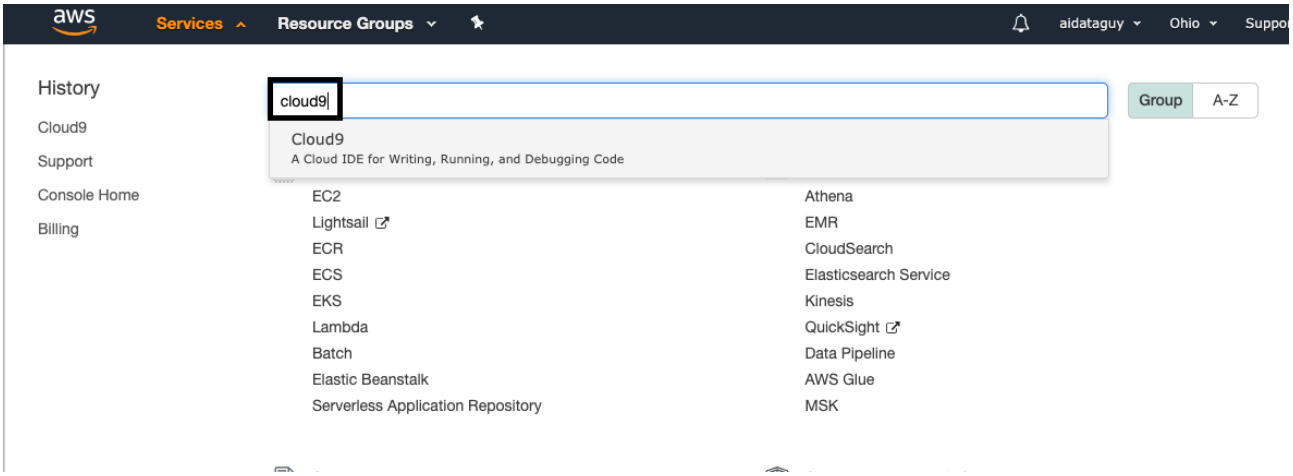
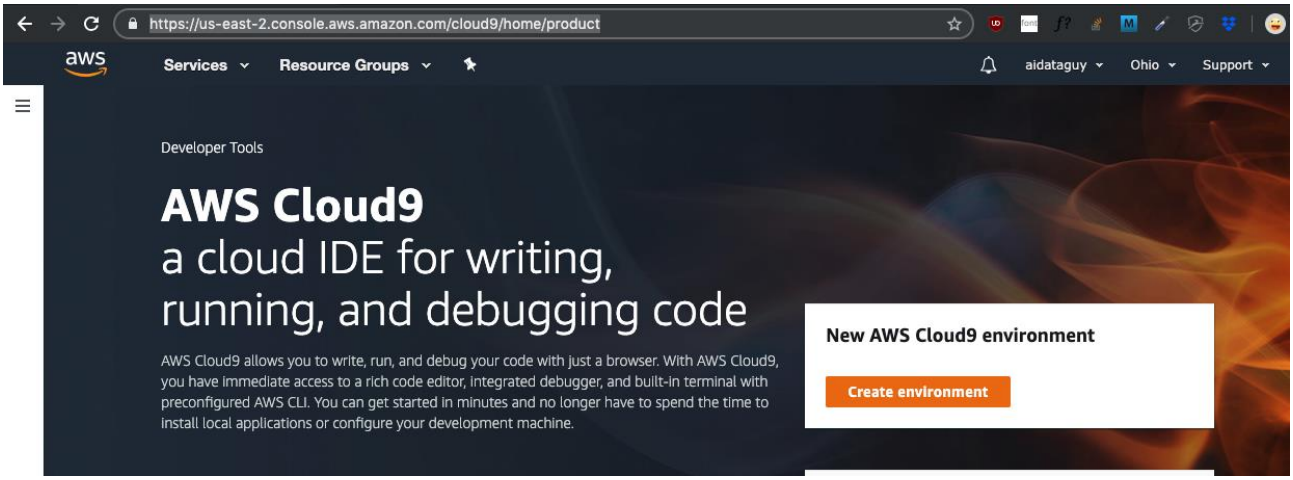
AWS Cloud 9 MySQL Setup Guide

In order to setup MySQL cloud 9, we need to perform following steps:

