Fundraising App - Deployment and Development Guidelines

Team 406 Not Acceptable

Mohamed Rahaman, Khang Tran, Jacob Roquemore, Natnael Tsegaselassie, Sam Smetana

Table of Contents

1 Development Instructions

Some Prerequisites:

- In order to follow this guide, it is assumed you have knowledge in various terminal programs and emulators (or terminals provided with IDE's like Visual Studio). It is also assumed that you have a basic understanding of technologies like Node.JS (and Express.JS), Angular, and MySQL. Most of the team's development was done on Linux machines in Bash based terminal emulators, with one developer using Visual Studio with a provided shell for development. Here are a few requirements and optional tools.

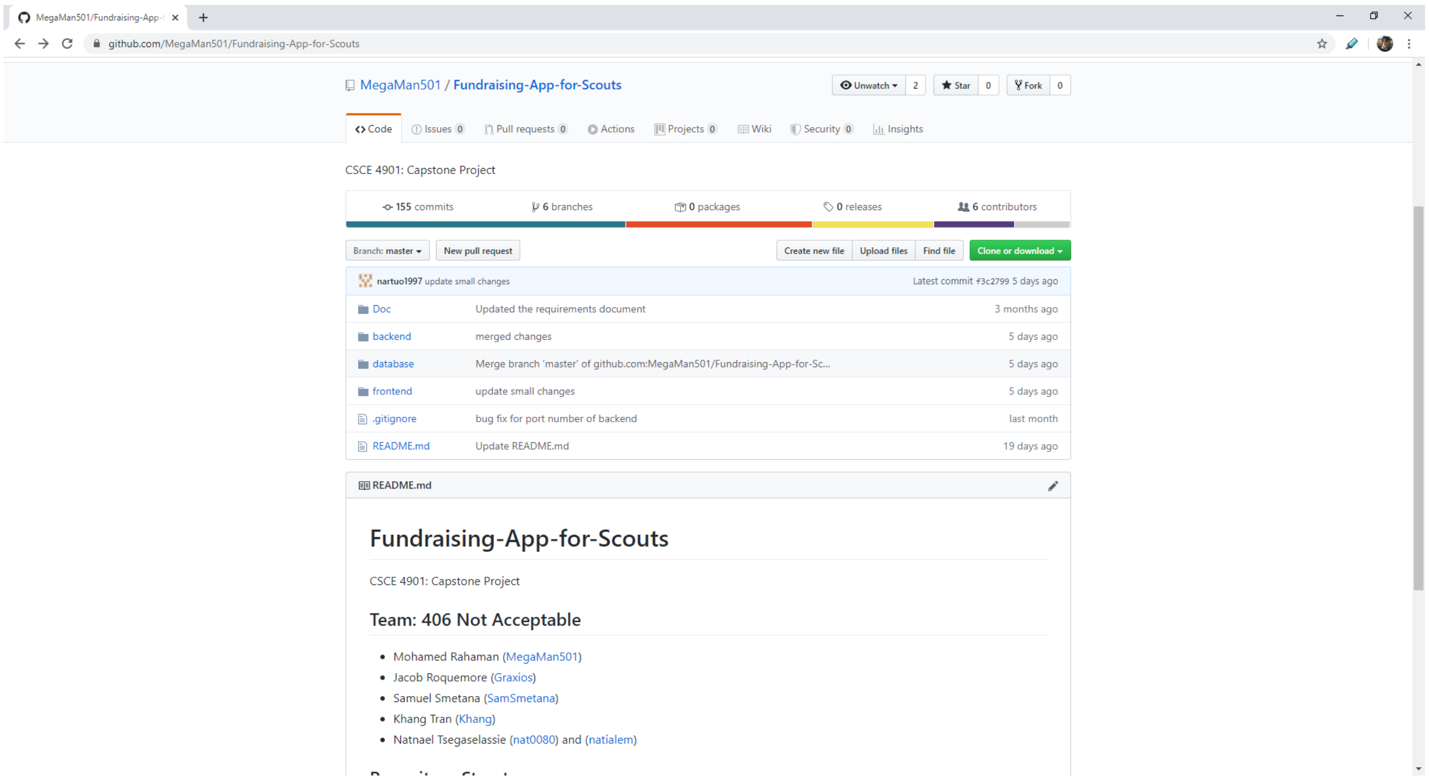
Additional requirements may be listed in other parts of the document depending on that part's contents:

- GitHub Desktop (optional)
  - https://desktop.github.com/
    - Simplified way of keeping track GitHub repositories in a manner similar to the web interface. Only has binaries for macOS and Windows though according to their website
- Git For Windows (optional)
  - https://gitforwindows.org/
    - Provides a terminal emulator through Git BASH for Windows which is great for following along with any git commands provided in this documentation. Also, if enabled during installation, allows git commands to be used in Windows based terminals like PowerShell
- Node.js + npm
  - Can be installed from many UNIX/Linux package repositories. Installers for Windows and Mac (as well as Linux binaries and source code) can be found here: https://nodejs.org/en/download/
  - (Note that npm is included with this installer for windows and mac. On some machines, npm may need to be installed separately)
- Python + pip
  - https://www.python.org/downloads/
  - (Can also be installed from many UNIX/Linux package repositories.)
  - (note that the latest versions of python already contain pip. If pip needs to be installed separately, it can be installed here: https://pip.pypa.io/en/stable/installing/)
- MySQL Workbench (optional, but highly recommended)
  - https://www.mysql.com/products/workbench/
  - Recommended tool used to form connections to either a local MySQL instance, or a cloud-based instance such as on the Google Cloud Platform
- MySQL (or MariaDB) Local Server
  - https://dev.mysql.com/downloads/
  - https://mariadb.com/kb/en/where-to-download-mariadb/
  - https://www.apachefriends.org/index.html (XAMPP; uses MariaDB, potentially easier install, especially on Windows)
- Gmail Account (optional, but highly recommended for following steps for Google Cloud Platform MySQL server as well as for simple integration with Nodemailer)

1.1 Setting Up Github Repository Online (using Fork Method)

The following is a simple set of steps (along with screenshots), which shows how to make a fork of Team 406 Not Acceptable's work and then start adding additional contributors for your group which can then be added to and edited as necessary (this guide assumes you already have a GitHub account along with :

1. Navigate to https://github.com/MegaMan501/Fundraising-App-for-Scouts if you have not already been provided a link. The page should appear like the following:



CSCE 4901: Capstone Project

nartuo1997 update small changes — Latest commit f3c2799 5 days ago

Doc	Updated the requirements document	3 months ago
backend	merged changes	5 days ago
database	Merge branch 'master' of github.com:MegaMan501/Fundraising-App-for-Sc...	5 days ago
frontend	update small changes	5 days ago
.gitignore	bug fix for port number of backend	last month
README.md	Update README.md	19 days ago

README.md

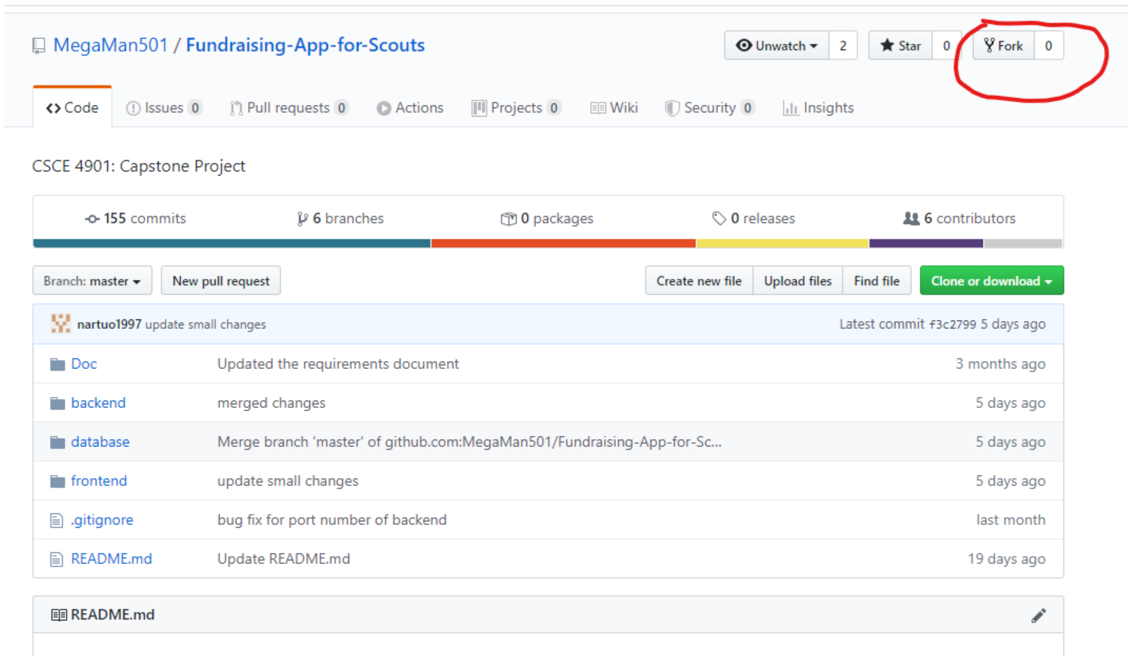
Fundraising-App-for-Scouts

CSCE 4901: Capstone Project

Team: 406 Not Acceptable

- Mohamed Rahaman (MegaMan501)
- Jacob Roquemore (Graxios)
- Samuel Smetana (SamSmetana)
- Khang Tran (Khang)
- Natnael Tsegaselassie (nat0080) and (natialem)

2. With a GitHub leader designated (the scrum/group leader, or the individual with the most git management experience), they can make a fork of our repository by selecting the Fork icon located in the top right hand corner:



MegaMan501 / Fundraising-App-for-Scouts

Unwatch 2 | Star 0 | Fork 0

Code | Issues 0 | Pull requests 0 | Actions | Projects 0 | Wiki | Security 0 | Insights

CSCE 4901: Capstone Project

155 commits | 6 branches | 0 packages | 0 releases | 6 contributors

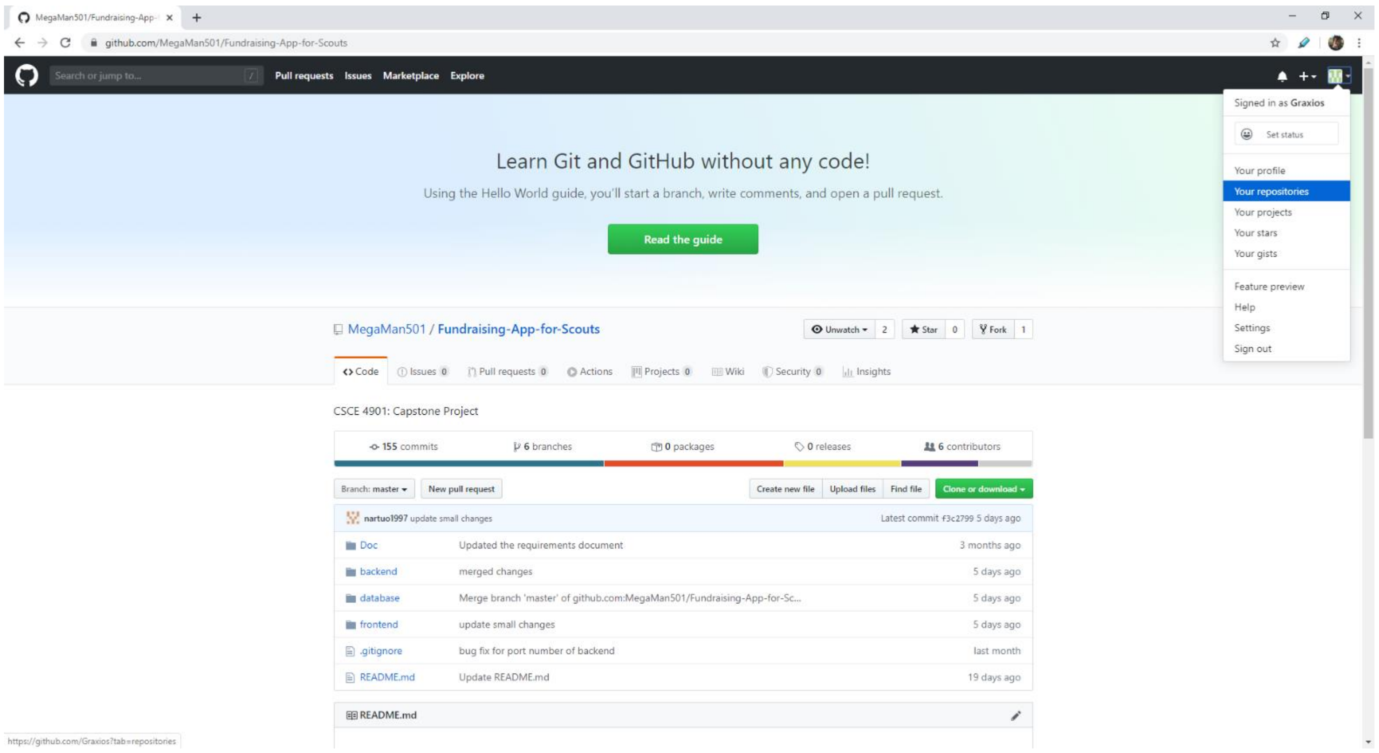
Branch: master | New pull request | Create new file | Upload files | Find file | Clone or download

nartuo1997 update small changes — Latest commit f3c2799 5 days ago

Doc	Updated the requirements document	3 months ago
backend	merged changes	5 days ago
database	Merge branch 'master' of github.com:MegaMan501/Fundraising-App-for-Sc...	5 days ago
frontend	update small changes	5 days ago
.gitignore	bug fix for port number of backend	last month
README.md	Update README.md	19 days ago

README.md

3. A copy of the repository can be found on the individual’s account. This can be seen by clicking on the currently logged in user’s profile in the top right corner and selecting, “Your repositories”, as can be seen below:



MegaMan501/Fundraising-App-

github.com/MegaMan501/Fundraising-App-for-Scouts

Search or jump to... | Pull requests | Issues | Marketplace | Explore

Signed in as Graxios

Set status

Your profile
Your repositories
Your projects
Your stars
Your gists

Feature preview
Help
Settings
Sign out

Learn Git and GitHub without any code!

Using the Hello World guide, you’ll start a branch, write comments, and open a pull request.

Read the guide

MegaMan501 / Fundraising-App-for-Scouts

Unwatch 2 | Star 0 | Fork 1

Code | Issues 0 | Pull requests 0 | Actions | Projects 0 | Wiki | Security 0 | Insights

CSCE 4901: Capstone Project

155 commits | 6 branches | 0 packages | 0 releases | 6 contributors

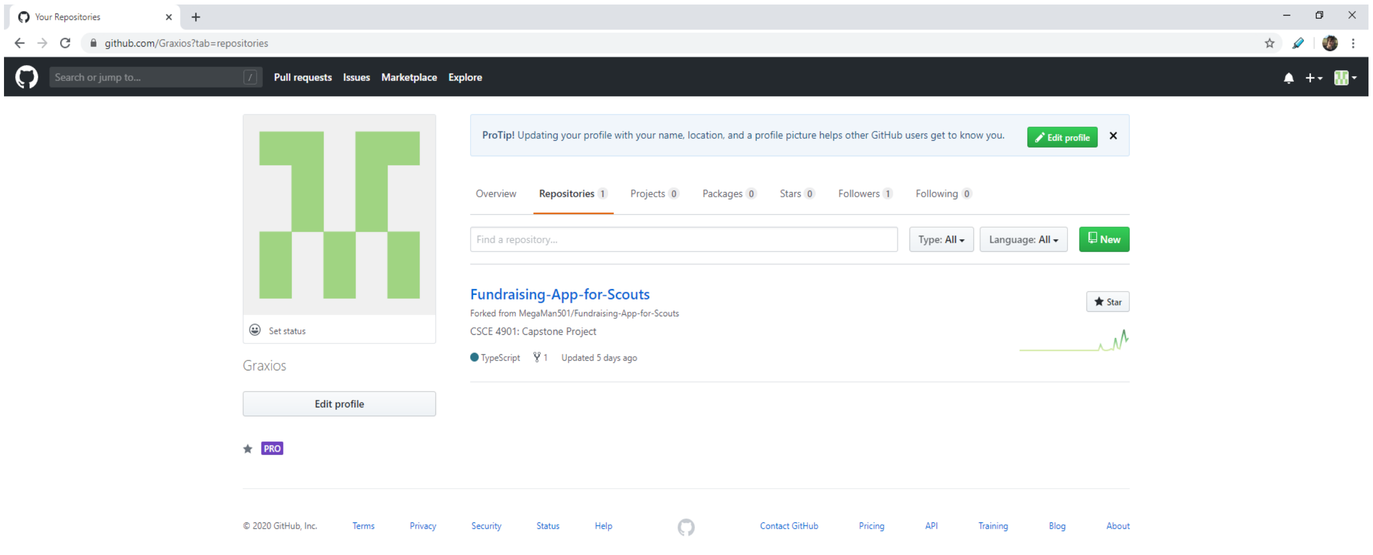
Branch: master | New pull request | Create new file | Upload files | Find file | Clone or download

nartuo1997 update small changes — Latest commit f3c2799 5 days ago

Doc	Updated the requirements document	3 months ago
backend	merged changes	5 days ago
database	Merge branch 'master' of github.com:MegaMan501/Fundraising-App-for-Sc...	5 days ago
frontend	update small changes	5 days ago
.gitignore	bug fix for port number of backend	last month
README.md	Update README.md	19 days ago

README.md

https://github.com/Graxios?tab=repositories



ProTip! Updating your profile with your name, location, and a profile picture helps other GitHub users get to know you.

