Tutorial/Laboratory 08 (Part 1)

Database Operations with PHP and MySQL

1. LEARNING OUTCOMES

Upon completion of these laboratory exercises, you should be able to:

- Configure a static IP address for an AWS instance using the Elastic IP option.
- Add and grant permissions to non-root MySQL users.
- Create a database server connection in MySQL Workbench.

2. REQUIRED SOFTWARE

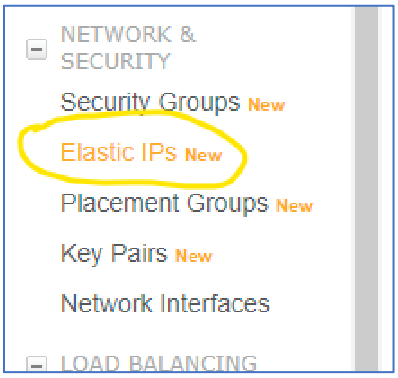
- Apache NetBeans 11.2 (or later): https://netbeans.apache.org/download/index.html
- FireFox (https://www.mozilla.org/en-US/firefox/new/) or Chrome (https://www.google.com/chrome/) web browser.
- AWS Educate account with EC2 instance running Ubuntu Server & LAMP stack (https://www.awseducate.com/signin/SiteLogin).
- MySQL Workbench (https://dev.mysql.com/downloads/workbench/)

3. EXERCISE 1: CONFIGURE A STATIC IP ADDRESS

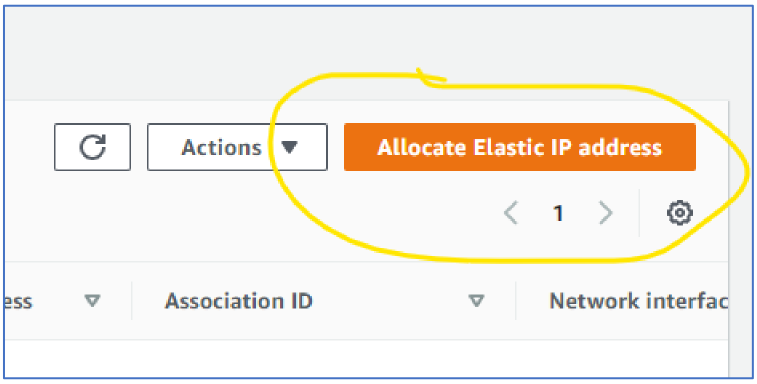
As we learned in the previous Lab exercise, AWS instances use dynamic IP addresses by default. This means each time you shut down and restart your instance, it'll be assigned a new IP address. Therefore, you would need to continuously update any links or references to the server's IP address, which of course is very cumbersome.

Instead, it is desirable to use a static IP address that never changes, especially for persistent instances such as web and database servers. In this exercise, we'll use the AWS "Elastic IP" feature to configure a static IP address for our LAMP stack.

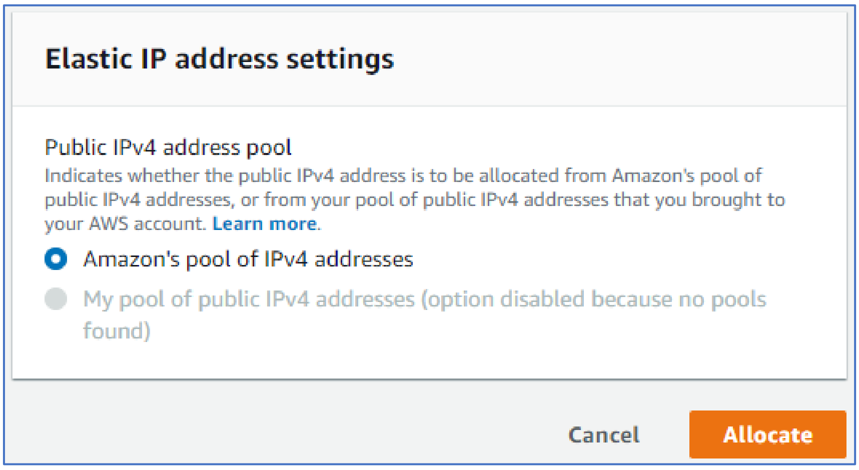
3.1 Go to the AWS console in your AWS Educate account (follow the instructions from the previous Lab exercise). From the menu on the left, click on "Elastic IPs" under the "NETWORK & SECURITY" category.



