Experiment 4: Install Google App Engine. Create a hello world app and other simple web applications using python/java.

Aim:

To Install Google App Engine. Create *hello world* app and other simple web applications using python/java.

Procedure:

1. Install Google Plugin for Eclipse

Read this guide – how to install Google Plugin for Eclipse. If you install the Google App Engine Java SDK together with "**Google Plugin for Eclipse**", then go to step 2, Otherwise, get the Google App Engine Java SDK and extract it.

2. Create New Web Application Project

In Eclipse toolbar, click on the Google icon, and select "**New Web Application Project…**"

Figure – New Web Application Project

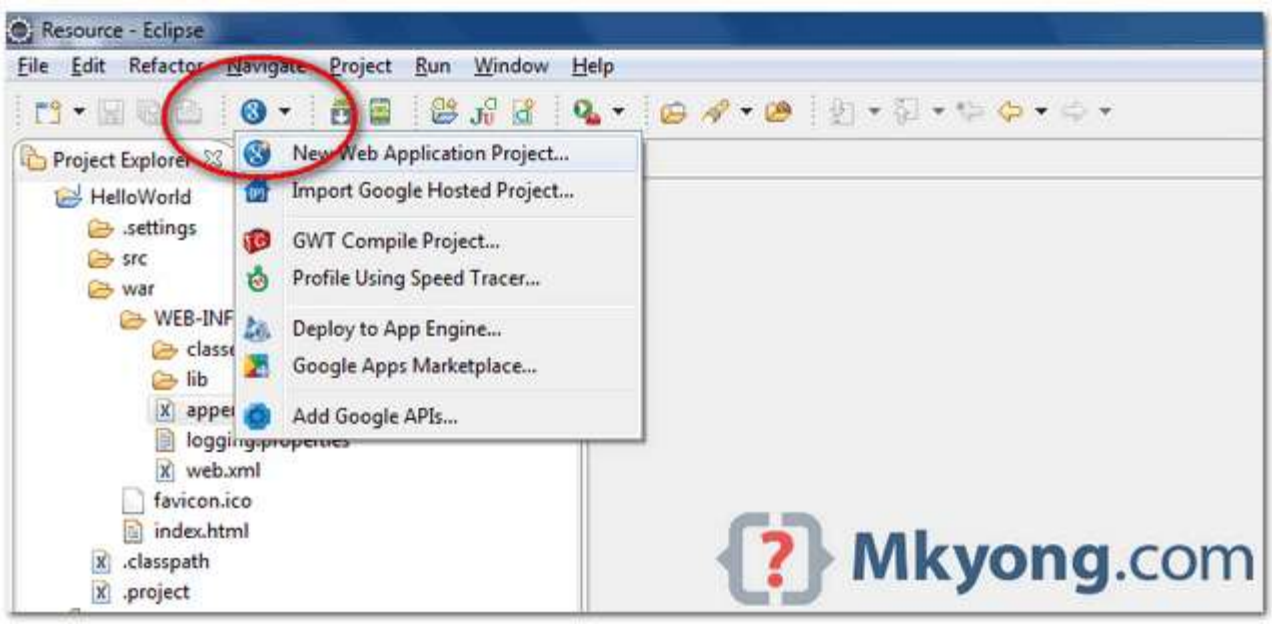
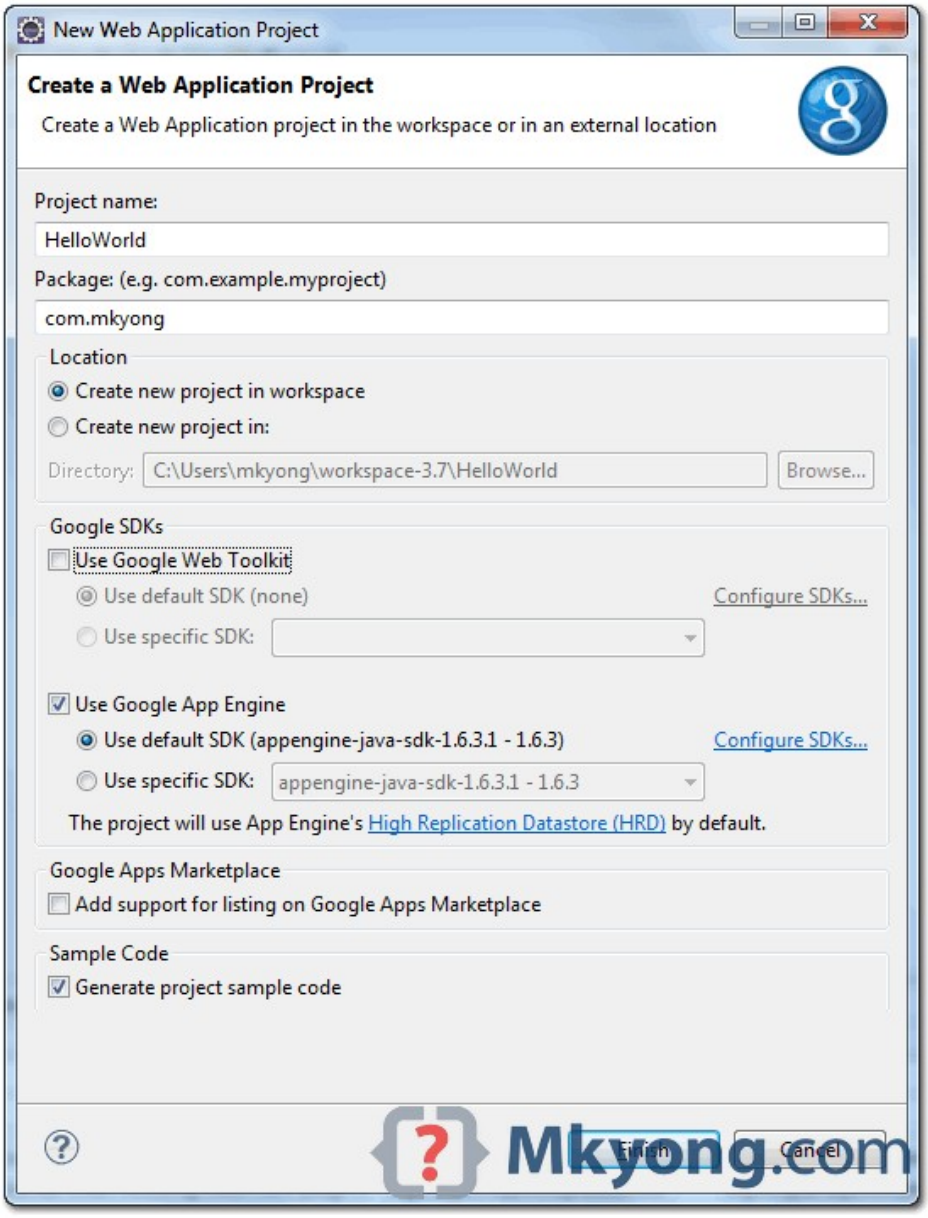


