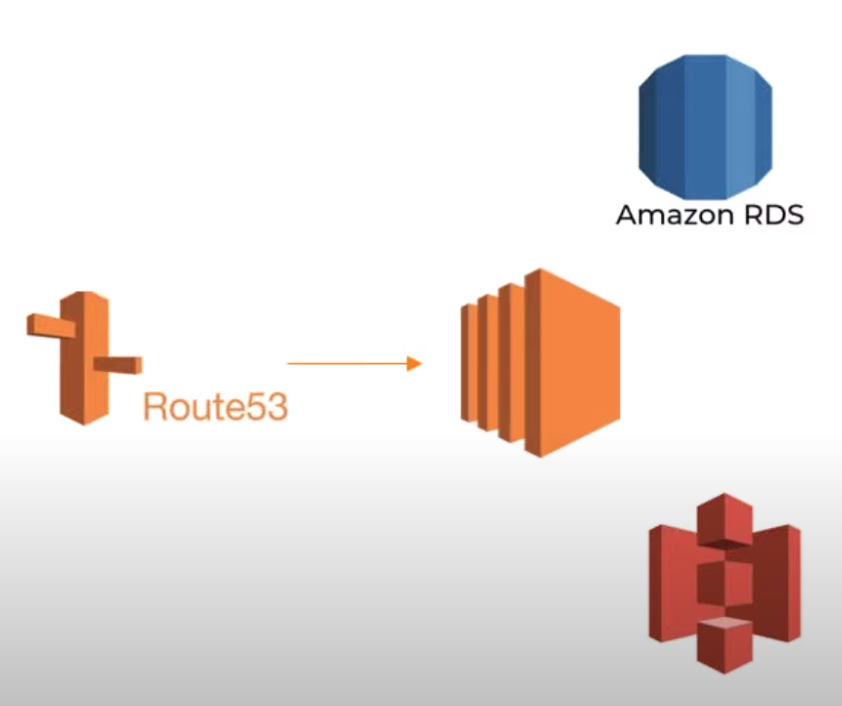
**AWS Deployment using EC2, Route 53 & RDS**

* User -> Website([www.aws-test.com](http://www.aws-test.com/))



**Create DB Instance**

* Go to RDS Service
* Create Database [Standard DB, Easy DB]
* Engine options- Amazon Aurora, MySQL, MariaDB, PostgreSQL, Oracle, Microsoft SQL Server
* Template- Production, Dev/Test, Free-Tier
* Settings- Name your DB Instance
* Credentials Settings- Provide user credentials
* Instance size, Storage, Storage Auto-Scaling
* VPC default, Security Group
* Password Authentication

**Create S3 Bucket**

* Go to S3 Service
* Create Bucket
* Select region [RDS should be created in same region]

**Create EC2 Server**

* Go to EC2 Service
* Launch Instance (Choose AMI from Linux/ Red Hat/ Windows/ Ubuntu)
* Instance Type- Configure Instance- Add Storage- Add Tags- Configure Security Group- Review

**Connect to EC2 Server from Terminal**

* *ssh -i /path/my-key-pair.pem user-name@instance-ip-address*
* Connect to RDS from EC2 instance by installing MySQL

**Steps for Linux Instance-**

*sudo wget https://dev.mysql.com/get/mysql57-community-release-el7-11.noarch.rpm*

*sudo yum localinstall mysql57-community-release-el7-11.noarch.rpm*

*sudo yum install mysql-community-server*

*sudo systemctl start mysqld.service*

**Steps for Ubuntu Instance-**

**Step 1:**

*sudo apt update*

*sudo apt-get install mysql-server -y*

**Step 2:**

To login on MySQL as root user, type command: *sudo mysql*

Create a database, type command: *CREATE DATABASE mynewdatabase;*

Show the databases using: *SHOW DATABASES;*

**Step 3:**

Add New User for MySQL Database

*CREATE USER 'myuser'@'localhost' IDENTIFIED BY 'mypassword';*

**Step 4:**

Grant User permission for MySQL database

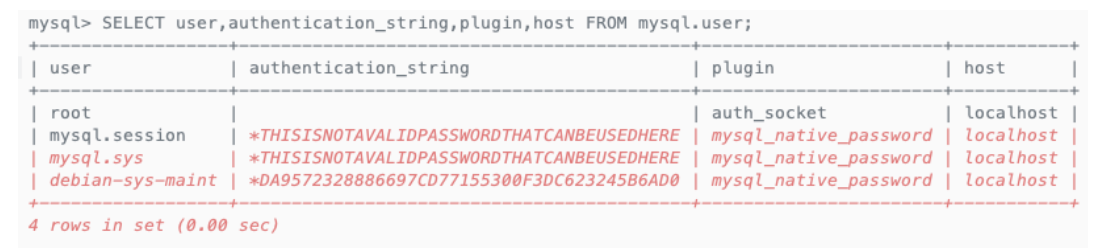
To apply the database changes without reloading your MySQL service. Type command: *FLUSH PRIVILEGES;*

