Networking Exercise

Goal:

The goal of this exercise is to create a user-defined network for a web application called Drupal.

Instructions:

Create a Network for the Drupal Web Application

Create a Docker bridged network named "drupal".

```
docker network create drupal
```

Create a Database Container.

First, create a volume that will be used to store the database data. Name the volume "db".

```
docker volume create db
```

Next, start a container named "db" based on the "postgres" image with a tag of "11.5". Make sure it is connected to the "drupal" network. Also, mount the "db" volume to "/var/lib/postgresql/data" inside the container.

Create a database named "drupal" that can be accessed by a "drupal" user with a password of "pw123". To do that, set the following environment variables when starting the container:

POSTGRES_USER=drupal

POSTGRES_PASSWORD=pw123

POSTGRES_DB=drupal

(NOTE: The environment variables were determined by looking at the documentation for this image as found on Docker Hub: https://hub.docker.com/_/postgres)

Here is the command: (NOTE: Type this all on one line as this is a single command.)

```
docker run -d --network drupal --name db -e POSTGRES_DB=drupal
-e POSTGRES_USER=drupal -e POSTGRES_PASSWORD=pw123
--mount src=db,dst=/var/lib/postgresql/data postgres:11.5
```

Inspect the "drupal" network to make sure the container is attached to it. You should see the "db" container listed in the "Containers" section of the output.

```
docker network inspect drupal
```

Create a Drupal Application Container

Start a container named "drupal" based on the "drupal" image with a tag of "8.7.7". Make sure it is connected to the "drupal" network. Also, publish port 80 on the host and map it to port 80 in the container.

```
docker run -d --network drupal --name drupal -p 80:80 drupal:8.7.7
```

Inspect the "drupal" network to make sure the container is attached to it. You should see the "drupal" container listed in the "Containers" section of the output.

```
docker network inspect drupal
```

Connect the Application to the Database

If you are running Docker on your local machine, then open up a web browser on your local machine and enter "http://localhost" into your web browser.

However, if the Docker host machine is a remote machine and you want to connect to from another system, then you'll need the IP address of your docker host machine. Determine the IP address of your Docker host machine and enter that into your web browser.

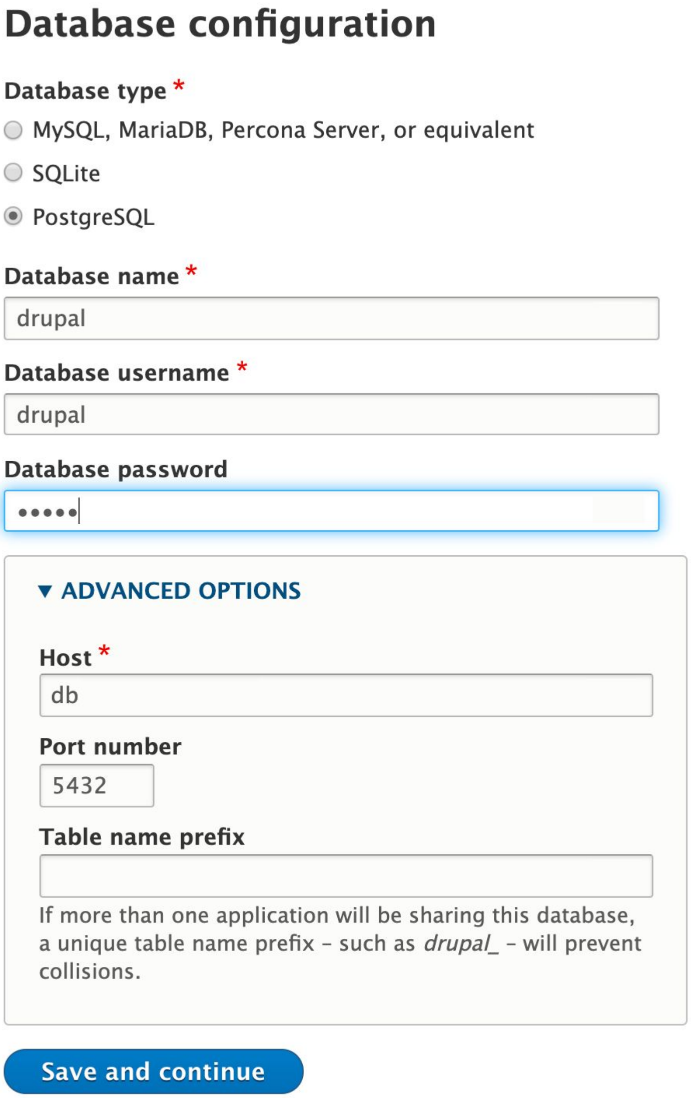
One way to get the IP address of your Docker host system is to use the "ip a" command:

```
ip a
```

(NOTE: If "`ip a`" doesn't work, try using the "`ifconfig`" command.)

Answer the installation prompts as follows.

Choose Language: English
Choose Profile: Standard
Set up Database:
- Select "PostgreSQL"
- Database name: drupal
- Database username: drupal
- Database password: pw123
- Click "Advanced Options."
  - Host: db
  - Port number: 5432
- It should look like this:



Database configuration

Database type *

○ MySQL, MariaDB, Percona Server, or equivalent
○ SQLite
● PostgreSQL

Database name *
drupal

Database username *
drupal

Database password
•••••

▼ ADVANCED OPTIONS

Host *
db

Port number
5432

Table name prefix

If more than one application will be sharing this database, a unique table name prefix – such as *drupal_* – will prevent collisions.

Click "Save and continue".

Configure Site:

Site name: Test
Site email address: root@localhost.localdomain
Username: admin
Password: admin123
Confirm Password: admin123
Click "Save and continue".

You will now be presented with Drupal web application.