*Figure – Deselect the "**Google Web ToolKit**", and link your GAE Java SDK via the "**configure SDK**" link*



Click finished, Google Plugin for Eclipse will generate a sample project automatically.

1. Hello World

Review the generated project directory.



Nothing special, a standard Java web project structure.

HelloWorld/

src/

...Java source code...

META-INF/

...other configuration...

war/

...JSPs, images, data files...

WEB-INF/

```
...app configuration...
 lib/
...JARs for libraries...
classes/
...compiled classes...
```

Copy

The extra is this file "appengine-web.xml", Google App Engine need this to run and deploy the application.

File : appengine-web.xml

```
<?xml version="1.0" encoding="utf-8"?>
<appengine-web-app xmlns="http://appengine.google.com/ns/1.0">
<application></application>
<version>1</version>

<!-- Configure java.util.logging -->
<system-properties>
<property name="java.util.logging.config.file" value="WEB-INF/logging.properties"/>
</system-properties>

</appengine-web-app>
```

Copy

4. Run it local

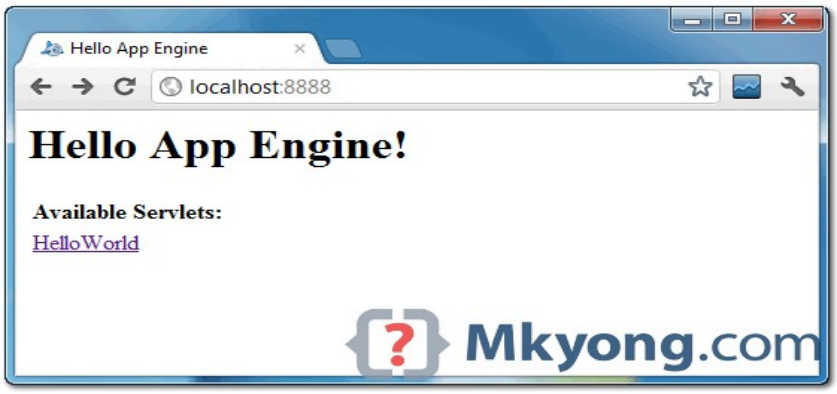
Right click on the project and run as "**Web Application**".

Eclipse console :

