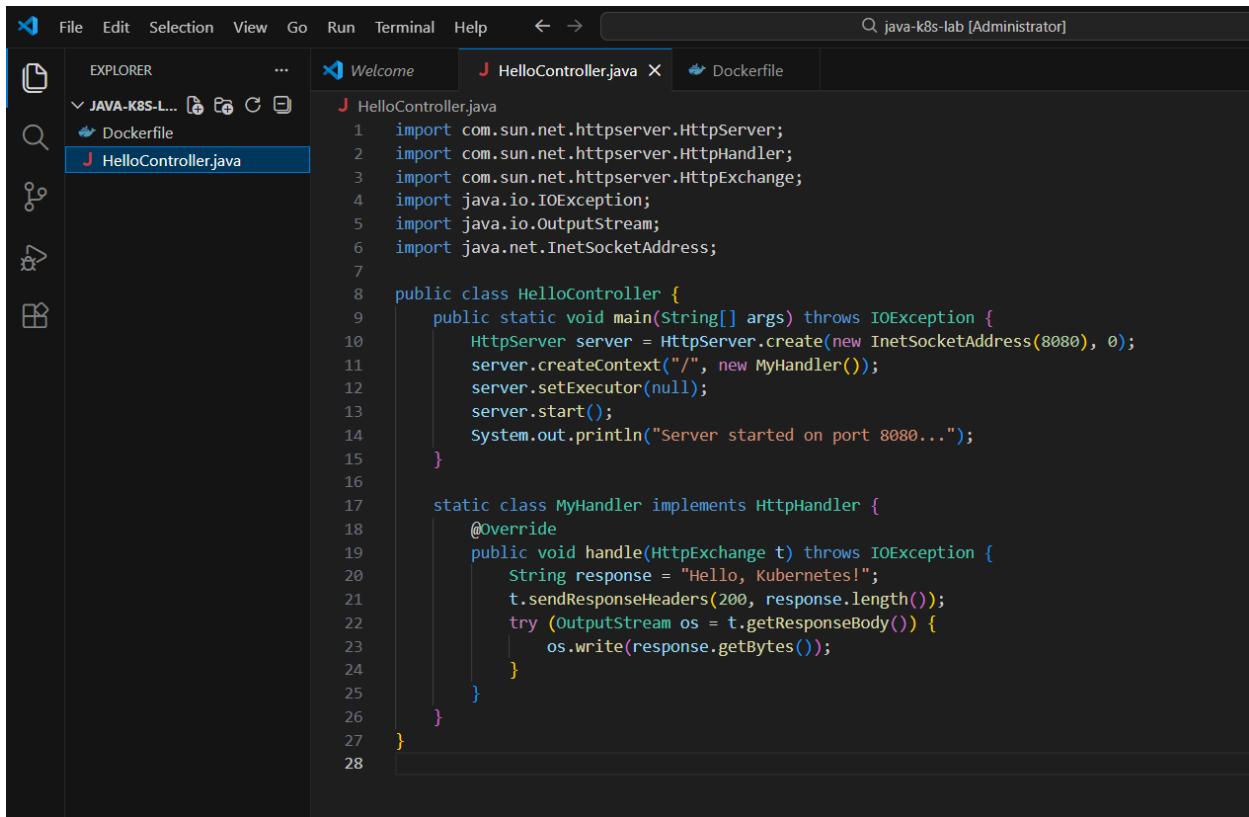


# Lab - Simple Java App on K8s

## Source Code/Dockerfile



The screenshot shows a code editor interface with the following details:

- File Menu:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Search Bar:** java-k8s-lab [Administrator].
- Explorer:** Shows a folder named "JAVA-K8S-L..." containing "Dockerfile" and "HelloController.java".
- Editor Tabs:** Welcome, HelloController.java (selected), Dockerfile.
- HelloController.java Content:**

```
1 import com.sun.net.httpserver.HttpServer;
2 import com.sun.net.httpserver.HttpHandler;
3 import com.sun.net.httpserver.HttpExchange;
4 import java.io.IOException;
5 import java.io.OutputStream;
6 import java.net.InetSocketAddress;
7
8 public class HelloController {
9     public static void main(String[] args) throws IOException {
10         HttpServer server = HttpServer.create(new InetSocketAddress(8080), 0);
11         server.createContext("/", new MyHandler());
12         server.setExecutor(null);
13         server.start();
14         System.out.println("Server started on port 8080...");
15     }
16
17     static class MyHandler implements HttpHandler {
18         @Override
19         public void handle(HttpExchange t) throws IOException {
20             String response = "Hello, Kubernetes!";
21             t.sendResponseHeaders(200, response.length());
22             try (OutputStream os = t.getResponseBody()) {
23                 os.write(response.getBytes());
24             }
25         }
26     }
27 }
28 }
```

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Search Bar:** java-k8s-lab [Administrator].
- Explorer:** JAVA-K8S-L... (containing Dockerfile and HelloController.java).
- Editor:** Dockerfile tab, displaying the following Dockerfile content:

```
1 # ----- Build Stage -----
2 FROM adoptopenjdk:11 as build
3 WORKDIR /app
4
5 # Copy the Java source file into the container
6 COPY HelloController.java .
7
8 # Compile the Java source file
9 RUN javac HelloController.java
10
11 # Package all compiled classes (including inner classes) into an executable jar
12 RUN jar cfe app.jar HelloController *.class
13
14 # ----- Run Stage -----
15 FROM adoptopenjdk:11-jre-hotspot
16 WORKDIR /app
17
18 # Copy the executable jar from the build stage
19 COPY --from=build /app/app.jar .
20
21 # Expose the port on which the app listens
22 EXPOSE 8080
23
24 # Run the jar file
25 ENTRYPOINT ["java", "-jar", "app.jar"]
26
```

