1). You need to write required yaml files for this service.

2). You need to create jar for this service first and then create dockerfile for it.

3). After that to deploy this service in kubernetes you need to install minikube in your local or if you have any cloud platform account then it is also fine.

4). You can access this application on any port you want to (it's your choice).

5). This application is deploying tomcat server in kubernetes.

Sol:

Step1: create deployment.yaml and service.yaml files

nano deployment.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: tomcat-deployment
spec:
  replicas: 2
  selector:
    matchLabels:
      app: tomcat
  template:
    metadata:
      labels:
        app: tomcat
    spec:
      containers:
      - name: tomcat
        image: krupaht/tomcatt-app # Use your built Docker image here
        ports:
        - containerPort: 8080
```

nano service.yaml

```
apiVersion: v1
kind: Service
metadata:
  name:tomcatt-service
spec:
  selector:
    app: tomcat
  ports:
    - protocol: TCP
    port: 80   # Expose on port 80
    targetPort: 8080  # Target Tomcat's internal port
  type: NodePort
```

Step 2: apt-get install maven

Building the jar file

```
mvn clean install
```

```
[INFO] Scanning for projects...
[INFO]
[INFO] --------------< com.sap.docker:docker-java-sample-webapp >--------------
[INFO] Building docker-java-sample-webapp 1.0-SNAPSHOT
[INFO] --------------------------------[ war ]---------------------------------
[INFO]
[INFO] --- maven-clean-plugin:2.5:clean (default-clean) @ docker-java-sample-webapp ---
[INFO] Deleting /home/sigmoid/Desktop/k8s_project_new/target
[INFO]
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @
docker-java-sample-webapp ---
[WARNING] Using platform encoding (UTF-8 actually) to copy filtered resources, i.e. build is
platform dependent!
[INFO] skip non existing resourceDirectory
/home/sigmoid/Desktop/k8s_project_new/src/main/resources
[INFO]
[INFO] --- maven-compiler-plugin:3.6.2:compile (default-compile) @
docker-java-sample-webapp ---
[INFO] Changes detected - recompiling the module!
[WARNING] File encoding has not been set, using platform encoding UTF-8, i.e. build is
platform dependent!
[INFO] Compiling 1 source file to /home/sigmoid/Desktop/k8s_project_new/target/classes
[INFO]
[INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @
docker-java-sample-webapp ---
[WARNING] Using platform encoding (UTF-8 actually) to copy filtered resources, i.e. build is
platform dependent!
[INFO] skip non existing resourceDirectory
/home/sigmoid/Desktop/k8s_project_new/src/test/resources
[INFO]
[INFO] --- maven-compiler-plugin:3.6.2:testCompile (default-testCompile) @
docker-java-sample-webapp ---
[INFO] No sources to compile
[INFO]
[INFO] --- maven-surefire-plugin:2.12.4:test (default-test) @ docker-java-sample-webapp ---
[INFO] No tests to run.
```

```
[INFO]
[INFO] --- maven-war-plugin:2.2:war (default-war) @ docker-java-sample-webapp ---
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by com.thoughtworks.xstream.core.util.Fields (file:/home/sigmoid/.m2/repository/com/thoughtworks/xstream/xstream/1.3.1/xstream-1.3.1.jar) to field java.util.Properties.defaults
WARNING: Please consider reporting this to the maintainers of com.thoughtworks.xstream.core.util.Fields
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
[INFO] Packaging webapp
[INFO] Assembling webapp [docker-java-sample-webapp] in [/home/sigmoid/Desktop/k8s_project_new/target/docker-java-sample-webapp-1.0-SNAPSHOT]
[INFO] Processing war project
[INFO] Copying webapp resources [/home/sigmoid/Desktop/k8s_project_new/src/main/webapp]
[INFO] Webapp assembled in [40 msecs]
[INFO] Building war: /home/sigmoid/Desktop/k8s_project_new/target/docker-java-sample-webapp-1.0-SNAPSHOT.war
[INFO] WEB-INF/web.xml already added, skipping
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  2.778 s
[INFO] Finished at: 2024-10-24T14:15:35+05:30
[INFO] ------------------------------------------------------------------------
```

Step 3: building docker image
Nano Dockerfile

```
# Use an official Tomcat base image
FROM tomcat:9.0.50-jdk11-openjdk

# Set environment variables
ENV JAVA_OPTS="-Xmx1024m"

# Copy your application JAR or WAR file into the Tomcat webapps directory
COPY target/docker-java-sample-webapp-1.0-SNAPSHOT.war /usr/local/tomcat/webapps/ROO>

# Expose port 8080 for Tomcat
EXPOSE 8080
```

```
# Start Tomcat
CMD ["catalina.sh", "run"]
```

To start minikube :**minikube start**

Docker build -t krupaht/tomcatt-app .