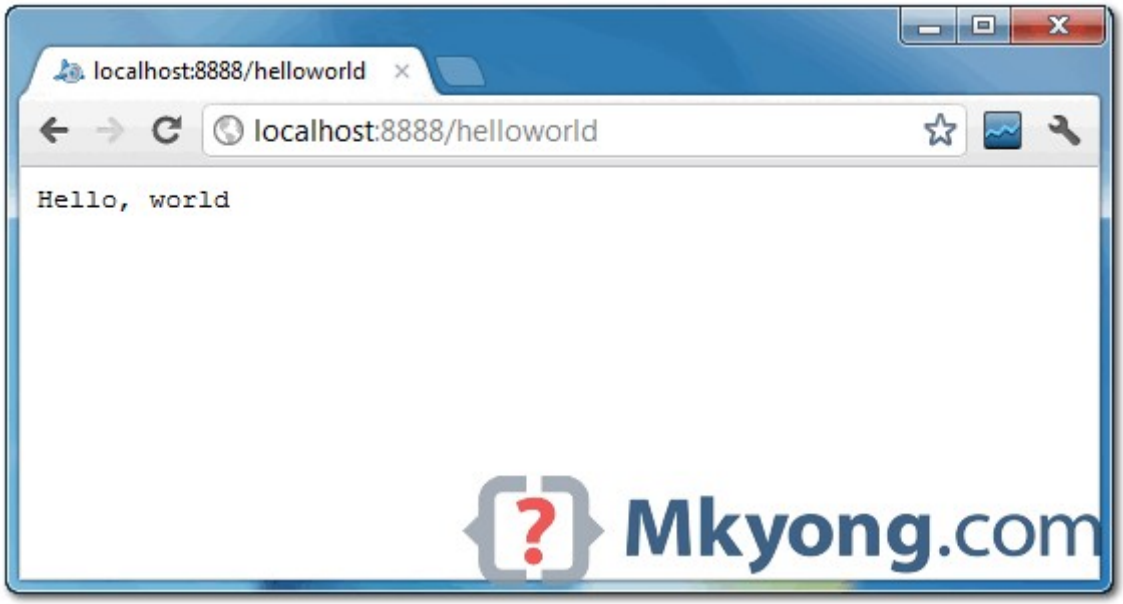
```
//...
INFO: The server is running at http://localhost:8888/
30 Mac 2012 11:13:01 PM
com.google.appengine.tools.development.DevAppServerImpl start INFO: The admin
console is running at http://localhost:8888/_ah/admin
```

Copy

Access URL http://localhost:8888/, see output



and also the hello world servlet – http://localhost:8888/helloworld



5. Deploy to Google App Engine

Register an account on https://appengine.google.com/, and create an application ID for your web application.

In this demonstration, I created an application ID, named "mkyong123", and put it in appengine web.xml.

File : appengine-web.xml

```
<?xml version="1.0" encoding="utf-8"?>
<appengine-web-app xmlns="http://appengine.google.com/ns/1.0">
  <application>mkyong123</application>
  <version>1</version>
```

```
  <!-- Configure java.util.logging -->
  <system-properties>
    <property name="java.util.logging.config.file" value="WEB-INF/logging.properties"/>
  </system-properties>

</appengine-web-app>
```

Copy

To deploy, see following steps:

Figure 1.1 – Click on GAE deploy button on the toolbar.

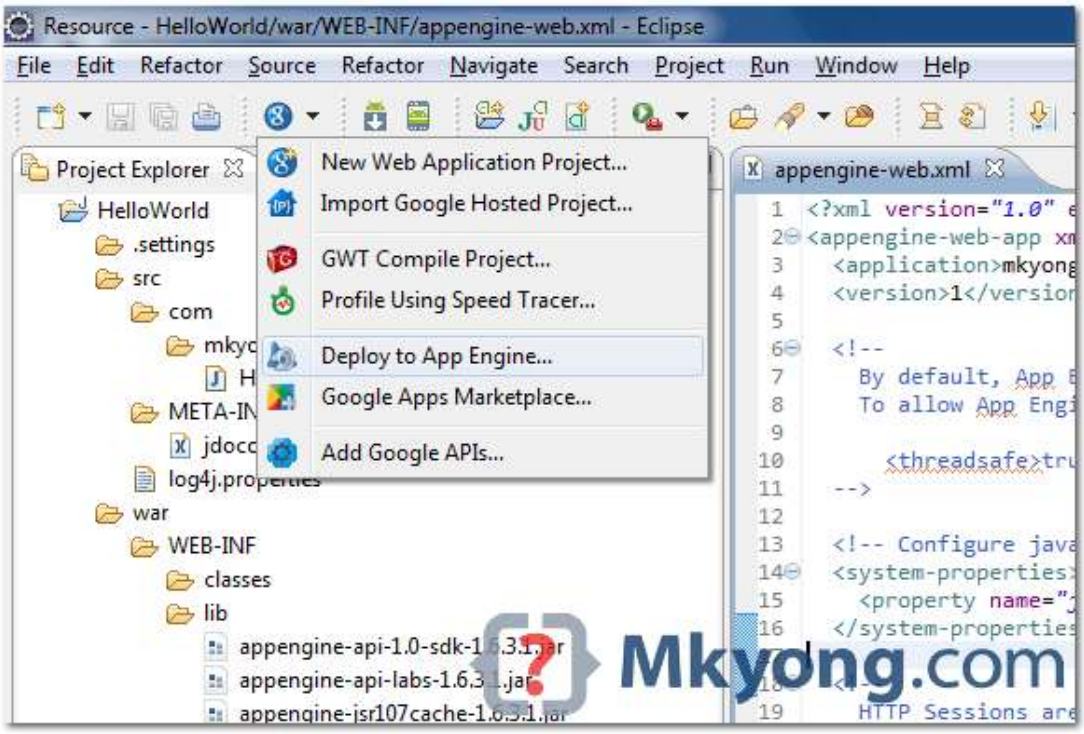


Figure 1.2 – Sign in with your Google account and click on the Deploy button.