Fundraising-App-for-Scouts
Forked from MegaMan501/Fundraising-App-for-Scouts
CSCE 4901: Capstone Project
TypeScript 1 Updated 5 days ago

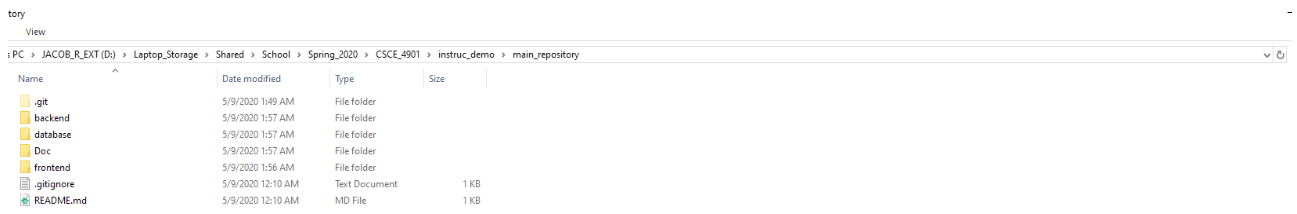
© 2020 GitHub, Inc. Terms Privacy Security Status Help Contact GitHub Pricing API Training Blog About

1.1.1 Making a Local Copy of the Repository to Push to Github (Manual Method)

The following steps detail how to make a new repository from scratch by using a ZIP file copy of the original repository (note – it is recommended that the designated repository leader perform these steps first):

1. Download a copy of our repository into a ZIP file by going to the following link and then selecting, “Download ZIP”, which will download a copy of the master branch’s content: https://github.com/MegaMan501/Fundraising-App-for-Scouts or use a zip file you were provided
2. Extract the contents of the zip folder
    a. (Note: You may run into an error with character limits on Windows or other platforms depending on how deeply nested your development directory is. It is important to make sure that the total character length of the path for all files does not exceed 255 characters on Windows. If issues still occur, it is recommended to use a tool like 7-Zip for zip file extraction instead)
3. Then on GitHub make a new public (or private if specified) repository. For demonstration purposes, this repository will be private, but the same rules will still apply
4. The following steps show how to manually put the extracted contents into the master branch on GitHub. This step assumes you are using a terminal based tool like or your terminal emulator on UNIX/Linux or Mac. Skip to Step 5 for additional steps on how to do this in GitHub Desktop on Windows:
    a. Make an empty folder/directory from where you will be doing your development. In this case it will be main_repository. An example of how this directory in PowerShell was created can be seen in the following screenshot (with the same command words such as mkdir and cd being usable on Unix/Linux systems):

b. Issue the command (without quotes) “git init” in the terminal in this location
c. Then copy the contents from the extracted zip location to this mostly empty directory (with the only file being a hidden file called .git)
    i. This can be accomplished with commands “cp -r” in Unix/Linux systems, or “Copy-Item” with the Recurse option in PowerShell (an example of which can be found in Example 2 here: https://docs.microsoft.com/en-us/powershell/module/microsoft.powershell.management/copy-item?view=powershell-7#examples), or using a graphical File Explorer on Windows, UNIX/Linux, and other machines
d. After step c, the folder should contain the following files:

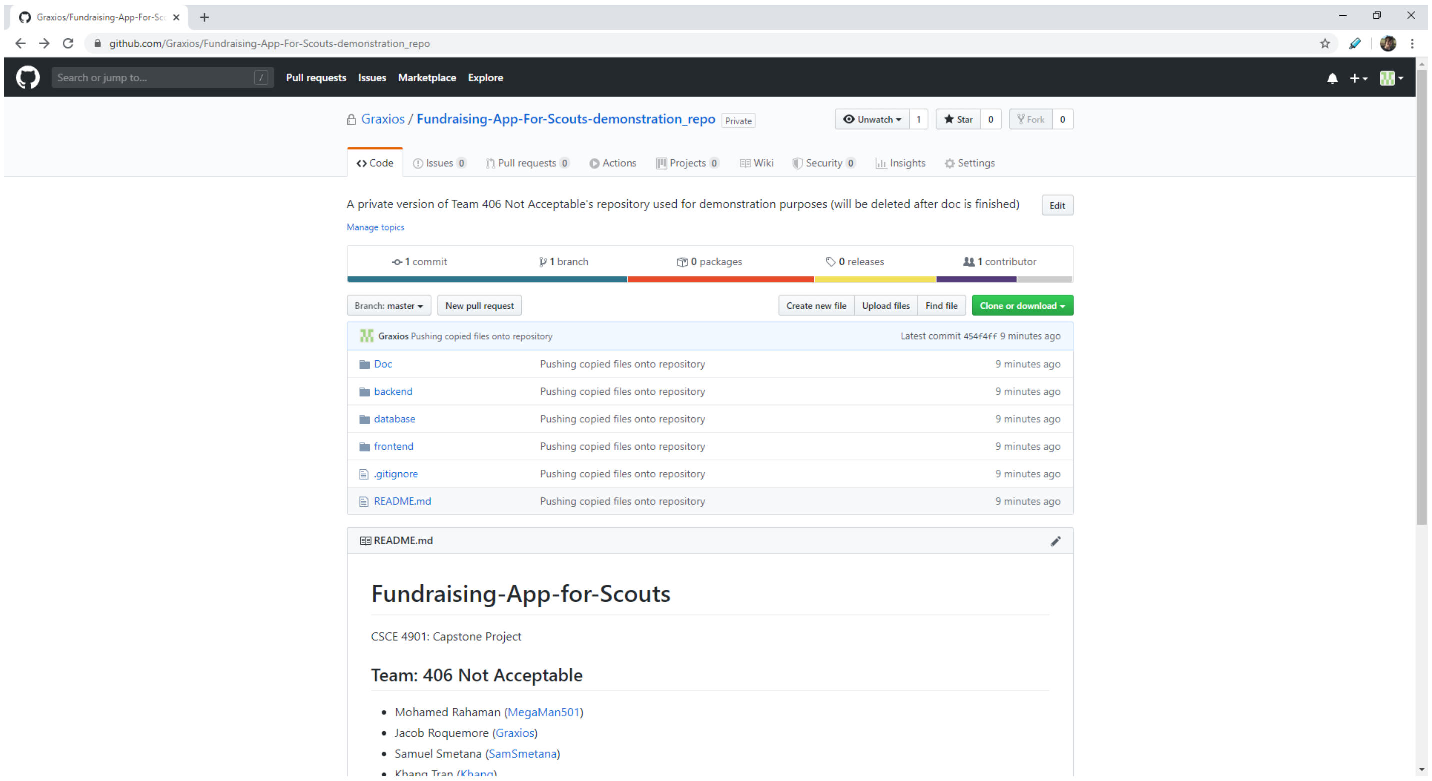


```
s PC > JACOB_R_EXT (D:) > Laptop_Storage > Shared > School > Spring_2020 > CSCE_4901 > instruc_demo > main_repository

Name            Date modified        Type            Size
.git            5/9/2020 1:49 AM     File folder
backend         5/9/2020 1:57 AM     File folder
database        5/9/2020 1:57 AM     File folder
Doc             5/9/2020 1:57 AM     File folder
frontend        5/9/2020 1:56 AM     File folder
.gitignore      5/9/2020 12:10 AM    Text Document   1 KB
README.md       5/9/2020 12:10 AM    MD File         1 KB
```

    i. (Note: .git will not be viewable unless you have Hidden Files set as viewable on your systems graphical file browser, or if you use a command like ls -la on UNIX/Linux based systems).
e. Then in a terminal issue the command (without quotes), “git add -A”, to get all of the copied files and directories ready for commit
    i. Note: On Windows based systems you may get several warning messages stating: “warning: LF will be replaced by CRLF in …”, and “The file will have its original line endings in your working directory”. This just means that because our code was primarily developed on Linux based systems, only linefields were used at the end of text lines, versus Windows using carriage returns and line fields to indicate line endings. Therefore, git is automatically accounting for this on Windows systems by adding CRLF. This error message can be suppressed using the following command, “git config core.autocrlf true”
f. Next in a terminal issue the command (without quotes), “git commit -m ‘Message here’”, where message here is a short message you want to use for your initial deployment
g. Then issue the command (without quotes): “git remote add origin https://github.com/user/repo_name.git”, where user is the username of the individual who has made the repository for the group, and repo_name is the name of the repository
h. Finally issue the command, “git push -u origin master”, to push the newly copied files to the master branch. When prompted for user details, provide the username and password of the individual who formed the initial repository

i. To ensure that the push was successful, the page should now similar to the following:



A private version of Team 406 Not Acceptable's repository used for demonstration purposes (will be deleted after doc is finished)

Graxios Pushing copied files onto repository — Latest commit 454f4ff 9 minutes ago

Doc	Pushing copied files onto repository	9 minutes ago
backend	Pushing copied files onto repository	9 minutes ago
database	Pushing copied files onto repository	9 minutes ago
frontend	Pushing copied files onto repository	9 minutes ago
.gitignore	Pushing copied files onto repository	9 minutes ago
README.md	Pushing copied files onto repository	9 minutes ago

README.md

Fundraising-App-for-Scouts

CSCE 4901: Capstone Project

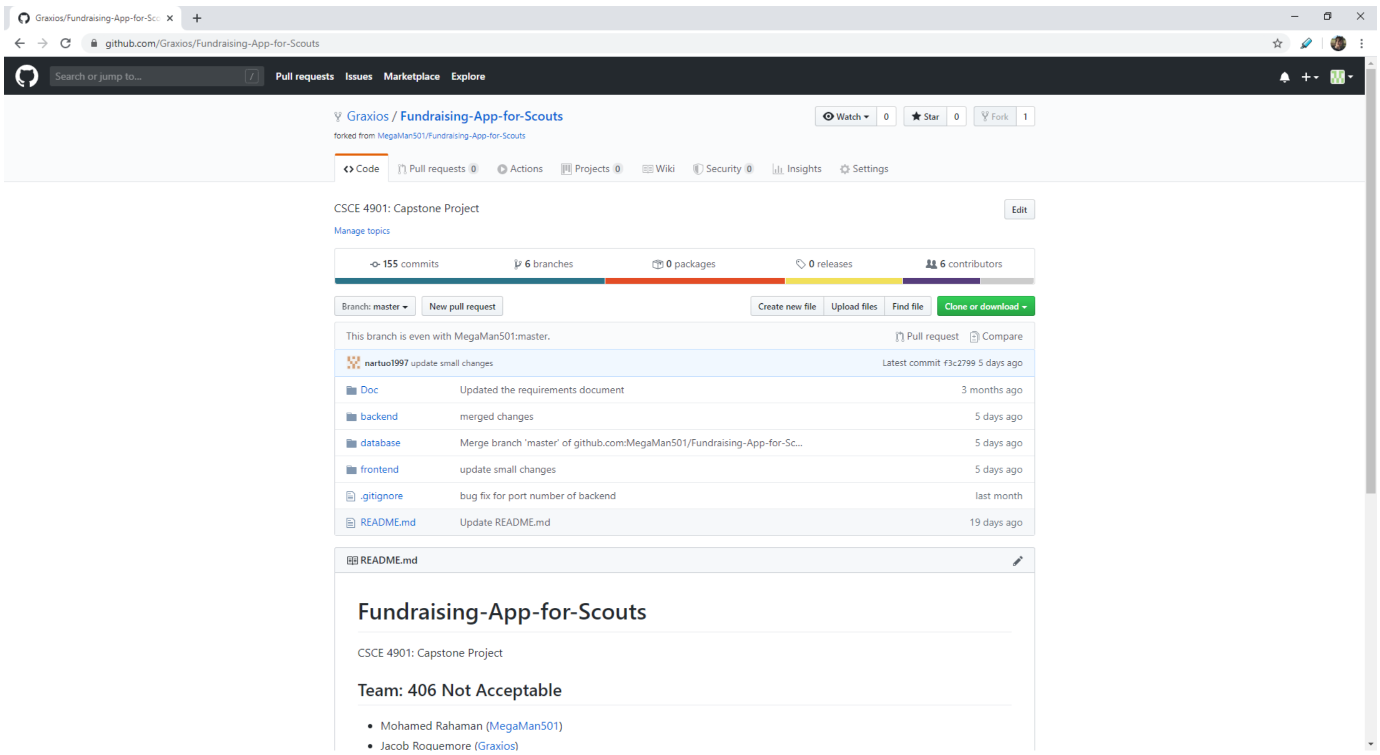
Team: 406 Not Acceptable

- Mohamed Rahaman (MegaMan501)
- Jacob Roquemore (Graxios)
- Samuel Smetana (SamSmetana)
- Khang Tran (Khang)

1.2 Adding Team Members as Collaborators to GitHub Repository

The following steps how to add additional group members and branches to the GitHub repository through the online interface:

1. The repository leader should navigate to the newly forked (or made) repository by clicking on it. This should bring them to the following screen:


Graxios / Fundraising-App-for-Scouts
forked from MegaMan501/Fundraising-App-for-Scouts

CSCE 4901: Capstone Project

155 commits | 6 branches | 0 packages | 0 releases | 6 contributors

This branch is even with MegaMan501:master.

nartuo1997 update small changes — Latest commit f3c2799 5 days ago

Doc	Updated the requirements document	3 months ago
backend	merged changes	5 days ago
database	Merge branch 'master' of github.com:MegaMan501/Fundraising-App-for-Sc...	5 days ago
frontend	update small changes	5 days ago
.gitignore	bug fix for port number of backend	last month
README.md	Update README.md	19 days ago

README.md

Fundraising-App-for-Scouts

CSCE 4901: Capstone Project

Team: 406 Not Acceptable

- Mohamed Rahaman (MegaMan501)
- Jacob Roquemore (Graxios)

2. The leader should then follow the documentation provided here: https://help.github.com/en/github/setting-up-and-managing-your-github-user-account/inviting-collaborators-to-a-personal-repository which should provide the most up to date method of performing this action

1.3 Installing and Setting Up Dependencies for the Project

It is important to note that there are some requirements needed before package dependencies for our project can be installed. The main requirements are:

- Node.js and npm
    - Can be installed from many UNIX/Linux package repositories. Installers for Windows and Mac (as well as Linux binaries and source code) can be found here: https://nodejs.org/en/download/
- Python

Important Side Note: While there have been relatively few issues with development on a Linux machine, when installing dependencies through PowerShell on Windows, there were errors regarding Visual Studio. CPHPython's answer at the following link provides instructions on how to potentially resolve this: https://stackoverflow.com/questions/57879150/how-can-i-solve-error-gypgyp-errerr-find-vsfind-vs-msvs-version-not-set-from-c. However, if this link is no longer available, then in summary the following steps could be performed if a Windows computer has a preexisting installation of Visual Studio (based on CPHPython's StackOverflow Answer):

1. Open Visual Studio Installer (can be found through a Windows search by pressing the Windows key and then typing the program name out)

2. Find the latest version of a Visual Studio Installation in the displayed list and press the Modify button on it

3. In the Workloads grid/list menu, select the checkbox next to, "Desktop development with C++"

4. Press the Modify button in the bottom right corner (could also be Download/Install)

The following steps outline how an individual user should install the dependencies for the backend components:

1. The user should navigate to where their local repository is located in their terminal (note: This terminal can be something like a UNIX/LINUX terminal, Windows Powershell, or a terminal provided in an IDE like Visual Studio). The current directory should be, "Fundraising-App-for-Scouts". The user should then change their directory to the backend/ directory if they are not already located there.
2. The user should then install the necessary dependencies using the command (without quotes), "npm i"
3. Then run, "npm update", to update any of the dependencies to the latest versions if needed
4. To test and make sure all dependencies are working run the command, "npm run dev-backend", or, "npm run backend". Use Ctrl+C when you are ready to stop the process as it is running
    a. Note: Error messages should be expected unless you have setup your .env file (more detail is covered in Section 1.9). Simply use Ctrl+C to terminate the process, or use the equivalent process sending tool provided in your terminal

The following steps outline how an individual should install the dependencies for the frontend components:

1. In any directory besides the frontend/ directory issue the command, "npm install -g @angular/cli" (the current directory in the terminal can be checked using a command like pwd in a Bash based terminal, PowerShell, or other terminals)
2. Then change your directory to the, "frontend", directory where your local repository is located and issue the command, "npm i" to install the necessary dependencies
3. Then run, "ng update", to perform any necessary updates to the latest version of angular

