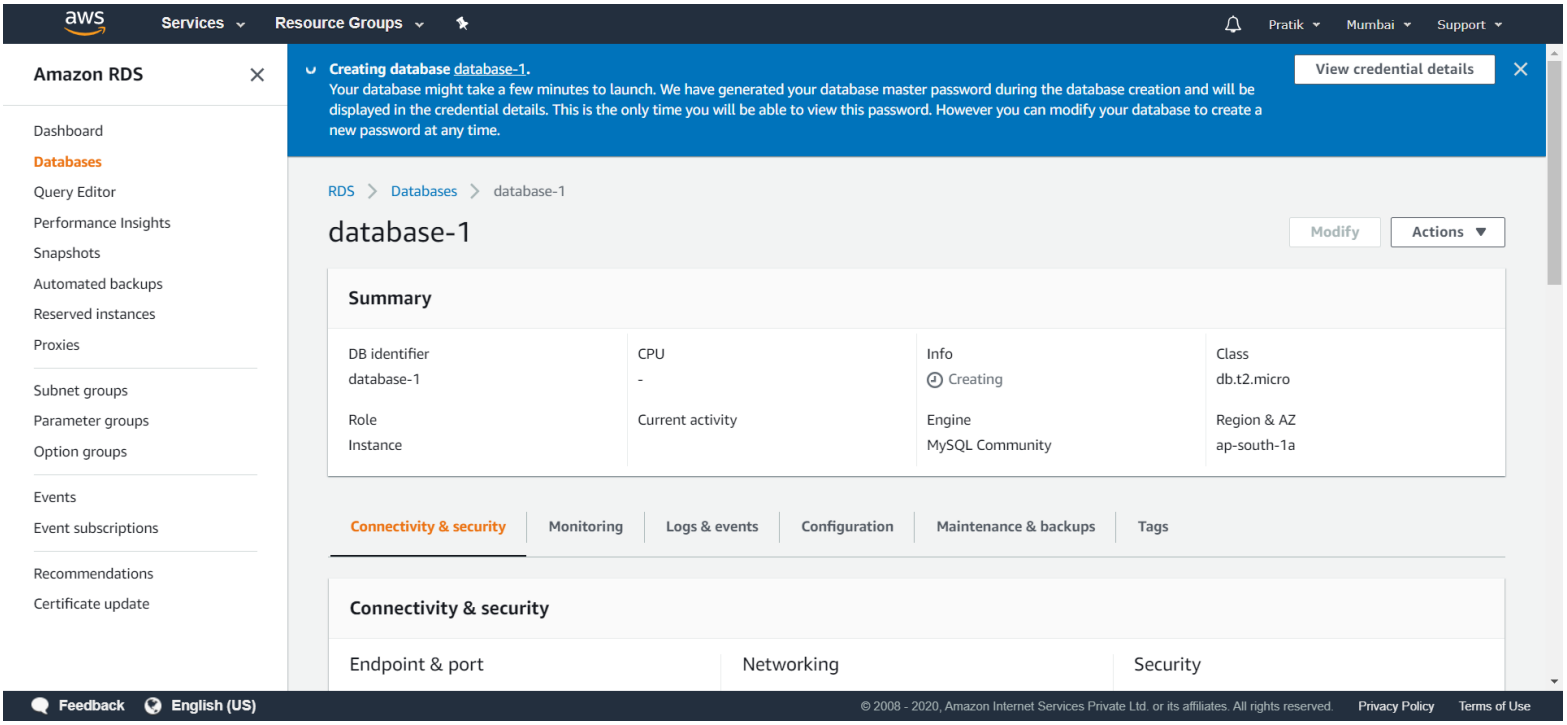
Step 3 :- Providing IP to connect to Wordpress Application

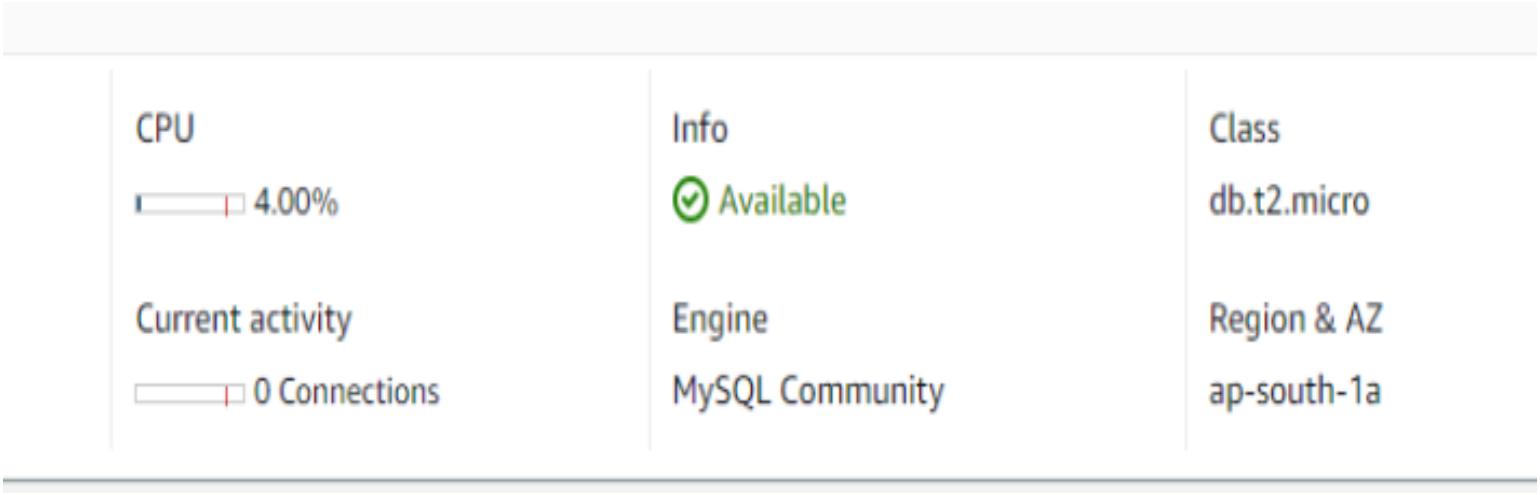
```
resource "null_resource" "minikubeservice"{
 provisioner "local-exec" {
      command = "minikube service list"
   }

depends_on = [
 kubernetes_deployment.wordpress,
 kubernetes_service.wordpresslb,
  aws_db_instance.mydb
 ]
}
```


aws
Services
Resource Groups
Pratik
Mumbai
Support
Amazon RDS
Dashboard
Databases
Query Editor
Performance Insights
Snapshots
Automated backups
Reserved instances
Proxies
Subnet groups
Parameter groups
Amazon Aurora
Amazon Aurora is a MySQL- and PostgreSQL-compatible enterprise-class database, starting at <$1/day. Aurora supports up to 64TB of auto-scaling storage capacity, 6-way replication across three availability zones, and 15 low-latency read replicas.
Learn more
Create database
Or, Restore Aurora DB cluster from S3
Resources
Refresh
You are using the following Amazon RDS resources in the Asia Pacific (Mumbai) region (used/quota)
Additional information
Getting started with RDS
Overview and features
Feedback
English (US)
© 2008 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use


aws
Services
Resource Groups
Pratik
Mumbai
Support
Amazon RDS
Dashboard
Databases
Query Editor
Performance Insights
Snapshots
Automated backups
Reserved instances
Proxies
Subnet groups
Parameter groups
RDS
Databases
Databases
Group resources
Modify
Actions
Restore from S3
Create database
Filter databases
1
DB identifier
Role
Engine
Feedback
English (US)
© 2008 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use


aws
Services
Resource Groups
Pratik
Mumbai
Support
Amazon RDS
Creating database database-1.
Your database might take a few minutes to launch. We have generated your database master password during the database creation and will be displayed in the credential details. This is the only time you will be able to view this password. However you can modify your database to create a new password at any time.
View credential details
Dashboard
Databases
Query Editor
Performance Insights
Snapshots
Automated backups
Reserved instances
Proxies
Subnet groups
Parameter groups
Option groups
Events
Event subscriptions
Recommendations
Certificate update
RDS > Databases
Databases
Group resources
Modify
Actions
Restore from S3
Create database
Filter databases
DB identifier
Role
Engine
Region & AZ
Size
Status
CPU
database-1
Instance
MySQL Community
-
db.t2.micro
Creating
-
Feedback
English (US)
© 2008 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use


aws
Services
Resource Groups
Pratik
Mumbai
Support
Amazon RDS
Creating database database-1.
Your database might take a few minutes to launch. We have generated your database master password during the database creation and will be displayed in the credential details. This is the only time you will be able to view this password. However you can modify your database to create a new password at any time.
View credential details
Dashboard
Databases
Query Editor
Performance Insights
Snapshots
Automated backups
Reserved instances
Proxies
Subnet groups
Parameter groups
Option groups
Events
Event subscriptions
Recommendations
Certificate update
RDS > Databases > database-1
database-1
Modify
Actions
Summary
DB identifier
database-1
CPU
-
Info
Creating
Class
db.t2.micro
Role
Instance
Current activity
Engine
MySQL Community
Region & AZ
ap-south-1a
Connectivity & security
Monitoring
Logs & events
Configuration
Maintenance & backups
Tags
Connectivity & security
Endpoint & port
Networking
Security
Feedback
English (US)
© 2008 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use