## Deploy to Kubernetes

```
vijayagagla@DESKTOP-FP4OP26:~/kubernetes/java-k8s-lab$ docker build -t my-java-app:1.0 .
failed to fetch metadata: fork/exec /usr/local/lib/docker/cli-plugins/docker-buildx: no such file or directory

DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
Install the buildx component to build images with BuildKit:
https://docs.docker.com/go/buildx/

Sending build context to Docker daemon 4.096kB

vijayagagla@DESKTOP-FP4OP26:~/kubernetes/java-k8s-lab$ docker images
REPOSITORY          TAG      IMAGE ID      CREATED     SIZE
my-java-app         1.0      4aae8f8d24ab   4 seconds ago  244MB

vijayagagla@DESKTOP-FP4OP26:~/kubernetes/java-k8s-lab$ docker run -d -p 8080:8080 --name test-java-app my-java-app:1.0
4c66a529572efd0437a67db55375350b618332a0dd10c781f5ddd36fb17405a5
vijayagagla@DESKTOP-FP4OP26:~/kubernetes/java-k8s-lab$ curl http://localhost:8080
Hello, Kubernetes!vijayagagla@DESKTOP-FP4OP26:~/kubernetes/java-k8s-lab$ docker rm -f test-java-app
test-java-app
```

```

vijayagagla@DESKTOP-FP4OP26:~/kubernetes/java-k8s-lab$ docker tag my-java-app:1.0 vijayagagla123/my-java-app:1.0
vijayagagla@DESKTOP-FP4OP26:~/kubernetes/java-k8s-lab$ docker push vijayagagla123/my-java-app:1.0
The push refers to repository [docker.io/vijayagagla123/my-java-app]
b8c689a7dab2: Pushed
ed51034fefbd: Pushed
83b767b06655: Mounted from library/adoptopenjdk
14fdb8039ba4: Mounted from library/adoptopenjdk
da55b45d310b: Mounted from library/adoptopenjdk
1.0: digest: sha256:d0a1329b1f10b030e9b78157401e286fd5735b68979300e7ca4015e2f0e2c260 size: 1367
vijayagagla@DESKTOP-FP4OP26:~/kubernetes/java-k8s-lab$ kubectl apply -f deployment.yaml
deployment.apps/java-app-deployment created
vijayagagla@DESKTOP-FP4OP26:~/kubernetes/java-k8s-lab$ kubectl get deployments
NAME          READY   UP-TO-DATE   AVAILABLE   AGE
hello          3/3     3            3           3d6h
java-app-deployment  0/1     1            0           9s
nginx-deployment    10/10   10           10          6h32m
nginx-webserver     1/1     1            1           4h56m
redis-deployment    1/1     1            1           109m
vijayagagla@DESKTOP-FP4OP26:~/kubernetes/java-k8s-lab$ kubectl apply -f service.yaml
service/java-app-service created
vijayagagla@DESKTOP-FP4OP26:~/kubernetes/java-k8s-lab$ kubectl get svc
NAME        TYPE        CLUSTER-IP      EXTERNAL-IP      PORT(S)        AGE
hello       NodePort    10.99.6.155    <none>         8080:32345/TCP  3d6h
java-app-service  NodePort    10.106.46.159   <none>         80:30080/TCP   8s
kubernetes   ClusterIP   10.96.0.1      <none>         443/TCP       3d6h
redis-service  NodePort    10.101.251.26   <none>         6379:30079/TCP  110m
vijayagagla@DESKTOP-FP4OP26:~/kubernetes/java-k8s-lab$ minikube service java-app-service

```

NAMESPACE	NAME	TARGET PORT	URL
default	java-app-service	80	http://192.168.49.2:30080

## Test the Application

```
vijayagagla@DESKTOP-FP4OP26:~/kubernetes/java-k8s-lab$ minikube service java-app-service
```

NAMESPACE	NAME	TARGET PORT	URL
default	java-app-service	80	http://192.168.49.2:30080

⌚ Starting tunnel for service java-app-service.

NAMESPACE	NAME	TARGET PORT	URL
default	java-app-service		http://127.0.0.1:44247

🌐 Opening service default/java-app-service in default browser...

```
/usr/bin/xdg-open: 882: x-www-browser: not found
/usr/bin/xdg-open: 882: firefox: not found
/usr/bin/xdg-open: 882: iceweasel: not found
/usr/bin/xdg-open: 882: seamonkey: not found
/usr/bin/xdg-open: 882: mozilla: not found
/usr/bin/xdg-open: 882: epiphany: not found
/usr/bin/xdg-open: 882: konqueror: not found
/usr/bin/xdg-open: 882: chromium: not found
/usr/bin/xdg-open: 882: chromium-browser: not found
/usr/bin/xdg-open: 882: google-chrome: not found
/usr/bin/xdg-open: 882: www-browser: not found
/usr/bin/xdg-open: 882: links2: not found
/usr/bin/xdg-open: 882: elinks: not found
/usr/bin/xdg-open: 882: links: not found
/usr/bin/xdg-open: 882: lynx: not found
/usr/bin/xdg-open: 882: w3m: not found
xdg-open: no method available for opening 'http://127.0.0.1:44247'
```

