Bethe OPS Project

Developer Instructions: Hosting

May 16, 2019

Table of Contents

1 Overview

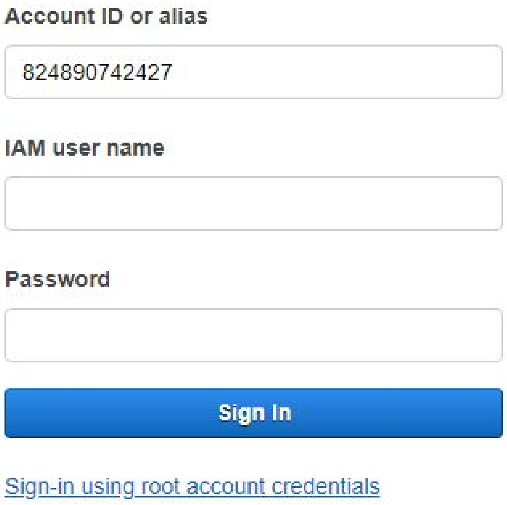
This document describes how the Bethe OPS system is hosted. There are two main components of the system that require hosting: the front-end application that users interact with and the back-end database that contains the actual information. Amazon Web Services, otherwise known as AWS, is used to host both of these components. AWS has many different cloud computing services at different prices. For the system, free-tier versions were used.

The front-end was hosted on AWS EC2 and the database was hosted on AWS RDS. Details on how to set up the hosting, accessing the hosting, and modifying different aspects of the hosting are described below.

1.1 Logging Into the Current System on AWS

The details in the sections 3 to 6 detail setting up hosting and logging in from scratch. Since the system is already in place, the details of logging into the current system will be put here.

To use the student account Erica created, visit this URL. This should result in the following page.



Account ID or alias: 824890742427

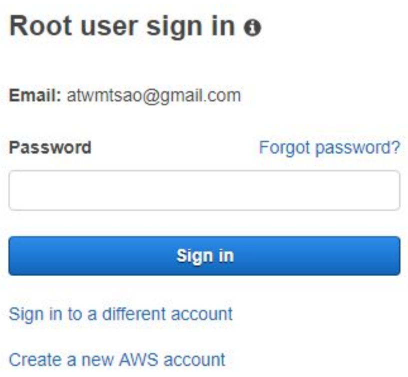
IAM user name:

Password:

The credentials are:

- **IAM user name**: StudentTech
- **Password**: BetheOPS

For Erica, visit this URL and make sure the page says “Root user sign in” as shown below.



Root user sign in

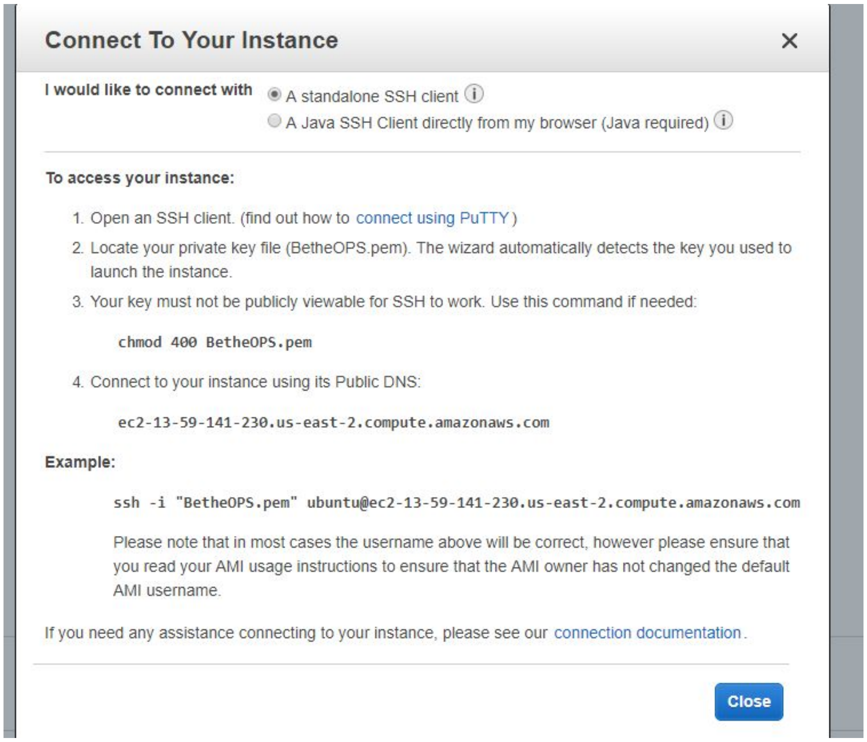
Email: atwmtsao@gmail.com

Password Forgot password?

1.2 Logging Into the Current EC2 System

Follow the steps in section 4 below. *BetheOPS.pem* is for Mac/Linux users and *BetheOPSConverted.ppk* is for Windows users. These instructions are also included in the README.

For Mac users, follow the instructions:



Connect To Your Instance

I would like to connect with ● A standalone SSH client ○ A Java SSH Client directly from my browser (Java required)

To access your instance:

1. Open an SSH client. (find out how to connect using PuTTY)
2. Locate your private key file (BetheOPS.pem). The wizard automatically detects the key you used to launch the instance.
3. Your key must not be publicly viewable for SSH to work. Use this command if needed:

```
chmod 400 BetheOPS.pem
```

4. Connect to your instance using its Public DNS:

```
ec2-13-59-141-230.us-east-2.compute.amazonaws.com
```

Example:

```
ssh -i "BetheOPS.pem" ubuntu@ec2-13-59-141-230.us-east-2.compute.amazonaws.com
```

Please note that in most cases the username above will be correct, however please ensure that you read your AMI usage instructions to ensure that the AMI owner has not changed the default AMI username.

If you need any assistance connecting to your instance, please see our connection documentation.

For Windows users, use PuTTY as in section 4.1.2. The important details are:

- **Host Name**: ec2-13-59-141-230.us-east-2.compute.amazonaws.com
- **Auto-login Username** (in the "Data" section): ubuntu
- Use *BetheOPSConverted.ppk*

1.3 Connecting to the Current Database on RDS

Follow the steps in section 4.2 using MySQLWorkbench. The important details are:

- **Hostname**: betheops-db-instance.cymmf1dfezhv.us-east-2.rds.amazonaws.com
- **Username**: StudentTech
- Click on "Store on Vault". **Password**: BetheOPS

2 Required Technologies

The required technologies for hosting the system are:

- AWS Elastic Compute Cloud (EC2)
- AWS Relational Database Service (RDS)
- Basic knowledge of navigating the Linux terminal
- A terminal that can use SSH for accessing the EC2 instance
- MySQL Workbench for accessing the RDS instance
- PM2

3 The Setup

Before detailing the steps in implementing the hosting, it is worth noting that Amazon has many tutorials and documentations on hosting. This tutorial was extremely helpful: https://wtmimura.com/post/aws-ec2-rds-wordpress-site/.

This tutorial will be referred to as the **setup tutorial** in later sections. Many of the steps in this tutorial are the same as those taken by the team to implement hosting so many references will be made to this tutorial.

The following three tutorials were also helpful in the individual steps of hosting:

1. Setting security groups and VPCs
2. Creating the EC2 instance and configuring settings
3. Creating the RDS instance to host the database

The sections that follow will go in depth into setting up the hosting. Some of the details in the tutorials above will be modified to fit the use of the system.

