**Questions:**

Screen shot of docker containers:

A black background with white text

Description automatically generated

Screenshot of Wordpress application in browser:

A screenshot of a computer

Description automatically generated

**As you've noticed we've put some credentials in the docker-compose file using environment variables. Imagine that this file ends up in source control. What is the potential security risk?**

There is a high security risk with having secrets contained within the docker-compose file. If this ends up in source code, anyone with access to the source code can also gain access to the SQL container we created. Any individuals gain that have access to the source code repository, can retrieve the Docker Compose file and obtain the sensitive credentials. Or, if the repository is public, an attacker could easily gain unauthorized access to the SQL container and compromise the database, leading to data breaches, unauthorized modifications, or other malicious activities.

**Three development practices that GitGaurdian has suggests**

1. Name sensitive files in .gitignore and .npmignore

In our case, we would add the docker-compose file in the .gitignore, so it is not added to the repository.

1. Store your secrets in a safer place

GitGaurdian recommends using local environment variables, storing secrets encrypted in a git repository, or using a "secrets as a service" solution like Hashicorp's Vault or Square's Keywhiz.

1. Whitelist your Ips

In our case, we could attach the docker containers to a network and only allow communication to the SQL container from the wordpress container by only allowing the IP of the wordpress container.

**Using an environment file**

We can protect our secrets by using a .env file. We can store our secrets here, and reference them in our docker-compose file using ${variable}. Then, we can add the .env file to our .gitignore so that it’s not included in the code repository. By doing this, we can keep the secrets private on our local machine. Notice the difference in the docker-compose file and the new .env file:

.env file:

A screen shot of a computer

Description automatically generated

New docker-compose file:

A screenshot of a computer program

Description automatically generated

The commands to run the docker-compose file stays the same, and the containers boot up the same as before:

A screenshot of a computer

Description automatically generated