CONFIGURING GOORM FOR MYSQL/PHPMYADMIN

Credits: *these notes were originally prepared by a colleague, Ian Tomey, and modified by myself.*

These notes expand upon the Goorm introduction given last week, to show you how to setup a MySQL database server environment. They also explain how to setup the database administration tool PHPMyAdmin, for which we need to install the **Apache2 web server**.

You will only have to do this once. Once installed and configured, your container will run MySQL and PHPMyAdmin without any manual intervention needed.

CREATE CONTAINER

First step is to create a container on Goorm. We covered this last week and hopefully you have a container set up already. Please see the week 1 notes at https://nwcourses.github.io/COM518/topic1.html.

CONFIGURING MYSQL

At the command prompt in Goorm, enter `mysql-ctl start`.

This will start the MySQL database server as a background process.

```
root@goorm:/workspace/MyNodeContainer# mysql-ctl start

MySQL 5.7 database added.  Please make note of these credentials:

       Root User: root
   Database Name: mysql

 * Starting MySQL database server mysqld
No directory, logging in with HOME=/
   ...done.
root@goorm:/workspace/MyNodeContainer#
```

Note that the root (admin) database user cannot be used for security reasons, so we will go into MySQL and create a new user. So enter `mysql` at the command prompt to start the MySQL client:

```
root@goorm:/workspace/MyNodeContainer# mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 4
Server version: 5.7.33-0ubuntu0.18.04.1 (Ubuntu)

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

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

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

mysql>
```

From the MySQL command prompt, we create the new user. We then create a database for the new user to use and grant all privileges on that database to the new user.

Enter the queries exactly as below. Change the username and password to whatever you want it to be. This example is creating the database user ‘myuser’ with a password of ‘mypassword’, creating database ‘waddb’, and granting user ‘myuser’ access to the database ‘waddb’:

```
CREATE USER 'myuser' IDENTIFIED BY 'mypassword';

CREATE DATABASE waddb;
GRANT ALL PRIVILEGES ON waddb.* TO 'myuser’;
FLUSH PRIVILEGES;

exit;
```

The

`exit;`

exits the MySQL client and returns you to the command prompt.

INSTALLING PHPMYADMIN (MYSQL WEB FRONTEND)

PHPMyAdmin is a web-based administration tool for a MySQL database. We need to install it to our container. Note that to use PHPMyAdmin, you need the Apache2 web server installed, as PHPMyAdmin is written in the PHP programming language, which works with Apache2. Apache2 is a general-purpose HTTP server; just like an Express application it processes HTTP requests and delivers responses, but does a whole lot more. It will run on port 80, rather than port 3000 (we will consider this further later on).

To install PHPMyAdmin (and Apache2), please enter the following from the container's command prompt:

```
sudo apt-get update

sudo apt-get install -y phpmyadmin
```

During the installation process you will see this:

```
Configuring phpmyadmin
----------------------

Please choose the web server that should be automatically configured to run phpMyAdmin.

  1. apache2  2. lighttpd

(Enter the items you want to select, separated by spaces.)

Web server to reconfigure automatically:
```

Enter 1 for the Apache2 webserver.

Later, you will see this prompt:

```
Configuring phpmyadmin
----------------------

The phpmyadmin package must have a database installed and configured before it can be used. This can be optional
ly handled with dbconfig-common.

If you are an advanced database administrator and know that you want to perform this configuration manually, or
if your database has already been installed and configured, you should refuse this option.
Details on what needs to be done should most likely be provided in /usr/share/doc/phpmyadmin.

Otherwise, you should probably choose this option.

Configure database for phpmyadmin with dbconfig-common? [yes/no]
```

Select "no". (as we have already created a user)

When the installation has finished we have to make a slight patch to phpMyAdmin to make it work correctly. Copy and paste this directly into the Goorm command prompt:

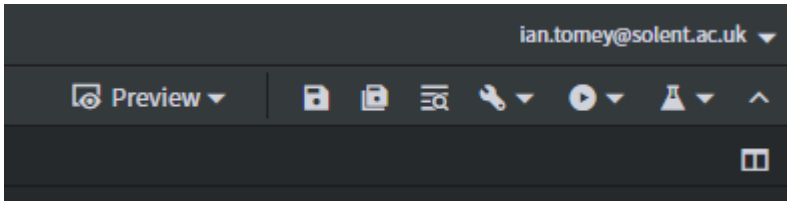
```
sudo sed -i "s/|\s*\((count(\$analyzed_sql_results\['select_expr'\]\)/| (\
1)/g" /usr/share/phpmyadmin/libraries/sql.lib.php
```

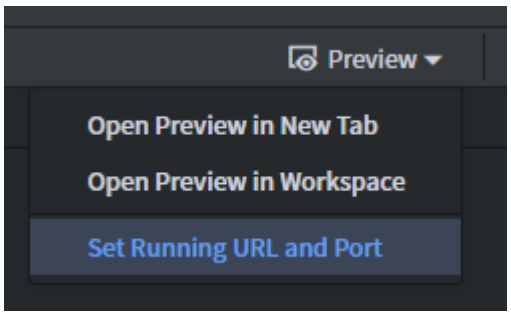
Then start the Apache webserver with

```
service apache2 start
```

RUNNING PHPMYADMIN

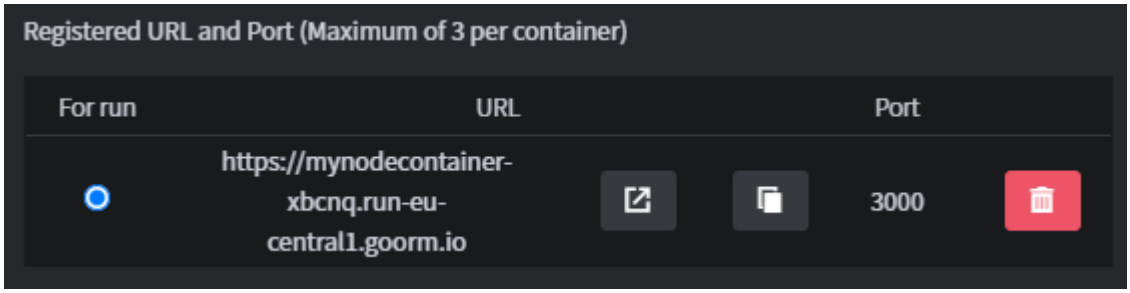
In Goorm, flick down the preview tab on the top right and select 'Set Running URL and Port'