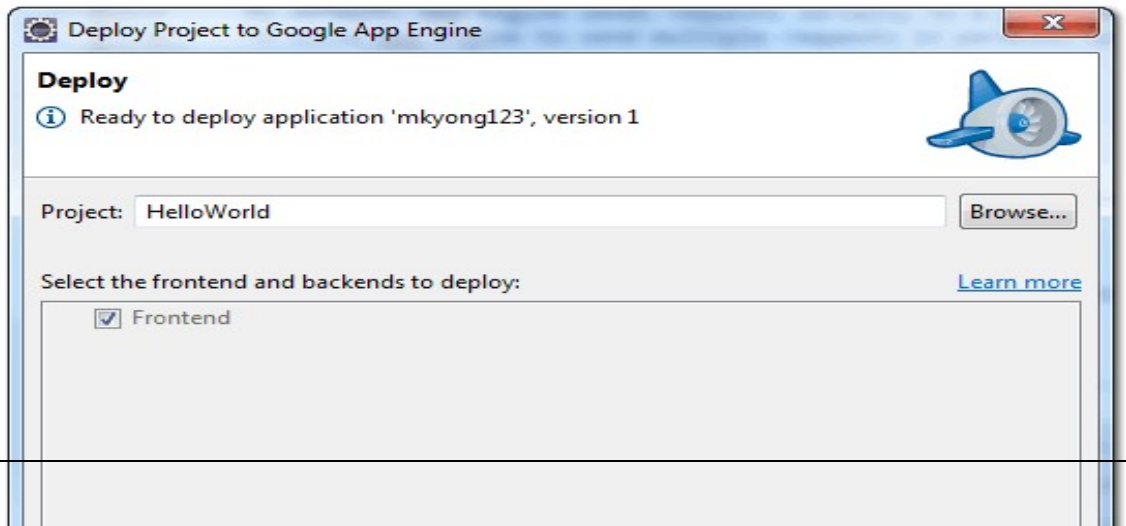


Figure 1.3 – If everything is fine, the hello world web application will be deployed to this URL – http://mkyong123.appspot.com/



Result:

Thus the simple application was created successfully.

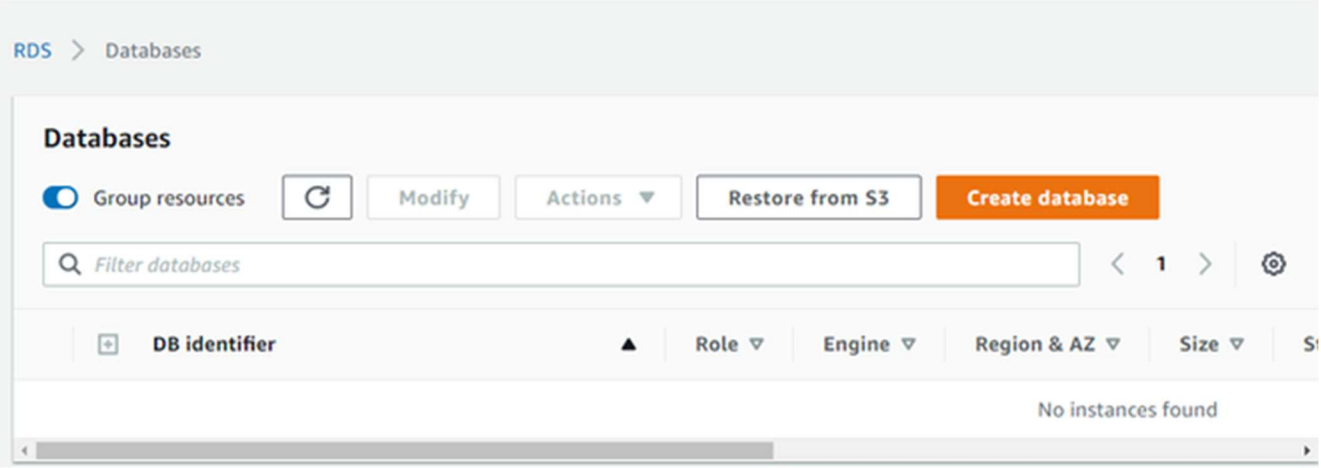
Experiment 9: Create a database instance in the cloud using Amazon RDS.

Steps to Create a Amazon RDS DB instance

Follow these steps to create an RDS DB instance using the AWS Management Console.

