Kubernetes Hello World Demo

This project is a simple demo application that connects a React frontend with a Java Spring Boot backend to deploy and expose services on a local Kubernetes cluster (via Minikube). - The **frontend** has a single button labeled *Create A*. - When clicked, it sends a request to the **backend**. - The backend creates a new Kubernetes Deployment and Service (using NGINX as the container). - The backend returns the external endpoint, which is displayed in the UI. - Visiting the endpoint shows the default NGINX page, confirming the pod is running in Kubernetes.

How to Run the Application
1. Start Minikube:
minikube start --driver=docker

2. Run the backend:
cd hello-backend
mvn spring-boot:run

3. Run the frontend:
cd hello-frontend
npm install
npm start

4. Open the frontend in the browser at http://localhost:3000 and click *Create A*.
5. The UI will display the Kubernetes service endpoint (e.g., http://192.168.49.2:31140).
6. Open that endpoint in a browser to see the NGINX welcome page.

Challenges Faced
- Setting up Minikube and Docker on macOS, dealing with driver compatibility issues.
- Backend originally hung when calling minikube service ... --url, solved by switching to NodePort and timeouts.
- React frontend required manual reconstruction of public/index.html and src/index.js after initialization problems.
- Dealing with CORS between localhost:3000 and localhost:8080.

What I Learned
- How to integrate a React frontend with a Spring Boot backend.
- How to programmatically create Kubernetes Deployments and Services from Java.
- How to debug Kubernetes rollout issues (ImagePullBackOff, NodePort access, etc.).
- How to troubleshoot local development environments on macOS (brew, Docker Desktop, Minikube).

Screenshots

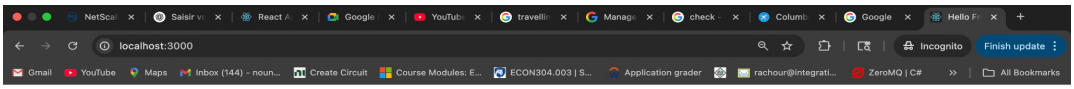
```
[INFO]
[INFO] --- spring-boot:3.3.2:repackage (repackage) @ hello-backend ---
[INFO] Replacing main artifact /Users/rayanachour/Desktop/IO-phase1/hello-world-k8s/hello-backend/target/hello-backend-1.0.0.jar with repackaged archive, adding nested dependencies in BOOT-INF/.
[INFO] The original artifact has been renamed to /Users/rayanachour/Desktop/IO-phase1/hello-world-k8s/hello-backend/target/hello-backend-1.0.0.jar.original
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  1.574 s
[INFO] Finished at: 2025-09-25T22:28:05-04:00
[INFO] ------------------------------------------------------------------------
(base) rayanachour@Rayans-MacBook-Pro hello-backend % mvn spring-boot:run

[INFO] Scanning for projects...
[INFO]
[INFO] ----------------------< com.example:hello-backend >-----------------------
[INFO] Building hello-backend 1.0.0
[INFO]   from pom.xml
[INFO] --------------------------------[ jar ]---------------------------------
[INFO]
[INFO] >>> spring-boot:3.3.2:run (default-cli) > test-compile @ hello-backend >>>
[INFO]
[INFO] --- resources:3.3.1:resources (default-resources) @ hello-backend ---
[INFO] skip non existing resourceDirectory /Users/rayanachour/Desktop/IO-phase1/hello-world-k8s/hello-backend/src/main/resources
[INFO] skip non existing resourceDirectory /Users/rayanachour/Desktop/IO-phase1/hello-world-k8s/hello-backend/src/main/resources
[INFO]
[INFO] --- compiler:3.13.0:compile (default-compile) @ hello-backend ---
[INFO] Nothing to compile - all classes are up to date.
[INFO]
[INFO] --- resources:3.3.1:testResources (default-testResources) @ hello-backend ---
[INFO] skip non existing resourceDirectory /Users/rayanachour/Desktop/IO-phase1/hello-world-k8s/hello-backend/src/test/resources
[INFO]
[INFO] --- compiler:3.13.0:testCompile (default-testCompile) @ hello-backend ---
[INFO] No sources to compile
[INFO]
[INFO] <<< spring-boot:3.3.2:run (default-cli) < test-compile @ hello-backend <<<
[INFO]
[INFO]
[INFO] --- spring-boot:3.3.2:run (default-cli) @ hello-backend ---
[INFO] Attaching agents: []

  .   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/

 :: Spring Boot ::                (v3.3.2)

2025-09-25T22:28:14.672-04:00  INFO 18451 --- [           main] com.example.App                          : Starting App using Java 21.0.7 with PID 18451 (/Users/rayanachour/Desktop/IO-phase1/hello-world-k
8s/hello-backend/target/classes started by rayanachour in /Users/rayanachour/Desktop/IO-phase1/hello-world-k8s/hello-backend)
2025-09-25T22:28:14.673-04:00  INFO 18451 --- [           main] com.example.App                          : No active profile set, falling back to 1 default profile: "default"
2025-09-25T22:28:14.884-04:00  INFO 18451 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat initialized with port 8080 (http)
2025-09-25T22:28:14.891-04:00  INFO 18451 --- [           main] o.apache.catalina.core.StandardService   : Starting service [Tomcat]
2025-09-25T22:28:14.891-04:00  INFO 18451 --- [           main] o.apache.catalina.core.StandardEngine    : Starting Servlet engine: [Apache Tomcat/10.1.26]
2025-09-25T22:28:14.912-04:00  INFO 18451 --- [           main] o.a.c.c.C.[Tomcat].[localhost].[/]       : Initializing Spring embedded WebApplicationContext
2025-09-25T22:28:14.913-04:00  INFO 18451 --- [           main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 322 ms
2025-09-25T22:28:15.067-04:00  INFO 18451 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat started on port 8080 (http) with context path '/'
2025-09-25T22:28:15.050-04:00  INFO 18451 --- [           main] com.example.App                          : Started App in 0.61 seconds (process running for 0.738)
2025-09-25T22:28:32.578-04:00  INFO 18451 --- [nio-8080-exec-1] o.a.c.c.C.[Tomcat].[localhost].[/]       : Initializing Spring DispatcherServlet 'dispatcherServlet'
2025-09-25T22:28:32.578-04:00  INFO 18451 --- [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet        : Initializing Servlet 'dispatcherServlet'
2025-09-25T22:28:32.580-04:00  INFO 18451 --- [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet        : Completed initialization in 2 ms
```