1. Signin to AWS console at https://console.aws.com

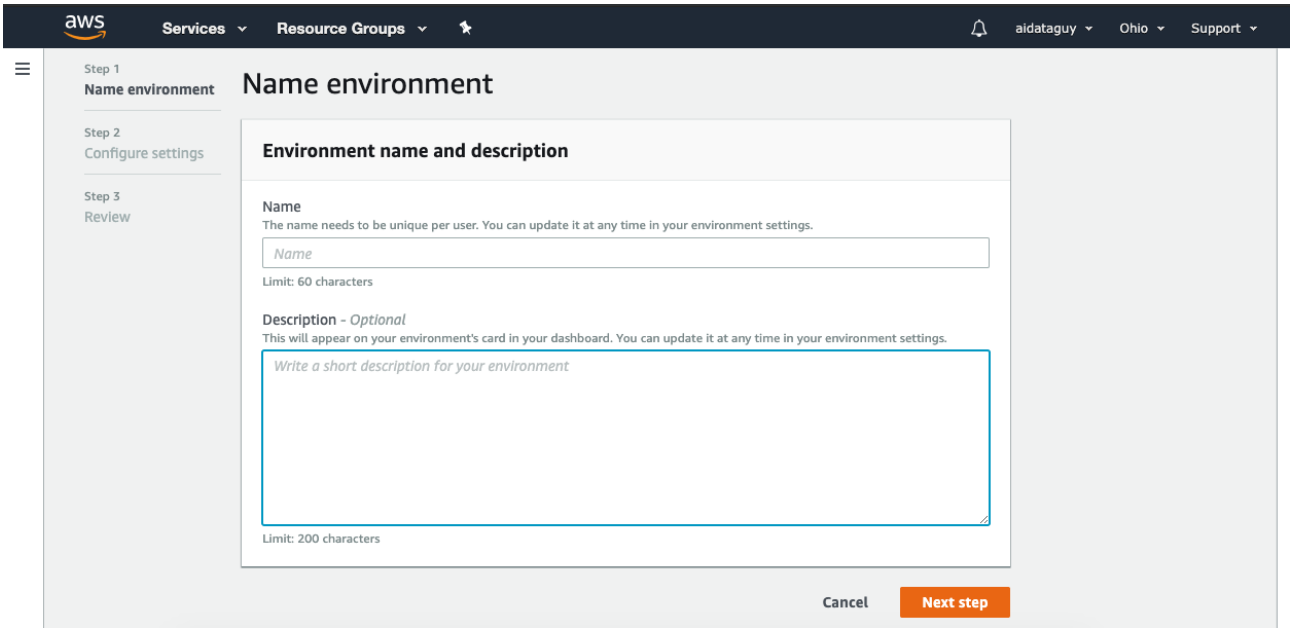


2. Once logged in search for cloud9 under Services and select cloud9 service:
3. Once you click on that you will be taken to a window something like this:



4. Click on the button you see below:

5. Once you click on it you will be redirected to next window to setup the environment. Creating IAM account user is the best practice but we're going to skip that part. To learn more about IAM roles, you can go here. Next, you would be required to fill the following form. Fill it to your heart's content and click "Next"



7. Now, we need to configure environment on next screen :

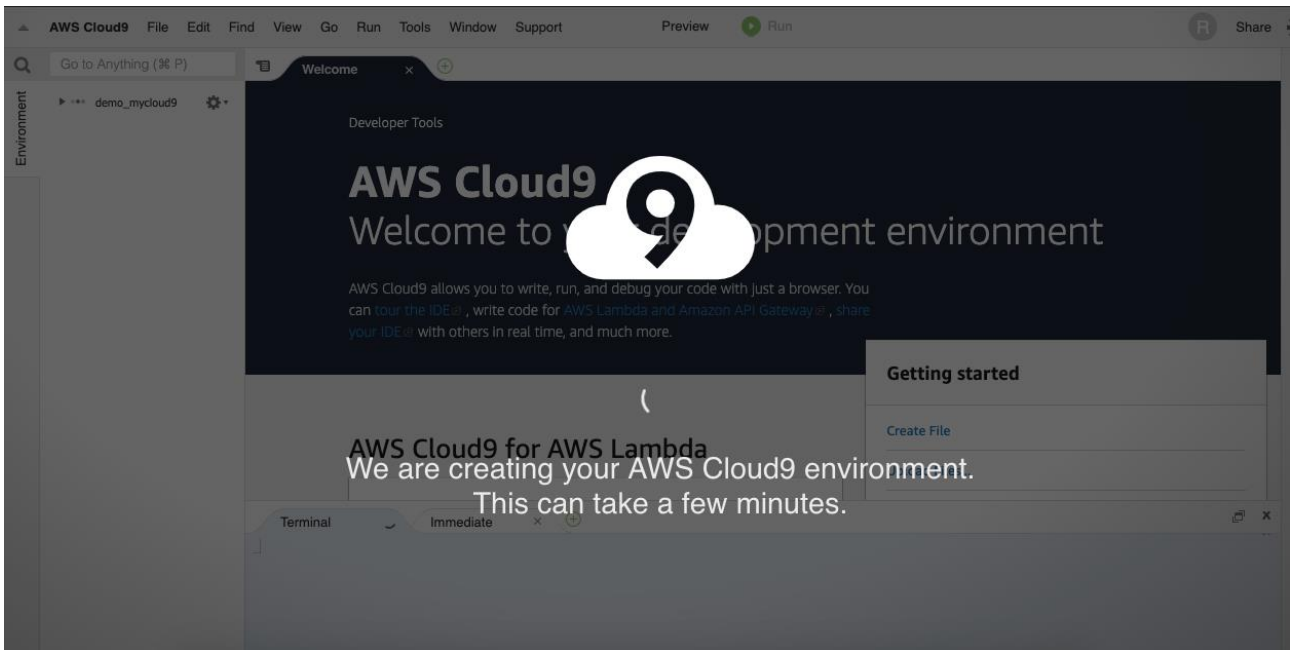
a. Environment type:
Select : Create a new instance for environment (EC2)

