Microservice name: spring-boot-postgres

```
public class TestConnection {

	public static void main(String args[]) throws Exception {
		String ipAddr = InetAddress.getLocalHost().getHostName();
		System.out.println("Printing IP address of the host " +
ipAddr);
		Map<String, String> env = System.getenv();
		for (String envName : env.keySet()) {
			System.out.format("%s=%s%n", envName,
env.get(envName));
		}
		Thread.sleep(10000);
		boolean connected = false;
		while (!connected) {
			try {
				// Note the way the postgres container is used
here. (db - database host/service name)
				String url = "jdbc:postgresql://db:5432/test?
autoReconnect=false&useSSL=false";
				String user = "postgres";
				String password = "123";
				System.out.println("Connecting to URL " +
url);
				// Load the Connector/J driver
				Class.forName("org.postgresql.Driver");
				// Establish connection to PostgreSQL
				Connection conn =
DriverManager.getConnection(url, user, password);
				if (conn != null) {
					System.out.println("Connection was
successful");
					connected = true;
				}
			} catch (Exception e) {
				System.err.println("Error connecting to
database");
				e.printStackTrace();
				Thread.sleep(5000);
			}
		}
	}
}
```

Dockerfile

```
FROM kharvinagaraj1/ubuntu-openjdk11:1.0

#Author of the Docker File
# MAINTAINER Pictolearn Note: MAINTAINER has been deprecated for
LABEL,
```

```
# LABEL is a key value pair
LABEL "Maintainer"="Nagaraj S Kharvi"

ADD . /usr/local/spring-boot-postgres
RUN cd /usr/local/spring-boot-postgres && mvn install
CMD ["java", "-cp", "/usr/local/spring-boot-postgres/target/spring-
boot-postgres-0.0.1-SNAPSHOT.jar.original",
"com.example.demo.connection.TestConnection"]
```

Above image is build by these steps

* Image is pulled from docker hub
* Microservice gets copied to docker image directory
* Jar gets generated once we build the image
* Calls database connection using db service