3.1 Creating Security Groups and VPC

3.1.1 What is a VPC?

VPCs are virtual networks tied to AWS accounts and are where the AWS services can be deployed. For example, when the EC2 instance is created, we will define a VPC network in which it will be deployed. With a VPC, we can limit communication between services in the network and the rest of the internet through the use of security groups.

3.1.2 What is a security group?

AWS security groups can be thought of as a set of rules that define who is allowed to talk to our services. For example, a security group can be configured to only allow certain traffic in and out of the service to which it is attached.

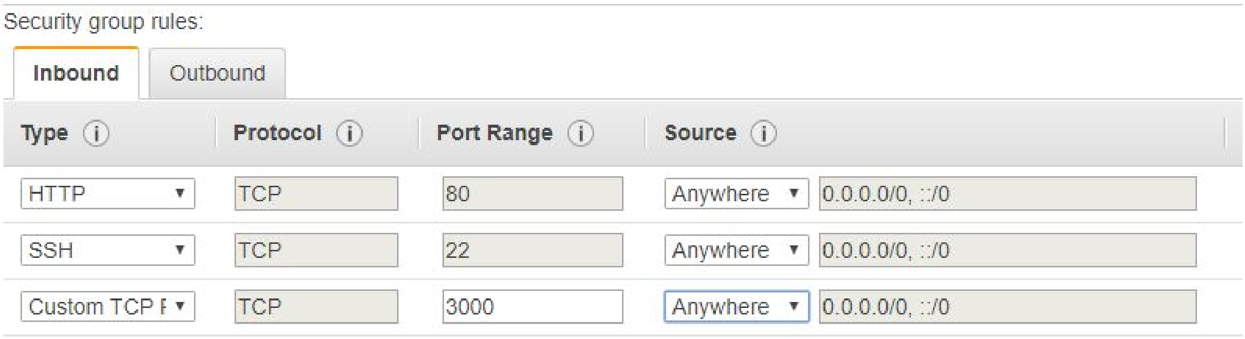
3.1.3 Setting Up the VPC and Security Groups for This Project

For all AWS users, there will be already be a default VPC that does not need to be explicitly configured and is available for use. To check this, go here. From there, click on “Your VPCs” in the left sidebar to see the available VPCs. There should at least be one VPC there.

For creating the security groups that will be used for the EC2 instance (server) and RDS (database), please follow the steps in section 2.1 in “2: Hands On” in the setup tutorial above. There are slight differences so the exact rules that should be created are described below.

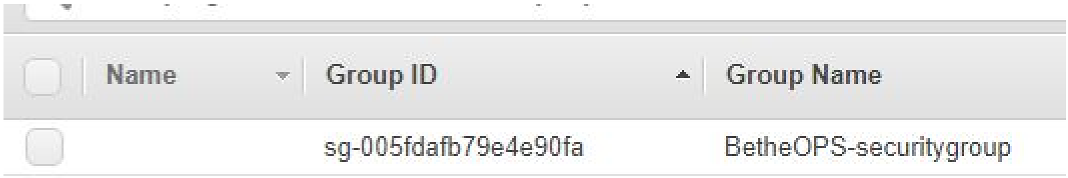
For the EC2 instance, the following rules should be added

- Type: HTTP, Protocol: TCP, Port: 80, Source: Anywhere
- Type: SSH, Protocol: TCP, Port: 22, Source: Anywhere or My IP depending on whether you want to allow other users to SSH into your EC2 instance.
- Type: Custom TCP Rule, Protocol: TCP, Port: 3000, Source: Anywhere



Type	Protocol	Port Range	Source	
HTTP	TCP	80	Anywhere	0.0.0.0/0, ::/0
SSH	TCP	22	Anywhere	0.0.0.0/0, ::/0
Custom TCP F	TCP	3000	Anywhere	0.0.0.0/0, ::/0

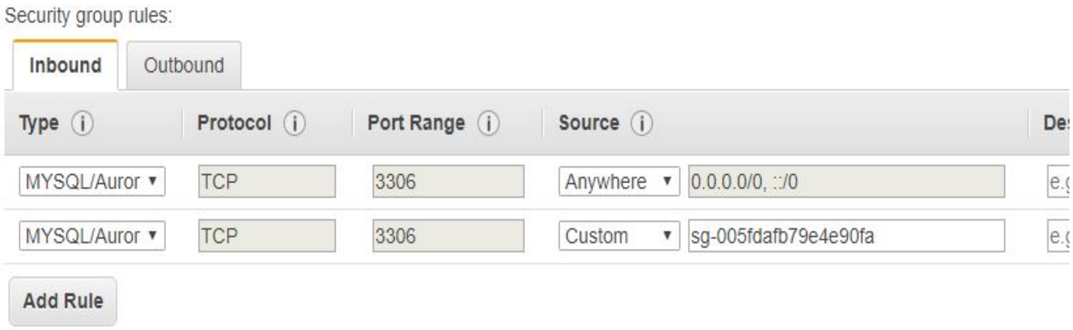
Once this security group is created, it will be assigned a "Group ID" as in the example below. In the example below, the group ID is sg-005fdafb79e4e90fa. Keep track of this as we will use this later.



Name	Group ID	Group Name
	sg-005fdafb79e4e90fa	BetheOPS-securitygroup

For the RDS instance, the following rules should be added

- Type: MySQL/Aurora, Protocol: TCP, Port: 3306, Source: Anywhere or My IP
- Type: MySQL/Aurora, Protocol: TCP, Port: 3306, Source: Custom with the group ID as shown below.



Type	Protocol	Port Range	Source		De
MYSQL/Auror	TCP	3306	Anywhere	0.0.0.0/0, ::/0	e.g
MYSQL/Auror	TCP	3306	Custom	sg-005fdafb79e4e90fa	e.g

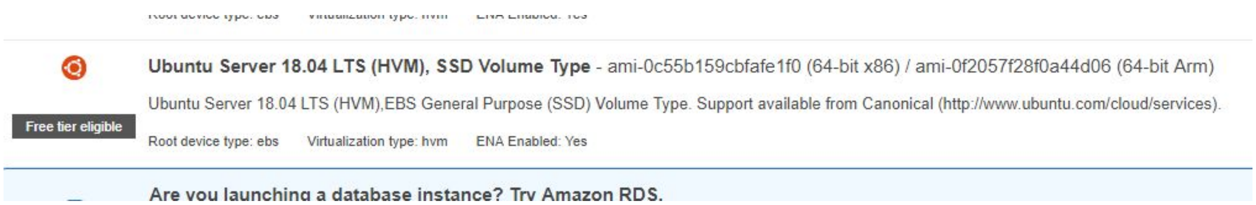
This custom MYSQL rule will allow apps run on the system's EC2 instance, such as the Bethe-OPS server, to access any database hosted on RDS instances with the above security group.

3.2 Creating the EC2 Instance

To create the EC2 instance, please follow the steps in section 2.3 in the setup tutorial as most of the steps are the same.

3.2.1 Which engine to create?

Instead of creating a Linux AMI distribution as in the tutorial, create an Ubuntu Server. This will create an instance of an Ubuntu system that developers can easily SSH into.



Root device type: ebs Virtualization type: hvm ENA Enabled: Yes

Ubuntu Server 18.04 LTS (HVM), SSD Volume Type - ami-0c55b159cbfafe1f0 (64-bit x86) / ami-0f2057f28f0a44d06 (64-bit Arm)

Ubuntu Server 18.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (http://www.ubuntu.com/cloud/services).