Run terraform init command

Command Prompt

```
Microsoft Windows [Version 10.0.19041.450]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\Asus>cd desktop

C:\Users\Asus\Desktop>cd terraformcodefiles

C:\Users\Asus\Desktop\terraformcodefiles>notepad task6.tf

C:\Users\Asus\Desktop\terraformcodefiles>terraform init

Initializing the backend...

Initializing provider plugins...
- Checking for available provider plugins...
- Downloading plugin for provider "null" (hashicorp/null) 2.1.2...

The following providers do not have any version constraints in configuration,
so the latest version was installed.

To prevent automatic upgrades to new major versions that may contain breaking
changes, it is recommended to add version = "..." constraints to the
corresponding provider blocks in configuration, with the constraint strings
suggested below.

* provider.aws: version = "~> 2.70"
* provider.kubernetes: version = "~> 1.12"
* provider.null: version = "~> 2.1"

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.

C:\Users\Asus\Desktop\terraformcodefiles>
```

Run terraform validate command

Run terraform apply or plan command

Command Prompt

```
C:\Users\Asus\Desktop\terraformcodefiles>terraform apply

An execution plan has been generated and is shown below.
Resource actions are indicated with the following symbols:
  + create

Terraform will perform the following actions:

  # aws_db_instance.mydb will be created
  + resource "aws_db_instance" "mydb" {
      + address                               = (known after apply)
      + allocated_storage                     = 20
      + apply_immediately                     = (known after apply)
      + arn                                   = (known after apply)
      + auto_minor_version_upgrade            = true
      + availability_zone                     = (known after apply)
      + backup_retention_period               = (known after apply)
      + backup_window                         = (known after apply)
      + ca_cert_identifier                    = (known after apply)
      + character_set_name                    = (known after apply)
      + copy_tags_to_snapshot                 = false
      + db_subnet_group_name                  = (known after apply)
      + delete_automated_backups              = true
      + endpoint                              = (known after apply)
      + engine                                = "mysql"
      + engine_version                        = "5.7.30"
      + hosted_zone_id                        = (known after apply)
      + iam_database_authentication_enabled   = true
      + id                                    = (known after apply)
      + identifier                            = "db-instance"
      + identifier_prefix                     = (known after apply)
      + instance_class                        = "db.t2.micro"
      + kms_key_id                            = (known after apply)
      + license_model                         = (known after apply)
      + maintenance_window                    = (known after apply)
      + monitoring_interval                   = 0
      + monitoring_role_arn                   = (known after apply)
      + multi_az                              = (known after apply)
      + name                                  = "mydb"
      + option_group_name                     = (known after apply)
      + parameter_group_name                  = "default.mysql5.7"
      + password                              = (sensitive value)
      + performance_insights_enabled          = false
      + performance_insights_kms_key_id       = (known after apply)
      + performance_insights_retention_period = (known after apply)
      + port                                  = (known after apply)
      + publicly_accessible                   = true
      + replicas                              = (known after apply)
```

Command Prompt

```
      + tags                                  = {
          + "Name" = "sqldb"
        }
      + timezone                              = (known after apply)
      + username                              = "pratik"
      + vpc_security_group_ids                = (known after apply)
    }

  # kubernetes_deployment.wordpress will be created
  + resource "kubernetes_deployment" "wordpress" {
      + id              = (known after apply)
      + wait_for_rollout = true

      + metadata {
          + generation       = (known after apply)
          + name             = "wp"
          + namespace        = "default"
          + resource_version = (known after apply)
          + self_link        = (known after apply)
          + uid              = (known after apply)
        }

      + spec {
          + min_ready_seconds         = 0
          + paused                    = false
          + progress_deadline_seconds = 600
          + replicas                  = 3
          + revision_history_limit    = 10

          + selector {
              + match_labels = {
                  + "App"    = "wordpress"
                  + "env"    = "production"
                  + "region" = "IN"
                }

              + match_expressions {
                  + key      = "env"
                  + operator = "In"
                  + values   = [
                      + "production",
                      + "webserver",
                    ]
                }
            }

          + strategy {
              + type = (known after apply)

              + rolling_update {
```

Command Prompt

```
                    }
                }
            }
        }
    }
  }

  # kubernetes_service.wordpresslb will be created
  + resource "kubernetes_service" "wordpresslb" {
      + id                   = (known after apply)
      + load_balancer_ingress = (known after apply)

      + metadata {
          + generation       = (known after apply)
          + name             = "wplb"
          + namespace        = "default"
          + resource_version = (known after apply)
          + self_link        = (known after apply)
          + uid              = (known after apply)
        }

      + spec {
          + cluster_ip                  = (known after apply)
          + external_traffic_policy     = (known after apply)
          + publish_not_ready_addresses = false
          + selector                    = {
              + "app" = "wordpress"
            }
          + session_affinity            = "None"
          + type                        = "NodePort"

          + port {
              + node_port   = (known after apply)
              + port        = 80
              + protocol    = "TCP"
              + target_port = "80"
            }
        }
    }

  # null_resource.minikubeservice will be created
  + resource "null_resource" "minikubeservice" {
      + id = (known after apply)
    }

  # null_resource.minikubestart will be created
  + resource "null_resource" "minikubestart" {
      + id = (known after apply)
    }
```

Command Prompt

```
  # null_resource.minikubeservice will be created
  + resource "null_resource" "minikubeservice" {
      + id = (known after apply)
    }

  # null_resource.minikubestart will be created
  + resource "null_resource" "minikubestart" {
      + id = (known after apply)
    }

Plan: 5 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

null_resource.minikubestart: Creating...
null_resource.minikubestart: Provisioning with 'local-exec'...
null_resource.minikubestart (local-exec): Executing: ["cmd" "/C" "minikube start"]
kubernetes_service.wordpresslb: Creating...
kubernetes_deployment.wordpress: Creating...
null_resource.minikubestart (local-exec): * minikube v1.11.0 on Microsoft Windows 10 Home Single Language 10.0.19041 Build 19041
null_resource.minikubestart (local-exec): * Using the virtualbox driver based on existing profile
null_resource.minikubestart (local-exec): * Starting control plane node minikube in cluster minikube
null_resource.minikubestart (local-exec): * Restarting existing virtualbox VM for "minikube" ...
aws_db_instance.mydb: Creating...
null_resource.minikubestart: Still creating... [10s elapsed]
kubernetes_service.wordpresslb: Still creating... [10s elapsed]
kubernetes_deployment.wordpress: Still creating... [10s elapsed]
null_resource.minikubestart: Still creating... [20s elapsed]
kubernetes_service.wordpresslb: Still creating... [20s elapsed]
kubernetes_deployment.wordpress: Still creating... [20s elapsed]
null_resource.minikubestart: Still creating... [30s elapsed]
null_resource.minikubestart: Still creating... [40s elapsed]
null_resource.minikubestart: Still creating... [50s elapsed]
null_resource.minikubestart: Still creating... [1m0s elapsed]
null_resource.minikubestart: Still creating... [1m10s elapsed]
null_resource.minikubestart: Still creating... [1m20s elapsed]
null_resource.minikubestart (local-exec): * Preparing Kubernetes v1.18.3 on Docker 19.03.8 ...
null_resource.minikubestart: Still creating... [1m30s elapsed]
null_resource.minikubestart: Still creating... [1m40s elapsed]
null_resource.minikubestart: Still creating... [1m50s elapsed]
null_resource.minikubestart: Still creating... [2m0s elapsed]
null_resource.minikubestart (local-exec): * Verifying Kubernetes components...
null_resource.minikubestart (local-exec): * Enabled addons: default-storageclass, storage-provisioner
null_resource.minikubestart (local-exec): * Done! kubectl is now configured to use "minikube"
null_resource.minikubestart: Creation complete after 2m7s [id=348086797605676458]
```

Now Run -minikube service list command in command prompt

```
null_resource.minikubeservice (local-exec): Executing: ["cmd" "/C" "minikube service list"]
null_resource.minikubeservice (local-exec): |---------------------|---------------------------|--------------|
null_resource.minikubeservice (local-exec): |      NAMESPACE      |           NAME            | TARGET PORT  |
null_resource.minikubeservice (local-exec): |---------------------|---------------------------|--------------|
null_resource.minikubeservice (local-exec): | default             | kubernetes                | No node port |
null_resource.minikubeservice (local-exec): | default             | wplb                      |           80 |
null_resource.minikubeservice (local-exec): | kube-system         | kube-dns                  | No node port |
null_resource.minikubeservice (local-exec): | kube-system         | tiller-deploy             | No node port |
null_resource.minikubeservice (local-exec): | kubernetes-dashboard | dashboard-metrics-scraper | No node port |
null_resource.minikubeservice (local-exec): | kubernetes-dashboard | kubernetes-dashboard      | No node port |
null_resource.minikubeservice (local-exec): |---------------------|---------------------------|--------------|
null_resource.minikubeservice: Creation complete after 6s [id=8079379263171424586]

Apply complete! Resources: 5 added, 0 changed, 0 destroyed.
```


```
Apply complete! Resources: 5 added, 0 changed, 0 destroyed.
```


Download a Popular SQL Client :-

1.Database instance creation is complete and status changes to available, you can connect to a database on the DB instance using any standard SQL client.