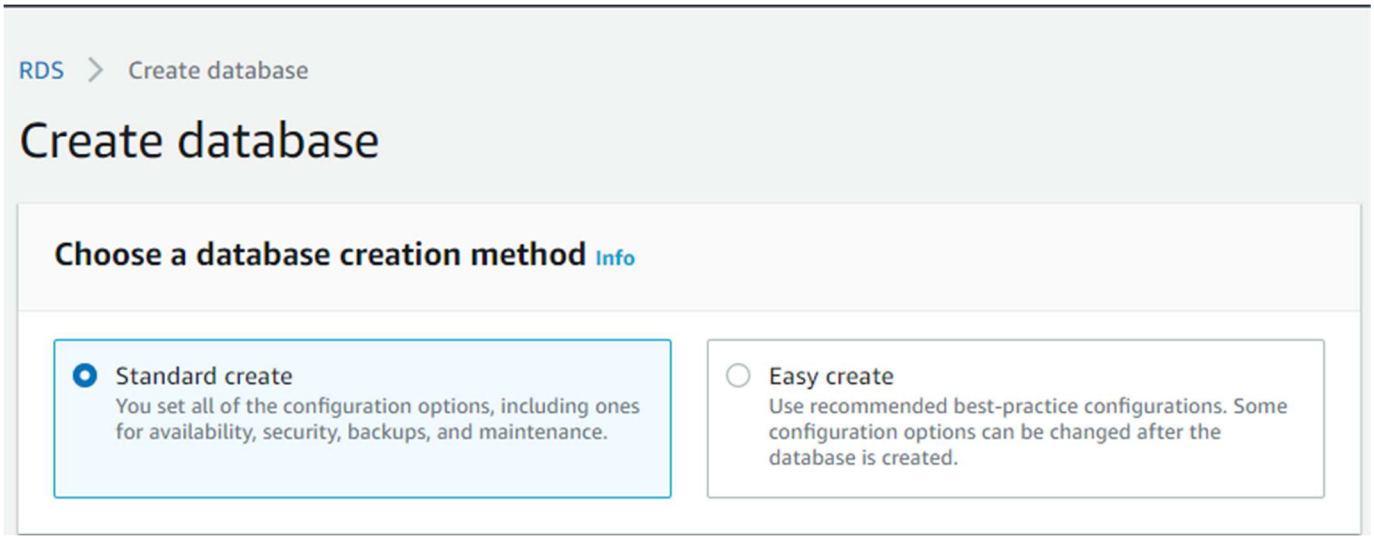
Step 1: Access the RDS Dashboard

- Log in to your **AWS Management Console**.
- Navigate to the **Amazon RDS** dashboard.
- Click on **Create Database** to start the setup process.



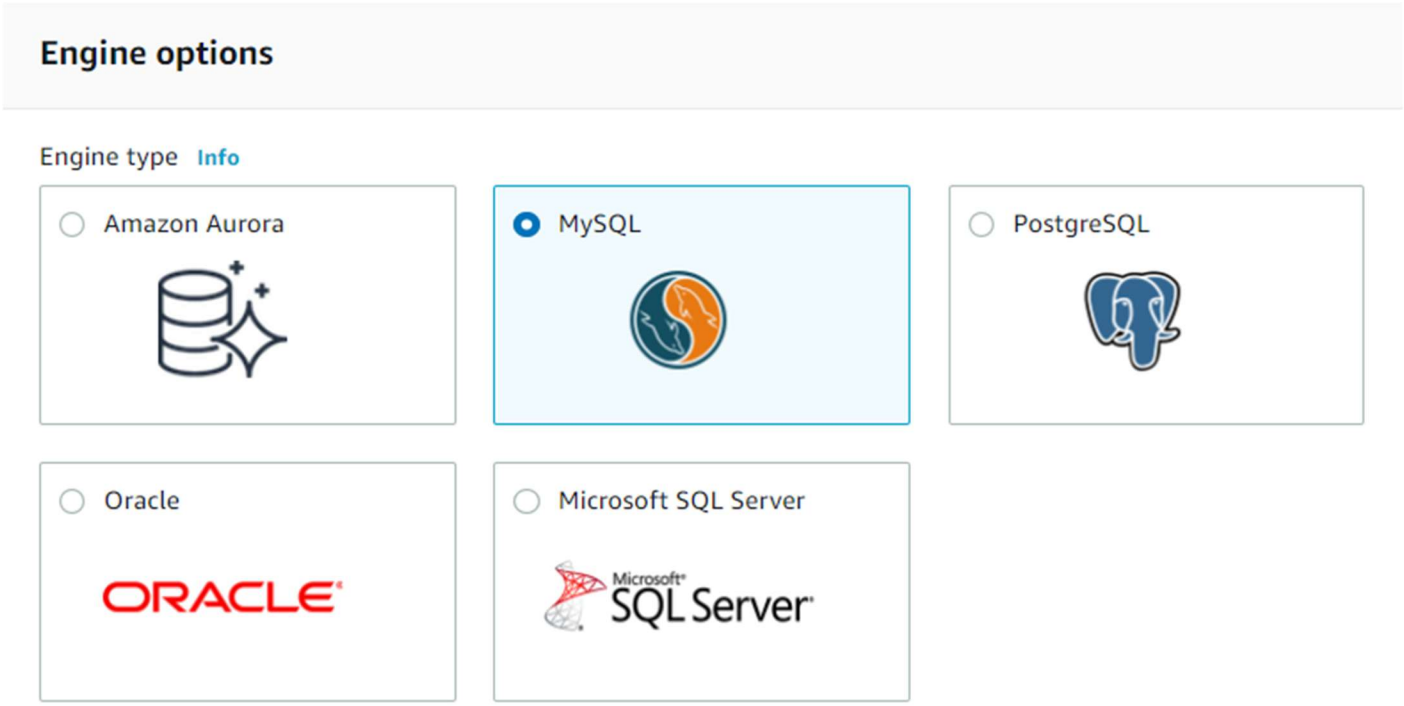
Step 2: Select a Database Engine

- Choose the **Standard database creation** method.