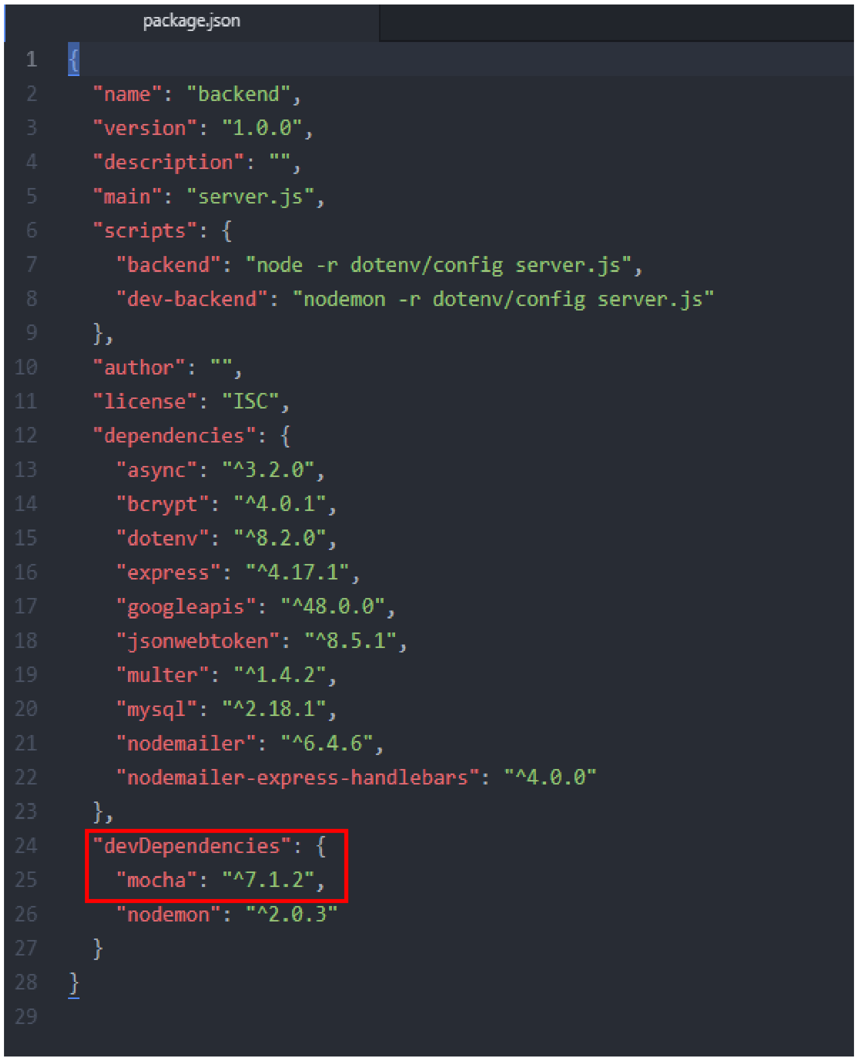
4. Then run, “npm update”, to update any of the remaining frontend dependencies
5. Then run, “ng serve”, to compile the frontend components and startup a test serve
    a. (Note: In Windows PowerShell or other Windows command line tools, the process may appear stuck. Press the Enter key, and output describing compiling and compilation should appear)
    b. The command should give an accessible url and port (most likely localhost:4200) which can be opened in a web browser
    c. Another Important Note: On some Windows installations, you may be prevented from running this command in environments like PowerShell due to Windows script execution policies. Here are two potential methods to fix this:
        i. A simple way to possibly do this (courtesy of Amir Makram’s answer at: https://stackoverflow.com/questions/58032631/why-powershell-does-not-run-angular-commands) is to remove ng.ps1 from the directory at C:\Users\username\AppData\Roaming\npm\, where user is the user name of the current user, and attempt the command again.
            1. (A way to access AppData is to first press Win + R or search for, “Run”, and then type in (without quotes) “C:\Users\%username%\AppData\Roaming\npm\”, and press enter
        ii. Another potential method to resolve this (courtesy of Stanley Mohlala at: https://stackoverflow.com/questions/58032631/why-powershell-does-not-run-angular-commands is to allow RemoteSigned Scripts through the following command in PowerShell, “Set-ExecutionPolicy -ExecutionPolicy RemoteSigned -Scope CurrentUser”, where CurrentUser is the user name of the Windows user). There may be some potential security implications from this method (more information can be found here: https://docs.microsoft.com/en-us/powershell/module/microsoft.powershell.core/about/about_execution_policies?view=powershell-7), so only run PowerShell commands which you trust whenever developing.

Lastly, whenever additional dependencies are needed for the project, they should be installed using the appropriate commands, depending on if a package is needed for development or for the production system.

The following steps cover an example of installing the, “mocha”, npm package which can be used for unit testing and regressive testing during Node.JS based development (note the final product will have the mocha package already listed as a development dependency and will not need to be installed again. This is just for demonstration purposes):

1. In the backend/ directory install the package using, “npm install –save-dev mocha”

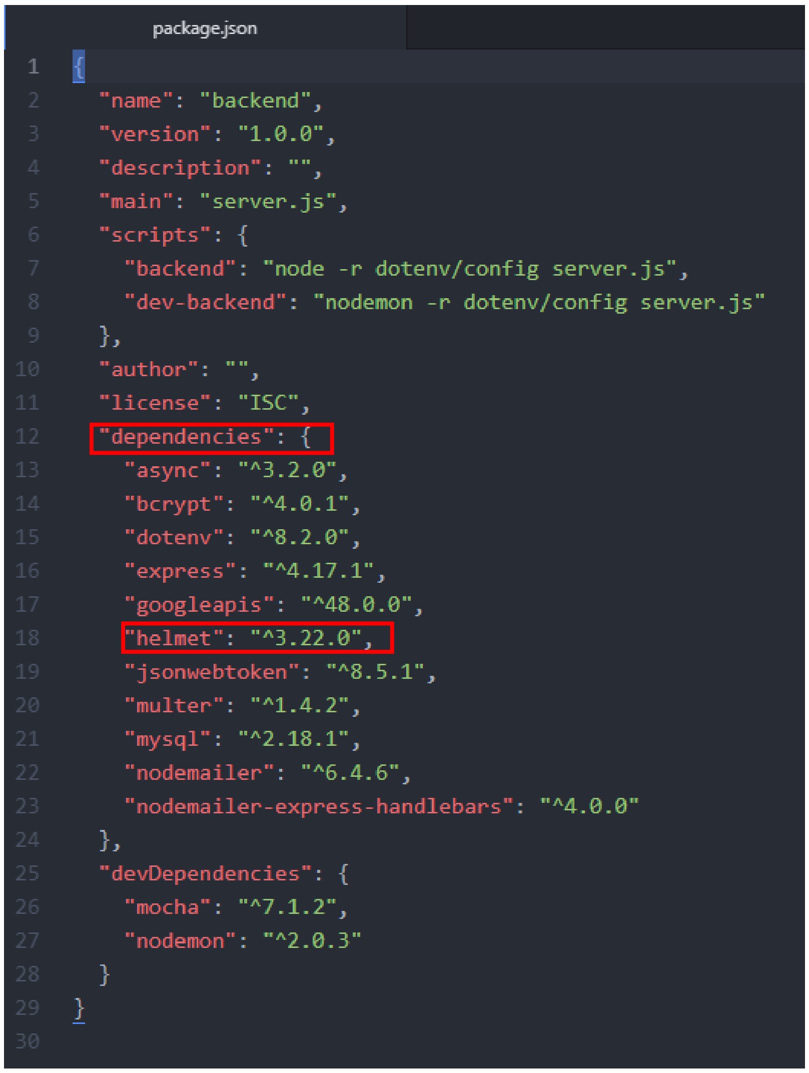
2. After installation, check the package.json file. It should now appear as a dependency. An example of this can be seen below:


package.json

```
{
  "name": "backend",
  "version": "1.0.0",
  "description": "",
  "main": "server.js",
  "scripts": {
    "backend": "node -r dotenv/config server.js",
    "dev-backend": "nodemon -r dotenv/config server.js"
  },
  "author": "",
  "license": "ISC",
  "dependencies": {
    "async": "^3.2.0",
    "bcrypt": "^4.0.1",
    "dotenv": "^8.2.0",
    "express": "^4.17.1",
    "googleapis": "^48.0.0",
    "jsonwebtoken": "^8.5.1",
    "multer": "^1.4.2",
    "mysql": "^2.18.1",
    "nodemailer": "^6.4.6",
    "nodemailer-express-handlebars": "^4.0.0"
  },
  "devDependencies": {
    "mocha": "^7.1.2",
    "nodemon": "^2.0.3"
  }
}
```

The following steps cover an example of installing the, “helmet”, npm package as a production requirement (note: Our team has not used helmet as a package in our production, this is just serving as an example):

1. In the backend/ directory install the package using, “npm install --save helmet”
2. After installation, check the package.json file. It should now appear as a dependency. An example of this can be seen below:



package.json

```
{
  "name": "backend",
  "version": "1.0.0",
  "description": "",
  "main": "server.js",
  "scripts": {
    "backend": "node -r dotenv/config server.js",
    "dev-backend": "nodemon -r dotenv/config server.js"
  },
  "author": "",
  "license": "ISC",
  "dependencies": {
    "async": "^3.2.0",
    "bcrypt": "^4.0.1",
    "dotenv": "^8.2.0",
    "express": "^4.17.1",
    "googleapis": "^48.0.0",
    "helmet": "^3.22.0",
    "jsonwebtoken": "^8.5.1",
    "multer": "^1.4.2",
    "mysql": "^2.18.1",
    "nodemailer": "^6.4.6",
    "nodemailer-express-handlebars": "^4.0.0"
  },
  "devDependencies": {
    "mocha": "^7.1.2",
    "nodemon": "^2.0.3"
  }
}
```

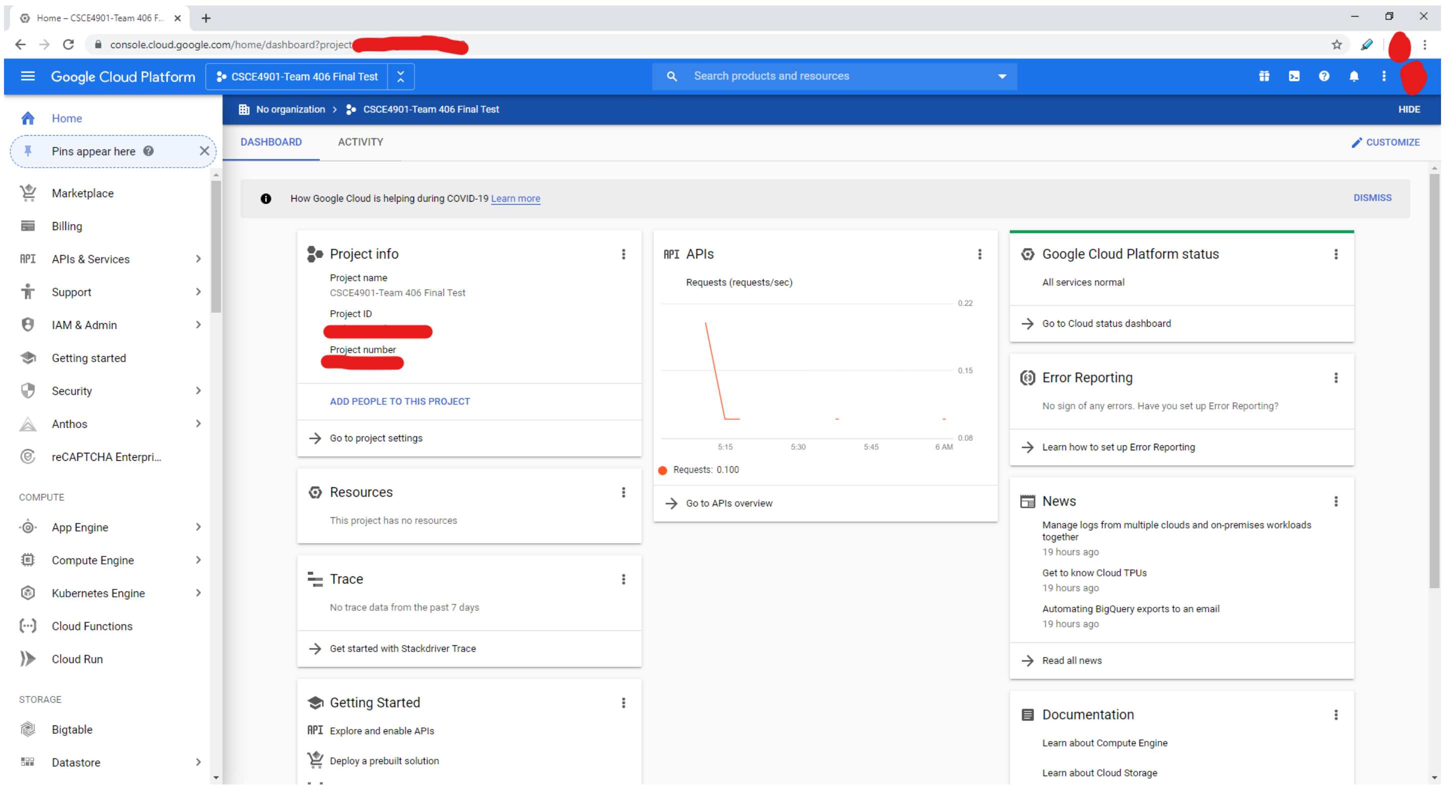
(It is important to form a regular habit of performing, “npm install”, and, “npm update”, after pulling any changes from GitHub, in case any other team members have added needed packages to the project. It is also encouraged to notify team members if any major packages have been added as a dependency as well.)

1.4 Setup for MySQL database

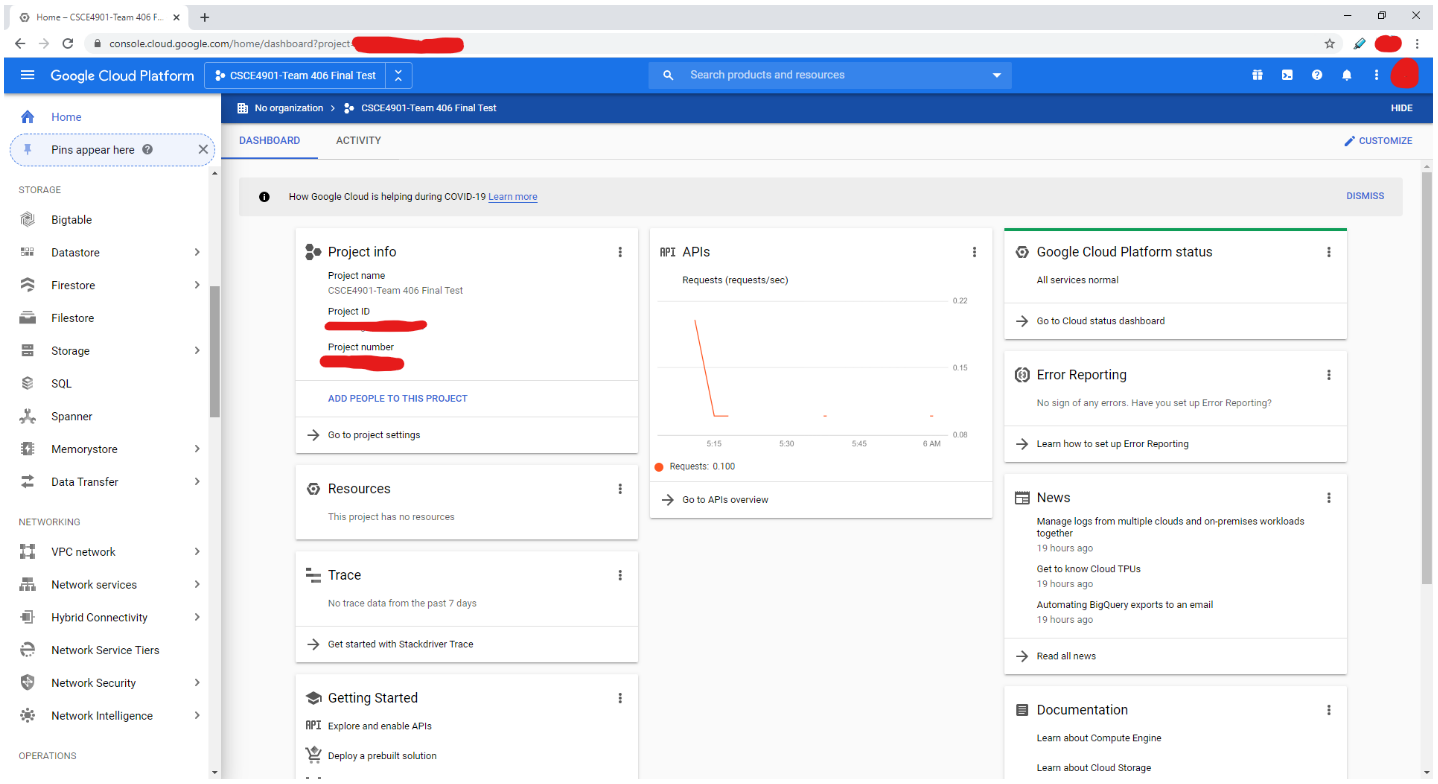
1.4.1 Google Cloud Platform (GCP) MySQL setup