3.2 Click on the "Allocate Elastic IP address" button.



3.3 Keep the default settings and click the “Allocate” button



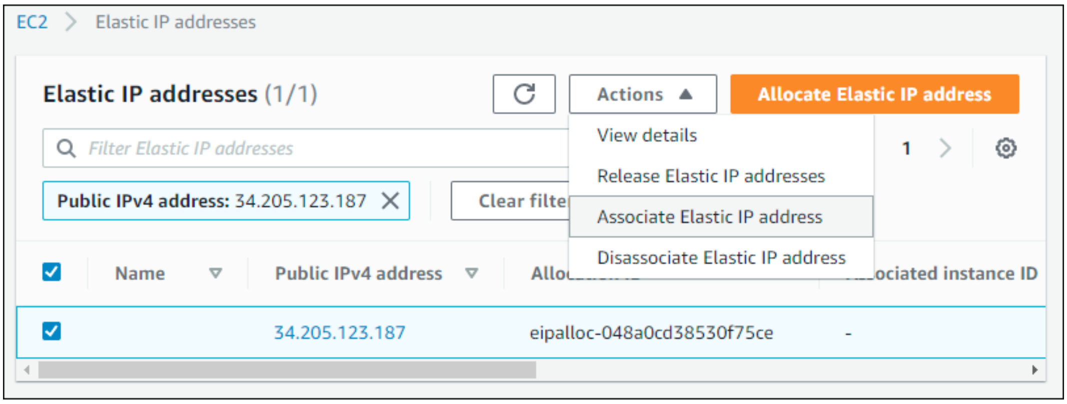
Elastic IP address settings

Public IPv4 address pool
Indicates whether the public IPv4 address is to be allocated from Amazon's pool of public IPv4 addresses, or from your pool of public IPv4 addresses that you brought to your AWS account. Learn more.

● Amazon's pool of IPv4 addresses

○ My pool of public IPv4 addresses (option disabled because no pools found)