Free tier eligible

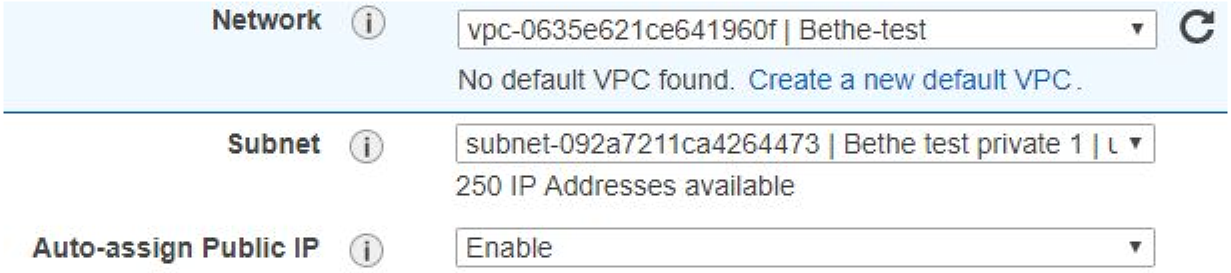
Root device type: ebs Virtualization type: hvm ENA Enabled: Yes

Are you launching a database instance? Try Amazon RDS.

As in the tutorial, the size of the instance should be t2.micro which is the free tier version.

3.2.2 VPC and Subnet Details

Move on to "Step 3: Configure Instance Details" and in the row labeled "Network", make sure the default (or the VPC you created) is selected. Additionally, for "Auto-Assign Public IP", change the option to "Enable".



Network vpc-0635e621ce641960f | Bethe-test

No default VPC found. Create a new default VPC.

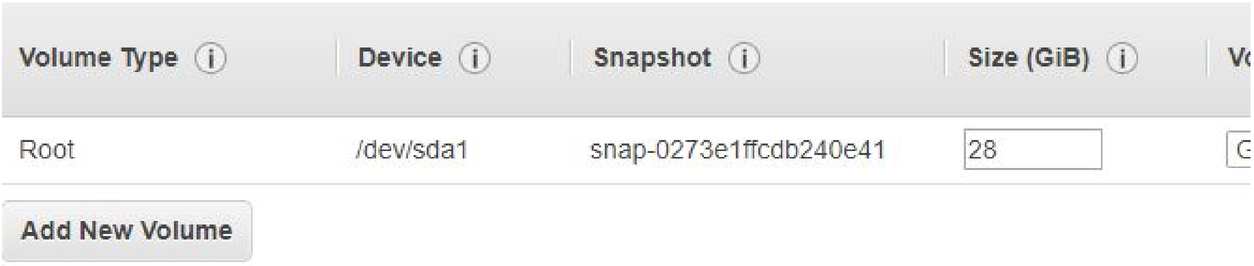
Subnet subnet-092a7211ca4264473 | Bethe test private 1 | ι

250 IP Addresses available

Auto-assign Public IP Enable

3.2.3 Storage

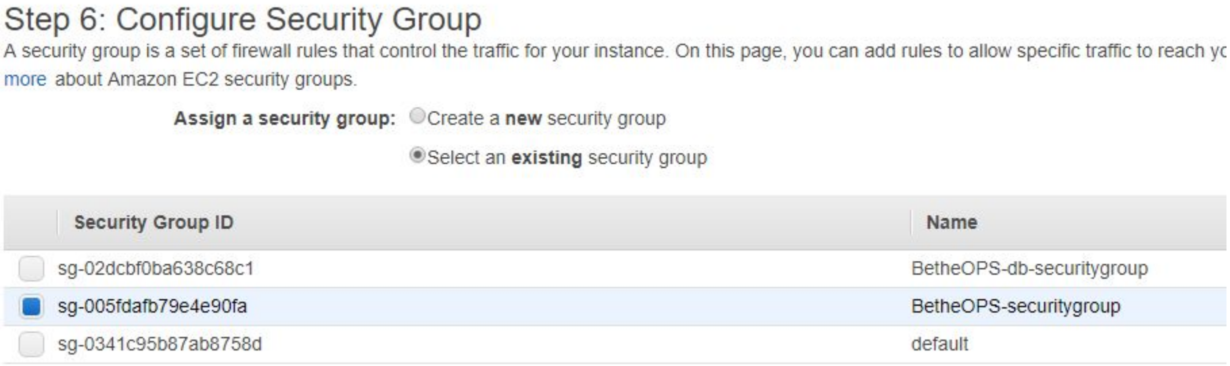
Next, move onto "Step 4: Add Storage" and change the "Size' field to be 20-30 GiB. Amazon allows for up to 30 GiB of free EBS storage which is what the EC2 instance uses.



Volume Type	Device	Snapshot	Size (GiB)	Vo
Root	/dev/sda1	snap-0273e1ffcdb240e41	28	G

3.2.4 Security Group Configuration

Next, skip “Step 5: Add Tags” and move on to “Step 6: Configure Security Group.” Click the button that says “Select an Existing Security Group” and select the security group that was created above for use with the server.



Step 6: Configure Security Group

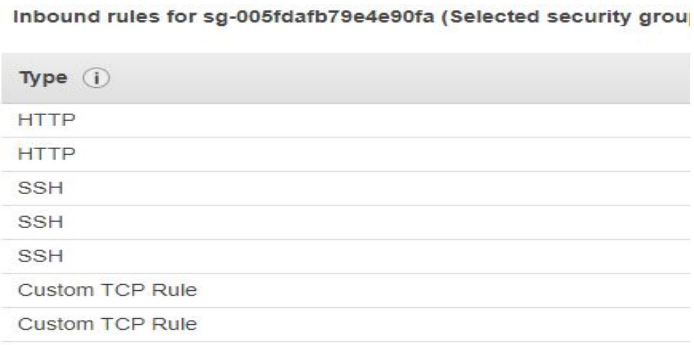
A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach yc more about Amazon EC2 security groups.

Assign a security group: ○ Create a **new** security group

● Select an **existing** security group

	Security Group ID	Name
□	sg-02dcbf0ba638c68c1	BetheOPS-db-securitygroup
☑	sg-005fdafb79e4e90fa	BetheOPS-securitygroup
□	sg-0341c95b87ab8758d	default

As a reminder, this security group has inbound rules for HTTP and SSH.