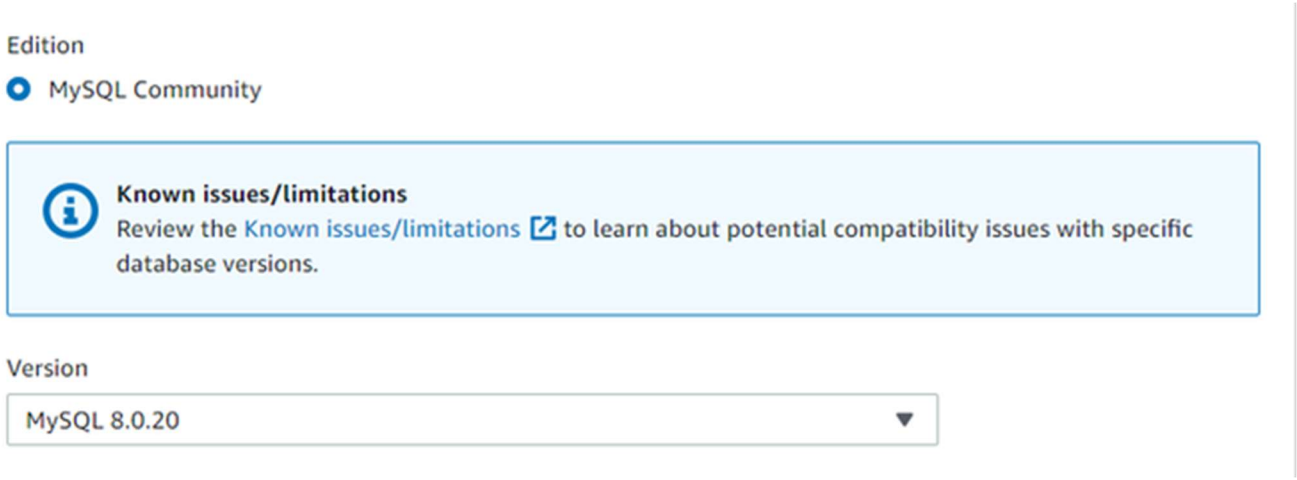


Creation Method

- Select the **MySQL** database engine (or any other engine based on your requirements).



- Select the version of the MySQL that you want to use.



Version of DB Engine

Step 3: Choose a Deployment Option

Under the **Templates** section, AWS offers three deployment options:

- **Production** – Suitable for high-performance applications.
- **Dev/Test** – Ideal for development and testing environments.
- **Free Tier** – Best for beginners and small-scale projects.