3.4 With the new IP address selected, click on “Associate Elastic IP address” from the “Actions” dropdown menu.



EC2 > Elastic IP addresses

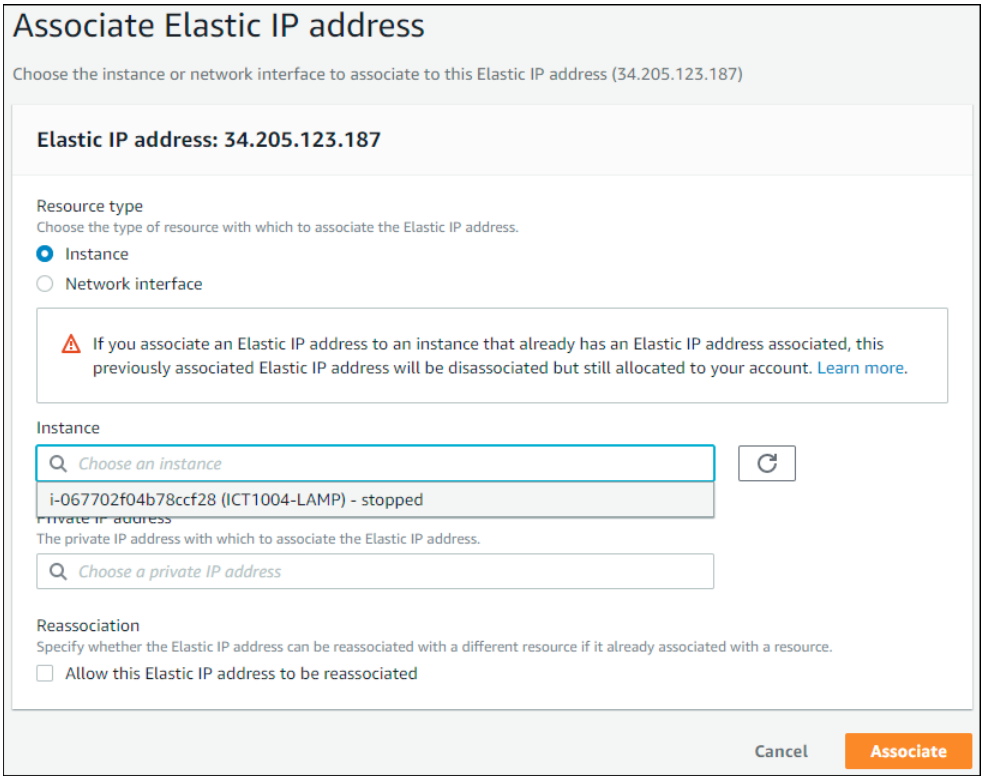
Elastic IP addresses (1/1)

Public IPv4 address: 34.205.123.187

☑	Name	Public IPv4 address	Allo…	…ociated instance ID
☑		34.205.123.187	eipalloc-048a0cd38530f75ce	-

Actions
Allocate Elastic IP address
Filter Elastic IP addresses
Clear filte
View details
Release Elastic IP addresses
Associate Elastic IP address
Disassociate Elastic IP address
1

3.5 On the next page, leave “Resource type” set to “Instance” and click in the box to select the Ubuntu Server instance where you configured the LAMP stack in the previous Lab. Leave the other settings as they are and click the “Associate” button.



Associate Elastic IP address

Choose the instance or network interface to associate to this Elastic IP address (34.205.123.187)

Elastic IP address: 34.205.123.187

Resource type

Choose the type of resource with which to associate the Elastic IP address.

● Instance

○ Network interface

If you associate an Elastic IP address to an instance that already has an Elastic IP address associated, this previously associated Elastic IP address will be disassociated but still allocated to your account. Learn more.

Instance

Choose an instance

i-067702f04b78ccf28 (ICT1004-LAMP) - stopped

Private IP address

The private IP address with which to associate the Elastic IP address.

Choose a private IP address

Reassociation

Specify whether the Elastic IP address can be reassociated with a different resource if it already associated with a resource.

□ Allow this Elastic IP address to be reassociated

3.6 After creating the IP address and associating it with your LAMP instance, click on “EC2 Dashboard” in the left-hand menu, then click “Running instances” in the “Resources” panel. Start up the Ubuntu Server instance with LAMP stack that you created in the previous Lab (‘Actions->Instance State->Start’).

3.7 Once the instance is spooled up and passes status checks, confirm that you can log in (SSH) to the server using the static IP address. Be sure to note down this IP address, as you will use it from now on to access your web server and MySQL database.

4. EXERCISE 2: ADD A NON-ROOT MYSQL USER AND GRANT PERMISSIONS

In the previous Lab, we installed MySQL with the default **root** user and locked it down for security reasons. We need to add a new non-root user for development purposes that has only the necessary permissions for managing databases.