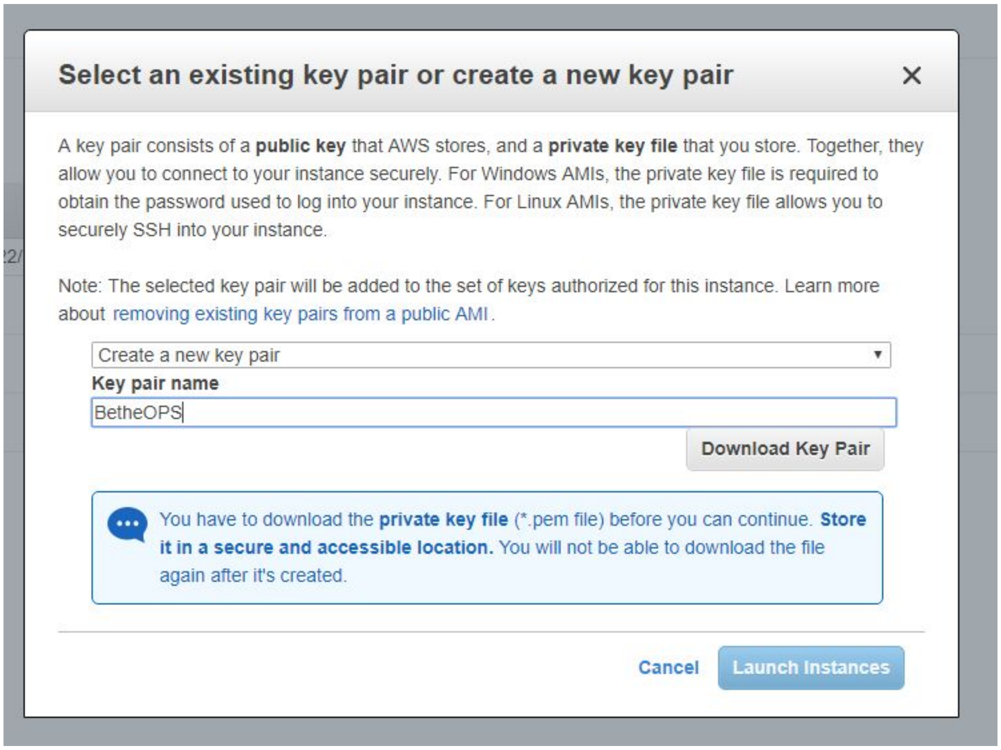


Inbound rules for sg-005fdafb79e4e90fa (Selected security grou

Type
HTTP
HTTP
SSH
SSH
SSH
Custom TCP Rule
Custom TCP Rule

3.2.5 Launching and Keys

Finally, select “Review and Launch”. A pop-up window asking you to create or use a key pair for secure connection should show up as shown below.



Select an existing key pair or create a new key pair

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about removing existing key pairs from a public AMI.

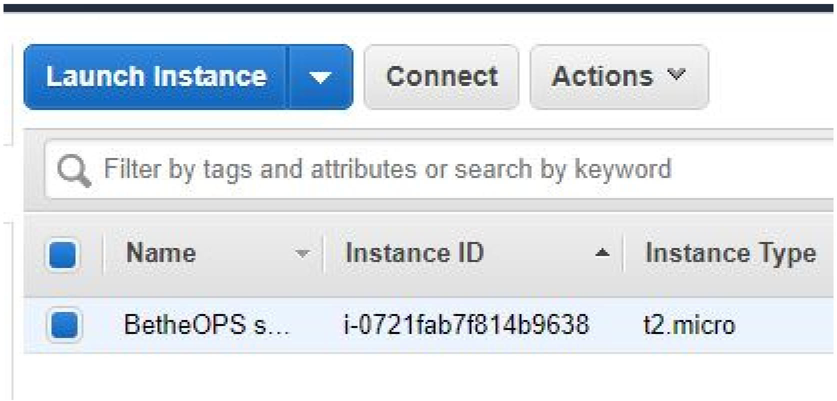
Create a new key pair

Key pair name

BetheOPS

You have to download the **private key file** (*.pem file) before you can continue. **Store it in a secure and accessible location.** You will not be able to download the file again after it's created.

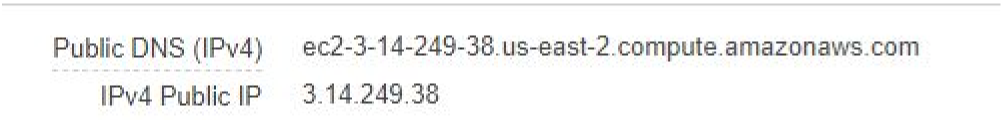
Create a new key pair, download it, and store it in a secure location as this will be used for connecting to the EC2 instance later. Click “Launch Instance” and after a while, there should be a new instance running in the “Instances” tab.



Name	Instance ID	Instance Type
BetheOPS s...	i-0721fab7f814b9638	t2.micro

3.2.6 Attaching an Elastic IP

Clicking on the instance will show additional details at the bottom of the screen including the instance ID, VPC ID, etc. The object of most interest is the "Public DNS" and "IPv4 Public IP".



Public DNS (IPv4) ec2-3-14-249-38.us-east-2.compute.amazonaws.com

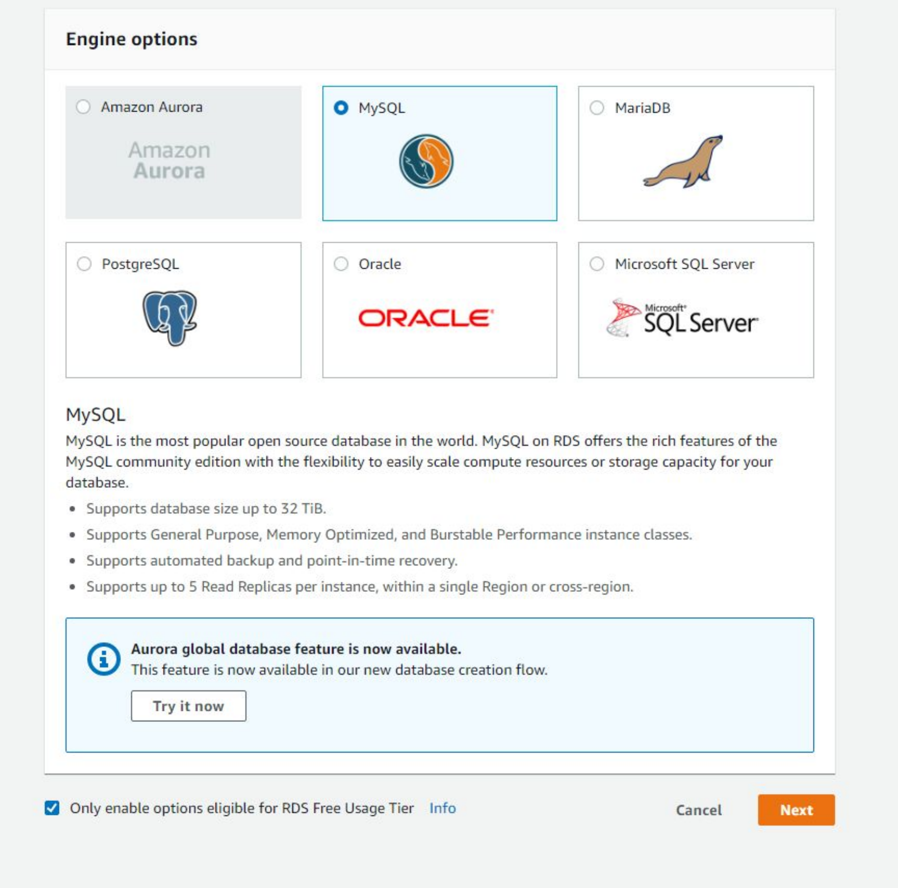
IPv4 Public IP 3.14.249.38

Typing either of these into the address bar of a browser is how any app running on this EC2 instance is accessed. However, these addresses will change any time the EC2 instance needs to reboot. Thus, what is known as an elastic IP, which will not change across reboots, should be attached.

Follow the instructions under "Allocating an Elastic IP address" here.

