ONLYFACTORIES DOCUMENTATION

V 1.0

Harris, Justin (harr1433@vandals.idaho.edu)
Lawson, Keller (laws1689@vandals.uidaho.edu)
Weisel, Parker (weis8961@vandals.uidaho.edu)

Technologies

Front-End Hosting: AWS Amplify

Front-End: React

Back-End Hosting: AWS EC2 Ubuntu Virtual Machine

Back-End: Node.js

MQTT Broker: Eclipse Mosquitto

Database: AWS RDS MySQL

Source Control: GitHub

Project Management: Trello

Front-End Hosting

The OnlyFactories web application is being hosted on AWS Amplify linked to a GitHub repository. This allows for the project to auto-build when the main or master branch is updated. To set up this build flow, follow the instructions below.

1. Select AWS Amplify from AWS services.
2. Select New app -> Host web app
3. Select the site that hosts your source control.
    a. For current project, select GitHub
4. Authorize AWS to have Read-Only access to your account.
5. Select install AWS Amplify
6. Install and Authorize AWS on the OnlyFactories repository containing only the React code.
7. Select recently updated repositories -> {your_username}/OnlyFactories
8. Select Branch -> Main
9. Leave default Build and Test settings.
10. Click Save and deploy.

Back-End Hosting

The Back-end for OnlyFactories is hosted on an AWS EC2 Ubuntu Virtual Machine(VM). This VM hosts the RESTful API and MQTT Broker. To set up the Ubuntu virtual machine, follow the instructions below.

1. Select EC2 from AWS services.
2. Select instances from left side navigation bar.
3. Select Launch Instance
4. Select Ubuntu from Amazon Machine Images (AMI)
    a. For capstone of Fall 2021 – Spring 2022, Ubuntu 20.04 LTS was used
5. Select Instance Type

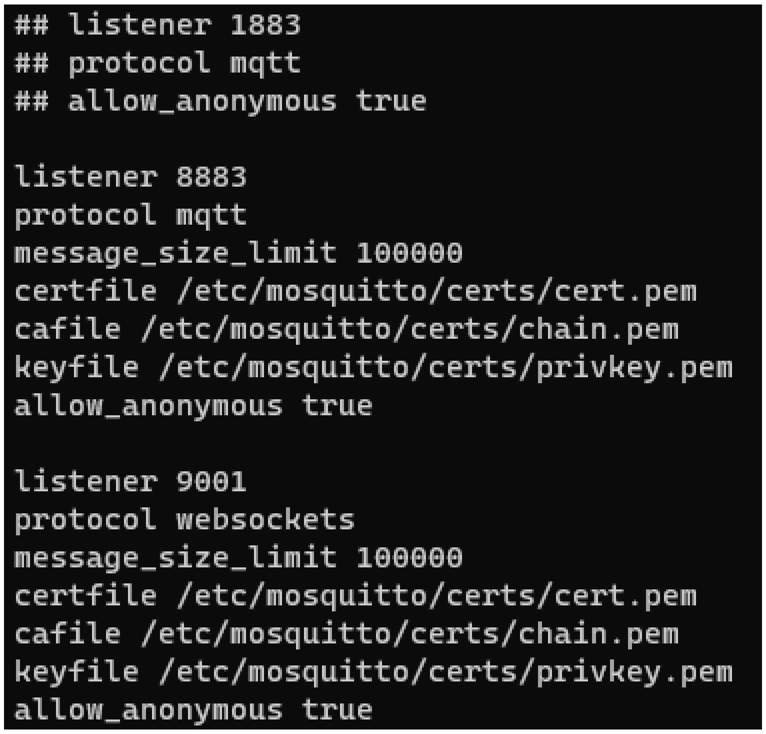
    a. For capstone of Fall 2021 – Spring 2022, t2.micro was used
6. Create a new key pair
    a. Save keypair locally or inside a folder called “private” inside git repo.
    b. Note: anything saved in the private folder will not be added with git add. This is a security measure to keep sensitive data private.
7. Leave the rest of the options as default.
8. Launch Instance

To Access the VM, navigate to the instance inside the EC2 service. Select connect from the options and use one of the methods listed.