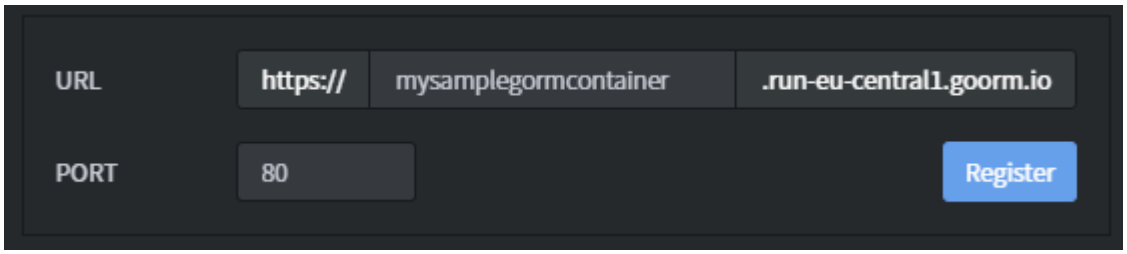


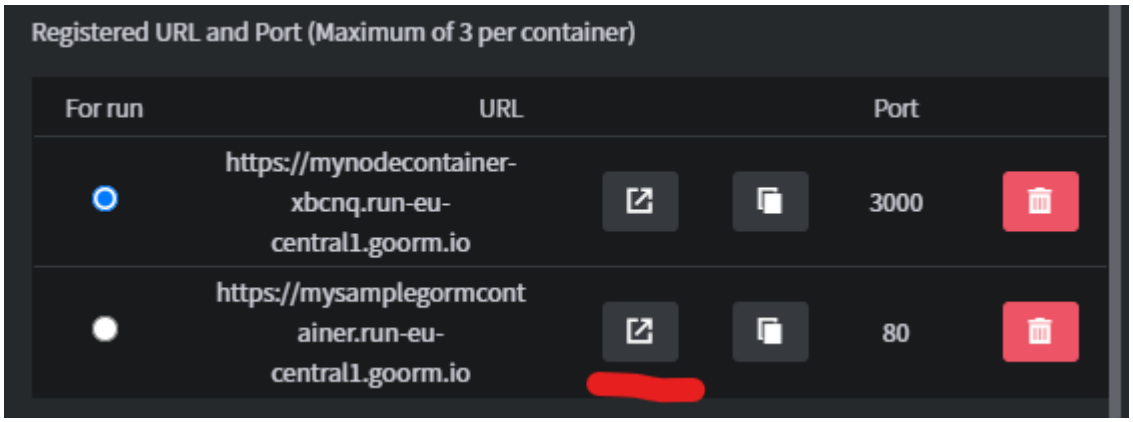
This page shows allows you to register a URL on the internet that exposes an additional port into your container. Note that as we saw last week, Goorm sets up one URL for you by default, which exposes your Node application on port 3000. However, Apache is running on a different port, port 80, so we need to setup a separate url to expose Port 80 of our container.

As we saw last week, a UR Lis already created with the port of 3000 (default for a node project).

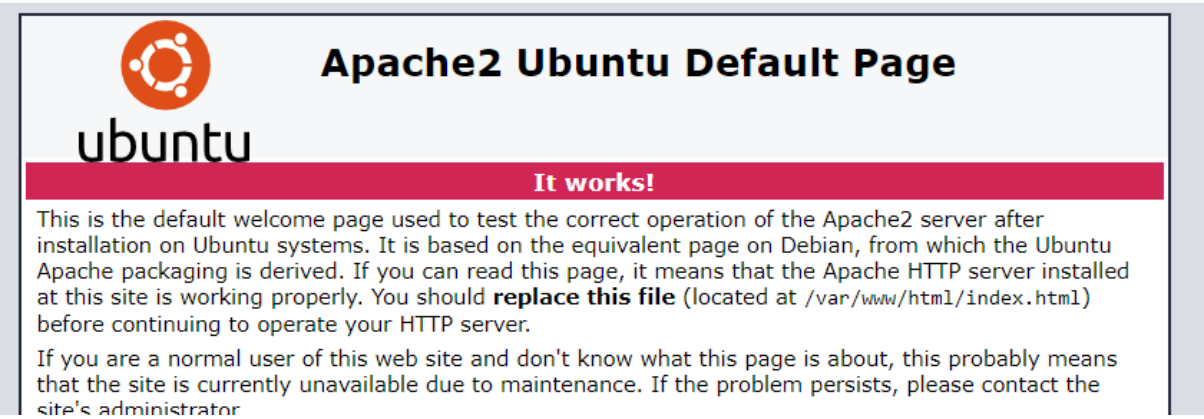


But we also need to communicate with the Apache webserver on port 80. So,, from the “Set running URL and port” screen, set up a URL on port 80 as shown below (choose the part before `.run-eu-central1.goorm.io`, note this has to be unique as it’s a public URL, so maybe choose a nickname personal to you) and hit register.





