**Orchestration using Kubernetes Part -2 case study**

**Problem Statement**

**You are working as a Devops Administrator. You’ve been tasked to deploy a multi-tier application on Kubernetes Cluster. The application is a NodeJS application available on Docker Hub with the following name:devopsedu/employee This NodeJS application works with a mongo database.**

**MongoDB image is available on DockerHub with the following name:mongoYou are required to deploy this application on Kubernetes:**

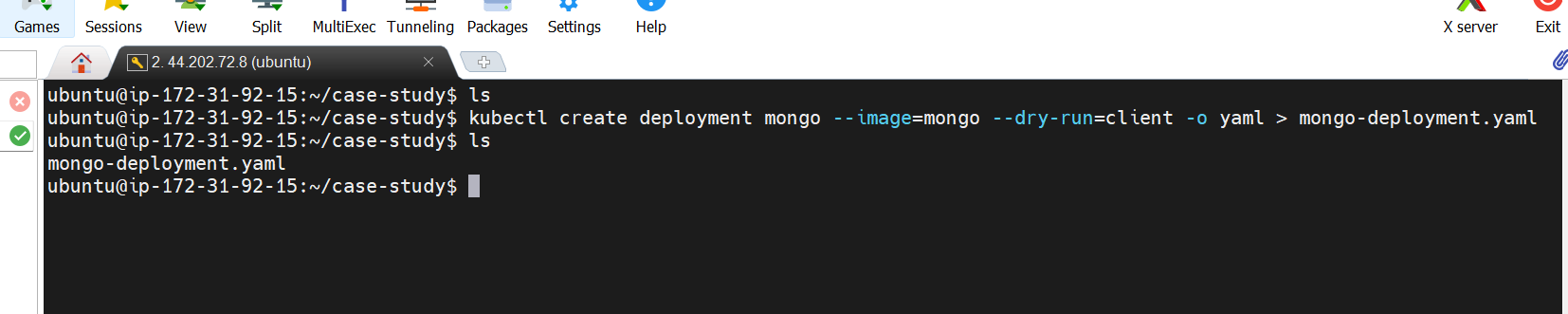
**•NodeJS is available on port 8888 in the containerand will be reaching out to port 27017 for mongo database connection**

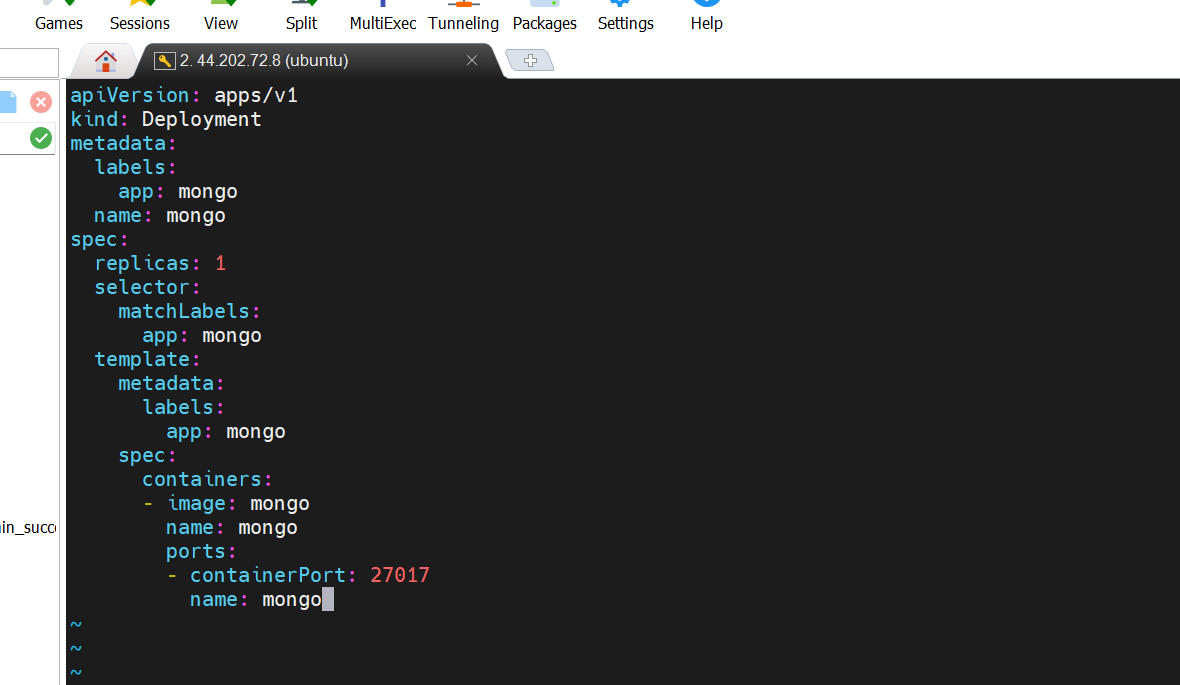
**•MongoDB will be accepting connections on port 27017You must deploy this application using the CLI.**