Express Server

To host the express server for the RESTful API, clone the OnlyFactories_Express_EC2 repository onto the server. Be sure to install pm2 using “sudo apt install pm2” to be able to run the server all the time. After it is installed, start the server and MQTT_Listener using “sudo pm2 start server.js MQTT_Listener.js”.

MQTT Broker

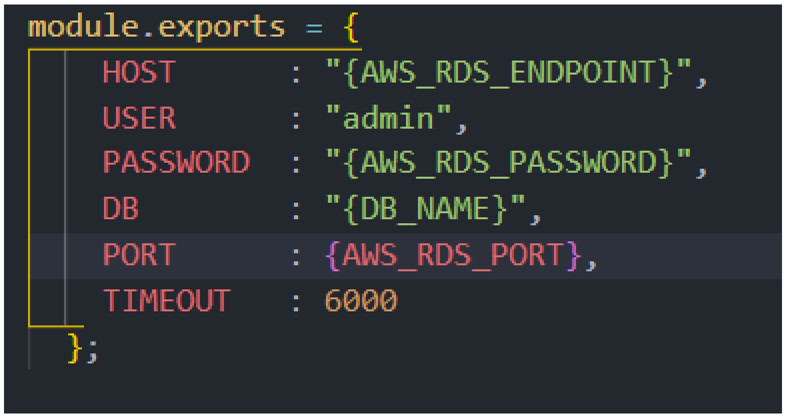


```
## listener 1883
## protocol mqtt
## allow_anonymous true

listener 8883
protocol mqtt
message_size_limit 100000
certfile /etc/mosquitto/certs/cert.pem
cafile /etc/mosquitto/certs/chain.pem
keyfile /etc/mosquitto/certs/privkey.pem
allow_anonymous true

listener 9001
protocol websockets
message_size_limit 100000
certfile /etc/mosquitto/certs/cert.pem
cafile /etc/mosquitto/certs/chain.pem
keyfile /etc/mosquitto/certs/privkey.pem
allow_anonymous true
```

Figure 1: MQTT Config file

Database: AWS RDS MySQL

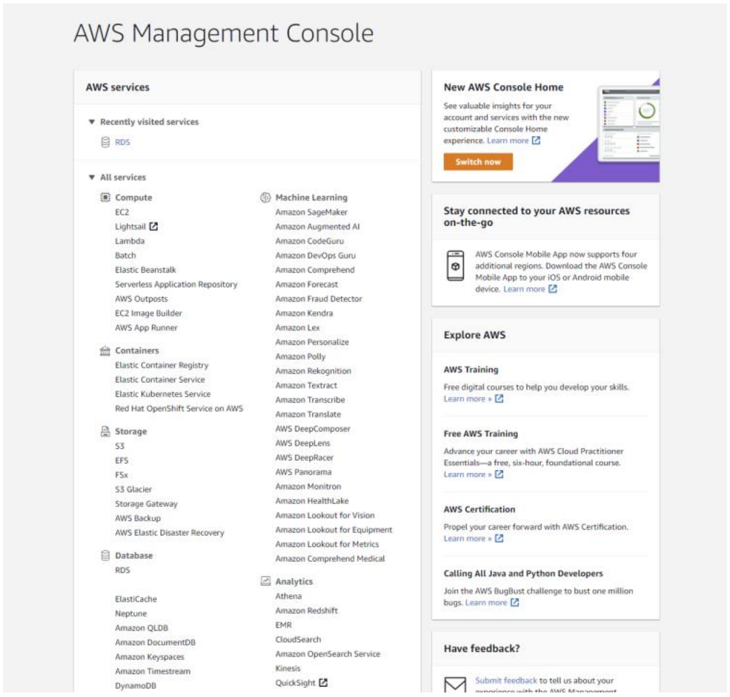
To establish a connection between the AWS RDS MySQL Database and the OnlyFactories back-end web application parameters in the db.config.js file need to be updated.



```
module.exports = {
    HOST      : "{AWS_RDS_ENDPOINT}",
    USER      : "admin",
    PASSWORD  : "{AWS_RDS_PASSWORD}",
    DB        : "{DB_NAME}",
    PORT      : {AWS_RDS_PORT},
    TIMEOUT   : 6000
  };
```