Create A

endpoint: http://192.168.49.2:31140

```
kind: Endpoints
metadata:
  creationTimestamp: "2025-09-25T18:44:00Z"
  labels:
    app: hello-1dc804
    endpoints.kubernetes.io/managed-by: endpoint-controller
  name: hello-1dc804
  namespace: default
  resourceVersion: "6211"
  uid: 51a6b4a0-23fa-4bfa-97f5-f2075cd0d64f
subsets:
- addresses:
  - ip: 10.244.0.11
    nodeName: minikube
    targetRef:
      kind: Pod
      name: hello-1dc804-66b7754bc4-njhrq
      namespace: default
      uid: c57182fc-6737-45fe-81a6-eda9b82cb3c1
  ports:
  - name: http
    port: 80
    protocol: TCP
(base) rayanachour@Rayans-MacBook-Pro hello-world-k8s % kubectl port-forward deploy/hello-1dc804 8081:80

Forwarding from 127.0.0.1:8081 -> 80
Forwarding from [::1]:8081 -> 80
Handling connection for 8081
Handling connection for 8081
^C                                                                                (base) rayanachour@Rayans-MacBook-Pro hello-world-k8s % kubectl rollout status deploy/hello-1dc804

deployment "hello-1dc804" successfully rolled out
(base) rayanachour@Rayans-MacBook-Pro hello-world-k8s % kubectl rollout status deploy/hello-1dc804

deployment "hello-1dc804" successfully rolled out
(base) rayanachour@Rayans-MacBook-Pro hello-world-k8s % curl -X POST http://localhost:8080/instances

curl: (52) Empty reply from server
(base) rayanachour@Rayans-MacBook-Pro hello-world-k8s % minikube start --driver=docker

😄  minikube v1.37.0 on Darwin 13.5 (arm64)
✨  Using the docker driver based on existing profile
👍  Starting "minikube" primary control-plane node in "minikube" cluster
🚜  Pulling base image v0.0.48 ...
🔄  Updating the running docker "minikube" container ...
🐳  Preparing Kubernetes v1.34.0 on Docker 28.4.0 ...
🔎  Verifying Kubernetes components...
    ▪ Using image gcr.io/k8s-minikube/storage-provisioner:v5
🌟  Enabled addons: storage-provisioner, default-storageclass
🏄  Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
(base) rayanachour@Rayans-MacBook-Pro hello-world-k8s % kubectl get nodes
NAME       STATUS   ROLES           AGE   VERSION
minikube   Ready    control-plane   10h   v1.34.0
(base) rayanachour@Rayans-MacBook-Pro hello-world-k8s % curl -X POST http://localhost:8080/instances

curl: (7) Failed to connect to localhost port 8080 after 6 ms: Connection refused
(base) rayanachour@Rayans-MacBook-Pro hello-world-k8s % lsof -i :8080

COMMAND   PID        USER   FD   TYPE             DEVICE SIZE/OFF NODE NAME
java    18451 rayanachour   36u  IPv6 0x7668be6b26d03665      0t0  TCP *:http-alt (LISTEN)
(base) rayanachour@Rayans-MacBook-Pro hello-world-k8s % curl -X POST http://localhost:8080/instances

(base) rayanachour@Rayans-MacBook-Pro hello-world-k8s %
```