The following set of steps show how to setup a MySQL instance on the Google Cloud Platform (GCP), assuming you already have used GCP in the past, or have at least setup an account. (If you have not, then go to: https://cloud.google.com/, and click on, “Get started for free”, and follow the necessary setup steps:

1. Navigate to cloud.google.com, assuming you are logged into your associated account, and select, “Go to console”
2. You should then see a dashboard like the one below:


Home – CSCE4901-Team 406 F...
console.cloud.google.com/home/dashboard?project[redacted]
Google Cloud Platform
CSCE4901-Team 406 Final Test
Search products and resources
No organization > CSCE4901-Team 406 Final Test
HIDE
DASHBOARD
ACTIVITY
CUSTOMIZE
Home
Pins appear here
Marketplace
Billing
APIs & Services
Support
IAM & Admin
Getting started
Security
Anthos
reCAPTCHA Enterpri...
COMPUTE
App Engine
Compute Engine
Kubernetes Engine
Cloud Functions
Cloud Run
STORAGE
Bigtable
Datastore
How Google Cloud is helping during COVID-19 Learn more
DISMISS
Project info
Project name
CSCE4901-Team 406 Final Test
Project ID
[redacted]
Project number
[redacted]
ADD PEOPLE TO THIS PROJECT
Go to project settings
Resources
This project has no resources
Trace
No trace data from the past 7 days
Get started with Stackdriver Trace
Getting Started
Explore and enable APIs
Deploy a prebuilt solution
APIs
Requests (requests/sec)
0.22
0.15
0.08
5:15
5:30
5:45
6 AM
Requests: 0.100
Go to APIs overview
Google Cloud Platform status
All services normal
Go to Cloud status dashboard
Error Reporting
No sign of any errors. Have you set up Error Reporting?
Learn how to set up Error Reporting
News
Manage logs from multiple clouds and on-premises workloads together
19 hours ago
Get to know Cloud TPUs
19 hours ago
Automating BigQuery exports to an email
19 hours ago
Read all news
Documentation
Learn about Compute Engine
Learn about Cloud Storage

3. Using the scrollbar on the left, scroll down to the STORAGE section until you see SQL. This should appear like the image below:


Home – CSCE4901-Team 406 F...

console.cloud.google.com/home/dashboard?project=[redacted] [redacted]

Google Cloud Platform | CSCE4901-Team 406 Final Test | Search products and resources [redacted]

No organization > CSCE4901-Team 406 Final Test HIDE

DASHBOARD ACTIVITY CUSTOMIZE

Home
Pins appear here

STORAGE
- Bigtable
- Datastore
- Firestore
- Filestore
- Storage
- SQL
- Spanner
- Memorystore
- Data Transfer

NETWORKING
- VPC network
- Network services
- Hybrid Connectivity
- Network Service Tiers
- Network Security
- Network Intelligence

OPERATIONS

How Google Cloud is helping during COVID-19 Learn more DISMISS

Project info
Project name
CSCE4901-Team 406 Final Test
Project ID
[redacted]
Project number
[redacted]
ADD PEOPLE TO THIS PROJECT
Go to project settings

Resources
This project has no resources

Trace
No trace data from the past 7 days
Get started with Stackdriver Trace

Getting Started
Explore and enable APIs
Deploy a prebuilt solution

APIs
Requests (requests/sec)
0.22, 0.15, 0.08
5:15, 5:30, 5:45, 6 AM
Requests: 0.100
Go to APIs overview

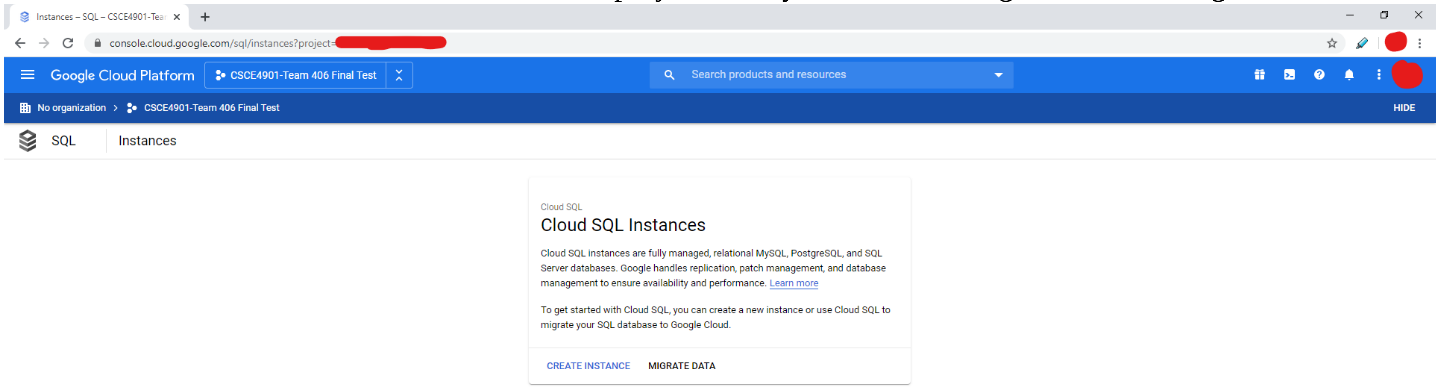
Google Cloud Platform status
All services normal
Go to Cloud status dashboard

Error Reporting
No sign of any errors. Have you set up Error Reporting?
Learn how to set up Error Reporting

News
Manage logs from multiple clouds and on-premises workloads together
19 hours ago
Get to know Cloud TPUs
19 hours ago
Automating BigQuery exports to an email
19 hours ago
Read all news

Documentation
Learn about Compute Engine
Learn about Cloud Storage

4. Then select SQL. If this is a new project, then you will be taking to the following screen:


Instances – SQL – CSCE4901-Tea

console.cloud.google.com/sql/instances?project=[redacted] [redacted]

Google Cloud Platform | CSCE4901-Team 406 Final Test | Search products and resources [redacted]

No organization > CSCE4901-Team 406 Final Test HIDE

SQL Instances

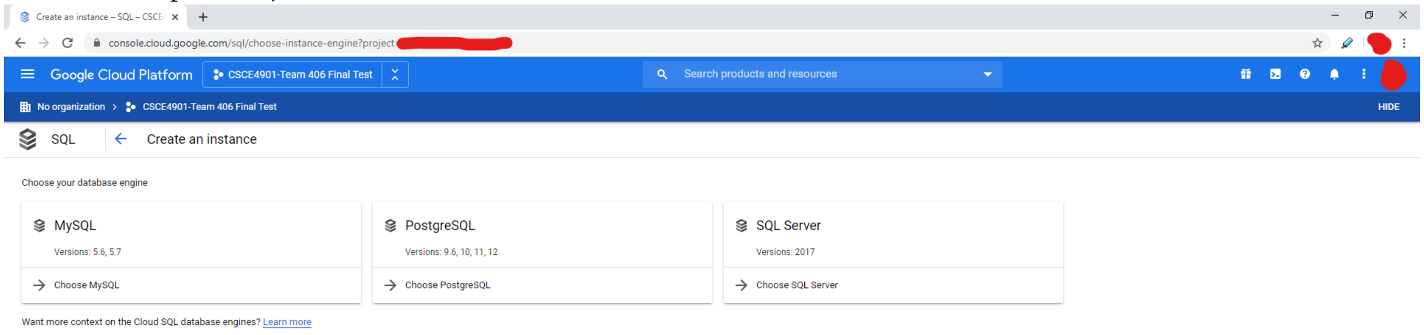
Cloud SQL
Cloud SQL Instances

Cloud SQL instances are fully managed, relational MySQL, PostgreSQL, and SQL Server databases. Google handles replication, patch management, and database management to ensure availability and performance. Learn more

To get started with Cloud SQL, you can create a new instance or use Cloud SQL to migrate your SQL database to Google Cloud.

CREATE INSTANCE MIGRATE DATA

5. Click on, "Create Instance", any you will be taken to the following screen (for our testing as a development team, we used the MySQL option. Select the option you prefer and then proceed):


Create an instance – SQL – CSCE

console.cloud.google.com/sql/choose-instance-engine?project=[redacted]

Google Cloud Platform | CSCE4901-Team 406 Final Test | Search products and resources

No organization > CSCE4901-Team 406 Final Test | HIDE

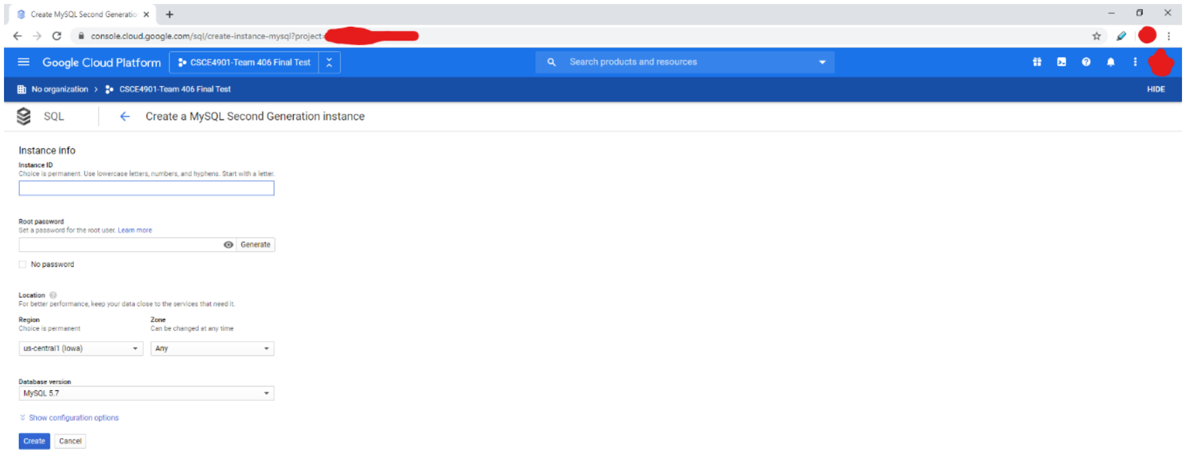
SQL | Create an instance

Choose your database engine

MySQL Versions: 5.6, 5.7 Choose MySQL	PostgreSQL Versions: 9.6, 10, 11, 12 Choose PostgreSQL	SQL Server Versions: 2017 Choose SQL Server

Want more context on the Cloud SQL database engines? Learn more

6. You will then be taken to this setup screen. Fill out the fields accordingly and choose the options that best fit your team's needs. It is important to note that, "Show configuration options", will provide you additional options, such as enabling or disabling automated backups, so it is important to look through those options as well:


Create MySQL Second Generatio

console.cloud.google.com/sql/create-instance-mysql?project=[redacted]

Google Cloud Platform | CSCE4901-Team 406 Final Test | Search products and resources

No organization > CSCE4901-Team 406 Final Test | HIDE

SQL | Create a MySQL Second Generation instance

Instance info

Instance ID
Choice is permanent. Use lowercase letters, numbers, and hyphens. Start with a letter.

Root password
Set a password for the root user. Learn more
Generate

☐ No password

Location
For better performance, keep your data close to the services that need it.

Region: Choice is permanent — us-central1 (Iowa)

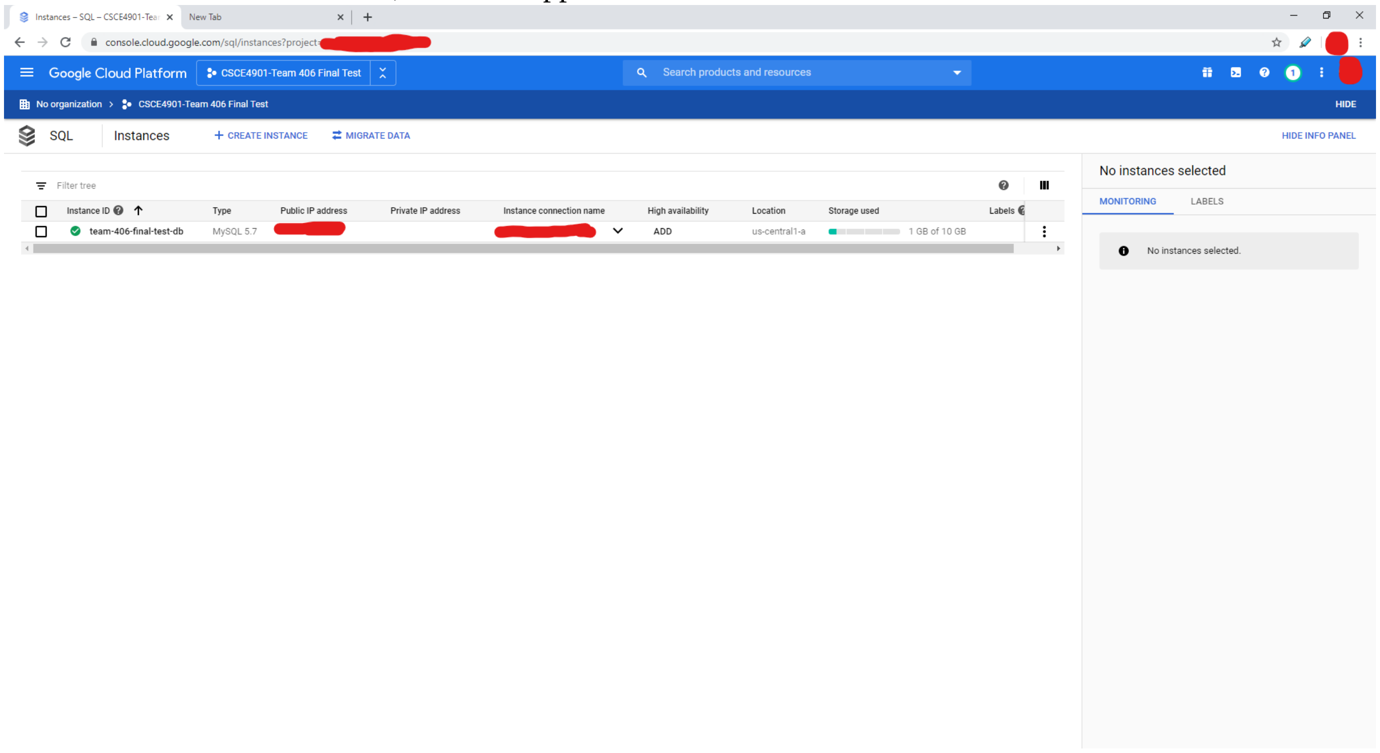
Zone: Can be changed at any time — Any

Database version: MySQL 5.7

Show configuration options

Create | Cancel

7. After creating the instance, you should be taken to the following screen. When the instance has finished, it should appear similar to the database below:



console.cloud.google.com/sql/instances?project[redacted]

Google Cloud Platform · CSCE4901-Team 406 Final Test

No organization > CSCE4901-Team 406 Final Test

SQL · Instances · CREATE INSTANCE · MIGRATE DATA · HIDE INFO PANEL

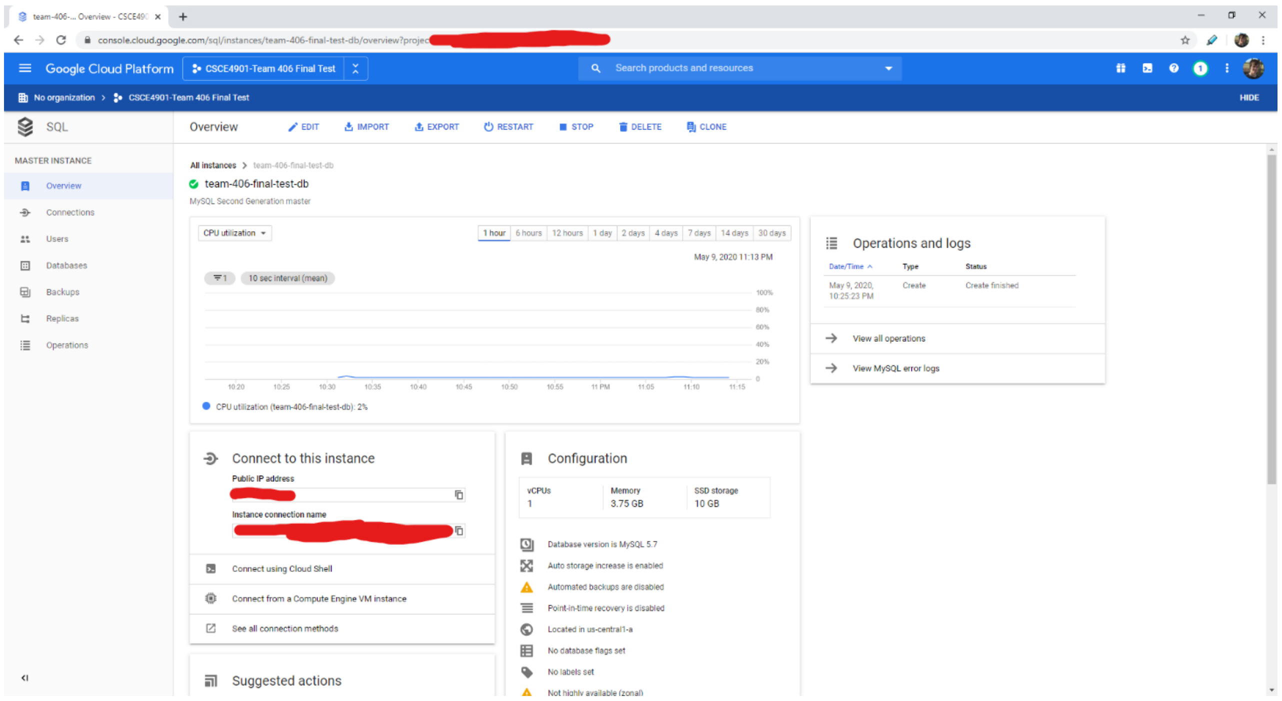
Instance ID	Type	Public IP address	Private IP address	Instance connection name	High availability	Location	Storage used	Labels
team-406-final-test-db	MySQL 5.7	[redacted]		[redacted]	ADD	us-central1-a	1 GB of 10 GB	

No instances selected

MONITORING · LABELS

No instances selected.