Figure 2: OnlyFactories back-end db.config.js

In this section HOST should be replaced with the AWS RDS Endpoint found in database instance. From the AWS Management Console, find the "RDS" option either under "Recently visited services" or the "All Services" drop down menu.



New AWS Console Home

See valuable insights for your account and services with the new customizable Console Home experience. Learn more

Stay connected to your AWS resources on-the-go

AWS Console Mobile App now supports four additional regions. Download the AWS Console Mobile App to your iOS or Android mobile device. Learn more

Explore AWS

AWS Training

Free digital courses to help you develop your skills. Learn more »

Free AWS Training

Advance your career with AWS Cloud Practitioner Essentials—a free, six-hour, foundational course. Learn more »

AWS Certification

Propel your career forward with AWS Certification. Learn more »

Calling All Java and Python Developers

Join the AWS BugBust challenge to bust one million bugs. Learn more

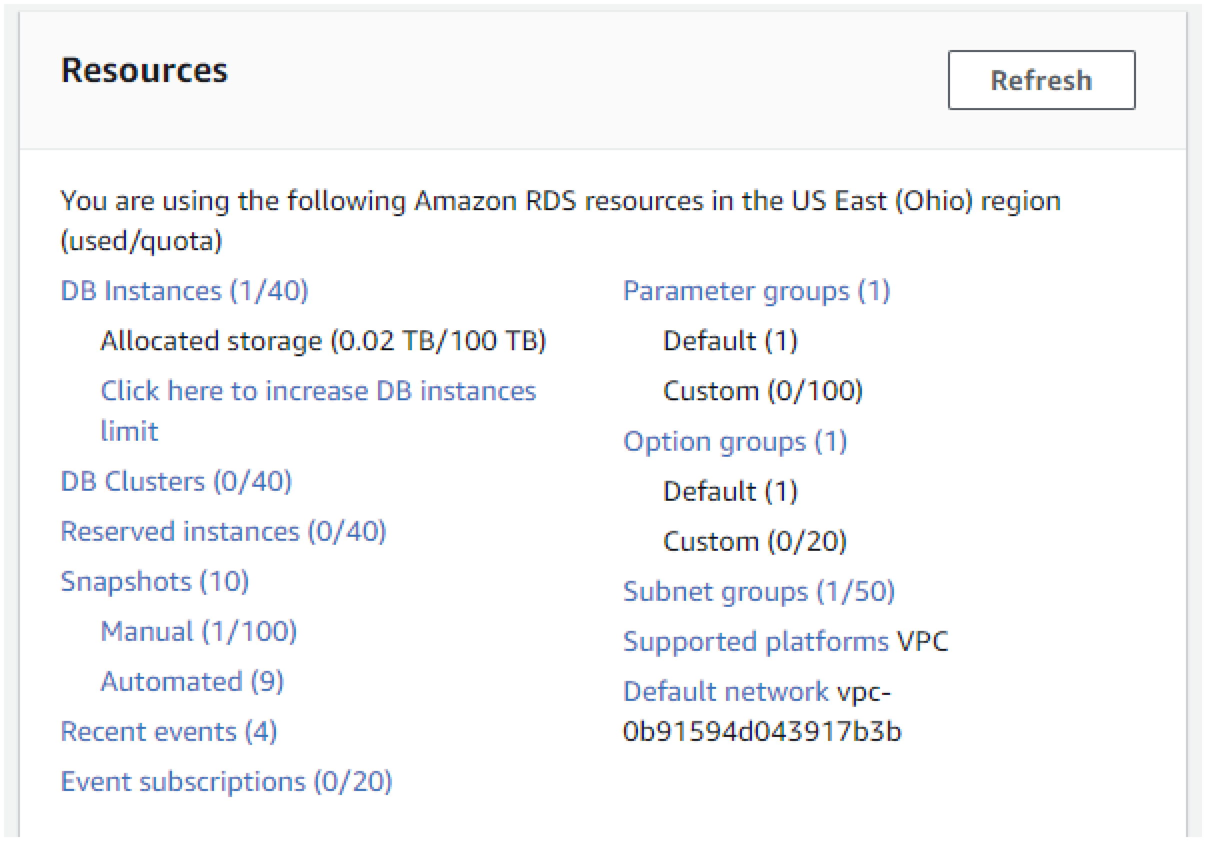
Have feedback?