4.1 Log into your Ubuntu Server instance using SSH and follow the steps below to create the new user.

a. Run the MySQL shell (logging in as **root**).

```
$ sudo mysql
```

b. Create the new user - replace ‘password’ with the password of your choice, and be sure to remember it.

```
mysql> CREATE USER 'sqldev'@'localhost' IDENTIFIED BY 'password';
```

c. Grant privileges to the user. It’s good practice to only grant the minimum privileges required. The **sqldev** account will be used for carrying out database operations for our web applications, so we need to give it all the CRUD related privileges.

```
mysql> GRANT CREATE, SELECT, INSERT, UPDATE, DELETE, DROP, SHOW DATABASES ON * . *
TO 'sqldev'@'localhost';
```

d. Lastly, always flush privileges to force new ones to take effect.

```
mysql> FLUSH PRIVILEGES;
```

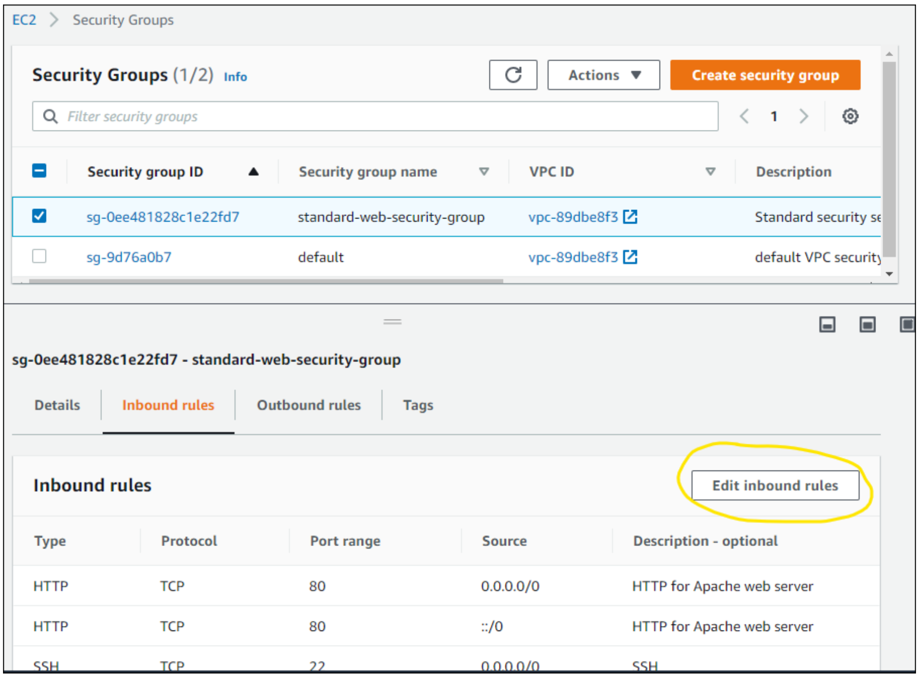
e. To check that privileges for **sqldev** were set up properly, log out of the MySQL shell (type ‘exit’ at the shell prompt) and log back in as **sqldev**, then execute the SHOW GRANTS command.

```
mysql> exit
Bye

$ mysql -u sqldev -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 19
Server version: 5.7.29-0ubuntu0.18.04.1 (Ubuntu)

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> SHOW GRANTS;
+--------------------------------------------------------------------------------
-----------------+
| Grants for sqldev@localhost
|
+--------------------------------------------------------------------------------
-----------------+
| GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, SHOW DATABASES ON *.* TO
'sqldev'@'localhost' |
+--------------------------------------------------------------------------------
-----------------+
1 row in set (0.00 sec)

mysql>
```

5. EXERCISE 3: OPEN THE FIREWALL FOR MYSQL

Next, we must edit the Security Group and open port 3306 to allow database connections to our server.

5.1 In the AWS console, under the “NETWORK & SECURITY” category on the left, click “Security Groups”. Select the web server security group that you created in the previous Lab exercise, then click “Edit inbound rules.”