3.3 Creating the RDS Instance

To create the RDS instance, follow section 2.2 "Creating RDS Instance" in the setup tutorial. Make sure to select the option labeled "Only enable options eligible for RDS Free Usage Tier" at the bottom of the page when selecting the engine as shown below.



Engine options

○ Amazon Aurora

● MySQL

○ MariaDB

○ PostgreSQL

○ Oracle

○ Microsoft SQL Server

MySQL

MySQL is the most popular open source database in the world. MySQL on RDS offers the rich features of the MySQL community edition with the flexibility to easily scale compute resources or storage capacity for your database.

- Supports database size up to 32 TiB.
- Supports General Purpose, Memory Optimized, and Burstable Performance instance classes.
- Supports automated backup and point-in-time recovery.
- Supports up to 5 Read Replicas per instance, within a single Region or cross-region.

Aurora global database feature is now available.
This feature is now available in our new database creation flow.

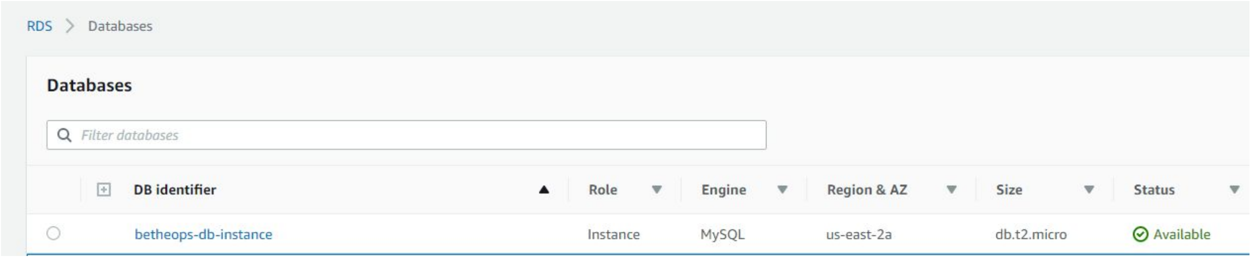
☑ Only enable options eligible for RDS Free Usage Tier

After clicking "Next", scroll down to the section labeled "Settings" and fill out the fields as follows:

- DB instance identifier: betheops-db-instance
- Master Username: StudentTech
- Master Password: BetheOPS

Move on to the next page. Under "Network and Security", make sure the VPC is the default one (or the one you created), which should be the same VPC used when creating the EC2 instance. For "Public Accessibility", select "Yes." For the "VPC Security Group", select "Choose existing VPC security group" and select the security group created for the database.

Leave the rest of the settings default and create the database. After a few minutes, the name of the database should show up as available in the RDS console.



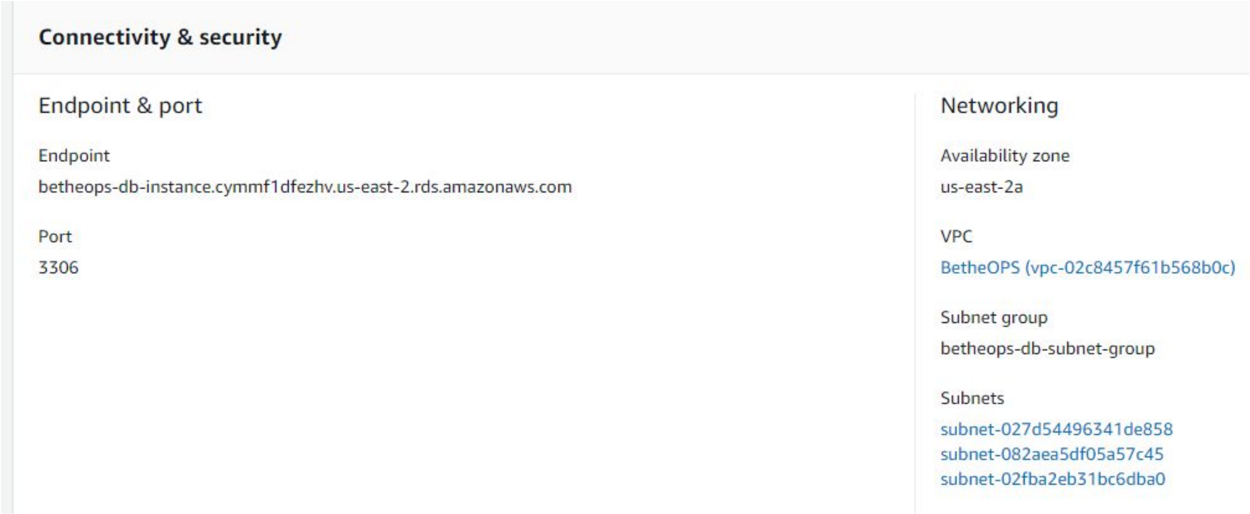
RDS > Databases

Databases

Filter databases

	DB identifier	Role	Engine	Region & AZ	Size	Status
○	betheops-db-instance	Instance	MySQL	us-east-2a	db.t2.micro	Available

Clicking on the name of the database gives us additional important information. Specifically, keep a note of the endpoint of the database as shown below.



Connectivity & security

Endpoint & port

Endpoint
betheops-db-instance.cymmf1dfezhv.us-east-2.rds.amazonaws.com

Port
3306

Networking

Availability zone
us-east-2a

VPC
BetheOPS (vpc-02c8457f61b568b0c)

Subnet group
betheops-db-subnet-group

Subnets
subnet-027d54496341de858
subnet-082aea5df05a57c45
subnet-02fba2eb31bc6dba0

4 Connection

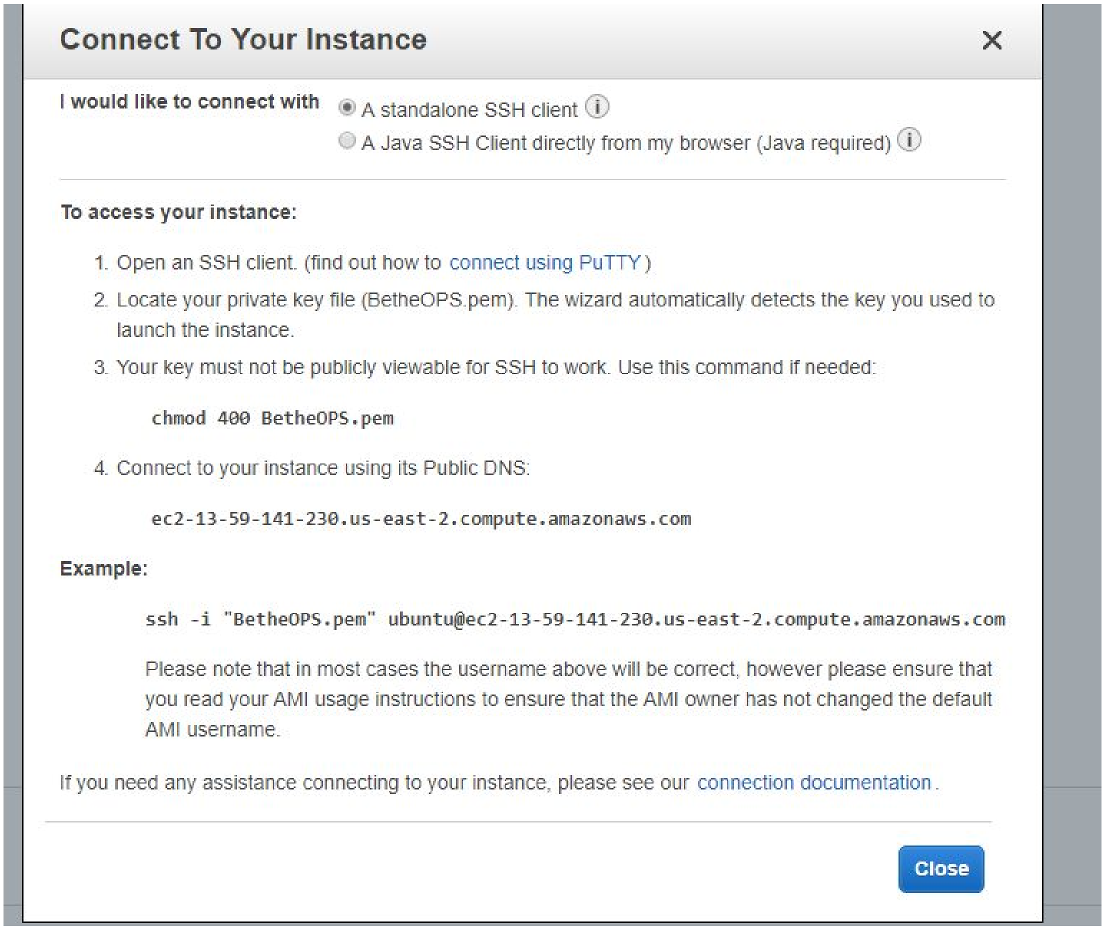
This section will detail how to connect to the EC2 instance and RDS instance in order to set up the hosting.