Submit feedback to tell us about your

Figure 3: AWS Management Console

If there is not a database already set up or a new database needs to be set up, start by clicking "Create database" towards the top of the screen. Additional guidance for this process can be found at the following link: https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP_Tutorials.WebServerDB.CreateDBInstance.html.

Once the database instance is created or if the database already exists, click on "DB Instances" under the "Resources" panel. Next, click on the name of the database under "DB identifier" on the following page.

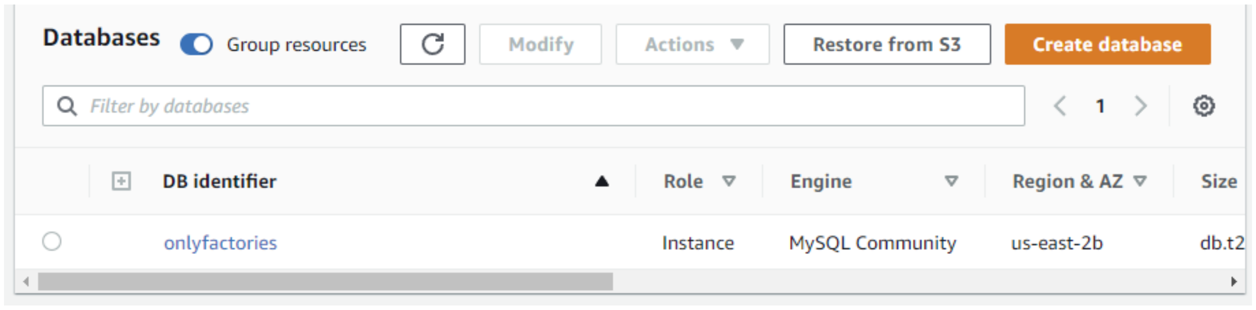


You are using the following Amazon RDS resources in the US East (Ohio) region (used/quota)

DB Instances (1/40)
Allocated storage (0.02 TB/100 TB)
Click here to increase DB instances limit
DB Clusters (0/40)
Reserved instances (0/40)
Snapshots (10)
Manual (1/100)
Automated (9)
Recent events (4)
Event subscriptions (0/20)

Parameter groups (1)
Default (1)
Custom (0/100)
Option groups (1)
Default (1)
Custom (0/20)
Subnet groups (1/50)
Supported platforms VPC
Default network vpc-0b91594d043917b3b

Figure 4: AWS RDS Resources Panel



	DB identifier	Role	Engine	Region & AZ	Size
	onlyfactories	Instance	MySQL Community	us-east-2b	db.t2