8. Click on the database to access it’s overview and settings. It should appear similar to the following screenshot below:



console.cloud.google.com/sql/instances/team-406-final-test-db/overview?project[redacted]

Google Cloud Platform · CSCE4901-Team 406 Final Test

No organization > CSCE4901-Team 406 Final Test

SQL · Overview · EDIT · IMPORT · EXPORT · RESTART · STOP · DELETE · CLONE

MASTER INSTANCE: Overview, Connections, Users, Databases, Backups, Replicas, Operations

All instances > team-406-final-test-db

team-406-final-test-db

MySQL Second Generation master

CPU utilization · 1 hour · 6 hours · 12 hours · 1 day · 2 days · 4 days · 7 days · 14 days · 30 days

May 9, 2020 11:13 PM

10 sec interval (mean)

CPU utilization (team-406-final-test-db): 2%

Operations and logs

Date/Time	Type	Status
May 9, 2020, 10:25:23 PM	Create	Create finished

View all operations

View MySQL error logs

Connect to this instance

Public IP address: [redacted]

Instance connection name: [redacted]

Connect using Cloud Shell

Connect from a Compute Engine VM instance

See all connection methods

Configuration

vCPUs	Memory	SSD storage
1	3.75 GB	10 GB

Database version is MySQL 5.7

Auto storage increase is enabled

Automated backups are disabled

Point-in-time recovery is disabled

Located in us-central1-a

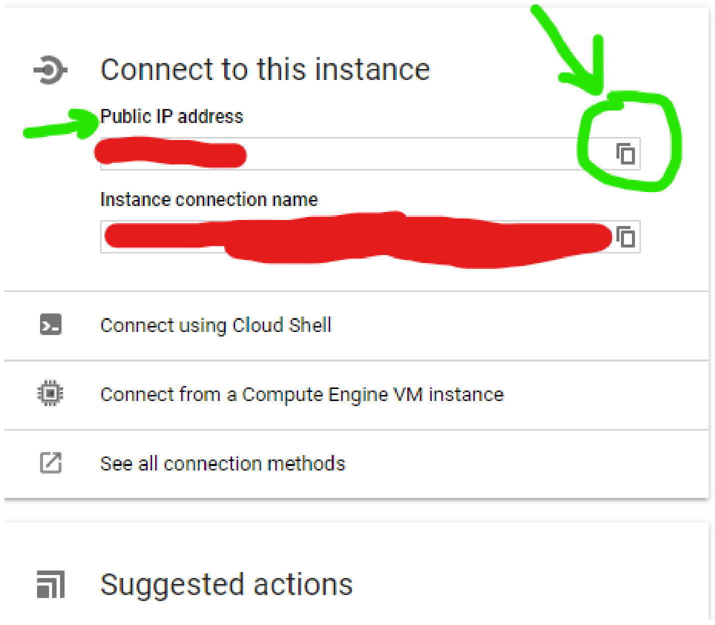
No database flags set

No labels set

Not highly available (zonal)

Suggested actions

9. In the Overview Screen, under, “Connect to this instance”, make note of the Public IP address by either manually selecting it and copying it, or by clicking on the copy icon as highlighted in the screenshot below:



Connect to this instance

Public IP address

[redacted]

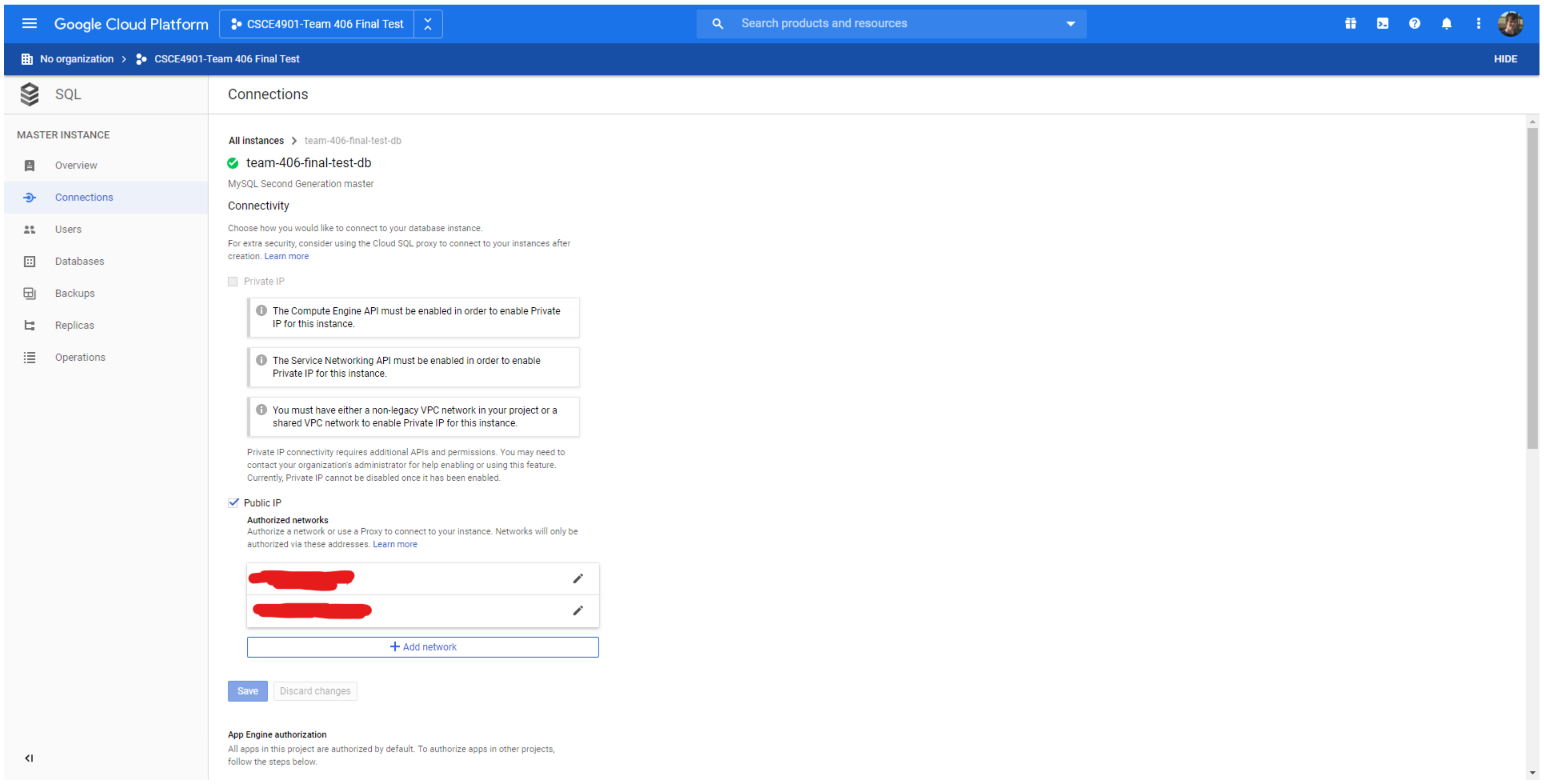
Instance connection name

[redacted]

Connect using Cloud Shell
Connect from a Compute Engine VM instance
See all connection methods
Suggested actions

   a. (Note: This public IP address, once all of the other security precautions below are finished, can be shared with your other team members for connecting to the database using tools like MySQL workbench and the MariaDB/mysql command line interfaces.)

10. Under the Master Instance bar on the right, select, “Connections”. Then on this screen, find, “Connectivity”. This should then appear similar to the following screen:


Google Cloud Platform
CSCE4901-Team 406 Final Test
Search products and resources
No organization > CSCE4901-Team 406 Final Test
HIDE
SQL
MASTER INSTANCE
Overview
Connections
Users
Databases
Backups
Replicas
Operations

Connections

All instances > team-406-final-test-db

team-406-final-test-db

MySQL Second Generation master

Connectivity

Choose how you would like to connect to your database instance.

For extra security, consider using the Cloud SQL proxy to connect to your instances after creation. Learn more

Private IP

The Compute Engine API must be enabled in order to enable Private IP for this instance.

The Service Networking API must be enabled in order to enable Private IP for this instance.

You must have either a non-legacy VPC network in your project or a shared VPC network to enable Private IP for this instance.

Private IP connectivity requires additional APIs and permissions. You may need to contact your organization's administrator for help enabling or using this feature. Currently, Private IP cannot be disabled once it has been enabled.

✓ Public IP

Authorized networks

Authorize a network or use a Proxy to connect to your instance. Networks will only be authorized via these addresses. Learn more

[redacted]

[redacted]

+ Add network

Save Discard changes

App Engine authorization

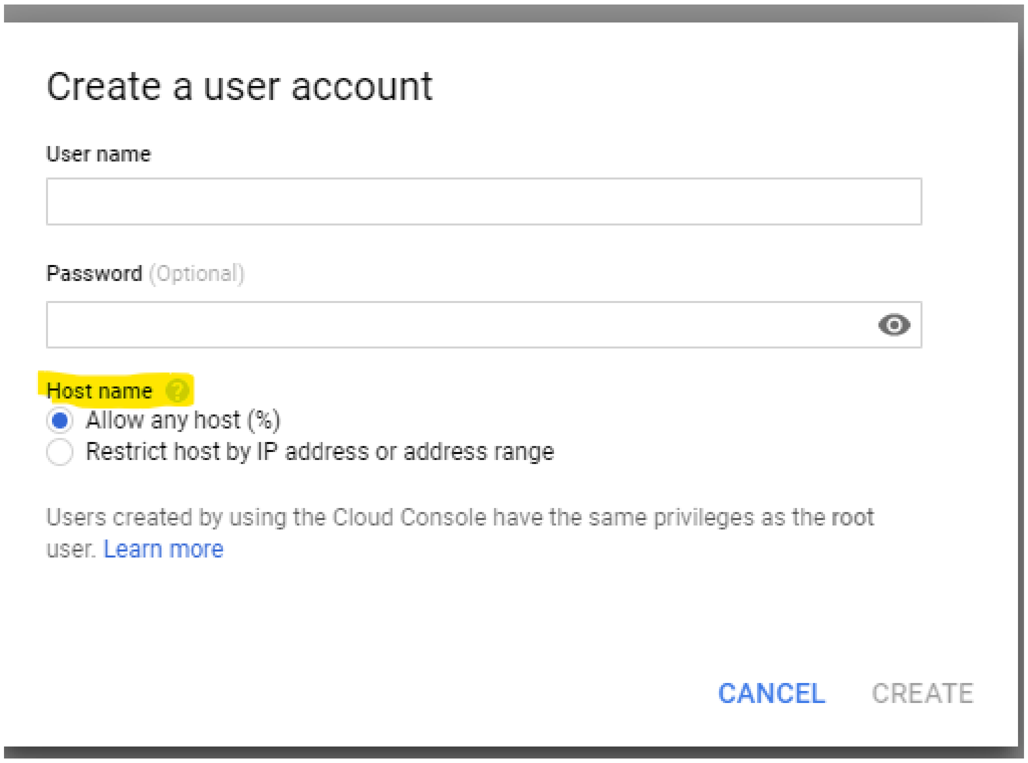
All apps in this project are authorized by default. To authorize apps in other projects, follow the steps below.

11. Under this section, and under the Public IP subsection, select, the, “Add Network”, button. This is where you should add you and your teammates public IP addresses for their home networks or networks which they most commonly develop on (you will have to privately discuss and exchange these IP addresses, and some public institutions may not allow traffic over port 3306 through their public facing IP address. If this may result in development issues in the future, then it is recommended that team members stick to a local MySQL instance. The setup for this is described in Section 1.4.2)

12. Then on the same, “Connections”, screen, scroll down to SSL. Under the SSL section perform the following steps for more secure connections to the cloud based MySQL database:
    a. Click on, “Allow only SSL connections”. (A refresh may be necessary if other options appear greyed out)
    b. Once this is done, click on, “Create new certificate”, under, “Configure SSL server certificates”. This will form a certificate authority which is required for SSL connections through MySQL connection tools
    c. Finally, click on, “Create a client certificate”, under, “Configure SSL client certificates”. Enter a unique identifier for this ssl certificate set when prompted. Finally, there will be a dialog window which pops up containing three important files: client-key.pem, client-cert.pem, and server-ca.pem. Download all of these files, and then transport them to a private, non-publicly accessible folder on your computer which can be referred to by any tools used for MySQL connections, as

well as by your node backend (more detail on this in Section 1.9). NOTE: ALWAYS HAVE THE FILES IN A PRIVATE, NON-PUBLICLY ACCESSIBLE LOCATION IN A DIRECTORY/PATH SEPARATE FROM YOUR LOCAL REPOSITORY DURING DEVELOPMENT AND YOUR DEPLOYMENT MACHINE'S FRONTEND AND BACKEND DIRECTORIES. ADDITIONALLY, ADD *.pem TO YOUR .gitignore FILE AT THE ROOT OF YOUR LOCAL REPOSITORY BEFORE PUSHING CHANGES (OR MAKE SURE THAT IT HAS BEEN ADDED)

13. It is recommended to have one or more separate users from your root user. Therefore, the following steps cover how to create a new user. In this example, a user called, "db-admin", will be created
    a. Under the, "Master Instance" side bar, select the, "Users" option
    b. There should already be two primary users (at least for a MySQL setup): mysql.sys and root.
    c. Add a new user by selecting, "Create user account". The following prompt should appear:



Create a user account

User name

Password (Optional)

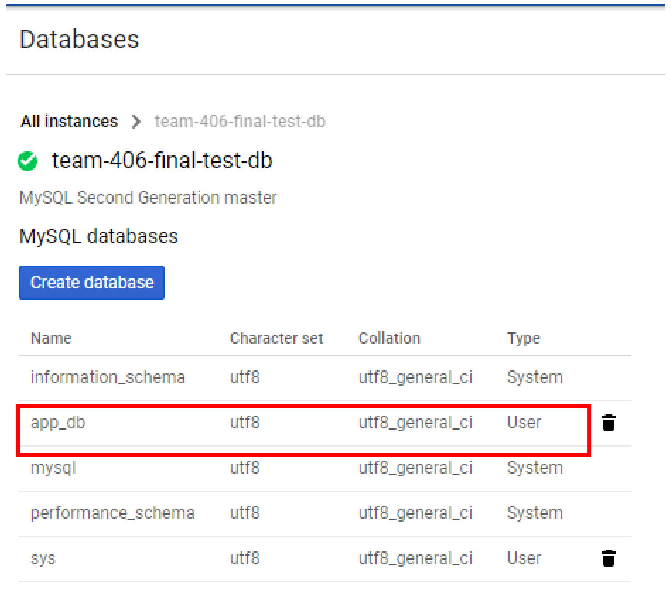
Host name
[✓ Allow any host (%)]
Restrict host by IP address or address range

Users created by using the Cloud Console have the same privileges as the root user. Learn more

        i. (Security Note: As highlighted above, Host name can be changed from Allow any host (which will allow connections to the database from any IP address, although they cannot access DB content without the associated SSL files and username and password) to Restrict host by IP Address or address range. With the restrict option, you can restrict the connection to only your team mate's public IP addresses for their home networks (which you will have to privately discuss and exchange information if decided upon)

    d. After entering a User name and password (and specifying restrictions to a select number of IP Addresses if agreed upon), click Create

14. Finally, it is necessary to have a database which can then store SQL tables, stored procedures, and other content needed for the application. In this example, a database called, “app_db”, will be created. The following steps below show how to make a new database in the GCP instance:
    a. Under the, “Master Instance” side bar, select the, “Databases” option
    b. There should already be system associated databases created automatically
    c. Click the, “Create database”, button
    d. Enter a database name, and select utf8 for the Character set. For this example, the collation will be left as Default collation (which is typically set to utf8_general_ci for MySQL instances), although you’re needs my vary.
    e. After adding the database, your database should now appear. An example of this can be seen below:


Databases
All instances > team-406-final-test-db
team-406-final-test-db
MySQL Second Generation master
MySQL databases
Create database

Name	Character set	Collation	Type
information_schema	utf8	utf8_general_ci	System
app_db	utf8	utf8_general_ci	User
mysql	utf8	utf8_general_ci	System
performance_schema	utf8	utf8_general_ci	System
sys	utf8	utf8_general_ci	User

1.4.2 Local MySQL server setup

There are two common tools for hosting an MySQL based server: MySQL server and MariaDB. The way in which they are setup depends on the platform. For a Linux-based manual installation (either for local development, or on a VPS like Digital Ocean), the following guides should be of use: MySQL - https://www.digitalocean.com/community/tutorials/how-to-install-linux-apache-mysql-php-lamp-stack-ubuntu-18-04#step-2-%E2%80%94-installing-mysql and MariaDB - https://www.digitalocean.com/community/tutorials/how-to-install-mariadb-on-ubuntu-18-04.

