

### ## Prerequisite

Docker: Ensure Docker is installed and running on your local machine.

Java 21: Ensure Java 21 is installed and configured on your system.

To verify your Java installation, run:

```
**java -version**
```

Docker Hub Account: Make sure you have a Docker Hub account and are logged in.

To verify Docker is running and you're logged in:

```
**docker info**
```

### Setting Up the Project

Open Terminal: Navigate to the project's root directory.

```
**cd /path/to/your/project**
```

### ## Generate jar artifacts

Build the Project: Run the following command to build the project and generate the necessary JAR files:

```
**./gradlew build**
```

This command will compile your Java code, run tests, and package the application into a JAR file, which is required for creating Docker images.

### ## Generate docker images

Verify Docker Setup: Ensure Docker is up and running and that you are signed in to your Docker Hub account.

Build and Push Docker Images:

Build the Hello service Docker image:

```
**docker build -t hello ./hello**
```

Push the Hello service Docker image to Docker Hub:

```
**docker push <dockerhub-username>/hello**
```

Replace <dockerhub-username> with your actual Docker Hub username.

Build the World service Docker image:

```
**docker build -t world ./world**
```

Push the World service Docker image to Docker Hub:

```
**docker push <dockerhub-username>/world**
```

Replace <dockerhub-username> with your actual Docker Hub username.

## To run application in minikubestall Minikube:

Install Minikube based on your operating system.

```
**brew install minikube**
```

Start Minikube:

After installation, start Minikube by running:

```
**minikube start**
```

Install kubectl:

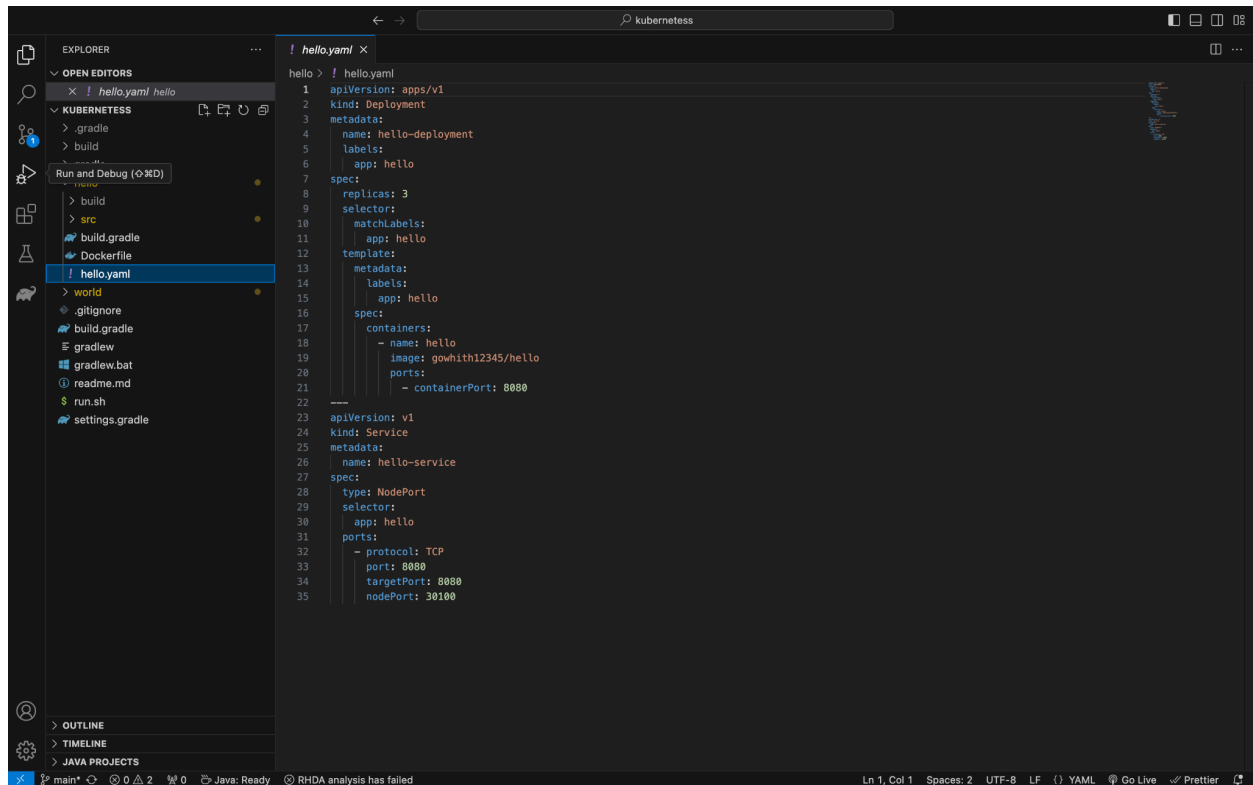
If kubectl is not installed, install it by running:

```
**minikube kubectl -- get po -A**
```

Deploy the Services:

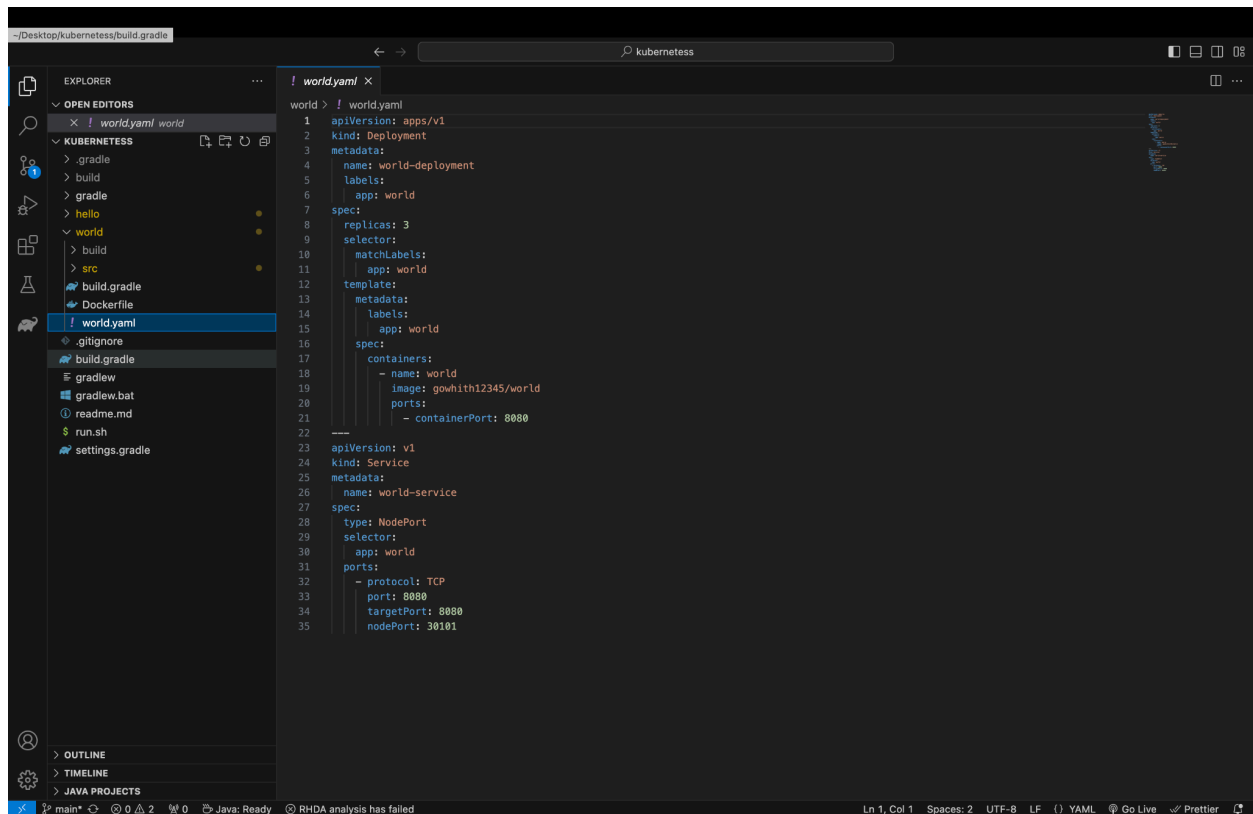
Deploy the Hello service:

```
**kubectl apply -f ./hello/hello.yaml**
```



Deploy the World service:

**\*\*kubectl apply -f ./world/world.yaml\*\***



Monitor Pods and Services:

Wait for the pods to warm up and come to a ready state. You can monitor the status by running:

**\*\*kubectl get all\*\***

Obtain Service URLs:

Get the URL for the Hello service:

**\*\*minikube service hello-service -n default --url\*\***

Copy the output URL and replace it in the `run.sh` script under the `HELLO_URL` variable.

Get the URL for the World service:

**\*\*minikube service world-service -n default --url\*\***

Copy the output URL and replace it in the `run.sh` script under the `WORLD_URL` variable.

Execute the `run.sh` Script:

Once the URLs have been set, run the `run.sh` script to make API calls to both the Hello and World services and print the combined "Hello World" message in the terminal:

Convert `run.sh` to an Executable Script

Create the `run.sh` File: Copy the above script into a file named `run.sh`.

