Actionable docker

Using docker to run packages locally.

Docker let's you do a lot of things, here's my tutorial on the same -

This tutorial is on `actionable docker` to start `packages` locally.

Installing Docker

`Docker GUI` is the easiest way to get off the ground.

You can find instructions to install docker on https://docs.docker.com/engine/install/

At the end of the installation, you need to make sure you're able to run the following command
-

What are we using docker for?

Docker let's you do a lot of things.

It let's you `containerise` your applications.

It let's you run other people's `code + packages` in your machine.

It let's you run common software packages inside a `container` (For eg - Mongo, Postgres etc)

Where can we get packages from?

Just like you can push your `code` to Github/Gitlab.

You can push `images` to `docker registries`

Common commands to know

1. docker run
2. docker ps
3. docker kill

Running an image

1. Running a simple image

Let's say you wan't to run MongoDB locally https://hub.docker.com/_/mongo

```
docker run mongo
```
Copy

You will notice you can't open it in `MongoDB Compass` .

Adding a port mapping

The reason is that you haven't added a `port mapping`

```
docker run  -p 27017:27017 mongo
```
Copy

Starting in detached mode

Adding `-d` will ensure it starts in the background

```
docker run -d -p 27017:27017 mongo
```
Copy

Inspecting a container

```
docker ps
```
Copy

This will show you all the containers you are running.

Stopping a container

```
docker kill <container_id>
```
Copy

Will stop the container that you are running

In the end, this is the flow of commands -

Common packages

Mongo

```
docker run -d -p 27017:27017 mongo
```
Copy

Postgres

```
docker run -e POSTGRES_PASSWORD=mysecretpassword -d -p 5432:5432 postgres
```
Copy

The connection string for this postgres would be

```
postgresql://postgres:mysecretpassword@localhost:5432/postgres
```
Copy

▼ Code to test it out

Copy

```
// Import the pg library
const { Client } = require('pg');

// Define your connection string (replace placeholders with your actual dat
const connectionString = 'postgresql://postgres:mysecretpassword@localhost:

// Create a new client instance with the connection string
const client = new Client({
  connectionString: connectionString
});

// Connect to the database
client.connect(err => {
  if (err) {
    console.error('connection error', err.stack);
  } else {
    console.log('connected to the database');
  }
});

// Run a simple query (Example: Fetching the current date and time from Pos
client.query('SELECT NOW()', (err, res) => {
  if (err) {
    console.error(err);
  } else {
    console.log(res.rows[0]);
  }

  // Close the connection
  client.end();
});
```