Security Groups (1/2) Info

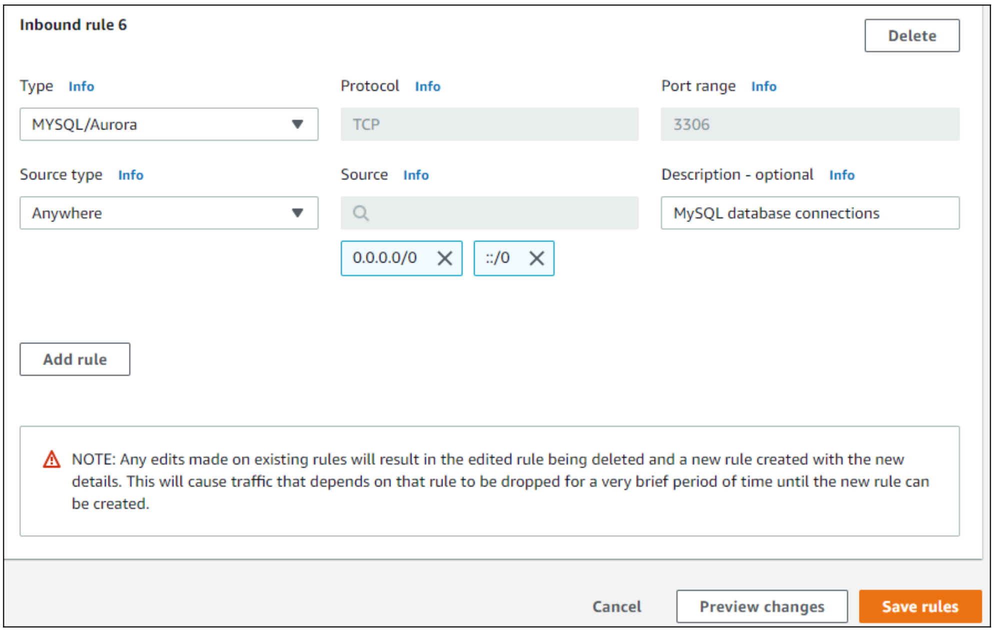
☐	Security group ID	Security group name	VPC ID	Description
☑	sg-0ee481828c1e22fd7	standard-web-security-group	vpc-89dbe8f3	Standard security se
☐	sg-9d76a0b7	default	vpc-89dbe8f3	default VPC security

sg-0ee481828c1e22fd7 - standard-web-security-group

Inbound rules

Type	Protocol	Port range	Source	Description - optional
HTTP	TCP	80	0.0.0.0/0	HTTP for Apache web server
HTTP	TCP	80	::/0	HTTP for Apache web server
SSH	TCP	22	0.0.0.0/0	SSH

Click “Add rule” and select “MYSQL/Aurora” as the Type. For “Source type” select “Anywhere” and enter an optional description.



Inbound rule 6

Type Info: MYSQL/Aurora

Protocol Info: TCP

Port range Info: 3306

Source type Info: Anywhere

Source Info: 0.0.0.0/0 ::/0

Description - optional Info: MySQL database connections

NOTE: Any edits made on existing rules will result in the edited rule being deleted and a new rule created with the new details. This will cause traffic that depends on that rule to be dropped for a very brief period of time until the new rule can be created.

Click “Save rules” to save the changes.

6. EXERCISE 4: CONNECTING TO THE DATABASE SERVER USING MYSQL WORKBENCH

Now that the MySQL server is configured, we can try connecting to it. MySQL Workbench is a popular tool for managing databases securely using SSH tunneling.

6.1 Download MySQL Workbench. You can skip the Oracle account login/signup and click on the ‘No thanks, just start my download.’ link near the bottom of the page. Once the download is complete, run the installer (choose the “Complete” option when prompted). After the installation is complete, launch MySQL Workbench.

6.2 In MySQL Workbench, create a database connection. Click on the ‘+’ sign next to “MySQL Connections” to add a connection.

a. In the “Setup New Connection” dialog box, enter a connection name of your choice (use a name that helps you remember what that database is for).

b. Change “Connection Method” to “Standard TCP/IP over SSH”.