2.In this step, we will download MySQL Workbench.

3. Go to the Download MySQL Workbench page to download and install MySQL Workbench. For Additional information ,go through the MySQL Docs.

4.Note: Remember to run MySQL Workbench from the same device from which you created the DB Instance. The security group your database is placed in is configured to allow connection only from the device from which you created the DB instance.

dev.mysql.com/downloads/workbench/

General Availability (GA) Releases | Archives

MySQL Workbench 8.0.21

Select Operating System:

Microsoft Windows

Recommended Download:

MySQL Installer for Windows

All MySQL Products. For All Windows Platforms. In One Package.

Starting with MySQL 5.6 the MySQL Installer package replaces the standalone MSI packages.

Windows (x86, 32 & 64-bit), MySQL Installer MSI — Go to Download Page >

Other Downloads:

Windows (x86, 64-bit), MSI Installer	8.0.21	35.7M	Download
(mysql-workbench-community-8.0.21-winx64.msi)		MD5: bea0696dd7b8cbab25357ee8bf725639 | Signature	

We suggest that you use the MD5 checksums and GnuPG signatures to verify the integrity of the packages you download.

dev.mysql.com/downloads/file/?id=497505

MySQL Community Downloads

Login Now or Sign Up for a free account.

An Oracle Web Account provides you with the following advantages:

- Fast access to MySQL software downloads
- Download technical White Papers and Presentations
- Post messages in the MySQL Discussion Forums
- Report and track bugs in the MySQL bug system

Login » using my Oracle Web account

Sign Up » for an Oracle Web account

MySQL.com is using Oracle SSO for authentication. If you already have an Oracle Web account, click the Login link. Otherwise, you can signup for a free account by clicking the Sign Up link and following the instructions.

No thanks, just start my download.

dev.mysql.com/downloads/file/?id=497505

MySQL Community Downloads

Login Now or Sign Up for a free account.

An Oracle Web Account provides you with the following advantages:

- Fast access to MySQL software downloads
- Download technical White Papers and Presentations
- Post messages in the MySQL Discussion Forums
- Report and track bugs in the MySQL bug system

Login » using my Oracle Web account

Sign Up » for an Oracle Web account

MySQL.com is using Oracle SSO for authentication. If you already have an Oracle Web account, click the Login link. Otherwise, you can signup for a free account by clicking the Sign Up link and following the instructions.

No thanks, just start my download.

mysql-workbench-....msi
3.9/35.7 MB, 40 secs left

Show all

dev.mysql.com/downloads/file/?id=497505

MySQL Community Downloads

Login Now or Sign Up for a free account.

An Oracle Web Account provides you with the following advantages:

- Fast access to MySQL software downloads
- Download technical White Papers and Presentations
- Post messages in the MySQL Discussion Forums
- Report and track bugs in the MySQL bug system

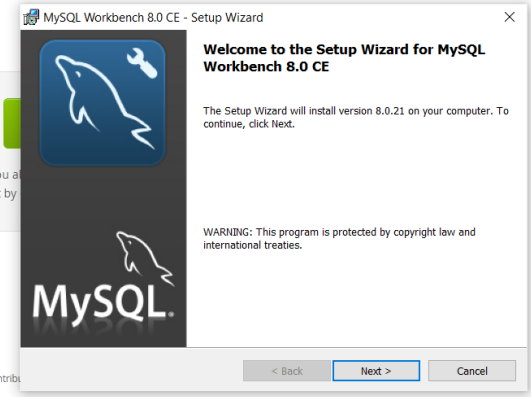
Login » using my Oracle Web account

MySQL.com is using Oracle SSO for authentication. If you a
Login link. Otherwise, you can signup for a free account by
instructions.

No thanks, just start my download.

ORACLE © 2020, Oracle Corporation and/or its affiliates

Legal Policies | Your Privacy Rights | Terms of Use | Trademark Policy | Contribu



mysql-workbench-....msi

Show all

dev.mysql.com/downloads/file/?id=497505

MySQL Community Downloads

Login Now or Sign Up for a free account.

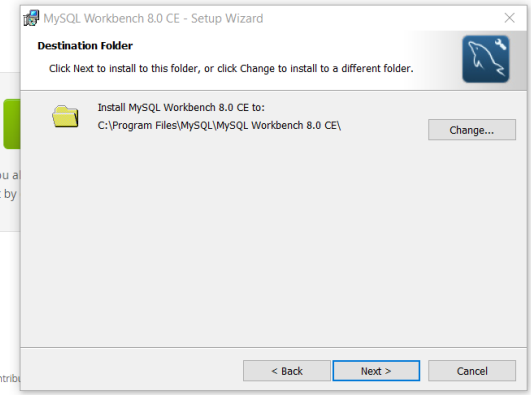
An Oracle Web Account provides you with the following advantages:

- Fast access to MySQL software downloads
- Download technical White Papers and Presentations
- Post messages in the MySQL Discussion Forums
- Report and track bugs in the MySQL bug system

Login »
using my Oracle Web account

MySQL.com is using Oracle SSO for authentication. If you a
Login link. Otherwise, you can signup for a free account by
instructions.

No thanks, just start my download.



dev.mysql.com/downloads/file/?id=497505

MySQL Community Downloads

Login Now or Sign Up for a free account.

An Oracle Web Account provides you with the following advantages:

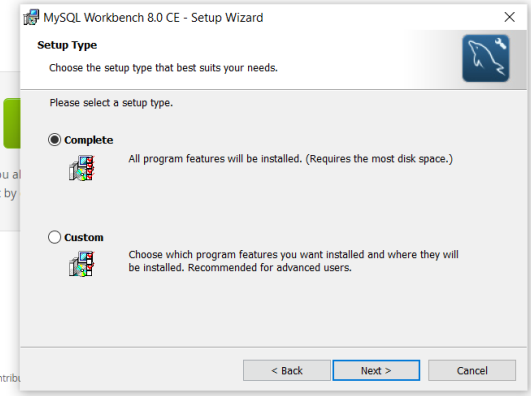
- Fast access to MySQL software downloads
- Download technical White Papers and Presentations
- Post messages in the MySQL Discussion Forums
- Report and track bugs in the MySQL bug system

Login »
using my Oracle Web account

MySQL.com is using Oracle SSO for authentication. If you a
Login link. Otherwise, you can signup for a free account by
instructions.

No thanks, just start my download.

ORACLE © 2020, Oracle Corporation and/or its affiliates
Legal Policies | Your Privacy Rights | Terms of Use | Trademark Policy | Contribu



dev.mysql.com/downloads/file/?id=497505

MySQL Community Downloads

Login Now or Sign Up for a free account.

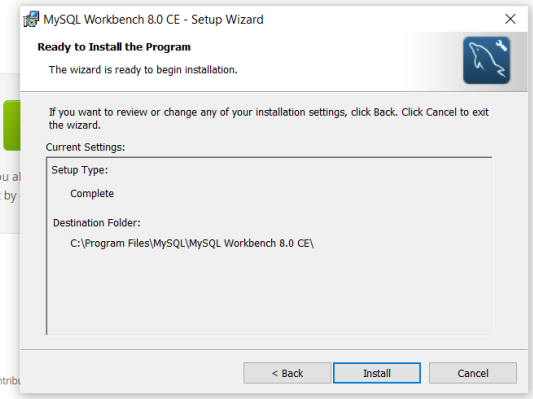
An Oracle Web Account provides you with the following advantages:

- Fast access to MySQL software downloads
- Download technical White Papers and Presentations
- Post messages in the MySQL Discussion Forums
- Report and track bugs in the MySQL bug system

Login »
using my Oracle Web account

MySQL.com is using Oracle SSO for authentication. If you a
Login link. Otherwise, you can signup for a free account by
instructions.