```
docker build -t krupaht/tomcatt-app .
[+] Building 4.2s (8/8) FINISHED                              docker:default
 => [internal] load build definition from Dockerfile                    0.4s
 => => transferring dockerfile: 414B                                    0.0s
 => [internal] load metadata for docker.io/library/tomcat:9.0.50-jdk11-op  2.9s
 => [auth] library/tomcat:pull token for registry-1.docker.io            0.0s
 => [internal] load .dockerignore                                        0.2s
 => => transferring context: 2B                                          0.0s
 => [internal] load build context                                        0.3s
 => => transferring context: 102.16kB                                    0.1s
 => CACHED [1/2] FROM docker.io/library/tomcat:9.0.50-jdk11-openjdk@sha25  0.0s
 => [2/2] COPY target/docker-java-sample-webapp-1.0-SNAPSHOT.war /usr/loc  0.1s
 => exporting to image                                                   0.1s
 => => exporting layers                                                  0.0s
 => => writing image sha256:d8ad2245154b781f9fd0f11294a9b35f92c5b326e90b0  0.0s
 => => naming to docker.io/krupaht/tomcatt-app                           0.0s
```

Push that to docker hub

Tp push

```
docker push krupaht/tomcatt-app
```

```
docker push krupaht/tomcatt-app
Using default tag: latest
The push refers to repository [docker.io/krupaht/tomcatt-app]
644a7489dd7e: Pushed
31e9fb42fe0e: Mounted from krupaht/tomcat-app
ec329d1fff1d: Mounted from krupaht/tomcat-app
22fb506c4d03: Mounted from krupaht/tomcat-app
f42aed5f7feb: Mounted from krupaht/tomcat-app
89819bafde36: Mounted from krupaht/tomcat-app
f3d5b8f65132: Mounted from krupaht/tomcat-app
ad83f0aa5c0a: Mounted from krupaht/tomcat-app
5a9a65095453: Mounted from krupaht/tomcat-app
4b0edb23340c: Mounted from krupaht/tomcat-app
afa3e488a0ee: Mounted from krupaht/tomcat-app
```

latest: digest: sha256:50dbc74076bfc326bf08949c0b5083cb7c126c1056f43dc6092a9c04c3db3369 size: 2631

Step 4: Deploying the images
kubectl apply -f deployment.yaml
deployment.apps/tomcat-deployment configured
sigmoid@sigmoid-ThinkPad-T460:~/Desktop/k8s_project_new$ **kubectl apply -f service.yaml**
service/tomcatt-service created

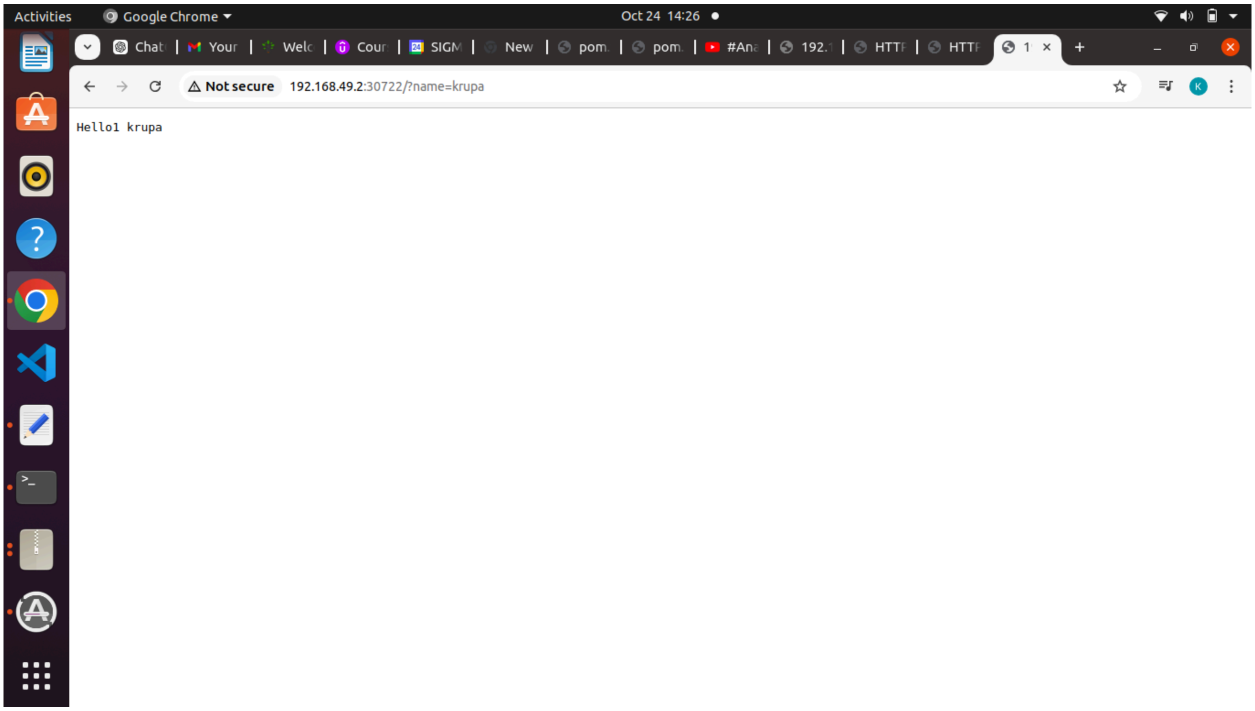
Step 5: To access the application

Kubectl get svc

Copy the port number there
Then
minikube ip

Copy that ip and paste in the url webpage

URL IS:http://192.168.49.2:30722/?name=krupa



192.168.49.2:30722/?name=krupa

```
Hello1 krupa
```