**Step 5:**

Configure MySQL root password

Verify which authentication method each of your MySQL user account, type command:

*SELECT user,authentication\_string,plugin,host FROM mysql.user;*



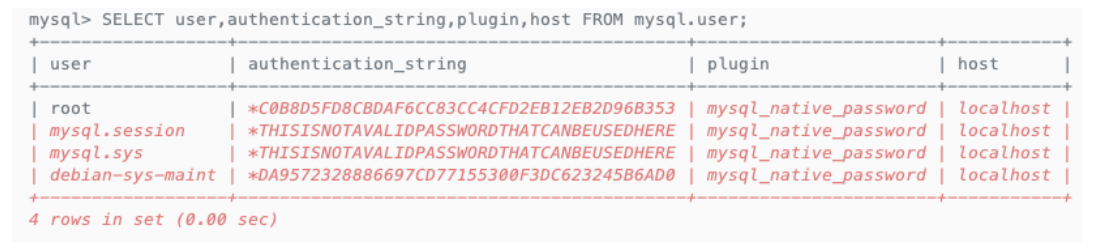
You can see the root user has no authentication\_string and using the auth\_socket plugin.

Type ALTER USER command to configure the root account to authenticate using password access.

*ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql\_native\_password BY 'mypassword';*

Verify your root user and confirm, if root user is no longer authenticated with auth\_socket plugin, run the command.

*SELECT user,authentication\_string,plugin,host FROM mysql.user;*



**Step 6:**

Verify MySQL Database

*sudo service mysql status*

To manage your MySQL server on Ubuntu 18.04 server, use the following command:

*sudo service mysql start*

*sudo service mysql stop*

*sudo service mysql reload*

*sudo service mysql restart*

For an additional check, you can try connecting to the database using the mysqladmin tool and return the version.

*sudo mysqladmin -p -u root version*

**Step 7:**

Basic MySQLdump command and importing database

If you want to backup your database within Ubuntu 18.04 server, type this command.

*mysqldump -u username -p database\_name > database\_name.sql*

To restore/import database from backup sql file, use this command:

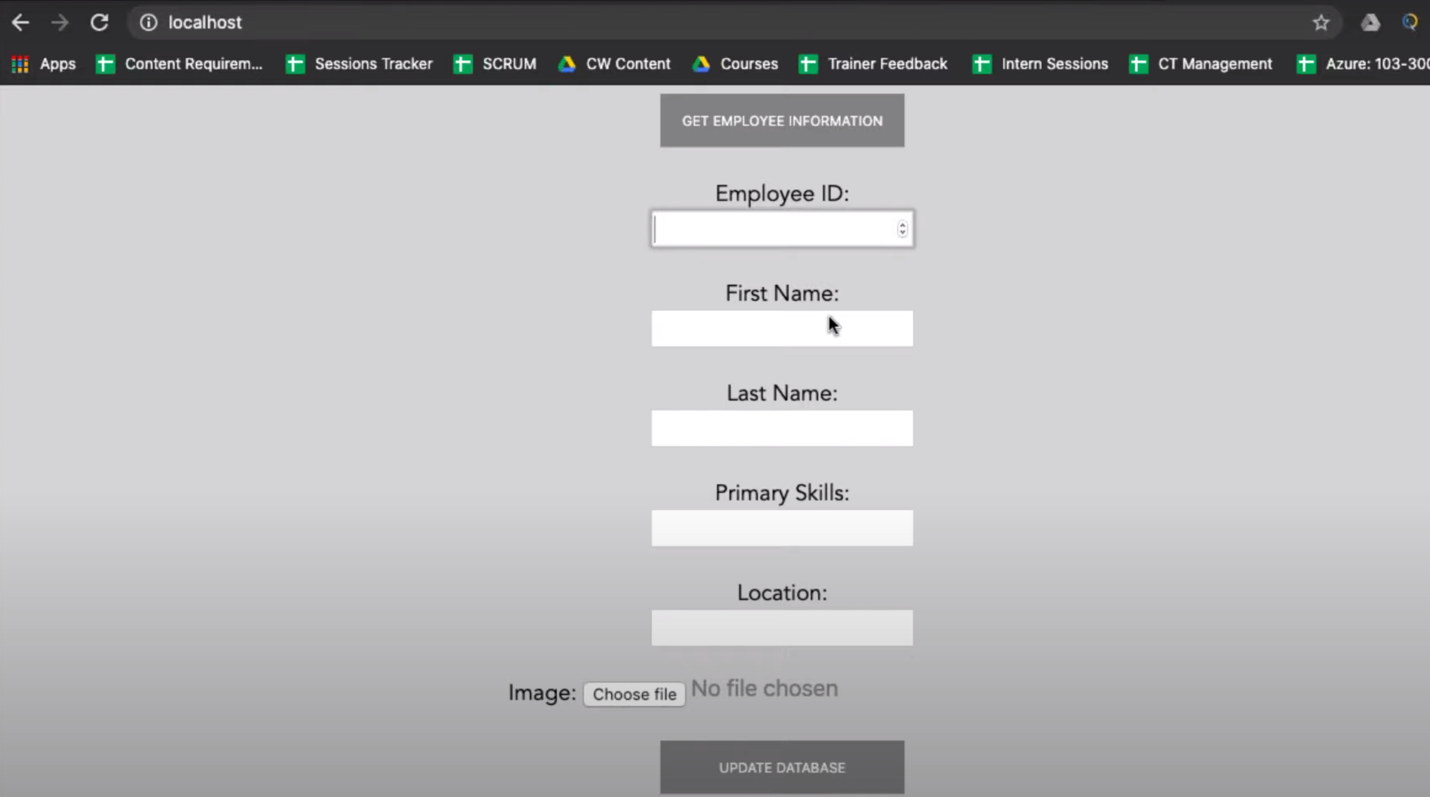
*mysql -u username -p database\_name < database\_name.sqlCopy*

To audit and troubleshoot error, use this command:

*sudo tail -f /var/log/mysql/error.log*

Next, exit from the MySQL console using: EXIT;

Useful- <https://linuxbeast.com/tutorials/aws/how-to-install-mysql-on-amazon-ec2-ubuntu-18-04/>



*mysql> create database student;*

*mysql> use student;*

*mysql> create table student(*

*studentid varchar(20),*

*fname varchar(20),*

*lname varchar(20),*

*pri\_skill varchar(20),*

*location varchar(20));*

*mysql> show tables;*

**Sample python project-**

Go to config.py

customhost = "RDS-instance-name"

customuser = "prabha"

custompass = "prabha123"

customdb = "student"

customuser = "addstudent"

customuser = "us-east-2"

EC2 Server:

*git clone https://github.com/pranavi9/aws-live.git*

*sudo apt-get update*

*# For Sql-client*

*sudo apt-get install mysql-client*

*# For python and related frameworks*

*sudo apt-get install python3 => lang version*

*sudo apt-get install python3-pip*

*sudo apt-get install python3-flask => Web frame*

*pip3 install flask*

*sudo apt-get install python3-pymysql => Library*

*sudo apt-get install python3-boto3 => AWS Python SDK to connect to S3 service*

*pip3 install pymysql boto3*

*# for running application*

*sudo python3 Empapp.py*

**End User wants to add information in website:**

In order to save end-user information in S3, set-up IAM role to EC2 Instance(where web-site is hosted)

Go to IAM Service

Role -> create Role -> select EC2 use case -> Select Policy (default provide admin access, it can connect to any AWS service)

Provide Role name "example" and create

After creating Role, go to EC2 instance -> Actions -> Instance settings -> Attach/Replace IAM Role -> Select our Role "example"

**Create Domain for our Website:**

<https://www.freenom.com/en/index.html?lang=en>

Provide name "aws-test.com" and check availability - free version is available for 3 months and place order.

Go to Service - My Domain - verify - Manage Domain - Management Tools - Nameservers (select use custom name server) -> Change Name servers

**Route 53 Service-**

Create Hosted Zone -> provide domain name ->You can use the values in Nameservers custom name

**Connect Route 53 to EC2 Instance:**

Create Record set

1)Keep default name (aws-test.com)Enter public IP address

2)Add www (www.aws-test.com)Enter public IP address