No thanks, just start my download.



dev.mysql.com/downloads/file/?id=497505

MySQL Community Downloads

Login Now or Sign Up for a free account.

An Oracle Web Account provides you with the following advantages:

- Fast access to MySQL software downloads
- Download technical White Papers and Presentations
- Post messages in the MySQL Discussion Forums
- Report and track bugs in the MySQL bug system

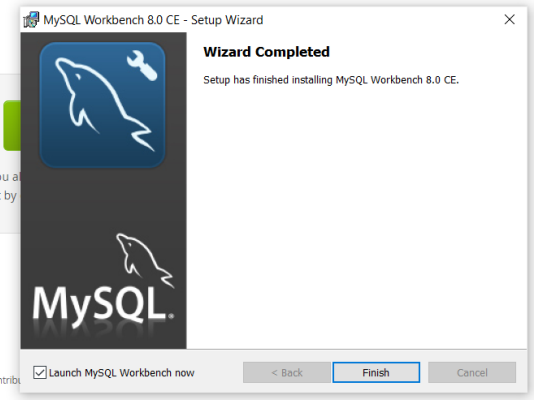
Login »
using my Oracle Web account

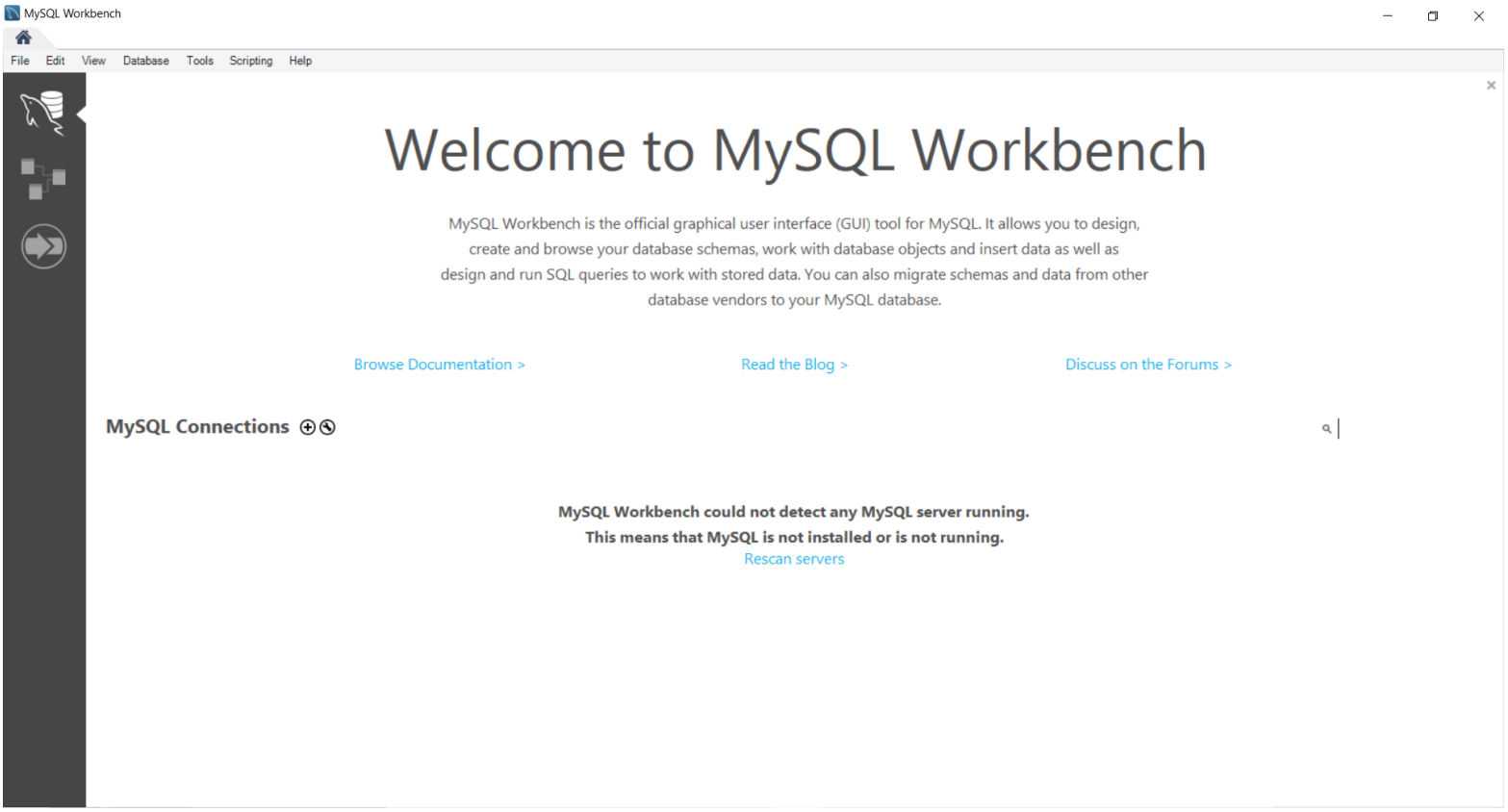
MySQL.com is using Oracle SSO for authentication. If you a
Login link. Otherwise, you can signup for a free account by
instructions.

No thanks, just start my download.

ORACLE © 2020, Oracle Corporation and/or its affiliates

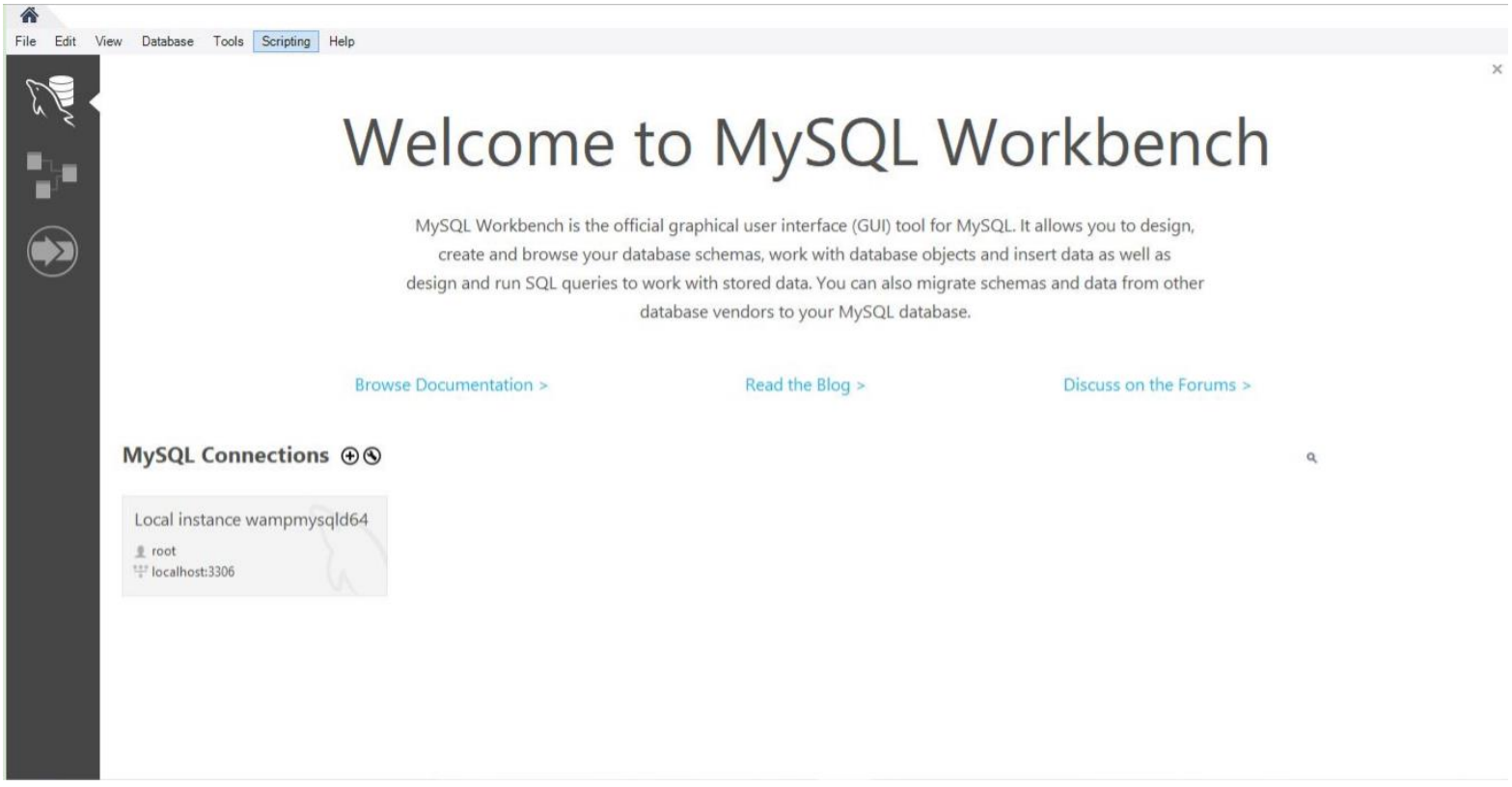
Legal Policies | Your Privacy Rights | Terms of Use | Trademark Policy | Contribu