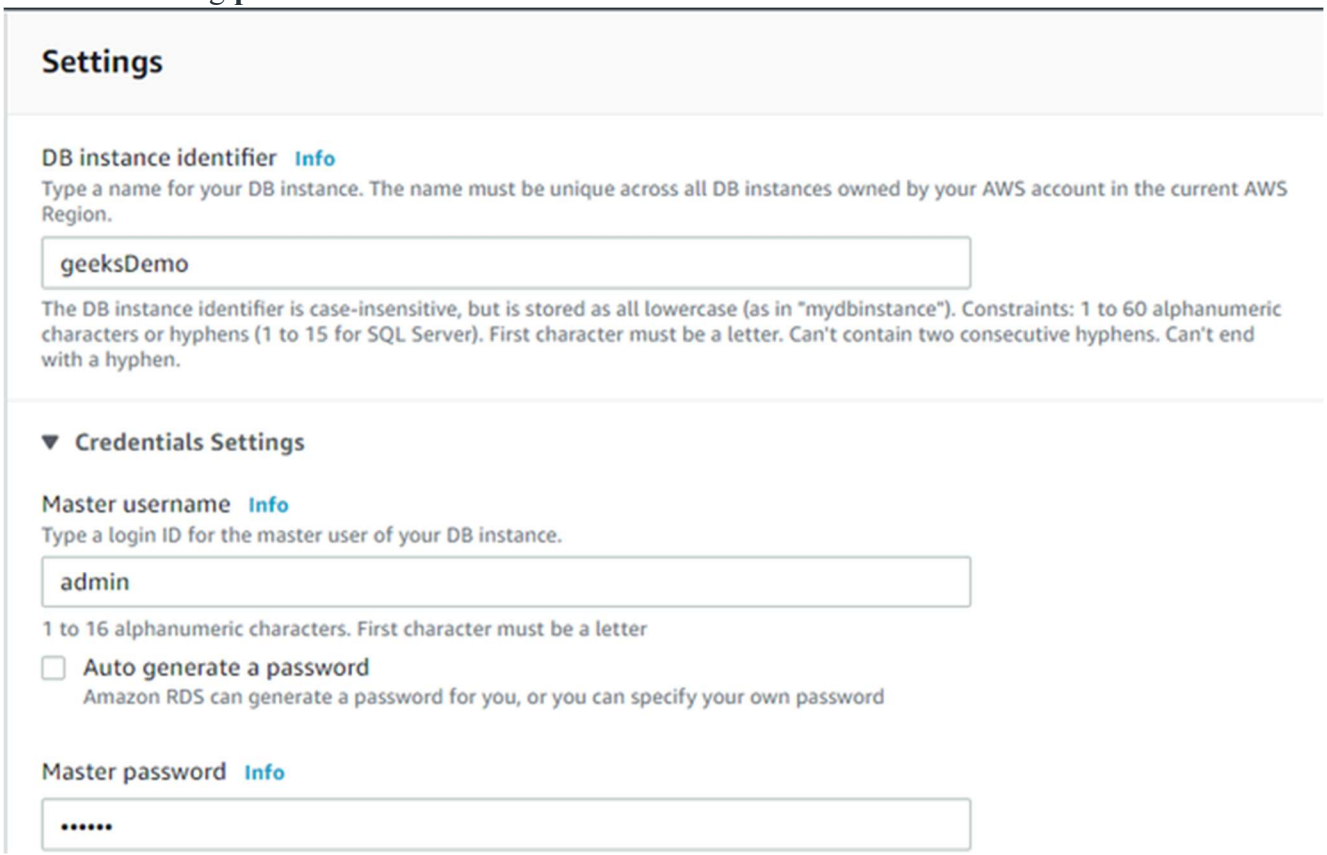


Template Types

We will opt for **Free tier.**

Step 4: Configure Database Settings

- Set the **DB instance identifier** (e.g., geeksDemo).
- Enter a **Master username** (e.g., admin).
- Choose a strong **password** for database access**.**

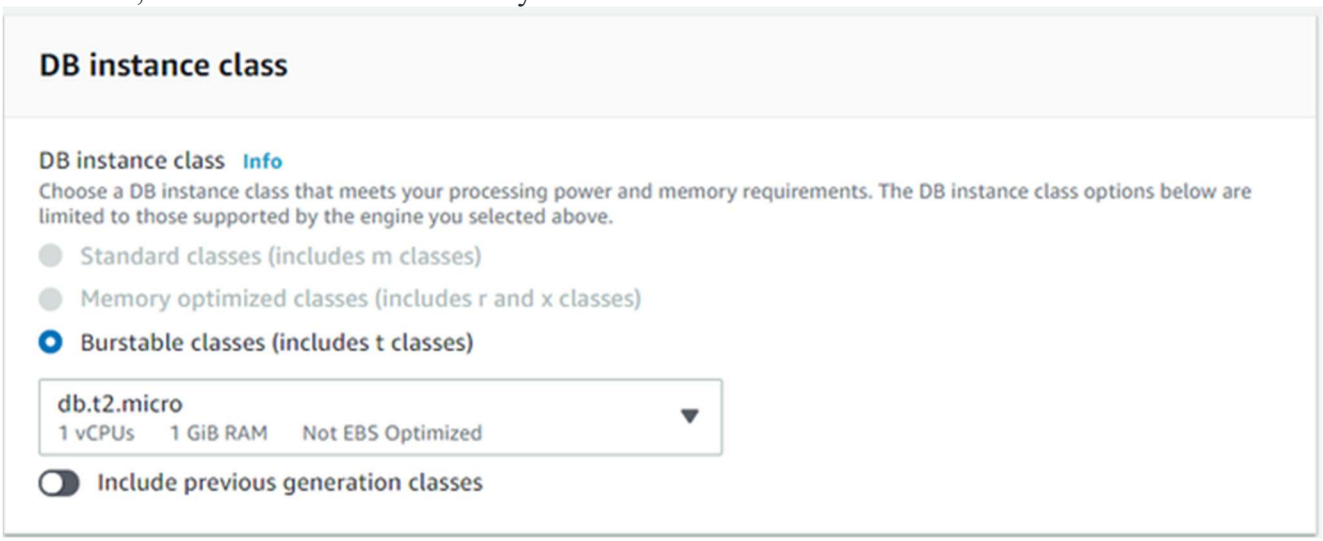


Settings

Step 5: Select an Instance Type

- The **db.t2.micro** instance type is automatically selected under the Free Tier.