4.1 Connecting to EC2

Section 2 in the following tutorial has instructions for connecting to EC2 using Mac or Windows. Instructions are also written below and included in the README in the Github repository for the project.

4.1.1 Mac Users

For Mac users, head to the EC2 console and click on “Instances” on the left sidebar. Select the EC2 instance and click on the button labeled “Connect” near the top. A pop-up should show up similar to the picture below.



Connect To Your Instance

I would like to connect with ● A standalone SSH client ○ A Java SSH Client directly from my browser (Java required)

To access your instance:

1. Open an SSH client. (find out how to connect using PuTTY)
2. Locate your private key file (BetheOPS.pem). The wizard automatically detects the key you used to launch the instance.
3. Your key must not be publicly viewable for SSH to work. Use this command if needed:

```
chmod 400 BetheOPS.pem
```

4. Connect to your instance using its Public DNS:

```
ec2-13-59-141-230.us-east-2.compute.amazonaws.com
```

Example:

```
ssh -i "BetheOPS.pem" ubuntu@ec2-13-59-141-230.us-east-2.compute.amazonaws.com
```

Please note that in most cases the username above will be correct, however please ensure that you read your AMI usage instructions to ensure that the AMI owner has not changed the default AMI username.

If you need any assistance connecting to your instance, please see our connection documentation.

Use this information to connect over the terminal.

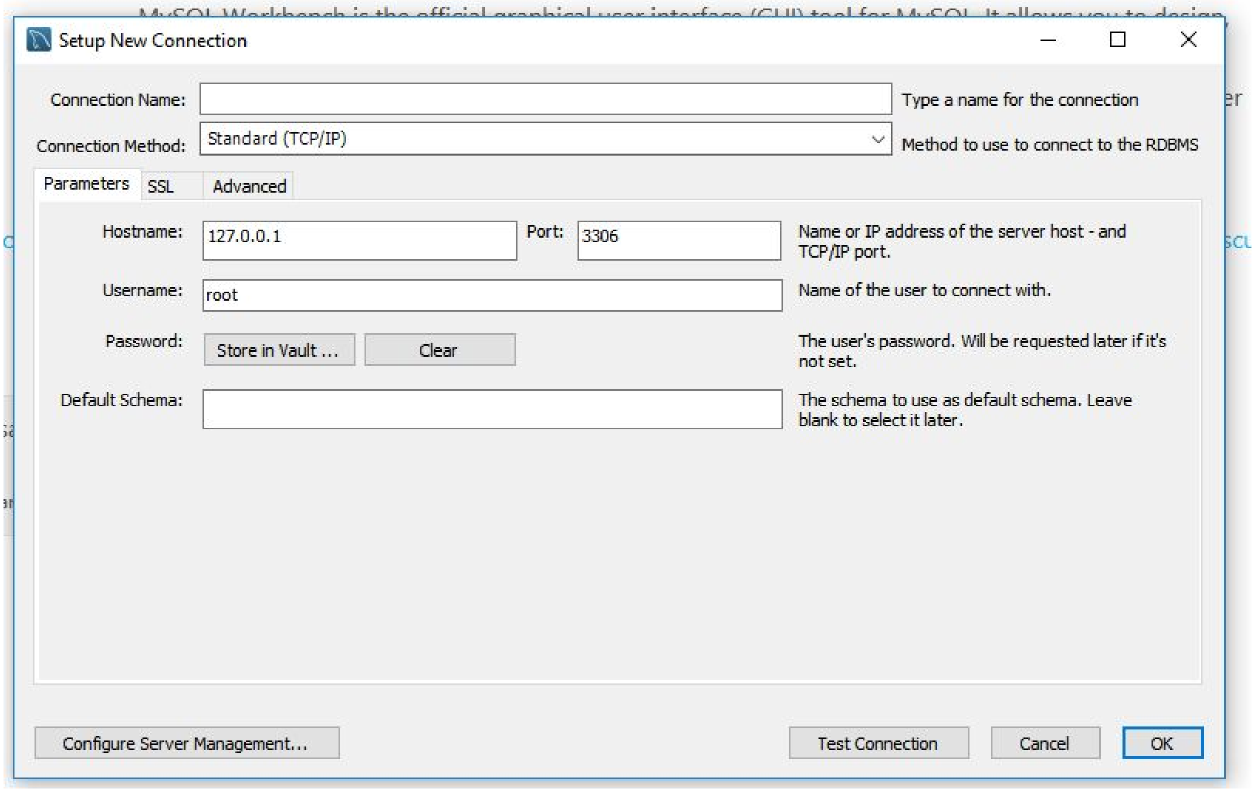
4.1.2 Windows Users

Connecting to EC2 using Windows is a bit more involved as the Windows terminal does not natively support SSH as with Macs. Thus, Windows users need to use PuTTY. The following tutorial explains this well.

4.2 Connecting to RDS

Connecting to RDS first requires installing MySQL Workbench. To do this, download the MySQL installer here. With the installer, install MySQL Workbench as well as MySQL Server.

Once MySQL Workbench is installed, open it up and create a new MySQL connection as by clicking the small plus icon next to “MySQL Connection”. The following window should pop up.



Setup New Connection

Connection Name: Type a name for the connection

Connection Method: Standard (TCP/IP) Method to use to connect to the RDBMS

Hostname: 127.0.0.1 Port: 3306 Name or IP address of the server host - and TCP/IP port.

Username: root Name of the user to connect with.