Finally, one of the easiest methods to setup MariaDB quickly on a Windows system (and also Linux and Mac), is to install XAMPP. XAMPP can be installed from the following site: https://www.apachefriends.org/download.html. (If installing XAMPP on a local machine for development purposes only, then it is also recommended to install phpMyAdmin for easier configuration. While this guide will not cover phpMyAdmin extensively, it's GUI based toolset is similar to MySQL Workbench). This section of the guide will cover similar setup steps to the one above, through the terminal (for manual MySQL/MariaDB installations). (Note the terminal for XAMPP can be accessed by selecting, "Shell", in the XAMPP window)

1.4.2.1 Setup using terminal command

(All of the commands listed below should be compatible with both MySQL and MariaDB.)

The following steps will cover how to setup a new database, a new user for the application, and allocating the user full privileges for the database. In this example the user will be, "db-admin", and the database will be, "app_db"
.(parts of this are based off the following guide: https://www.digitalocean.com/community/tutorials/how-to-install-wordpress-with-lamp-on-ubuntu-18-04#step-1-%E2%80%93-creating-a-mysql-database-and-user-for-wordpress):

1. mysql -u root -p
    a. Enter a password if one is required. If no password is required for the root account, then just hit Enter to continue
2. CREATE DATABASE app_db DEFAULT CHARACTER SET utf8 COLLATE utf8_general_ci;
3. GRANT ALL ON app_db.* TO 'db-admin'@'localhost' IDENTIFIED BY 'password';
    a. NOTE: password should be replaced with a password of the database creators choice
4. FLUSH PRIVILEGES;
5. EXIT;

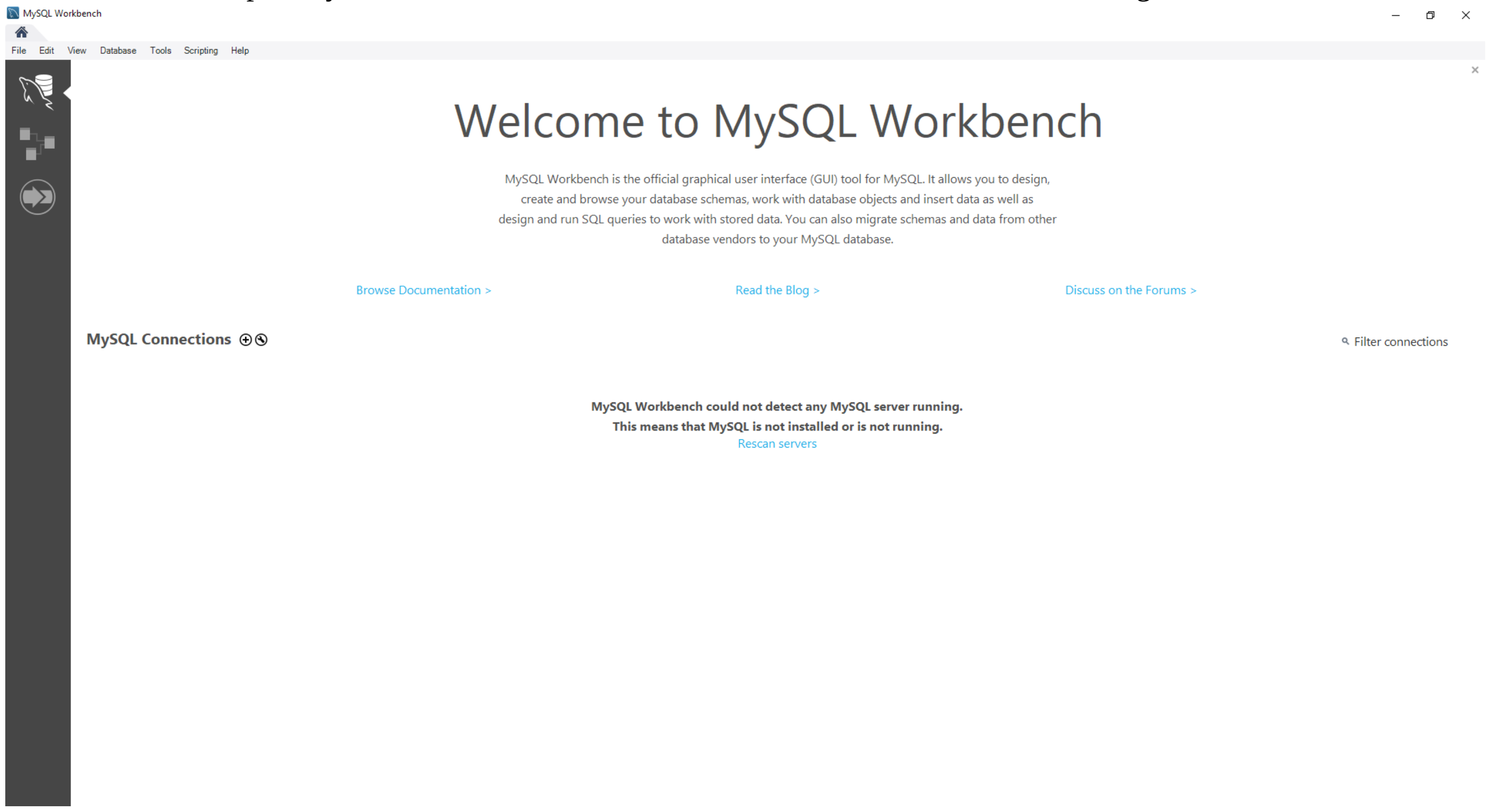
1.5 Connecting to MySQL database

For convenience and brevity, the rest of the MySQL setup portions will be performed using MySQL Workbench. The only exception will be the connection steps, in which examples will be provided for both

1.5.1 Google Cloud Platform (GCP) MySQL Connection Steps

1.5.1.1 Using MySQL Workbench

1. Open MySQL Workbench. The interface should be similar to the following:

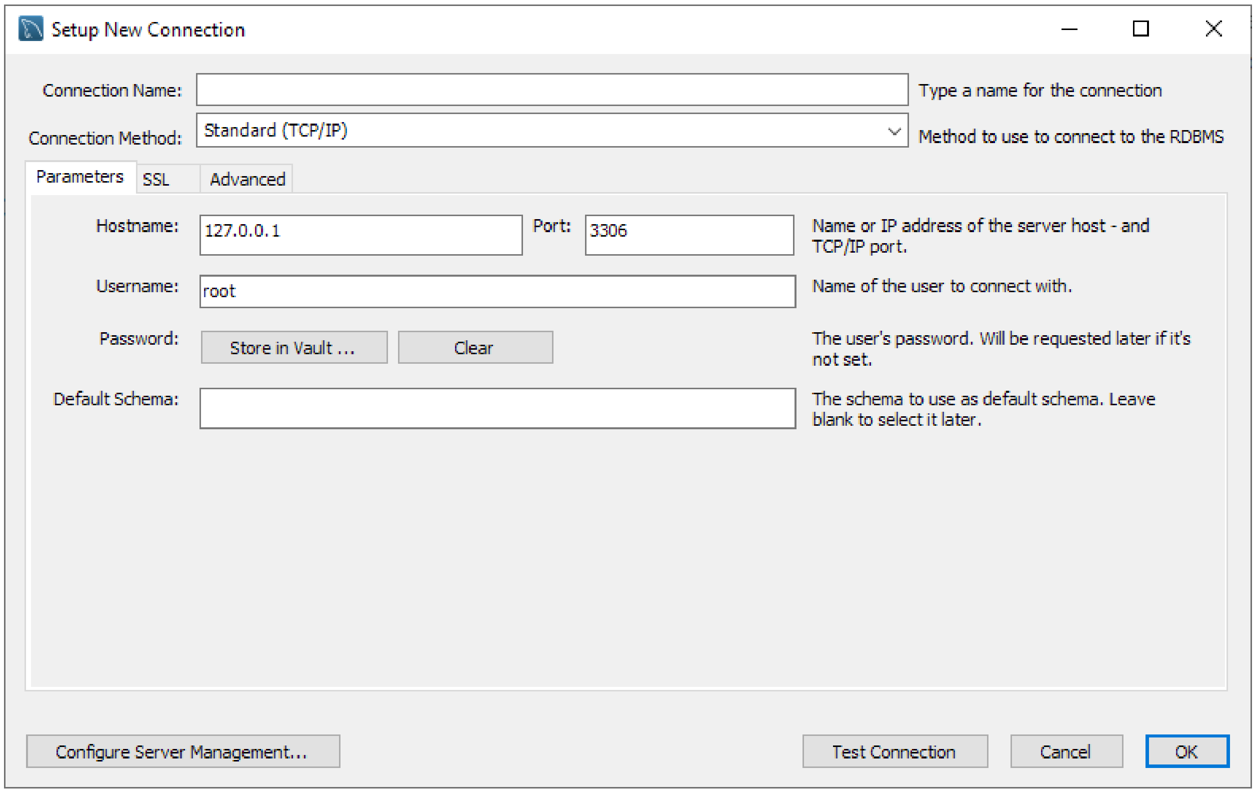


Welcome to MySQL Workbench

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.

MySQL Workbench could not detect any MySQL server running.
This means that MySQL is not installed or is not running.

2. Select the + button next to MySQL Connections. The following window should appear:


Setup New Connection

Connection Name: Type a name for the connection

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

Parameters | SSL | Advanced

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

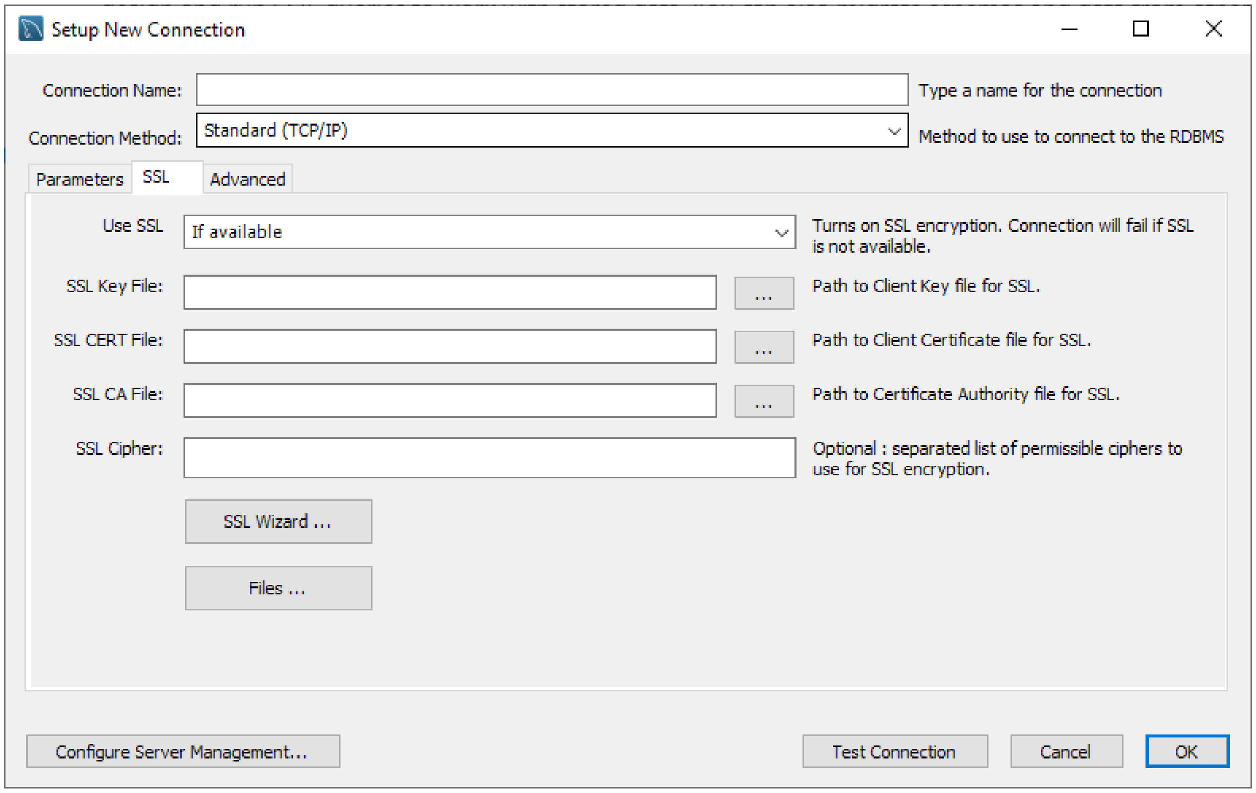
Username: root Name of the user to connect with.

Password: Store in Vault ... Clear 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.

Configure Server Management... Test Connection Cancel OK

3. The following instructions describe how to fill out the most important fields above:
    a. For Connection Name, type any nickname you want for these settings. In the future, you will be able to make connections easily by simply selecting the connection in the main Welcome Screen
    b. Leave Connection Method as Standard (TCP/IP)
    c. For Hostname: Replace the IP address with the one associated with your GCP database instance (refer to Section 1.4.1 Step 9 to find your IP in case you did not note it down)
    d. Leave Port as 3306, unless you changed it on your GCP instance
    e. Change Username to the one you used in Section 1.4.1
    f. Optional: For Password, select Store in Vault… and then enter the password with the password prompt
        i. (Note: Linux distributions may require the usage of a keyring on the system, most likely the Default keyring).
4. Next click on the SSL tab. It should appear similar to the screenshot below:


Setup New Connection

Connection Name: Type a name for the connection

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

Use SSL: If available — Turns on SSL encryption. Connection will fail if SSL is not available.

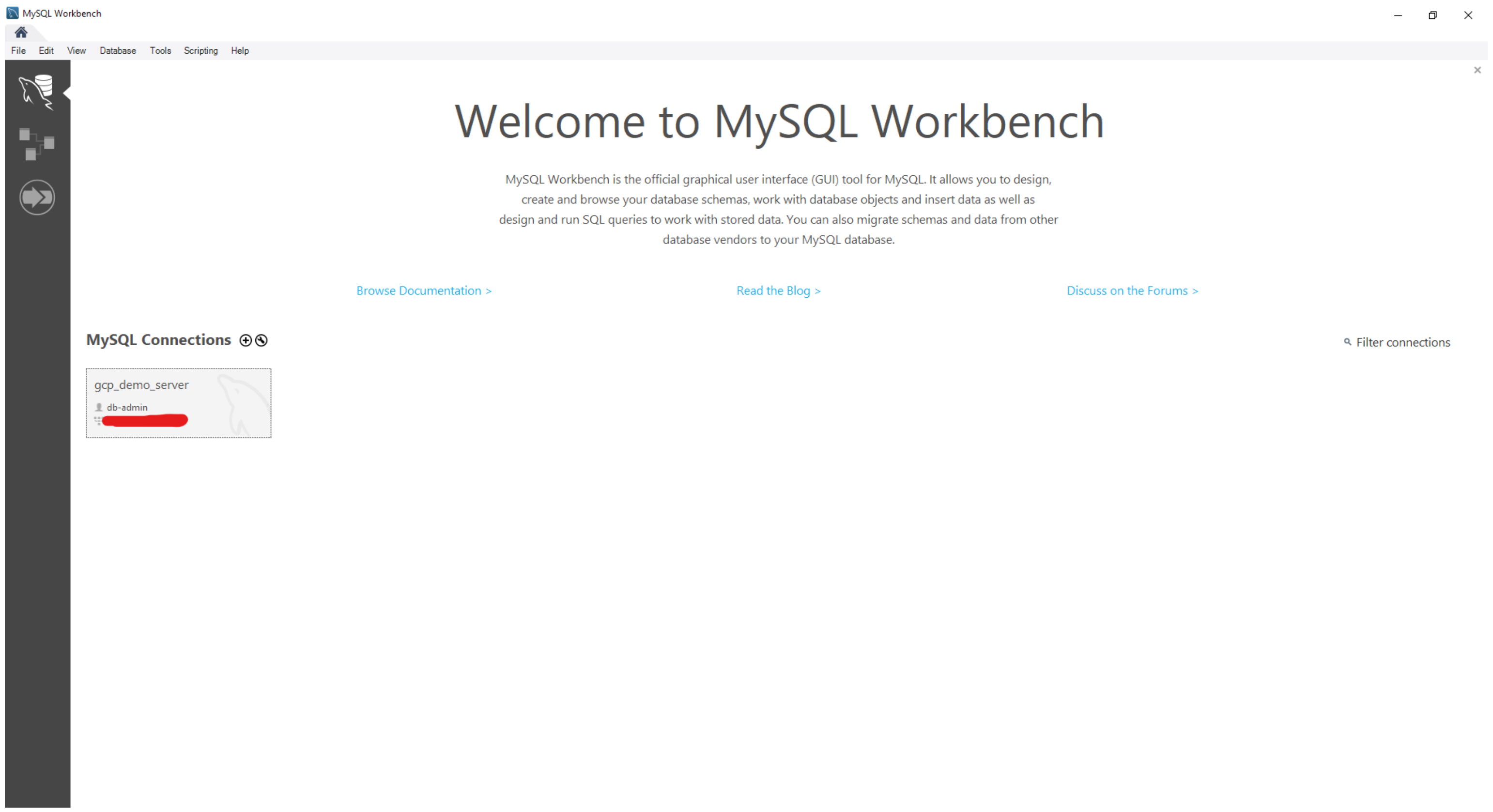
SSL Key File: Path to Client Key file for SSL.