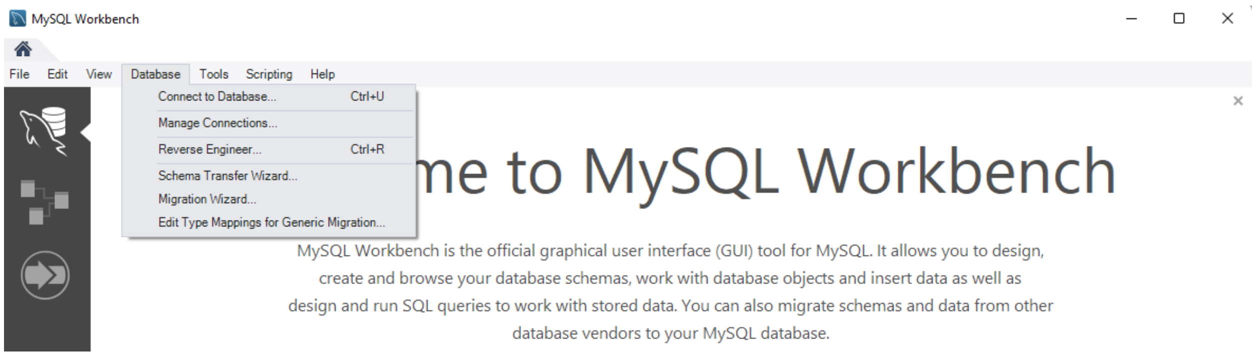
Figure 5: AWS Databases Table

This will lead to the database summary page for the instance selected. In this section, the database endpoint and port can be found. Replace the "HOST" section in the db.config.js file with the "Endpoint" URL under "Connectivity & Security" and the "PORT" section with the "Port" number listed in this section. The "USER" in the db.config.js file should be left as "admin", and the "PASSWORD" section needs to be replaced by the database authentication password. More information about the authentication password can be found at the following link: https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/database-authentication.html. Finally, the "DB" section of the db.config.js file needs to be replaced with the database name.

Additionally, a team taking over this project may want to connect to the database using a third-party tool to verify database contents or test the validity of SQL queries. Our team decided to establish a connection to the database through MySQL Workbench to verify the contents of the database tables and test the validity of SQL queries.

To start setting up MySQL Workbench with a connection to the AWS MySQL RDS database, use the following link to download the appropriate installer: https://dev.mysql.com/downloads/.

Once downloading and installation is complete open MySQL Workbench. Next, select the "Database" option from the top menu bar, followed by selecting "Connect to Database".