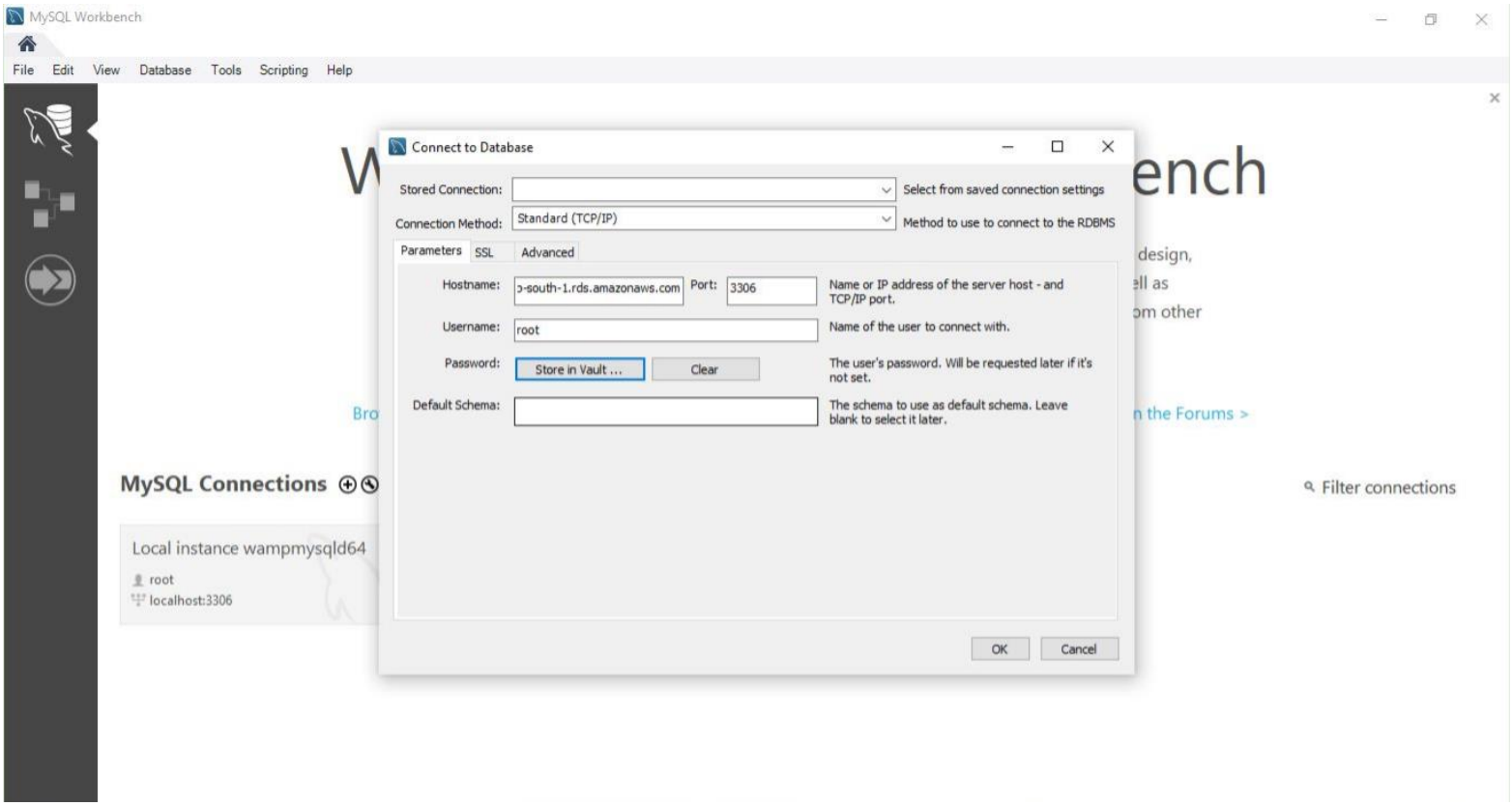
Step 3: Connect to the MySQL Database. In this step, we will connect to the database we created using MySQL Workbench.