Launch the URL with the highlighted button in your browser. It should give us the default Apache2 page as below

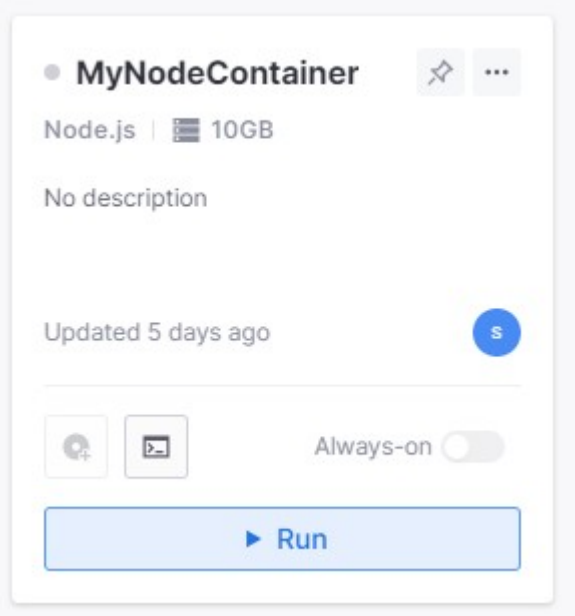


Now add phpmyadmin (all lower case) to the URL and you will have access to phpMyAdmin, using the username and password we set up for MySQL in an earlier step (myuser/mypassword in the example given above):

mysamplegormcontainer.run-eu-central1.goorm.io/phpmyadmin/

SETTING SERVICES TO AUTO RUN AT STARTUP

We can configure our Goorm container to automatically run the Apache web server and the MySQL server at startup. To do this, access the dashboard (the page which first appears when you login to Goorm) and select your container.

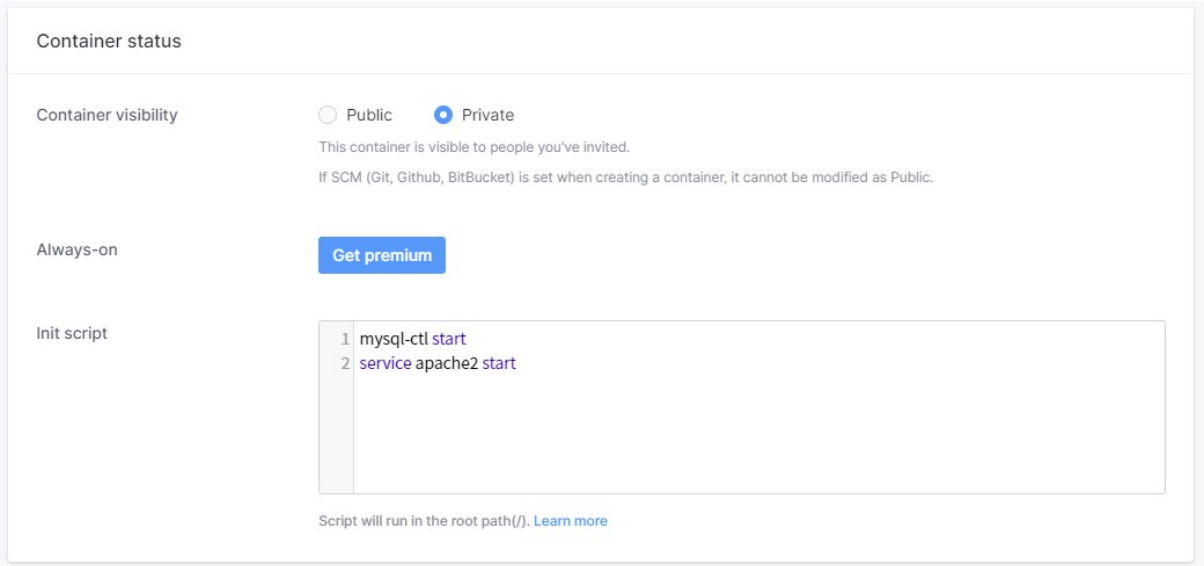


Click the three dots in the top right.

Edit the init script (see below) to start up MySQL and Apache by adding the following:

```
mysql-ctl start

service apache2 start
```



This will automatically startup MySQL and Apache2 when the container boots. Now whenever yourself *or somebody who have shared the container with who has root level access* the services should be started with the container.