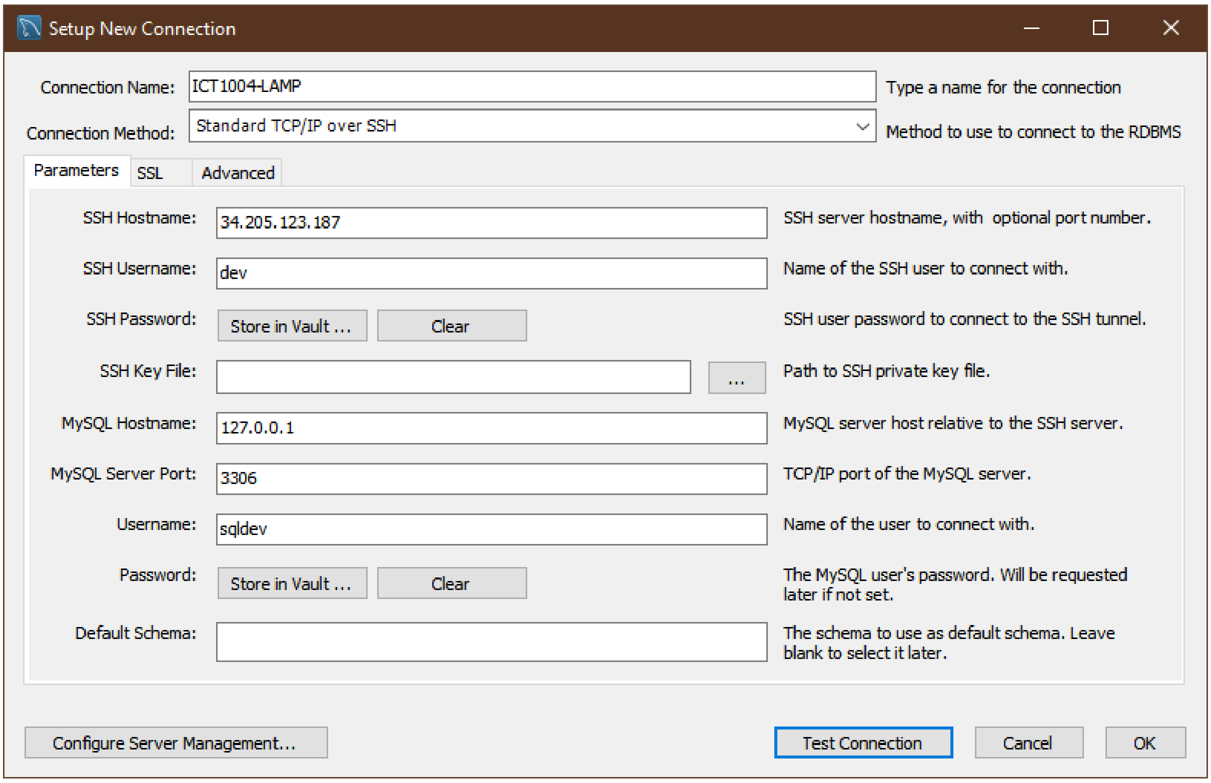
c. For “SSH Hostname” enter the static IP address of your LAMP server.

d. For “SSH Username” enter **dev** (or whichever user account you SSH into the server with). This is *not* the MySQL user (that’s further down).

e. “SSH Password” is the password you have been using to log in via SSH.

f. For “MySQL Hostname” and “MySQL Server Port” you can leave the default values. We’ll be logging into the database server as a local user (after tunneling through SSH).

g. For “Username” and “Password” enter the MySQL user that you created in 4.1 above.


Setup New Connection

Connection Name: ICT1004-LAMP — Type a name for the connection

Connection Method: Standard TCP/IP over SSH — Method to use to connect to the RDBMS

SSH Hostname: 34.205.123.187 — SSH server hostname, with optional port number.

SSH Username: dev — Name of the SSH user to connect with.