1. Launch the MySQL Workbench application and go to Database > Connect to DB (Ctrl+U) from the menu bar.

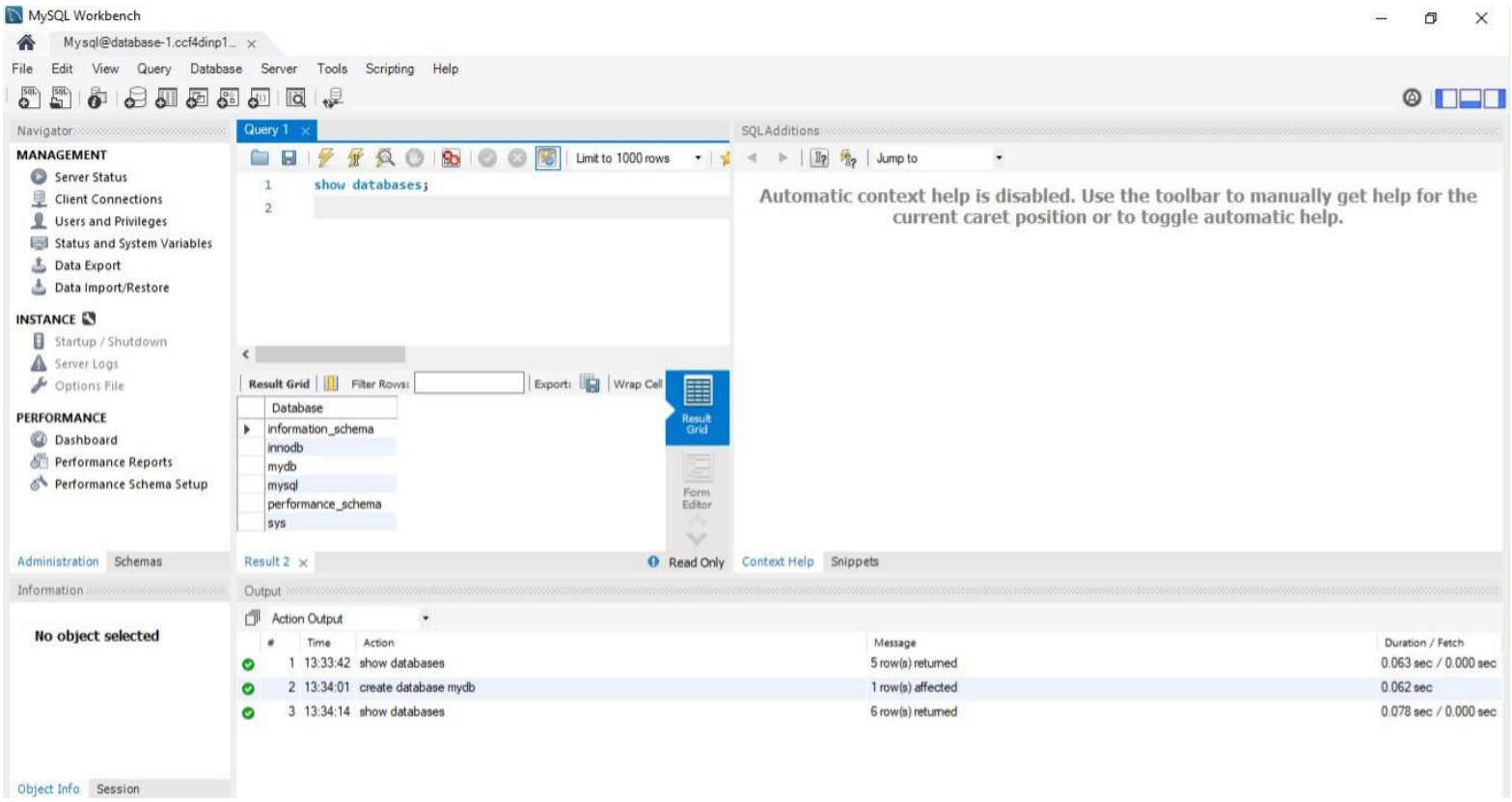


2. A dialog box appears.The following data should be entered in dialog box :

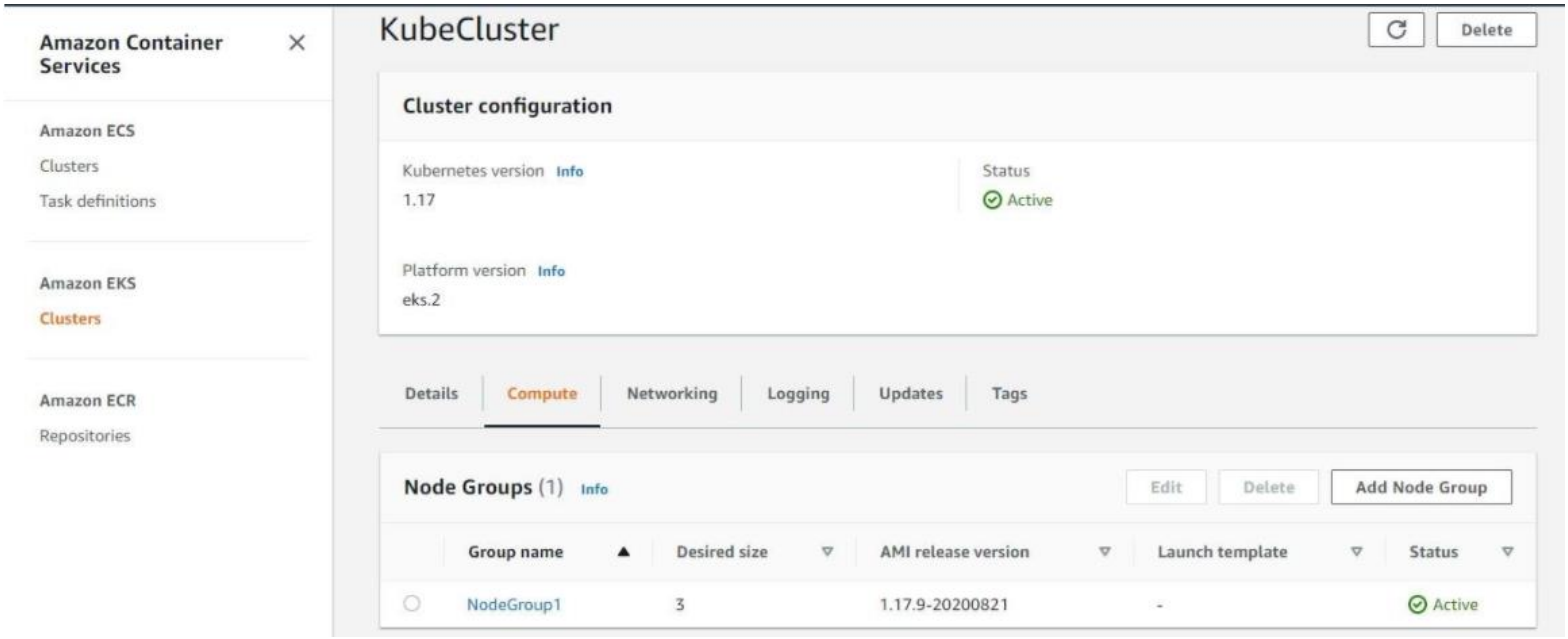
- **Hostname: You can find your hostname on the Amazon RDS**
- **Port: The default value should be 3306.**
- **Username: Type in the username you created for the Amazon RDS database. Password: Click Store in Vault (or Store in Keychain on macOS) and enter the password that you used when creating the Amazon RDS database. Click OK**
- **After entering password the following screenshot will appear.This shows your completion on MySQL workbench.**