SSL CERT File: Path to Client Certificate file for SSL.

SSL CA File: Path to Certificate Authority file for SSL.

SSL Cipher: Optional : separated list of permissible ciphers to use for SSL encryption.

5. The following instructions describe how to fill out the most important fields above:
    a. For the dropdown menu next to Use SSL, select the option, “Require and Verify CA”
    b. For SSL Key File, enter (or browse to using the … button) the path where you had saved your local copy of the, “client-key.pem”, file
    c. For SSL Cert File, enter (or browse to using the … button) the path where you had saved your local copy of the, “client-cert.pem”, file
    d. For SSL CA File, enter (or browse to using the … button) the path where you had saved your local copy of the, “server-ca.pem”, file
6. Finally, before saving the changes, click on, “Test Connection”. If you did not provide a password in the password vault, then you will receive a prompt to enter the password for the user you specified. You should then receive a successful connection message
7. In the welcome screen, you should then see your database similar to the screenshot below. You can connect to the database by either double clicking (or single clicking depending on your Operating System) on the database, or by right clicking on it and selecting, “Open Connection”

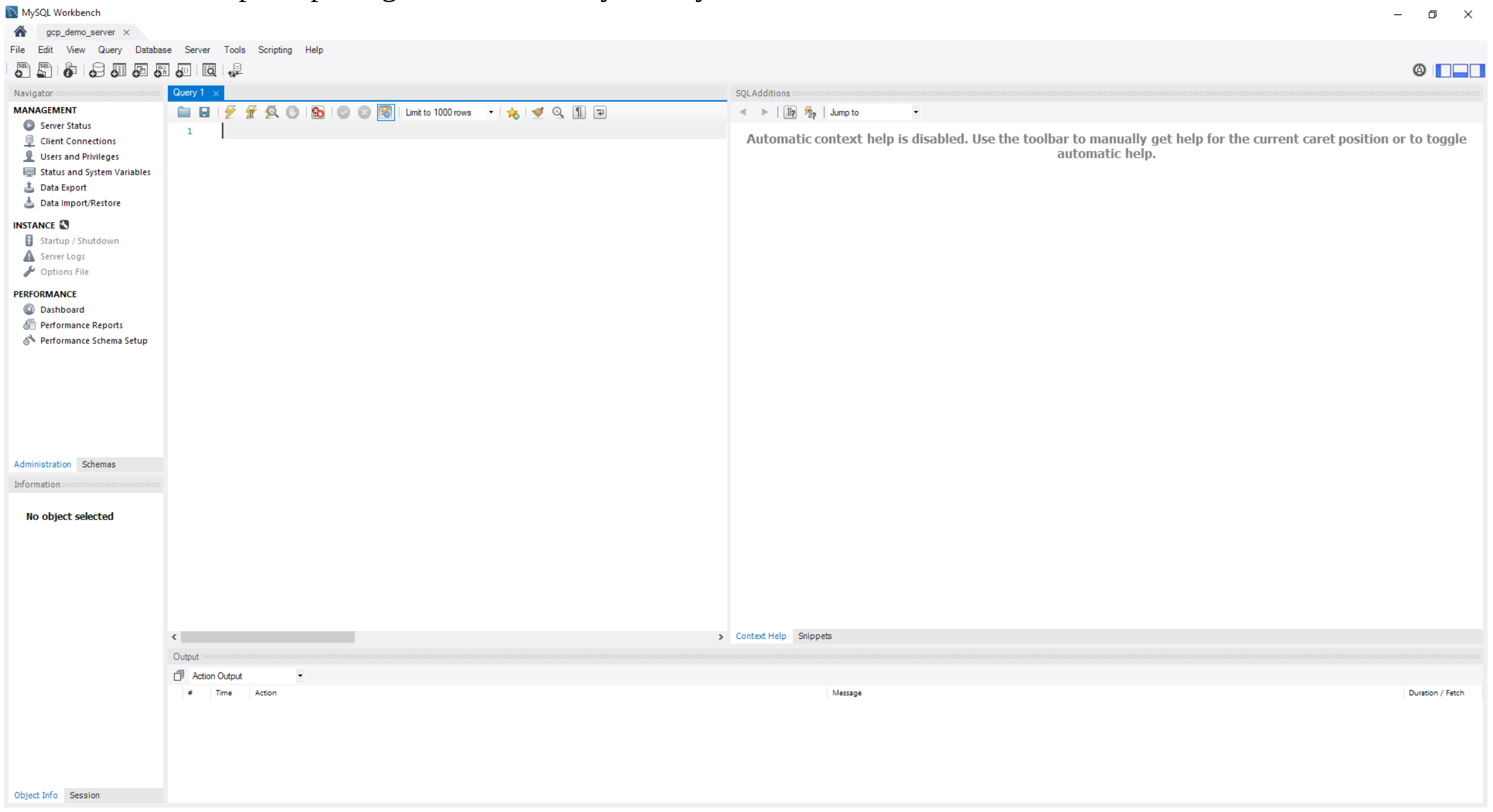


Welcome to MySQL Workbench

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.

[redacted]

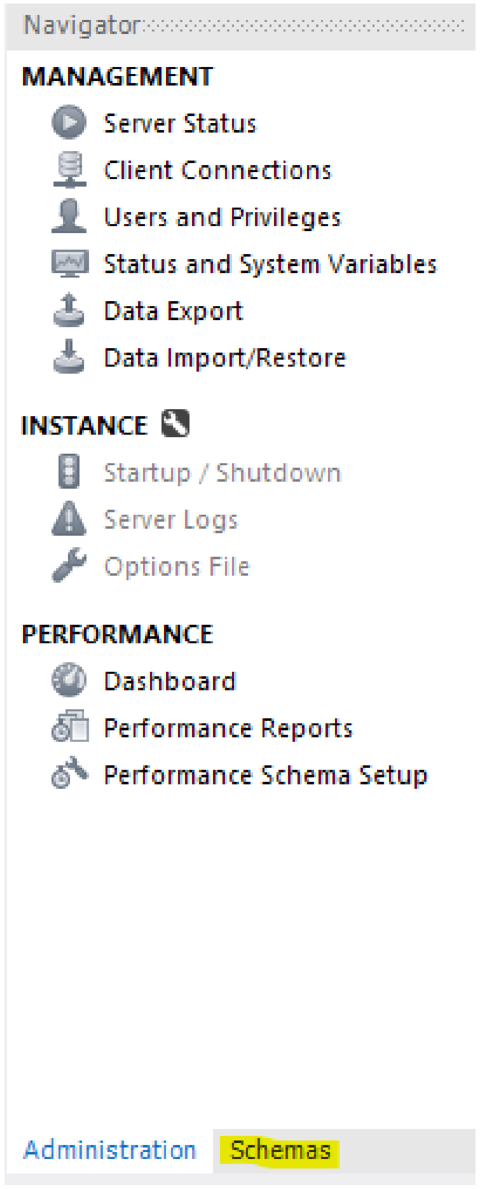
8. Upon opening the connection you may see a screen similar to the one below:



No object selected

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

9. To view the databases which your user can connect to and add tables, perform queries, etc., select the, “Schemas”, tab under the Navigator, as shown below:



1.5.1.2 Using MariaDB/MySQL Terminal Commands

A command to execute in a MySQL or MariaDB terminal is the following (without quotes): “mysql -u user -p -h host_name --ssl-ca=/path/to/server-ca.pem --ssl-cert=/path/to/client-cert.pem --ssl-key=/path/to/client-key.pem”, where user is the user you had setup in Section 1.4.1, host_name is the IP address of your GCP IP address gathered from the steps in Section

1.4.1, and "/path/to/server-ca.pem", "/path/to/client-cert.pem", and, "/path/to/client-key.pem", is the full directory path (or absolute path) to all of your locally saved ssl files. Upon pressing enter, you will be prompted for the password associated with the user you previously setup.

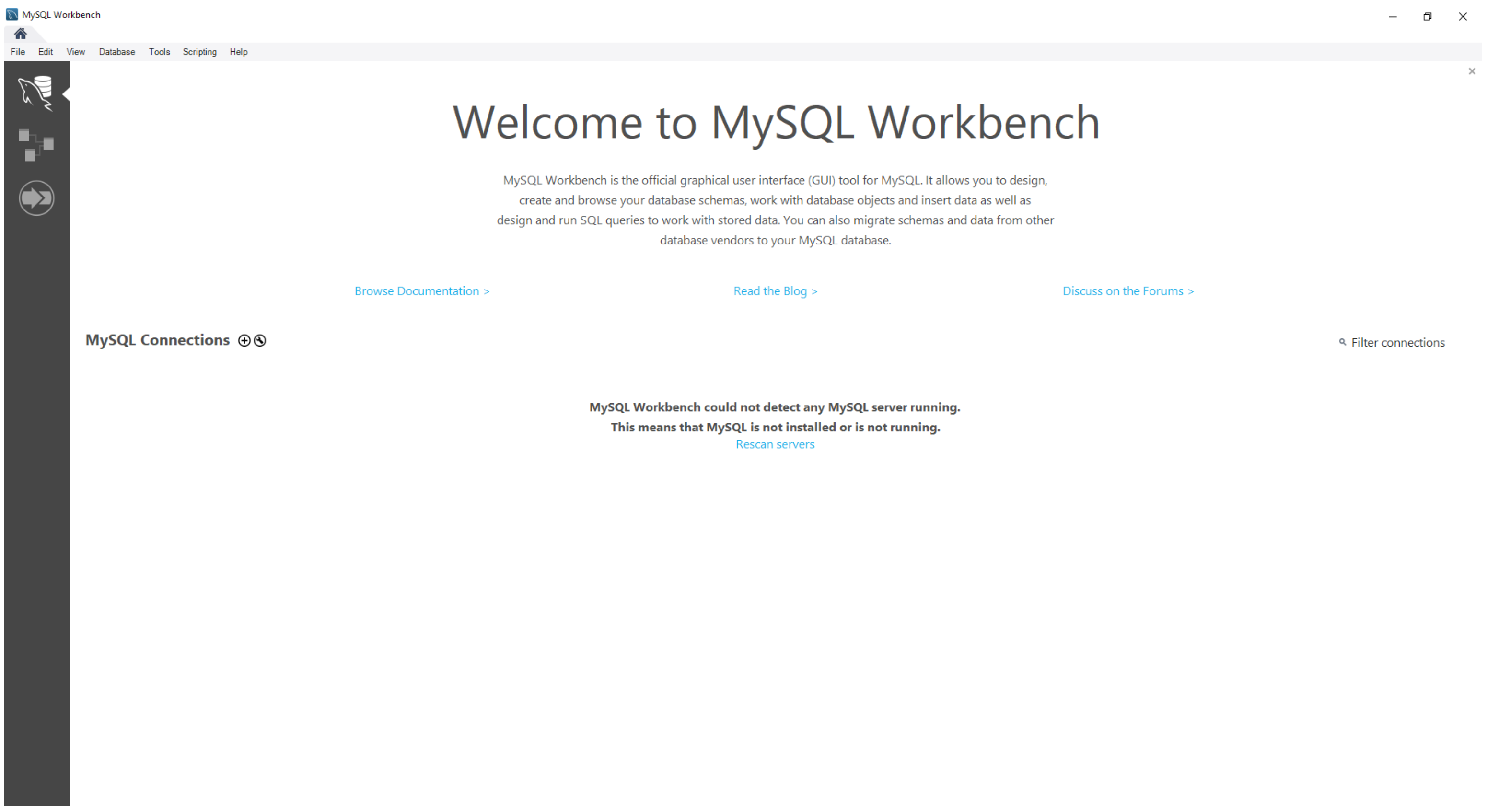
Important Notes:

- If the command is executed in the same directory as where the ssl files are located, then the full directory path (portion in /path/to), is not necessary
- On Windows based machines, the /path/to part will use \ instead of /. As an example, if the ssl files were stored in a directory in your Documents folder, then it would appear like the following: "C:\Users\User_name\Documents\ssl_files"

1.5.2 Local MySQL Connection

1.5.2.1 Using MySQL Workbench

1. Open MySQL Workbench. The interface should be similar to the following:

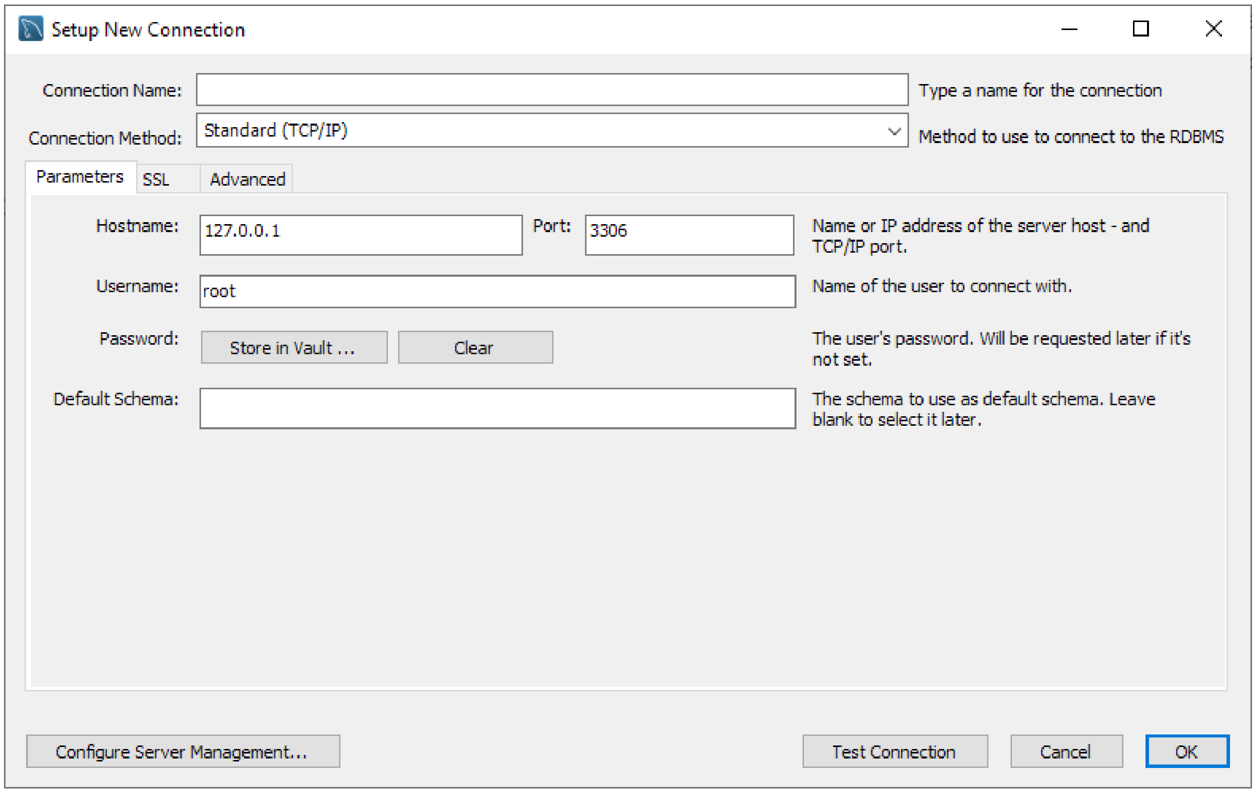


Welcome to MySQL Workbench

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.

MySQL Workbench could not detect any MySQL server running.
This means that MySQL is not installed or is not running.

2. Select the + button next to MySQL Connections. The following window should appear:



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.