SSH Password: — SSH user password to connect to the SSH tunnel.

SSH Key File: — Path to SSH private key file.

MySQL Hostname: 127.0.0.1 — MySQL server host relative to the SSH server.

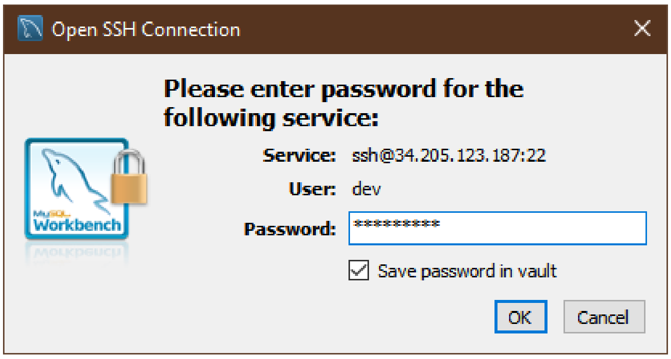
MySQL Server Port: 3306 — TCP/IP port of the MySQL server.

Username: sqldev — Name of the user to connect with.

Password: — The MySQL user's password. Will be requested later if not set.

Default Schema: — The schema to use as default schema. Leave blank to select it later.

6.3 Click the “Test Connection” button to see if everything works. Note that you should be prompted for TWO passwords, one when connecting via SSH and another for the MySQL login.


Open SSH Connection

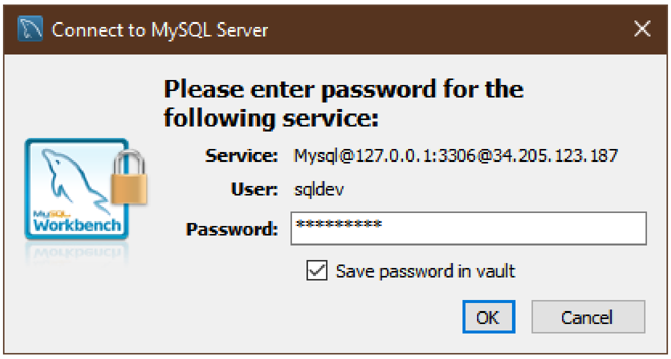
Please enter password for the following service:

Service: ssh@34.205.123.187:22

User: dev

Password: **********

☑ Save password in vault


Connect to MySQL Server

Please enter password for the following service:

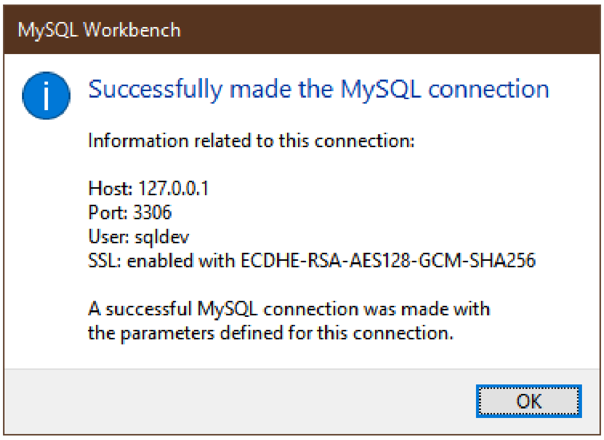
Service: Mysql@127.0.0.1:3306@34.205.123.187

User: sqldev

Password: **********

☑ Save password in vault

You should see the message below if the connection was successful.



Successfully made the MySQL connection

Information related to this connection:

Host: 127.0.0.1
Port: 3306
User: sqldev
SSL: enabled with ECDHE-RSA-AES128-GCM-SHA256

A successful MySQL connection was made with
the parameters defined for this connection.

7. CONCLUSION

This completes Part 1 of the Lab exercise. In Part 2, we'll proceed to create a database schema and tables using MySQL Workbench, then learn how to use PHP to execute database commands.