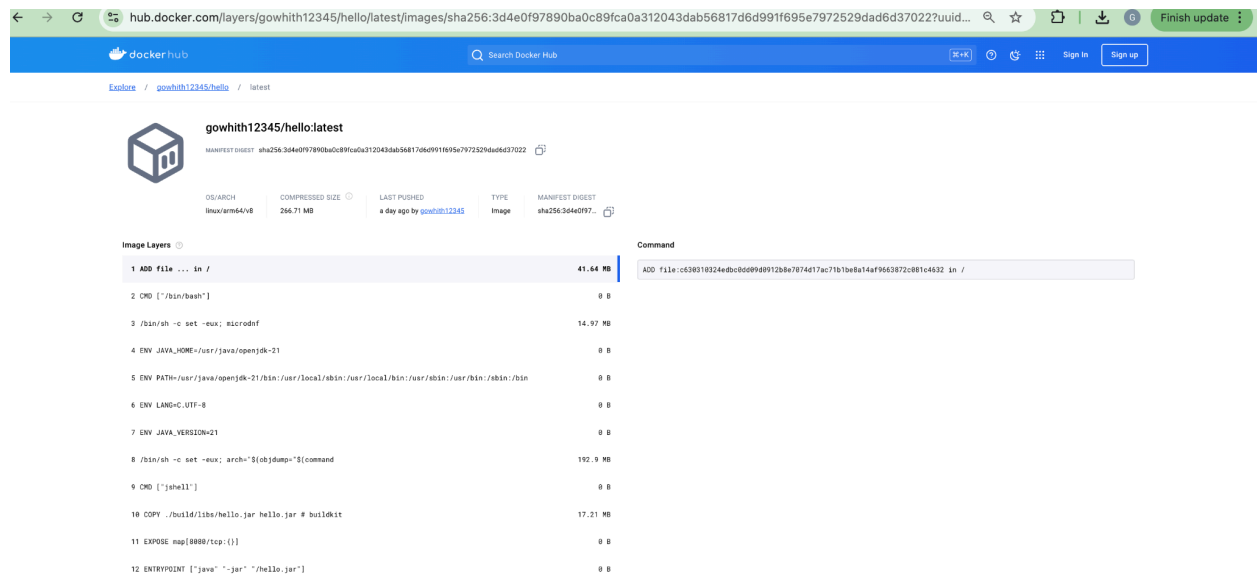
**\*\*chmod +x run.sh\*\***

Run the Script: You can now execute the script directly by typing:

**\*\*./run.sh\*\***

**##Docker images to Docker Hub:-**

hello :



hub.docker.com/layers/gowhith12345/hello/latest/images/sha256:3d4e0f97890ba0c89fca0a312043dab56817d6d991f695e7972529dad6d37022?uuid=...

gowhith12345/hello:latest

MANIFEST DIGEST sha256:3d4e0f97890ba0c89fca0a312043dab56817d6d991f695e7972529dad6d37022

OS/ARCH	COMPRESSED SIZE	LAST PUSHED	TYPE	MANIFEST DIGEST
linux/amd64/v8	266.71 MB	a day ago by gowhith12345	Image	sha256:3d4e0f97...

Image Layers

Layer	Command	Size
1 ADD file: c638318324e4b0c808948912b6784d7ac71b1b8a14af9563872c881c4632 in /	ADD file: c638318324e4b0c808948912b6784d7ac71b1b8a14af9563872c881c4632 in /	41.64 MB
2 CMD ["/bin/bash"]		0 B
3 /bin/sh -c set -eux; microdnf		14.97 MB
4 ENV JAVA_HOME=/usr/java/openjdk-21		0 B
5 ENV PATH=/usr/java/openjdk-21/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin		0 B
6 ENV LANG=C.UTF-8		0 B
7 ENV JAVA_VERSION=21		0 B
8 /bin/sh -c set -eux; arch=\$(objdump -f \$(command		192.9 MB
9 CMD ["\$shell"]		0 B
10 COPY ./build/lib/hello.jar hello.jar # buildkit		17.21 MB
11 EXPOSE map[8080/tcp:()]		0 B
12 ENTRYPOINT ["java" "-jar" "/hello.jar"]		0 B

[https://hub.docker.com/layers/gowhith12345/hello/latest/images/sha256:3d4e0f97890ba0c89fca0a312043dab56817d6d991f695e7972529dad6d37022?uuid=58c85301-c562-48f5-a59b-fb668bb07375%0A]

**\*\*gowhith12345/hello\*\***

world :

hub.docker.com/layers/gowhith12345/world/latest/images/sha256:1477c9b5a117d10cc4d0c3d79743d01470f05f0339b04d74acaaec7acee258e?uui...



gowhith12345/world:latest

MANIFEST DIGEST sha256:1477c9b5a117d10cc4d0c3d79743d01470f05f0339b04d74acaaec7acee258e

OS/ARCH	COMPRESSED SIZE ⓘ	LAST PUSHED	TYPE	MANIFEST DIGEST
linux/arm64/v8	266.71 MB	a day ago by <a href="#">gowhith12345</a>	image	sha256:f477c9b5...

Image Layers ⓘ

ADD FILE ... IN /	41.64 MB	ADD FILE:c630318324edbc8d4090912b8e7674417ac71b1bfa14af9663872c081c6632 IN /
2 CMD ["./bin/bash"]	0 B	
3 /bin/sh -c set -eux; microdnf	14.97 MB	
4 ENV JAVA_HOME=/usr/java/openjdk-21	0 B	
5 ENV PATH=/usr/java/openjdk-21/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin	0 B	
6 ENV LANG=C.UTF-8	0 B	
7 ENV JAVA_VERSION=21	0 B	
8 /bin/sh -c set -eux; arch=\$(objdump -Scommand	192.9 MB	
9 CMD ["\$shell"]	0 B	
10 COPY ./build/libs/world.jar world.jar # buildkit	17.21 MB	
11 EXPOSE mqp[0000/tcp:()]	0 B	
12 ENTRYPOINT ["java" "-jar" "/world.jar"]	0 B	