Scale a running Kubernetes deployment from 2 replicas to 5 replicas, and verify that the scaling was successful. i want to perform everything through docker give me steps and code

I don't have any project ready. Give me all the steps from scratch

Make sure Rancher is up and running

Create folder my-k8s-app

Inside the folder, create server.js

```
const express = require('express');
const app = express();
const port = 3000;

app.get('/', (req, res) => {
  res.send('Hello, Kubernetes!');
});

app.listen(port, () => {
  console.log(`App running on port ${port}`);
});

npm init -y
```

npm install express

This will create package.json file in the pwd

node server.js

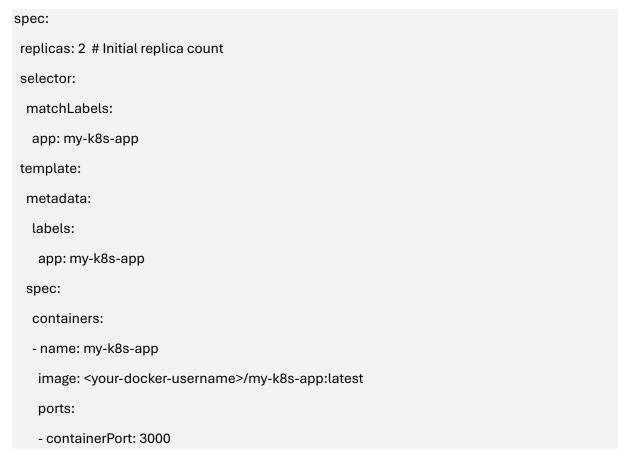
Open your browser and visit http://localhost:3000. You should see "Hello, Kubernetes!".

Create a Dockerfile in the root of your project folder (my-k8s-app):

Use an official Node.js runtime as a parent image

FROM node:14

Set the working directory in the container
WORKDIR /usr/src/app
Copy the current directory contents into the container
COPY
Install dependencies
RUN npm install
Make the app available on port 3000
EXPOSE 3000
Define the command to run the app
CMD ["node", "server.js"]
Build the Docker image : Run the following command to build your Docker image:
docker build -t my-k8s-app:latest .
Run the Docker container locally to test the image:
docker run -p 3000:3000 my-k8s-app:latest
Visit http://localhost:3000 in your browser to verify the app is running.
Log in to Docker Hub (or your preferred registry):
docker login
Tag your image for Docker Hub:
docker tag my-k8s-app:latest <your-docker-username>/my-k8s-app:latest</your-docker-username>
Push the image to Docker Hub:
docker push <your-docker-username>/my-k8s-app:latest</your-docker-username>
deployment.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
name: my-k8s-app-deployment



kubectl apply -f deployment.yaml

service.yaml

```
apiVersion: v1
kind: Service
metadata:
name: my-k8s-app-service
spec:
selector:
app: my-k8s-app
ports:
- protocol: TCP
port: 80
targetPort: 3000
type: LoadBalancer # This exposes the service outside the cluster
```

kubectl apply -f service.yaml

Get the external IP (if you're using a cloud provider): You can get the external IP of the service by running:

kubectl get svc my-k8s-app-service

kubectl get pods

Scale the deployment from 2 replicas to 5 replicas: You can scale the deployment using the following command:

kubectl scale deployment my-k8s-app-deployment --replicas=5

kubectl get pods

Do port forwarding in rancher

Visit url, you will see output

Clean Up (Optional)

If you no longer need the deployment and service, you can delete them:

kubectl delete deployment my-k8s-app-deployment

kubectl delete service my-k8s-app-service