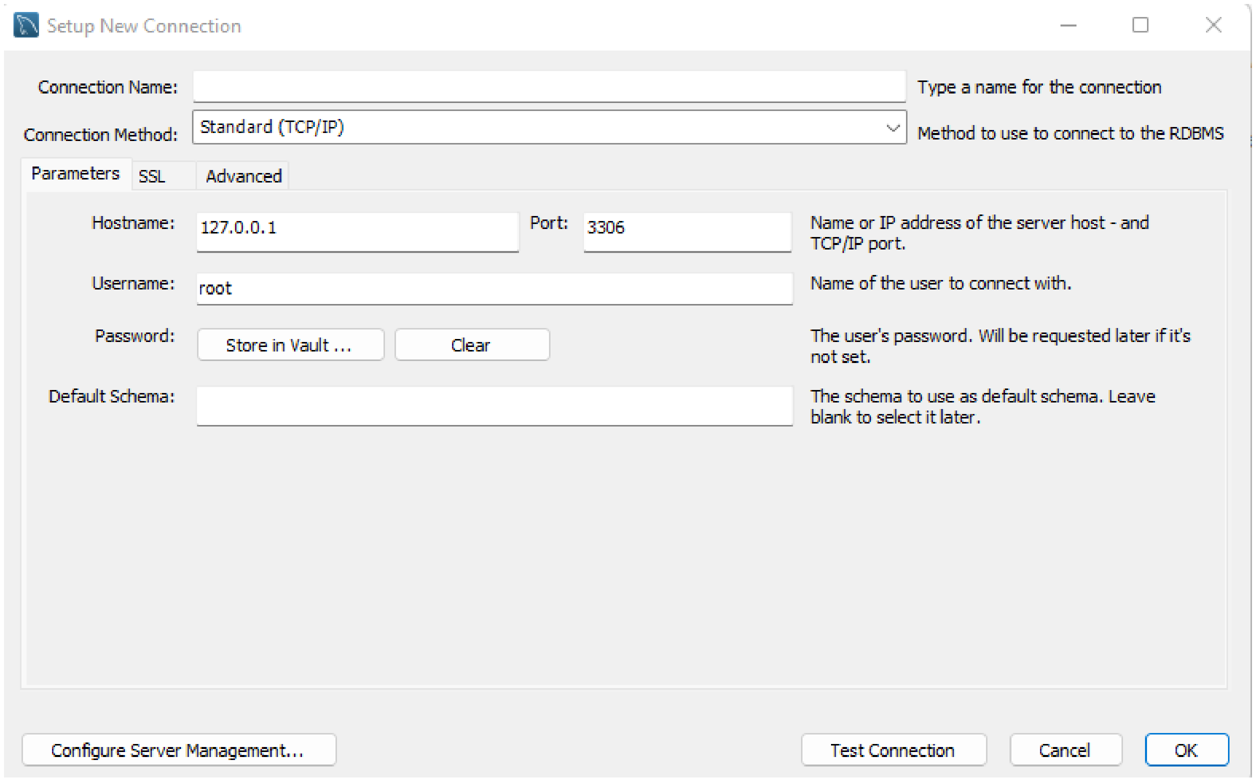


MySQL Workbench
File
Edit
View
Database
Tools
Scripting
Help
Connect to Database... Ctrl+U
Manage Connections...
Reverse Engineer... Ctrl+R
Schema Transfer Wizard...
Migration Wizard...
Edit Type Mappings for Generic Migration...
me to MySQL Workbench

MySQL Workbench is the official graphical user interface (GUI) tool for MySQL. It allows you to design, create and browse your database schemas, work with database objects and insert data as well as design and run SQL queries to work with stored data. You can also migrate schemas and data from other database vendors to your MySQL database.

Figure 6: MySQL Workbench Landing

This will bring up a window to enter the database credentials to establish the connection. The “Connection Name” can be anything, but it is recommended to use a naming scheme that allows the connection to be easily identified from the main page of MySQL Workbench. The “Connection Method” section should be set to “Standard (TCP/IP)”. The “Hostname”, “Port”, “Username”, and “Password” should be filled in with the same credentials used in the db.config.js file. To set a password at this stage, select the “Store in Vault…” option. Otherwise, if a password is required MySQL Workbench will request one when attempting to connect to the database instance.



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.

Figure 7: MySQL Workbench Database Connection

Finally select “OK” at the bottom right corner of the window to establish the connection.

Database Schema

The overall database schema for this project is titled “onlyfactories” and contains the following tables:

1) FactoryOrders: contains information relating to each order made by a user to the factory
2) Inventory: contains information related to the current inventory in the factory
3) CustomerAccounts: contains information related to accounts created for the OnlyFactories web app. (Note: the only account used during the project was an admin account to access reports).

4) FactoryStatus: contains information related to the current status of the factory, current job running, job queue length, and last updated information.
5) FactoryJobs: contains information regarding the current job, order, disk and status of the job that is being processed by the factory.
6) Transactions: contains information regarding the transaction (payment) for each factory order.
7) mqttMessages: contains information regarding mqtt messages sent and received, along with factory status.
8) ItemPrice: contains the customer and production cost for each disk produced by the factory.
9) Webcam: contains information about the webcam, such as status and image data received.

The exact schema for each table that makes up the OnlyFactories database can be found in the "OnlyFactories" repository under the "queries" folder, in the file titled "createTables.sql". This file is a script that can be run in a tool like MySQL Workbench after creating the initial database "onlyfactories", which will create all of the tables needed for this project.