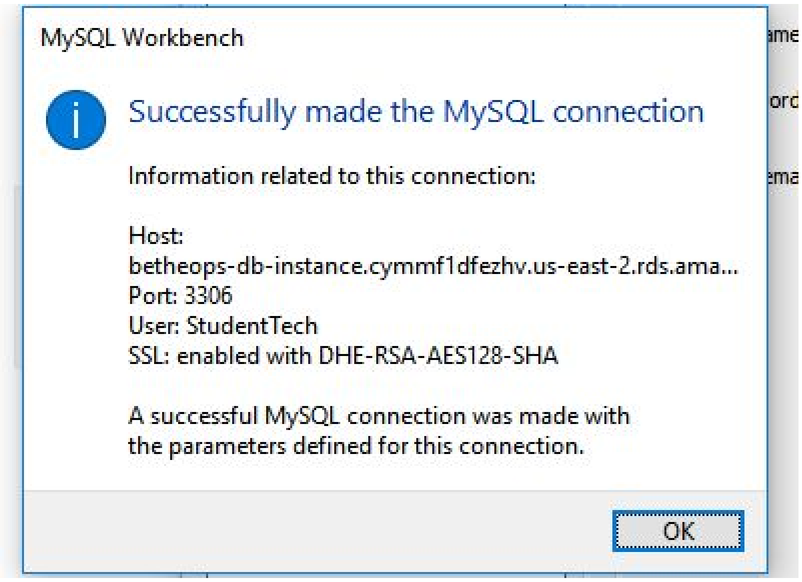
Password: The user's password. Will be requested later if it's not set.

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

Next, follow the following steps:

- Label the connection name BetheOPS.
- For the "Hostname", set this to be the endpoint of the RDS instance we created earlier.
- Put the Master Username of the RDS instance in the "Username" field
- Click on "Store in Vault" and type in the master password of the RDS instance

Click on "Test Connection" and if all goes well, a pop-up will indicate a successful connection.



MySQL Workbench

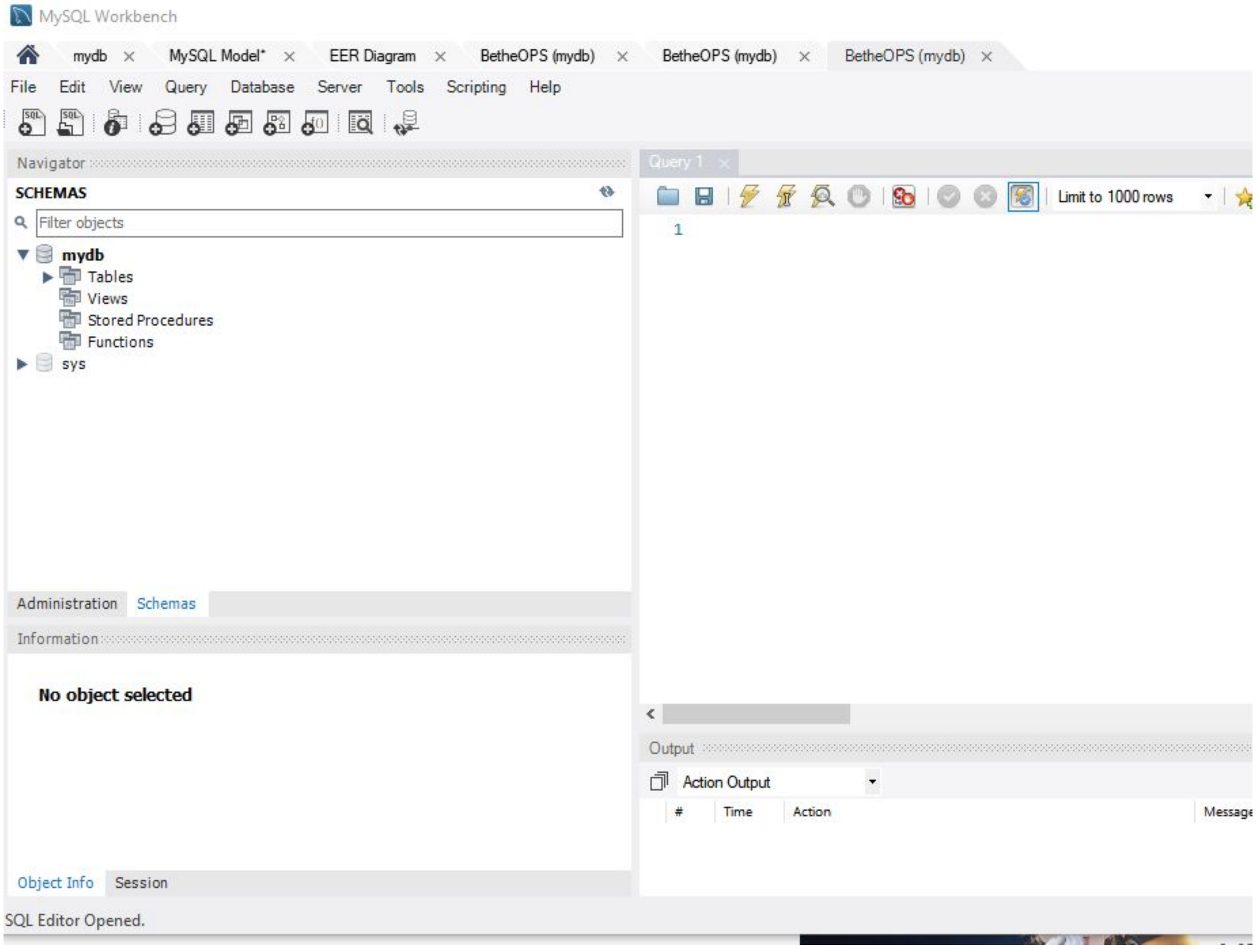
Successfully made the MySQL connection

Information related to this connection:

Host:
betheops-db-instance.cymmf1dfezhv.us-east-2.rds.ama...
Port: 3306
User: StudentTech
SSL: enabled with DHE-RSA-AES128-SHA

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

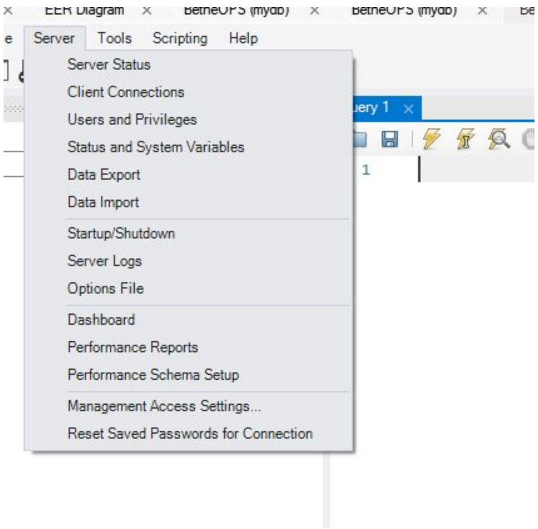
Close the windows and click on the new button labeeld “BetheOPS” under “MySQL Connections”. This should open up a new view of MySQL Workbench, which will allow you to create schemas and query databases stored in the RDS instance.