b. Instance Type:
Select: t2.micro (1 GiB RAM + 1 vCPU) (it's free tier eligible)

c. Platform:
Select: Ubuntu Server 18.04 LTS

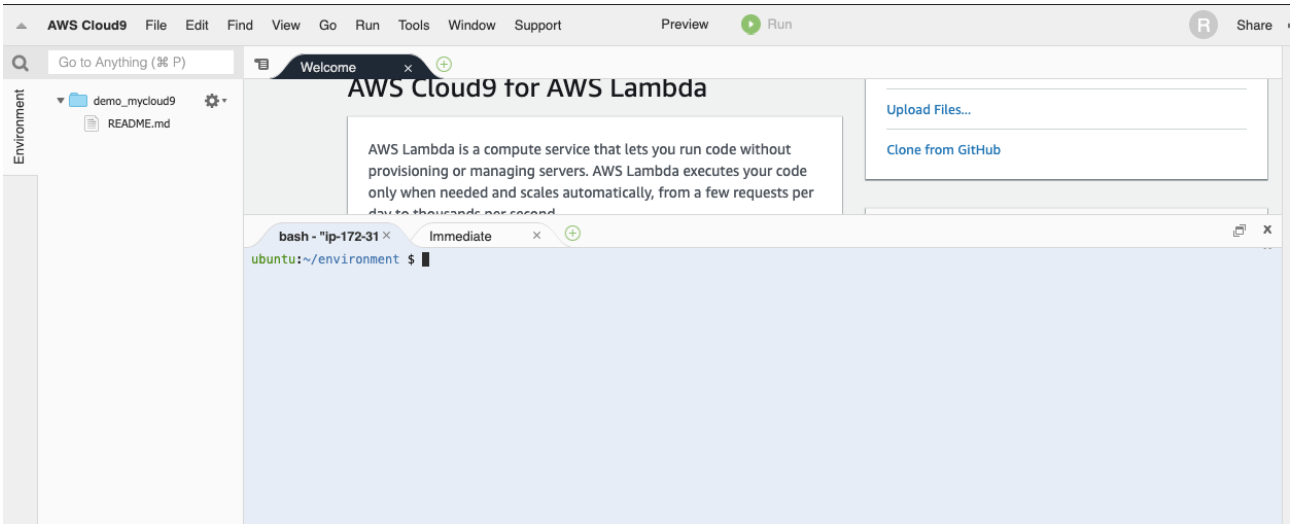
D: Cost-saving setting:
Select: After 30 minutes (default) - This setting is enough.

For other settings leave them default. Then click Next



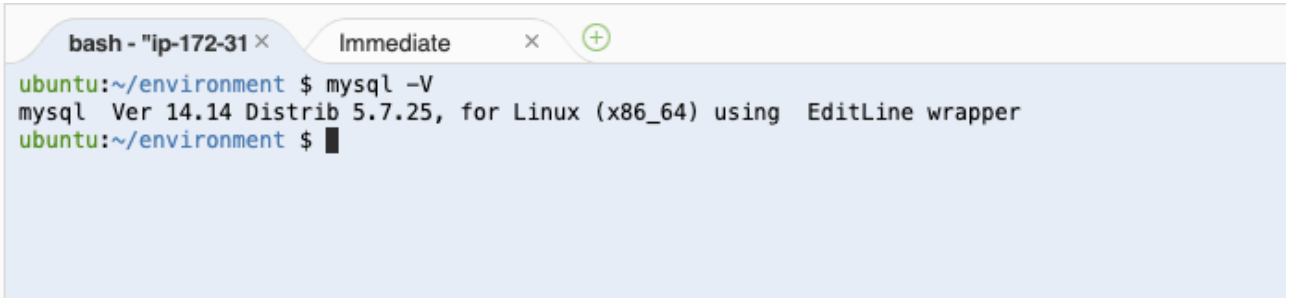
8. A default IAM role(If you don't have one will be generated with name "AWSServiceRoleForAWSCloud9 (generated)". Click on Create environment button, this will generate your environment. Now that your environment is ready. You will be greeted with this screen for first time. Setup takes a little time, have some patience.