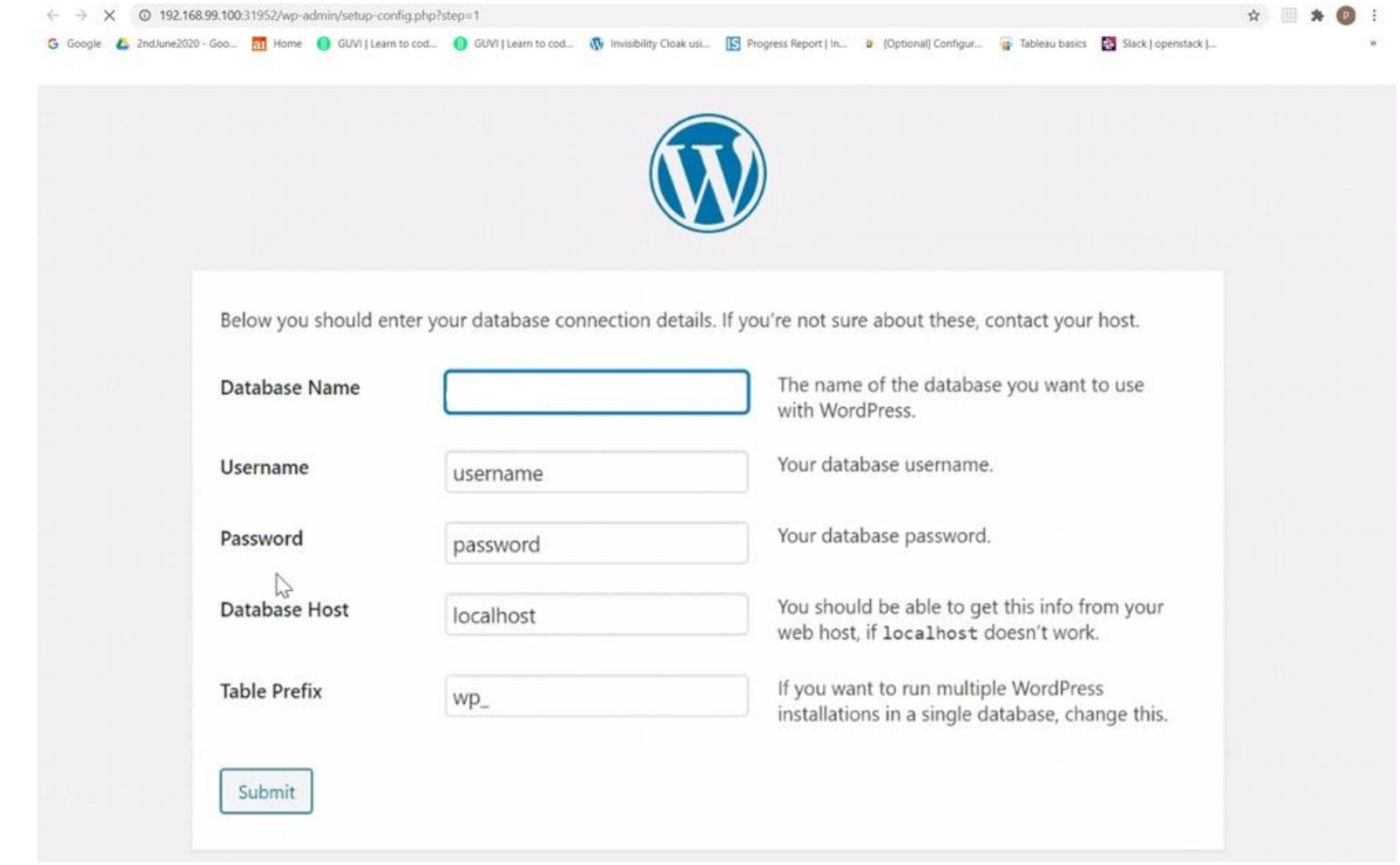


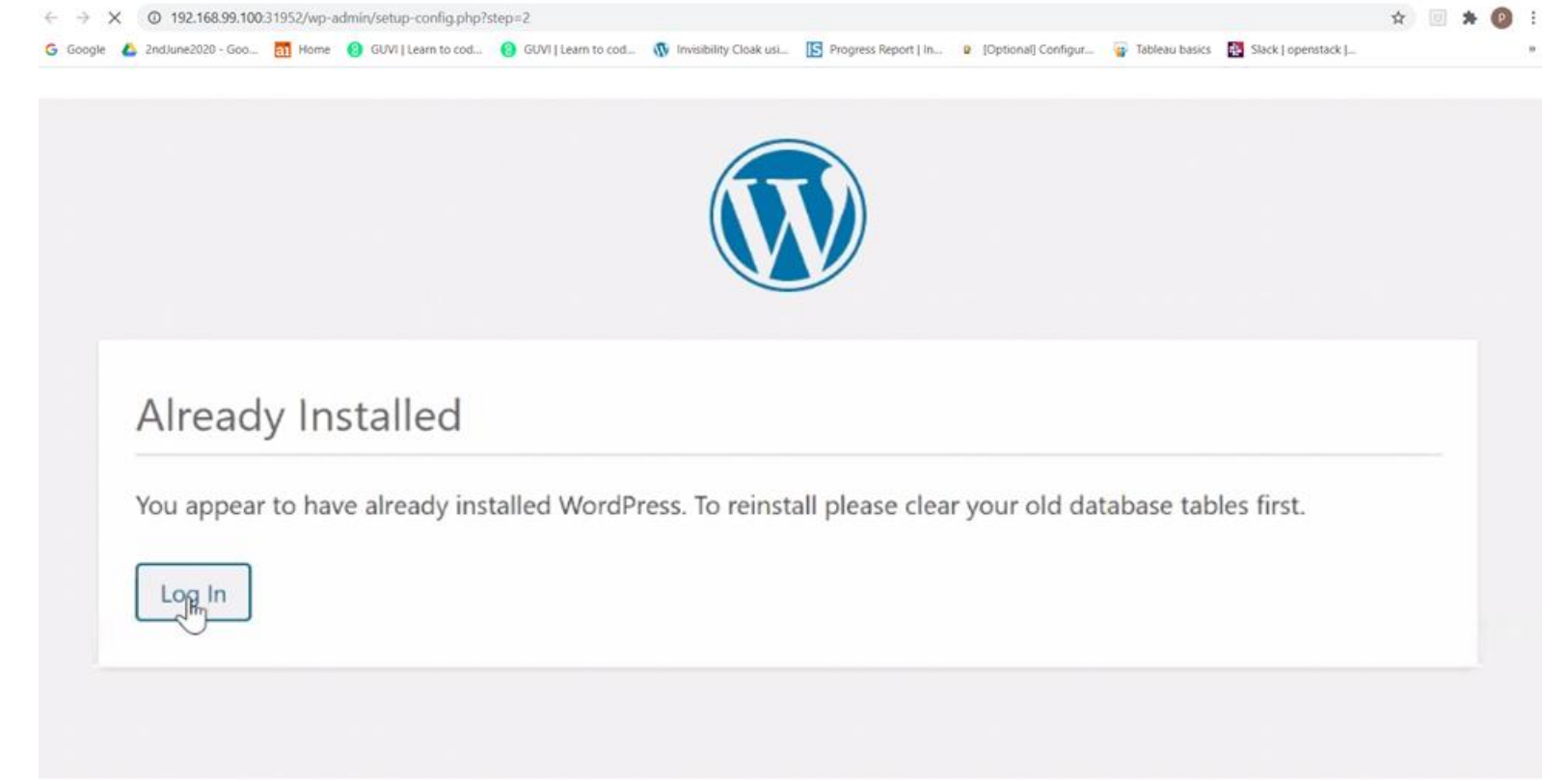
3. Now we are connected to the database! On the MySQL Workbench, you will be able start creating tables, insert data, and run queries.

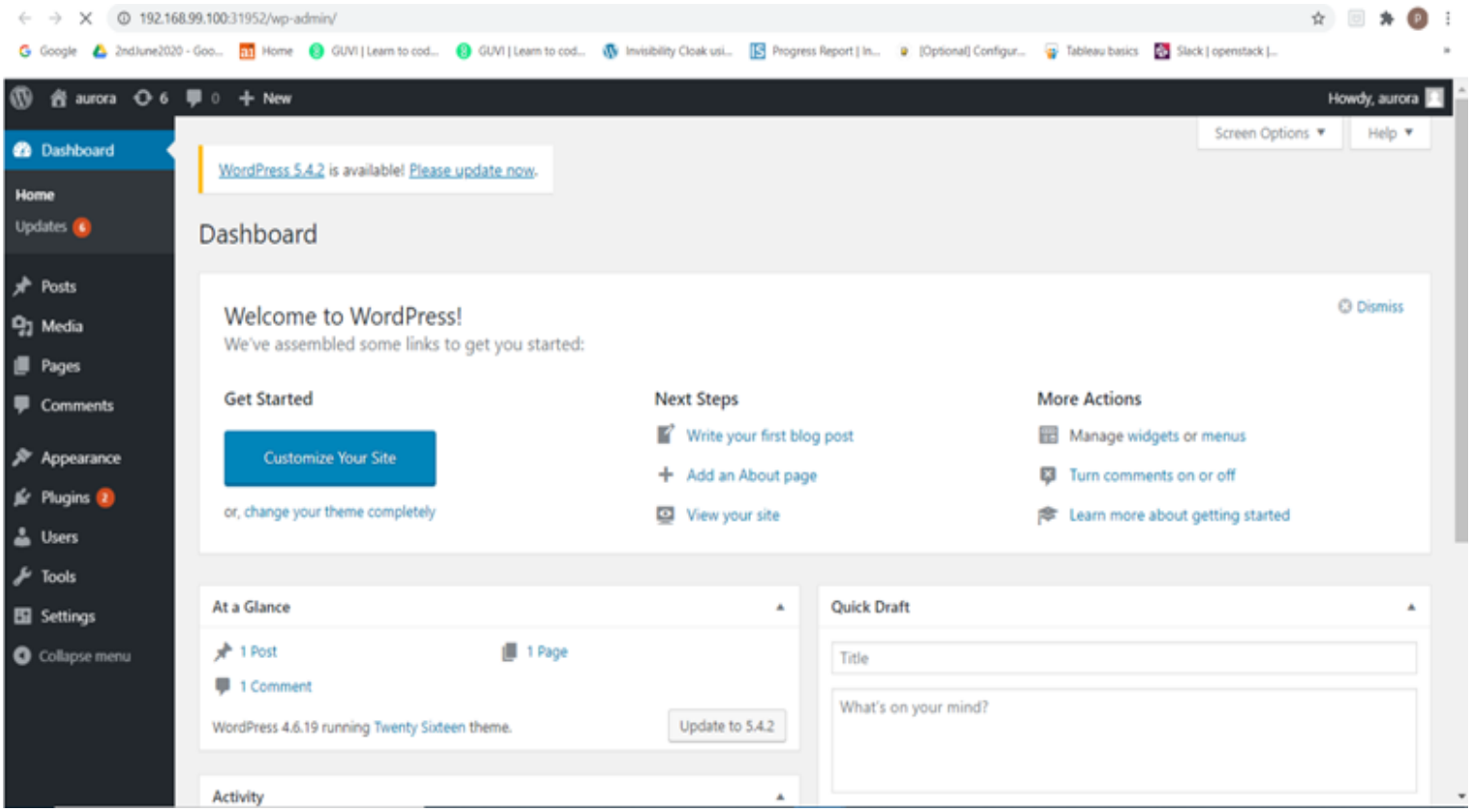


About Kube Cluster :-




192.168.99.100:31952/wp-admin/setup-config.php?step=1
Below you should enter your database connection details. If you're not sure about these, contact your host.
Database Name
The name of the database you want to use with WordPress.
Username
username
Your database username.
Password
password
Your database password.
Database Host
localhost
You should be able to get this info from your web host, if localhost doesn't work.
Table Prefix
wp_
If you want to run multiple WordPress installations in a single database, change this.
Submit


192.168.99.100:31952/wp-admin/setup-config.php?step=2
Already Installed
You appear to have already installed WordPress. To reinstall please clear your old database tables first.
Log In