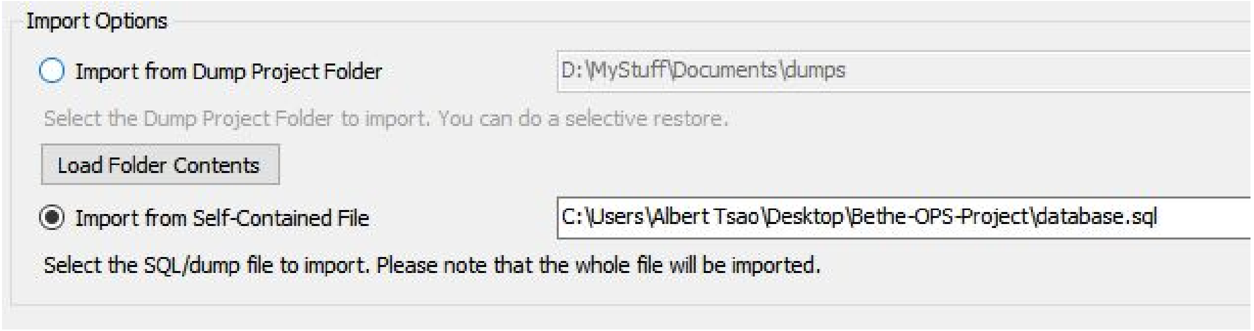
5 Hosting the Database on MySQL Workbench

With MySQL Workbench, creating a database on the RDS instance is incredibly simple. In the Github repository for this project, there should be a file named “database.sql”. Download this and store it on your computer. Next, connect to the RDS instance as above.

Once connected, go to the top and click on the dropdown menu labeled “Server” and click on the option labeled “Data Import.”



This should bring up a new screen. In the box labeled "Import Options", click the button labeled "Import from Self-Contained File" and set the file to be "database.sql."



Import Options

○ Import from Dump Project Folder D:\MyStuff\Documents\dumps

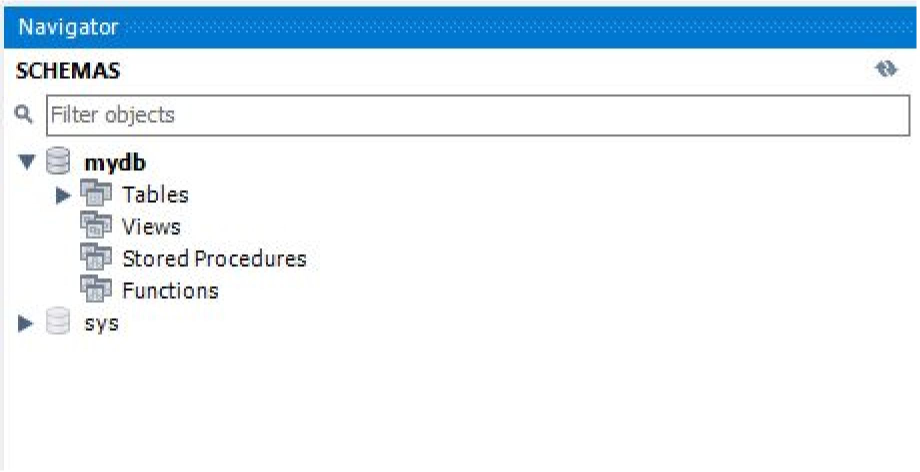
Select the Dump Project Folder to import. You can do a selective restore.

Load Folder Contents

● Import from Self-Contained File C:\Users\Albert Tsao\Desktop\Bethe-OPS-Project\database.sql

Select the SQL/dump file to import. Please note that the whole file will be imported.

Click on "Start Import" on the bottom right and the database should be uploaded to the RDS instance. To check this, go to the "navigator" sidebar on the left and click the refresh button and if everything worked, a database schema called "mydb" should show up.



The database has been successfully hosted on RDS!

6 Hosting the Bethe OPS Web Server on EC2

First, SSH into the EC2 instance using the methods discussed above. Next, clone the repository for Bethe OPS. Once this is done, enter the directory of the project. Before running the actual server, a few things need to be set up first.

6.1 Installing Prerequisites and Common Packages

Run the following command: *sudo apt-get install*.

6.2 Installing Node.js and System Dependencies

The backend of the server uses Node.js so this needs to be installed on our EC2 instance. To do this, run the following commands:

- *curl -o- https://raw.githubusercontent.com/creationix/nvm/v0.32.1/install.sh | bash*
- *source ~/.bashrc*
- *nvm install node*

6.3 Installing the Packages for the Server

Make sure that you are in the directory of the “Bethe-OPS-Project” Github repository. Next run the command: *npm install*. This will install all the necessary modules required by the server.

6.4 Installing and Configuring Nginx

Nginx is what redirects traffic from the IP address of the EC2 instance to the web server.

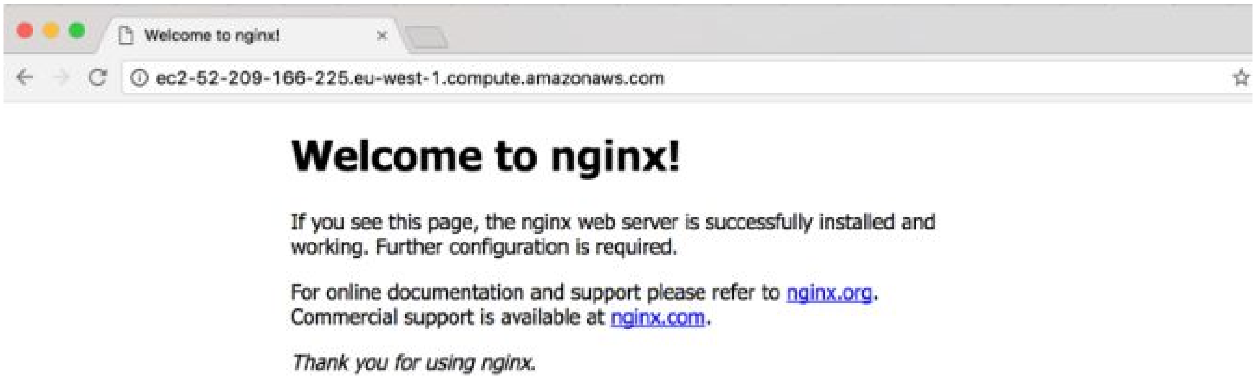
Follow the following tutorial to install Nginx.

There is one thing that is different in the tutorial above. In the tutorial, when creating the config file, the tutorial sets it to the following:

```
server {
 listen 80;
 server_name tutorial;
 location / {
   proxy_set_header  X-Real-IP  $remote_addr;
   proxy_set_header  Host       $http_host;
   proxy_pass        http://127.0.0.1:3000;
 }
}
```

Change "tutorial" to be the public DNS URL of the EC2 instance and change the highlighted numbers in the proxy_pass to be the private IP address of the EC2 instance. This can be found in the dashboard of AWS EC2.

As the tutorial mentions, to test that Nginx is working properly, type the public DNS URL of the EC2 instance into a browser. If Nginx installed correct, the default Nginx page will be displayed.



ec2-52-209-166-225.eu-west-1.compute.amazonaws.com

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

Default nginx page

6.5 Run the App

With Node.js and Nginx installed, the app is ready to run. To run the server, type the command *app.js*.

Next, type the public DNS of the AWS EC2 instance into the browser and the app should run.

6.6 Run the App Forever (Optional)

To keep the app running after closing the SSH connection, PM2 is used. The same link for installing Nginx also has details about installing and using PM2.

To keep the app running forever with PM2, do the following (instructions are also included in the README):

1. Connect to the EC2 instance as described in Section 4.1.
2. Navigate to the project folder by typing “cd Bethe/Bethe-OPS-Project”.
3. Run the command *pm2 start app.js*.

To check if the app is running, simply run the command *pm2 list*. If the app is running, it will have a status of “online”; otherwise, it will have a status of “stopped”.

To stop the app, simply run the command *pm2 stop app*.