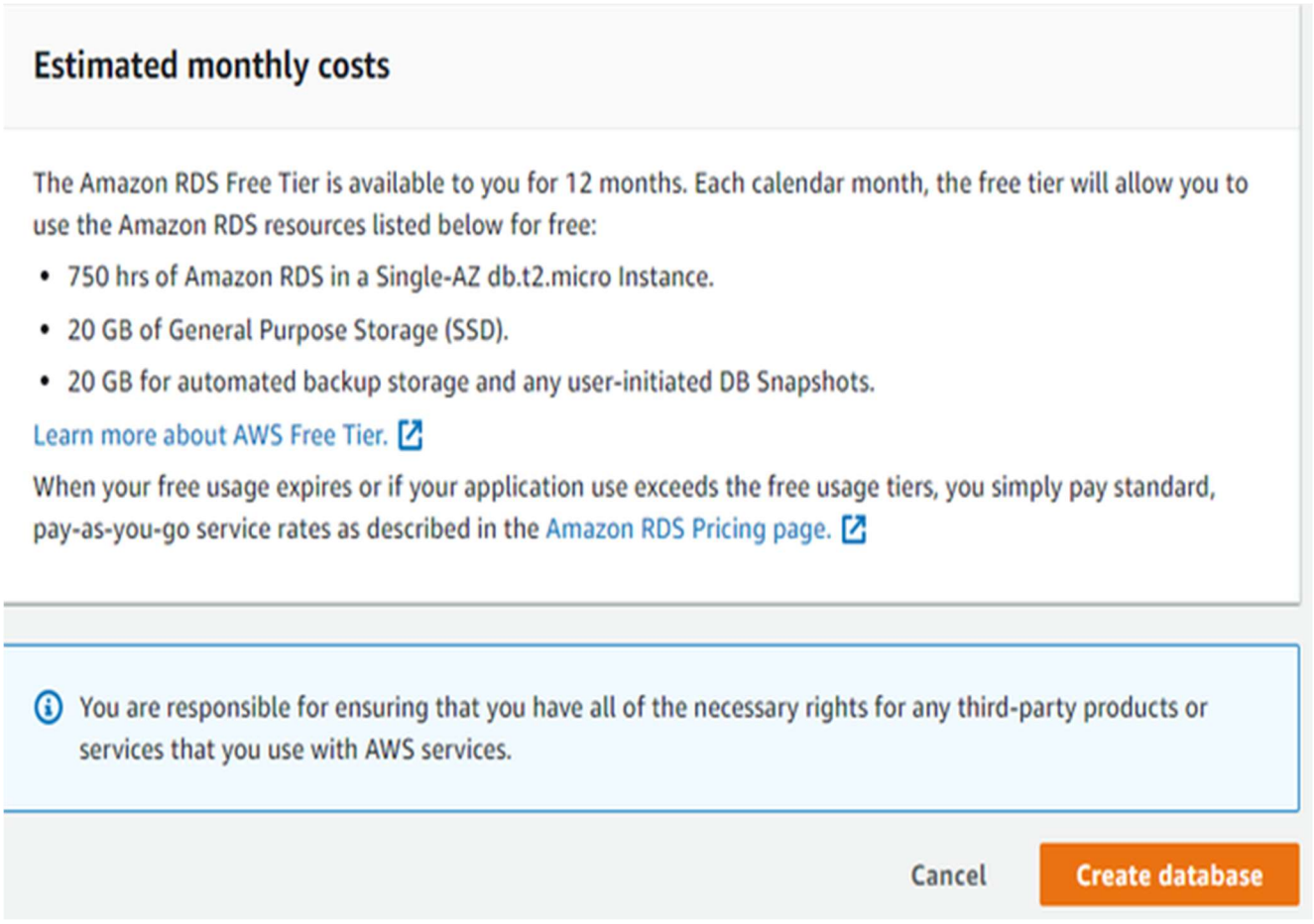
- If needed, choose an instance that fits your workload.

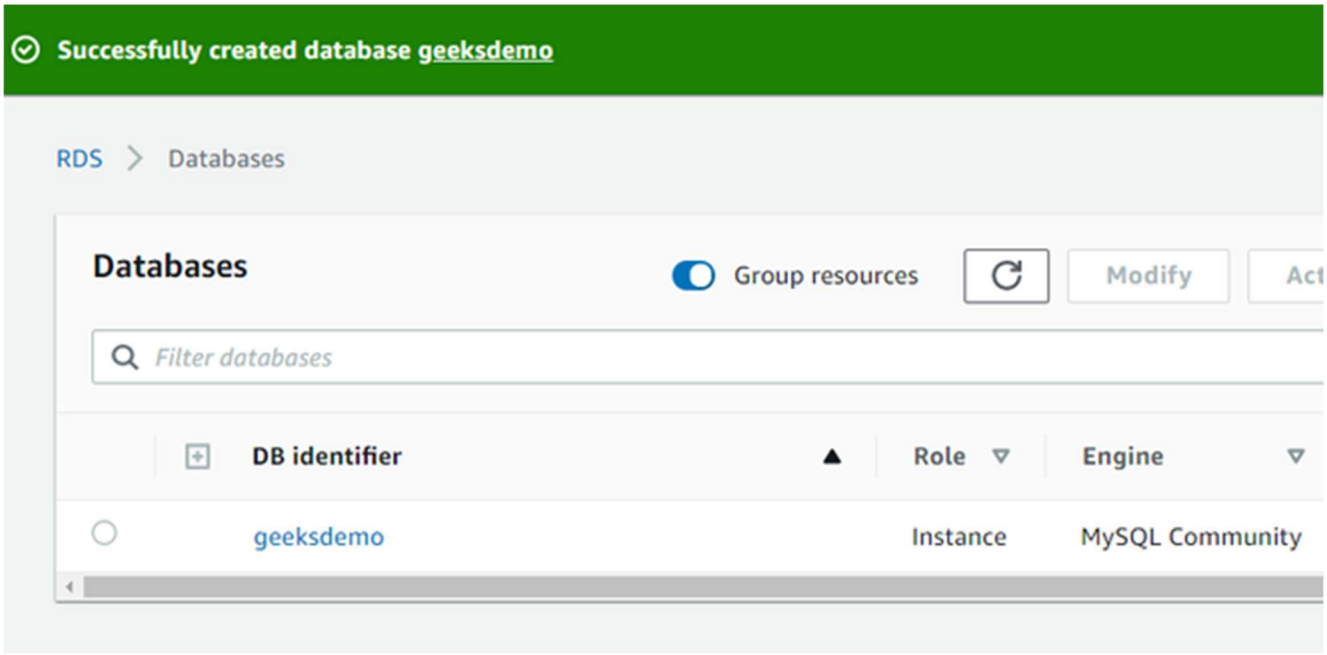


DB Instance

Step 6: Finalize and Create the Database

- Click **Create Database** to launch the instance**.**




Successfully created database geeksdemo
RDS
Databases
Databases
Group resources
Modify
Filter databases
DB identifier
Role
Engine
geeksdemo
Instance
MySQL Community