Now, once cloud9 instance is ready, you will see something like this.



Awesome!!! our environment is ready for use. Let's setup the MySQL!

By Default MySQL 5.7.25 is installed in the environment (remember we've selected UBUNTU 18.04 LTS), but we do have to make sure that we have correct mysql Installed, to check that run "mysql -V" . Running the command will show something like this:



Yay!! we're done… but nah!, we have to do additional installation steps to setup the MySQL.

Let's do Initial Installation to setup Mysql like setting up root password, enabling MySQL on start, starting mysql for first time etc…

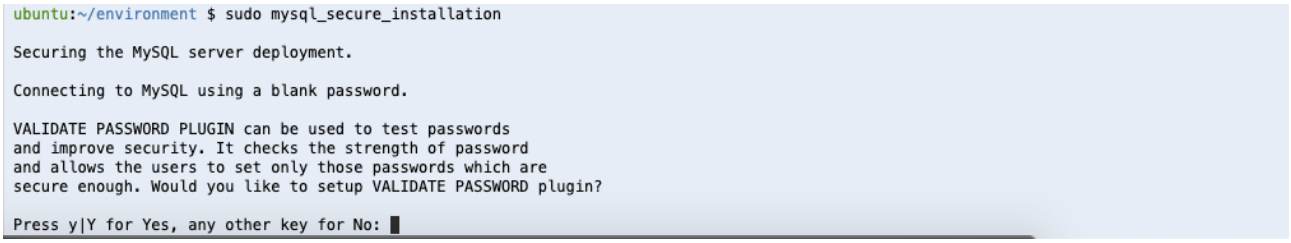
9. First we need to switch to root user and set the password for our "ubuntu" user as AWS creates on by default.

1. sudo su, to change to root user
   you will see a command prompt like this : root@ip-~~172-31-18-238~~:/home/ubuntu/environment#
2. now let's set the password for ubuntu user
   passwd ubuntu

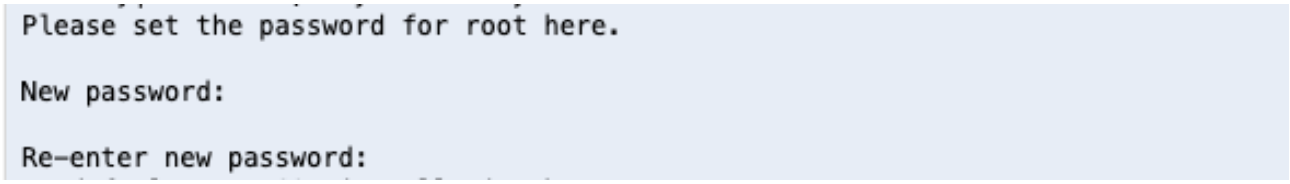
3. Also let's set our root password too, so as to be able to login as root to mysql
   passwd

4. On the prompt set the desired password.
5. Once the password is set logout doing hitting ctrl + d

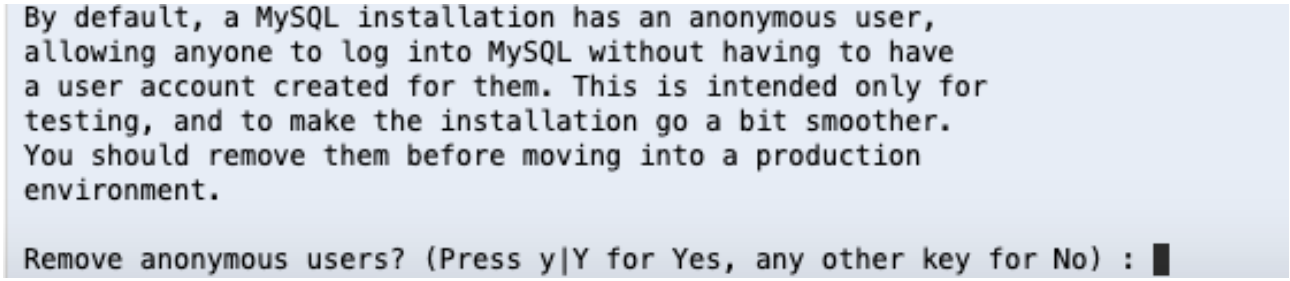
Once done we will do MySQL initial setup next run, sudo mysql_secure_installation and go through the steps as mentioned below.



