root@test:~/JavaWebAppp# history

1 ls

2 mvn

3 git clone https://github.com/ramannkhanna2/JavaWebAppp.git

4 ls

5 cd JavaWebAppp/

6 ls

7 apt install maven

8 apt update -y

9 apt install maven

10 ls

11 mvn clean package

**1. Verify the Installed Java Version**

Run the following command to check the current Java version on your system:

bash

Copy code

java -version

If it shows a version lower than 17, you'll need to install JDK 17.

**2. Install JDK 17**

If your system doesn't have JDK 17 installed, you can download and install it. Depending on your operating system, the steps may differ:

**On Ubuntu (or Debian-based systems):**

bash

Copy code

sudo apt update

sudo apt install openjdk-17-jdk

**3. Run the Spring Boot Application**

Once the JAR file is on the server and you have JDK 17 installed, you can start the Spring Boot application using the following command:

bash

Copy code

java -jar /root/JavaWebAppp/target/javaazurewebapp-0.0.1-SNAPSHOT.jar --server.port=8081

This will start your Spring Boot application with its embedded server (e.g., Tomcat). By default, it will listen on port 8080, unless specified otherwise in the application.properties or application.yml file.

**4. Run in the Background (Optional)**

If you want your Spring Boot application to run in the background, you can use nohup or screen.

**Using nohup:**

nohup java -jar /root/JavaWebAppp/target/javaazurewebapp-0.0.1-SNAPSHOT.jar --server.port=8081 &

This command will run the application in the background and you can safely log out without stopping the application. The output will be written to a file called nohup.out in the current directory.

Note : make sure open the port in firewall ….

Create and deploy Azure Resource Manager templates

* Go to vm and to automation >> download the template and parameter file ..
* Install az cli on local and deploy templates using az cli ..

az deployment --template-file template.json --parameters parameters.json

OR DEPLOY IT USING UI USING TEMPLATESPEC SERVICE