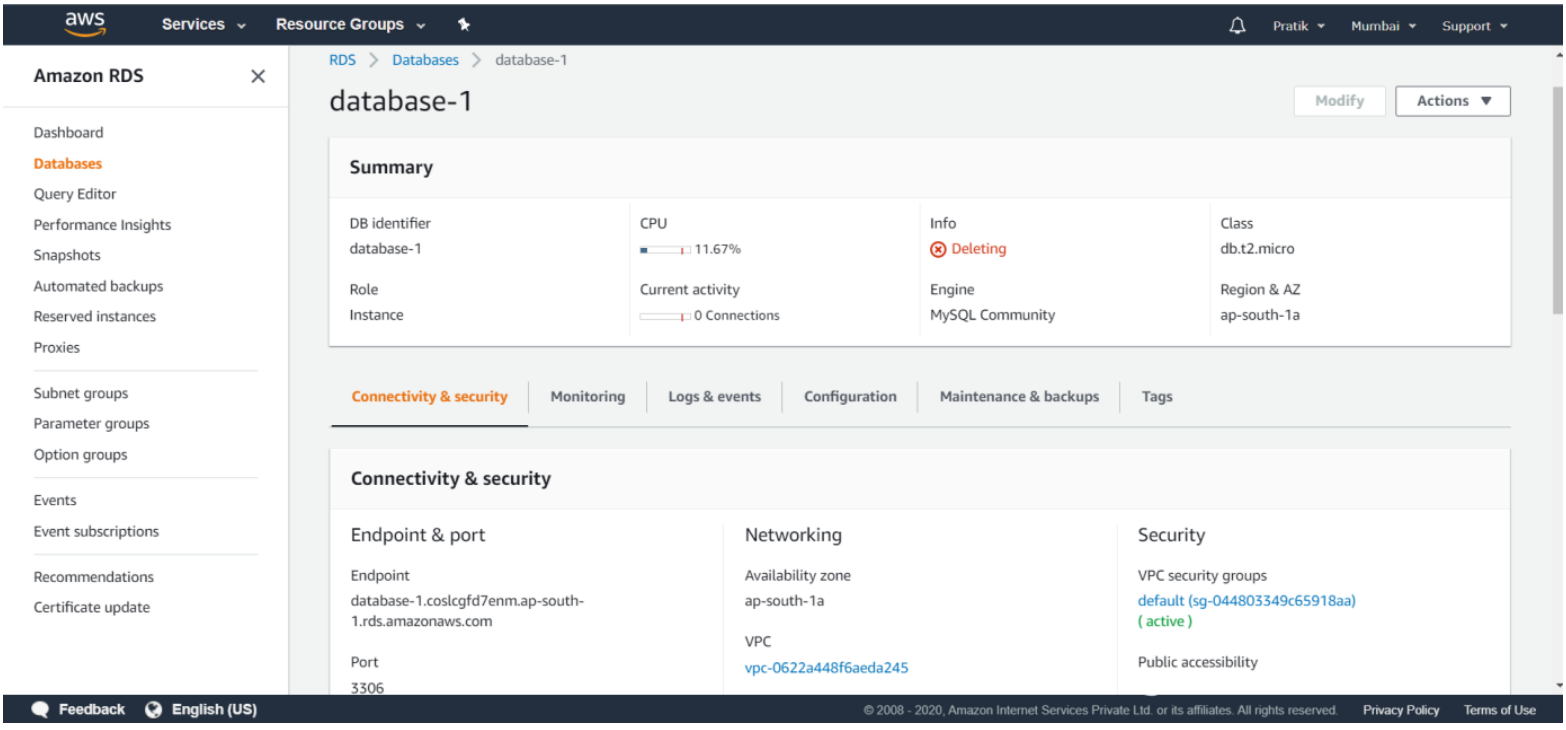
Automatically destroyed the task using terraform destroy command:-

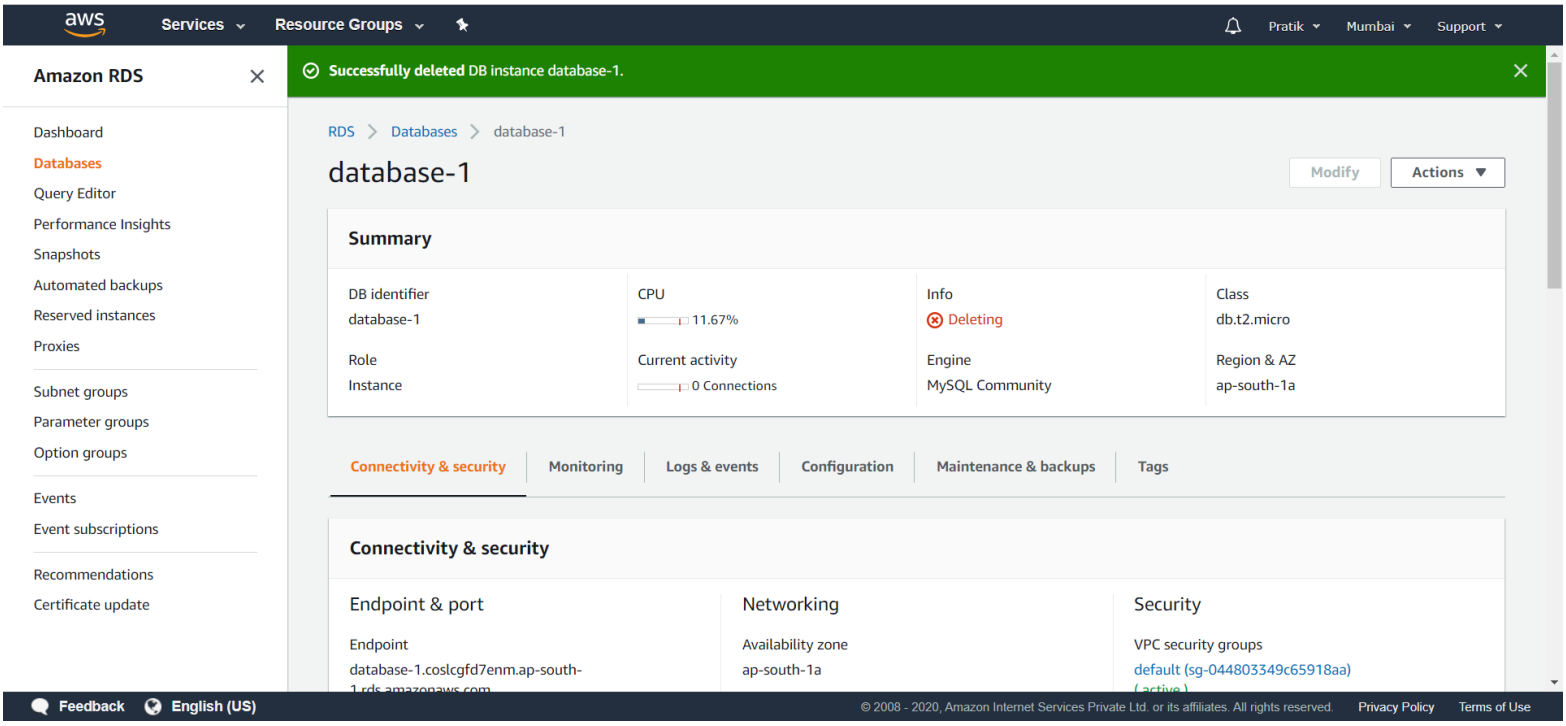
```
C:\Users\Asus\Desktop\terraformcodefiles>terraform destroy
null_resource.minikubestart: Refreshing state... [id=348086797605676458]
kubernetes_service.wordpresslb: Refreshing state... [id=default/wplb]
kubernetes_deployment.wordpress: Refreshing state... [id=default/wp]

An execution plan has been generated and is shown below.
Resource actions are indicated with the following symbols:
  - destroy

Terraform will perform the following actions:

  # kubernetes_deployment.wordpress will be destroyed
  - resource "kubernetes_deployment" "wordpress" {
      - id               = "default/wp" -> null
      - wait_for_rollout = true -> null

      - metadata {
          - annotations      = {} -> null
          - generation       = 1 -> null
          - labels           = {} -> null
          - name             = "wp" -> null
          - namespace        = "default" -> null
          - resource_version = "108322" -> null
          - self_link        = "/apis/apps/v1/namespaces/default/deployments/wp" -> null
          - uid              = "a1cc0936-858f-4c02-aa88-1996a631f1fe" -> null
        }

      - spec {
          - min_ready_seconds         = 0 -> null
          - paused                    = false -> null
          - progress_deadline_seconds = 600 -> null
          - replicas                  = 3 -> null
          - revision_history_limit    = 10 -> null

          - selector {
              - match_labels = {
                  - "App"    = "wordpress"
                  - "env"    = "production"
                  - "region" = "IN"
                } -> null

              - match_expressions {
```

Command Prompt

```
        - namespace        = "default" -> null
        - resource_version = "108267" -> null
        - self_link        = "/api/v1/namespaces/default/services/wplb" -> null
        - uid              = "6f2ea1ca-aa67-4a45-aa92-6a78fcd6b618" -> null
      }

    - spec {
        - cluster_ip                  = "10.105.77.61" -> null
        - external_ips                = [] -> null
        - external_traffic_policy     = "Cluster" -> null
        - load_balancer_source_ranges = [] -> null
        - publish_not_ready_addresses = false -> null
        - selector                    = {
            - "app" = "wordpress"
          } -> null
        - session_affinity            = "None" -> null
        - type                        = "NodePort" -> null

        - port {
            - node_port   = 30091 -> null
            - port        = 80 -> null
            - protocol    = "TCP" -> null
            - target_port = "80" -> null
          }
      }
  }

  # null_resource.minikubestart will be destroyed
  - resource "null_resource" "minikubestart" {
      - id = "348086797605676458" -> null
    }

Plan: 0 to add, 0 to change, 3 to destroy.

Do you really want to destroy all resources?
  Terraform will destroy all your managed infrastructure, as shown above.
  There is no undo. Only 'yes' will be accepted to confirm.

  Enter a value: yes

null_resource.minikubestart: Destroying... [id=348086797605676458]
null_resource.minikubestart: Destruction complete after 0s
kubernetes_service.wordpresslb: Destroying... [id=default/wplb]
kubernetes_service.wordpresslb: Destruction complete after 0s
kubernetes_deployment.wordpress: Destroying... [id=default/wp]
kubernetes_deployment.wordpress: Destruction complete after 0s

Destroy complete! Resources: 3 destroyed.
```

Manually destroying Amazon RDS:-





Completed Task-6 Successfully

Thanks for Reading!!!