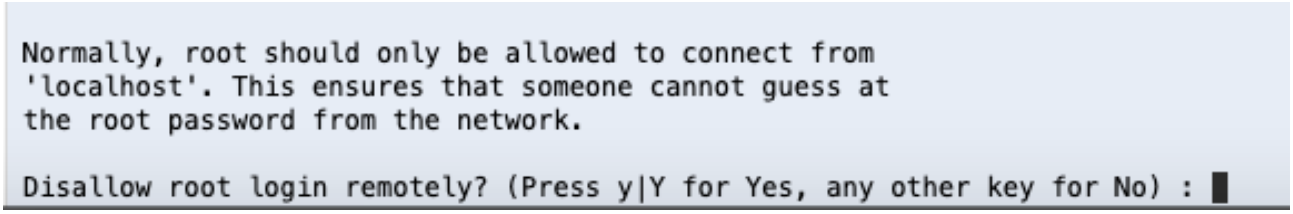
Hit No, in case you don't want to apply validations(like strong password or password requirement etc.).



Now let's set password for mysql root user at next prompt which look like this
Next, let's remove mysql anonymous users as they are not necessary at this prompt:



Once we're done with above, we shall actually disallow the remote login for root mysql user:



Now, we shall remove the test database as it's of no use, Hit Yes at the prompt you will get this message

```
By default, MySQL comes with a database named 'test' that
anyone can access. This is also intended only for testing,
and should be removed before moving into a production
environment.


Remove test database and access to it? (Press y|Y for Yes, any other key for No) : Y
 - Dropping test database...
Success.

 - Removing privileges on test database...
Success.

Reloading the privilege tables will ensure that all changes
made so far will take effect immediately.
```

Great, let's reload the privileges, hitting Yes

You will get following confirmation:

```
Reloading the privilege tables will ensure that all changes
made so far will take effect immediately.

Reload privilege tables now? (Press y|Y for Yes, any other key for No) : Y
Success.

All done!
ubuntu:~/environment $
```

Now let's start MySQL server.

To start MySQL server, run :

```
sudo service mysql restart
```

To enable MySQL server to start at each reboot/boot run:

```
sudo systemctl enable mysql
```

To check if our mysql is running fine or not, run :

```
sudo mysql -u root -p
```

You will get the following prompt (use password we've set for root user)

MySQL maintains separate user authentication other than root user. So you need to create a new user post logging into MySQL as root, sudo mysql -u root -p (use root user password we've created)

That's It, we have successful done mysql setup and installation.

Creating Database, Connecting to Database,

Connecting to MySQL and creating Database.

To connect to database you need to run following command:

mysql -u root -p

You will be prompted for root password, enter the same and hit enter.

Let's create new DB. run:

create

```
mysql> create database newdb;
Query OK, 1 row affected (0.01 sec)

mysql>
```

database newdb

Next, create a user and grant all privileges and create a new User. let's run the command below to do so:

```
mysql> use newdb;
Database changed
mysql> create table tasks(task_id INT, title VARCHAR(255) NOT NULL);
Query OK, 0 rows affected (0.08 sec)

mysql>
```

GRANT ALL PRIVILEGES ON newdb.* TO 'mashrur'@'localhost' identified by 'Password';

```
mysql> GRANT ALL PRIVILEGES ON newdb.* TO 'mashrur'@'localhost' identified by 'Password';
Query OK, 0 rows affected, 1 warning (0.01 sec)

mysql>
```

Now we're able to login to our database using following command:

mysql -u mashrur -p and than provide the password you shall see the following screen

Let's create the table :)

first use the new db, run :

use newdb;

than let's create table, run following command:

create table tasks(task_id INT, title VARCHAR(255) NOT NULL);

```
mysql> use newdb;
Database changed
mysql> create table tasks(task_id INT, title VARCHAR(255) NOT NULL);
Query OK, 0 rows affected (0.08 sec)

mysql>
```

The above table will create the table tasks with task id as INT, title as VARCHAR

That's all we've created our DB, Created a user and connected to it. Moving ahead, we had created created a table and granted privileges etc….