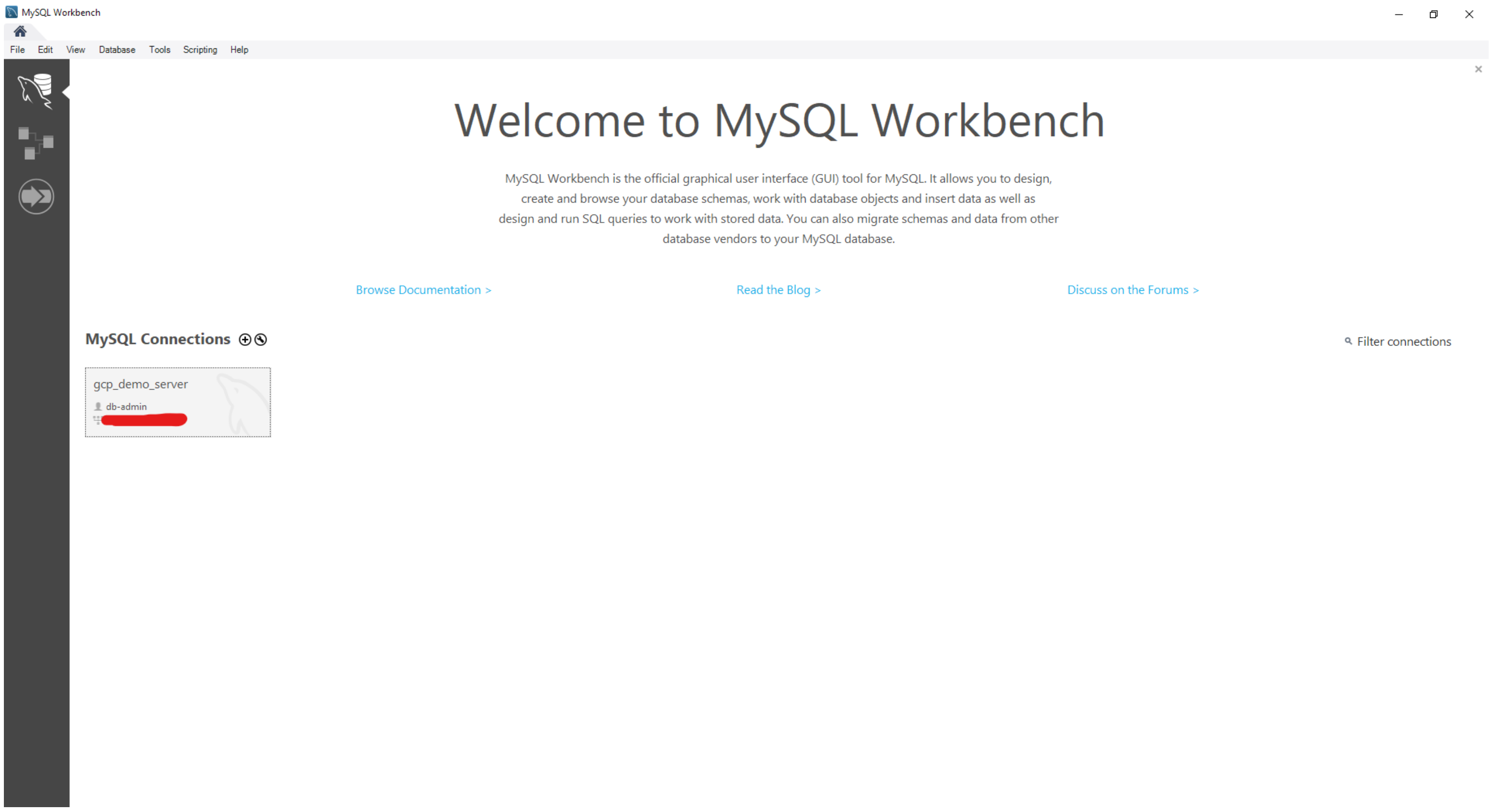
Username: root
Name of the user to connect with.

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.

3. The following instructions describe how to fill out the most important fields above:
   a. For Connection Name, type any nickname you want for these settings. In the future, you will be able to make connections easily by simply selecting the connection in the main Welcome Screen
   b. Leave Connection Method as Standard (TCP/IP)
   c. For Hostname: Leave the hostname as 127.0.0.1, or change it if you changed your MySQL server instance to a different IP Address (localhost is also a valid hostname)
   d. Leave Port as 3306, unless you changed it on your local instance
   e. Change Username to the one you used in Section 1.4.2.1
   f. Optional: For Password, select Store in Vault… and then enter the password with the password prompt
      i. (Note: Linux distributions may require the usage of a keyring on the system, most likely the Default keyring).
4. Finally, before saving the changes, click on, “Test Connection”. If you did not provide a password in the password vault, then you will receive a prompt to enter the password for the user you specified. You should then receive a successful connection message
5. In the welcome screen, you should then see your database similar to the screenshot below. You can connect to the database by either double clicking (or single clicking depending on your Operating System) on the database, or by right clicking on it and selecting, “Open Connection”



Welcome 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.

Browse Documentation >

Read the Blog >

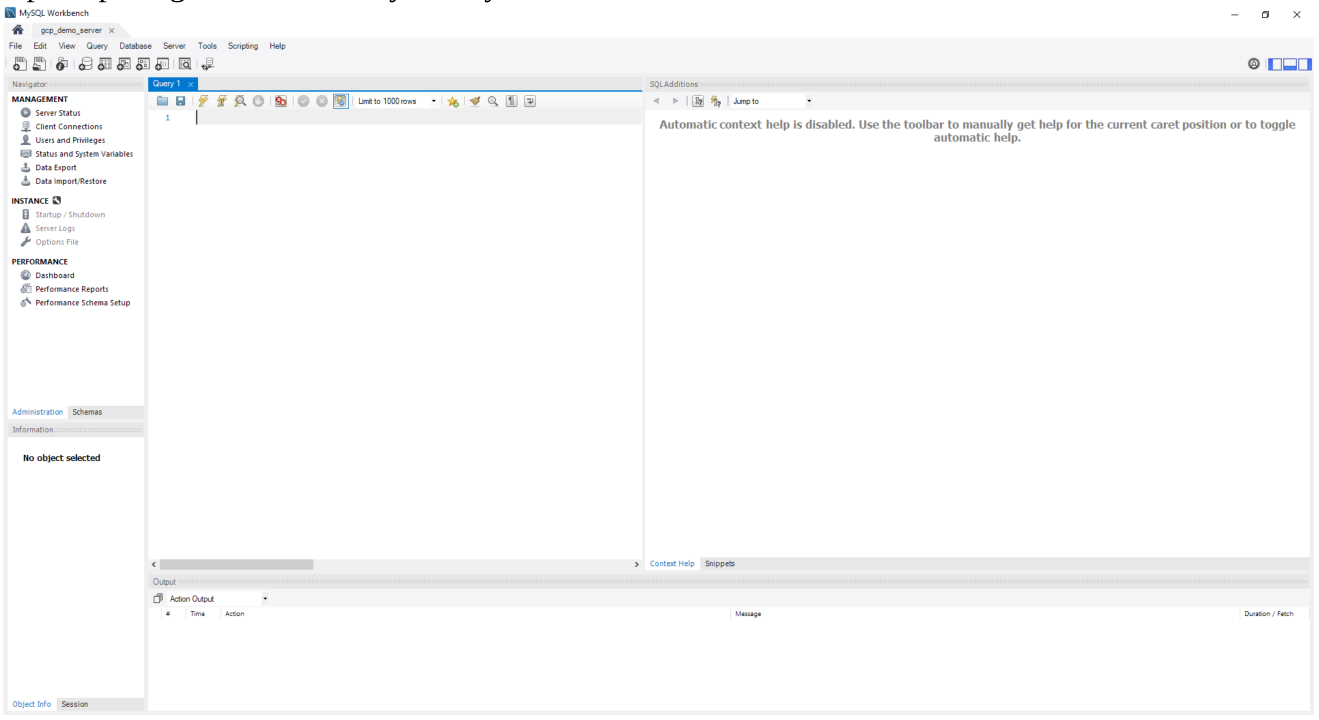
Discuss on the Forums >

MySQL Connections

Filter connections

gcp_demo_server
db-admin
[redacted]

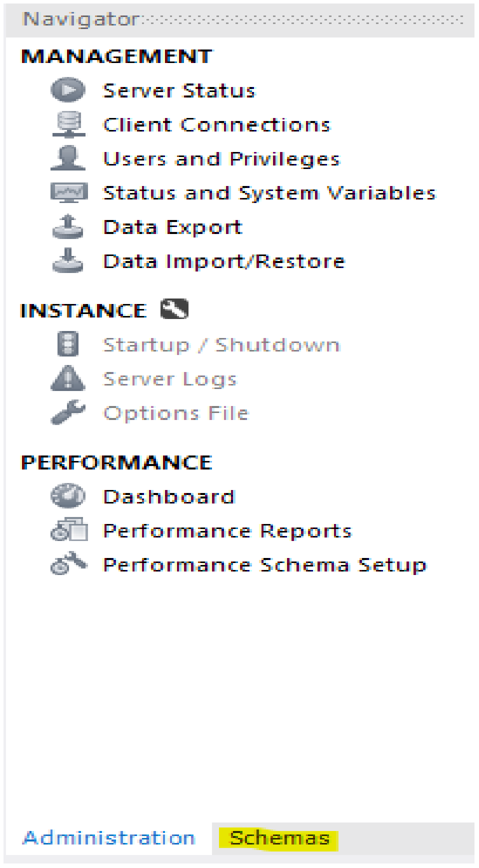
6. Upon opening the connection you may see a screen similar to the one below:



Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

No object selected

7. To view the databases which your user can connect to and add tables, perform queries, etc., select the, “Schemas”, tab under the Navigator, as shown below:



1.5.2.2 Using MariaDB/MySQL Terminal Commands

A command to execute in a MySQL or MariaDB terminal is the following (without quotes): “mysql -u user -p -h host_name”, where user is the username of the user you made in Section 1.4.2.1, and host_name can either be 127.0.0.1 or localhost.

1.6 Generating SQL Tables and Stored Procedures using Provided SQL Scripts

Open up the related files for the stored procedures and table schemas, in the DMBS of your choice. Once the files have been opened run them using the run button in the toolbar at the top of the DMBS window. Then refresh the tables on the schema browsers on the left to confirm that that schemas and stored procedures have been generated correctly. An example of how to open these files, if using MySQL Workbench while connected to an instance, is to select, “File” -> “Open SQL Script…”, and then select the necessary file. Then to execute the script, first double click on the database you want to execute the script on, and then click the first lightning icon, or "Execute..." icon.

1.7 Initial Setup for Nodemailer (Gmail + OAUTH guide, setting up the .env file for nodemailer, and other recommended services and guides for Nodemailer)

The following guide was used to setup a test Gmail Account for our development to use OAUTH for the automatic sending of mail through nodemailer: https://medium.com/@nickroach_50526/sending-emails-with-node-js-using-smtp-gmail-and-oauth2-316fe9c790a1. For this guide it is recommended you follow all steps through Step 4 and stop before, "Step 5: Write Some Code!". Then, following the .env file guidelines given in the backend/Readme.md file in your local repository, replace, "OAUTH2_CLIENT_ID", "OAUTH2_CLIENT_SECRET", "OAUTH2_REFRESH_TOKEN", and, "EMAIL_USER", with the information gathered from the guide above.

If you do not wish to use google services for nodemailer, then you can use other services, such as Twilio Sendgrid or even a regular SMTP server. More information for nodemailer and Twilio Sendgrid can be found here: https://nodemailer.com/about/ and here: https://sendgrid.com/blog/sending-email-nodemailer-sendgrid/. (Also: Twilio Sendgrid has free credits through the GitHub Student Developer pack which can be found here: https://education.github.com/pack)

1.8 Setting up Environment File (.env), and db.js File for Important Connections

In order to setup your environment file, first create a file called .env in your local repository please follow the, "Setup for .env file", guide in the Readme.md file in the backend portion, and fill in the values based on the information provided in quotes next to each variable. IMPORTANT: MAKE SURE THE ENVIRONMENT FILE IS EXACTLY NAMED (without quotes), ".env". IF IT IS NOT YOU MAY RISK PRIVATE INFORMATION BEING ACCIDENTALLY POSTED TO YOUR GITHUB AS .gitignore WILL NOT BE ABLE TO DETECT THIS SLIGHT DIFFERENCE.

1.8.2 Local MySQL Connection Differences

If you are using a Local MYSQL instance, then a few minor changes will need to be made to the .env file and db.js. First, in the .env file, remove the variables and values for: MYSQL_CA, MYSQL_KEY, and MYSQL_CERT. Then in the db.js file remove the ssl : {} portion, along with the contents inside the brackets.

1.9 Running the Components Locally to Ensure Proper Frontend/Backend Interaction

Once you have set everything up following the sections above, you can finally do proper testing. The following steps outline a simple first time testing procedure:

1. Ensure that your MySQL server is up and running (either local or cloud based)
2. Then in a terminal (either through a UNIX/Linux terminal emulator, Windows Powershell, or an IDE provided shell) navigate to your backend directory in your local repository if your shell was not already located there
3. Then issue the command, “npm run dev-backend”
4. Upon your first run, you will be prompted to create an admin account if no admin accounts were added to your database. Simply fill out the prompted information as needed (Note: This will have you fill in a password which may appear in plain text since we are not using any UNIX/Linux like system methods to hide the password. Make sure that an admin account is created in a private setting on a private machine. This password will be hashed and salted in the database, so it will not appear as plaintext in the future, and can be safely accessed in the future)
5. When finished, with the above step, press Ctrl+C to stop the process, and clear your terminal or start a new session. Then, run the command, “npm run dev-backend”, again. The admin account prompt should no longer appear
6. In a separate terminal or shell session, navigate to your frontend directory in your local repository
7. In this shell run the command, “ng serve”. Wait for everything to compile (refer to Section 1.3 if needed
8. Then in a browser navigate to http://localhost:4200. From there, navigate to the Login page by selecting the User icon in the top right and selecting the Login button
9. Then login using the email and password you setup earlier. You should be taken to a leader dashboard screen

Here are a few potential errors you can fix if you run into them:

- If you accidentally interrupted the admin addition process in the terminal, and the prompt to create an admin account does not come up again, then an account can be manually made by first issusing an INSERT statement into the user table, and then using our Forgot Your Password? functionality to reset your account password. Note that the account you create must be made with a valid, real, email address if you intend to use the password reset functionality

- If the frontend has difficulty connecting to backend, then make sure the port number in your front end service files matches the port your backend is running on. In our files, the variable is commonly, “const BACKEND_URL = 'http://localhost:45213/api/…';”, where 45213 is the port and /api/… is a path to a route on the backend

Lastly, unit tests (which can be found in backend/test/app.test.js), can be ran using, “npm test”

2 Recommended Resources For Development and Future Deployment

The following guides below are not extensive instructions on how to setup a machine for deployment, but are rather recommended external guides and resources for future deployment.

2.1 External Learning Resources

For the toolset we used, here are a few guides and documentation sources to get you familiarzed with the items in the MEAN stack (with M meaning MySQL instead of MongoDB in this instance):

- MySQL
  - https://dev.mysql.com/doc/ and https://dev.mysql.com/doc/refman/8.0/en/ - for documentation
  - https://mariadb.com/kb/en/documentation/ - for MariaDB
  - https://sqlbolt.com/ - great tutorial set for SQL queries
- Express
  - https://expressjs.com/en/starter/hello-world.html - getting started guide
  - https://expressjs.com/en/guide/routing.html - guides to using expressjs
- Angular
  - https://angular.io/docs
- Node.js
  - https://nodejs.org/en/docs/

2.2 Setup Code for Deployment Guides

The following guide is a great starting point for learning about ways to prepare your code for production: https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express_Nodejs/deployment

2.3 Simple Setup on a VPS (like DigitalOcean, AWS, etc.)

For a VPS, it is recommended to use a nginx (or Apache) reverse-proxy for backend (api) portions, and to serve the Angular frontend components through a /var/www directory. Here are a few guides to help you get started with research:

- Backend API
  - https://www.tecmint.com/nginx-as-reverse-proxy-for-nodejs-app/ - reverse proxy brief tutorial
  - https://medium.com/@utkarsh_verma/configure-nginx-as-a-web-server-and-reverse-proxy-for-nodejs-application-on-aws-ubuntu-16-04-server-872922e21d38 - another reverse proxy tutorial
  - https://www.digitalocean.com/community/tutorials/how-to-secure-nginx-with-let-s-encrypt-on-ubuntu-18-04 - securing nginx with ssl
- Frontend
  - https://medium.com/@hmurari/serving-angular-app-from-behind-a-nginx-web-server-3579df0b04b1 - brief tutorial on serving frontend
- General NGINX documentation:
  - http://nginx.org/en/docs/
- General APACHE documentation:
  - https://httpd.apache.org/docs/
  - https://www.digitalocean.com/community/tutorials/how-to-install-linux-apache-mysql-php-lamp-stack-ubuntu-18-04#step-1-%E2%80%94-installing-apache-and-updating-the-firewall – brief Apache installation guide
  - https://www.digitalocean.com/community/tutorials/how-to-secure-apache-with-let-s-encrypt-on-ubuntu-18-04 - securing apache with ssl