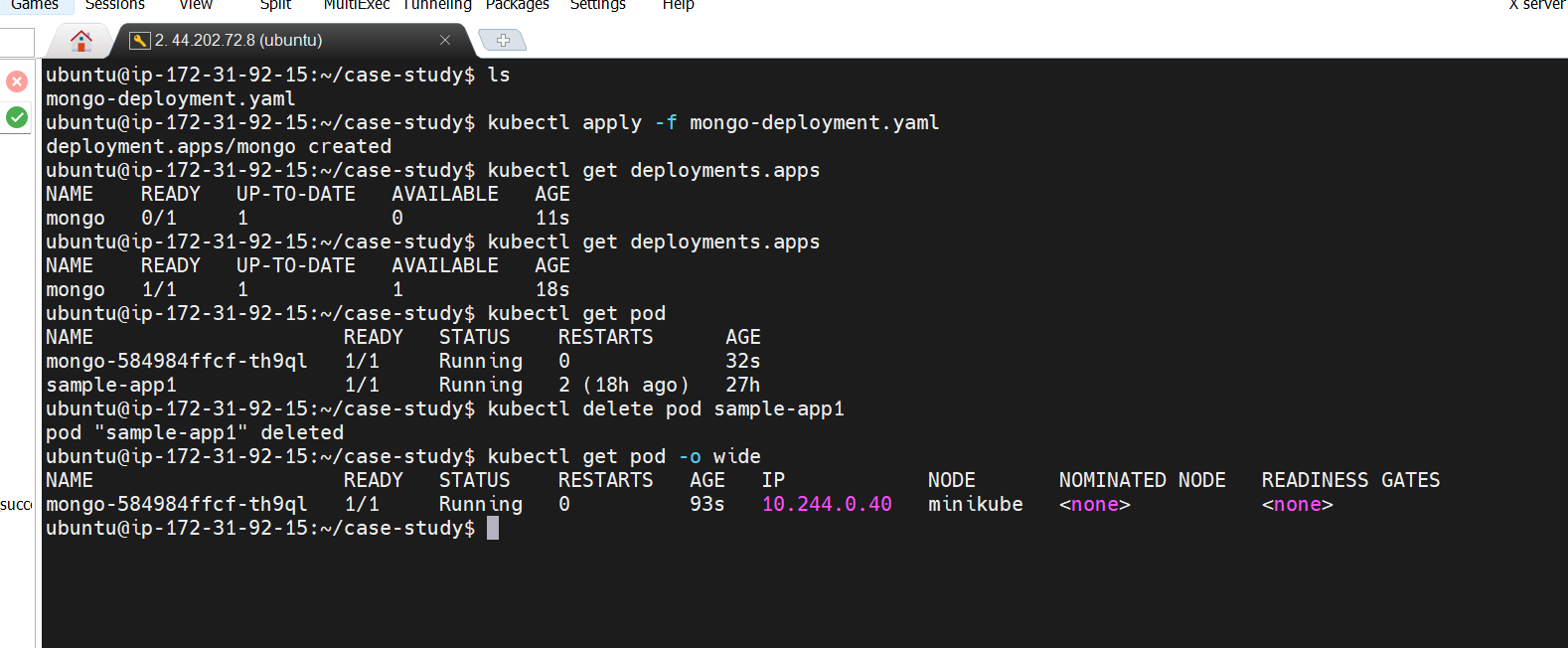
**Once your application is up and running, ensure you can add an employee from the NodeJS application and verify by going to Get Employee page and retrieving your input.**

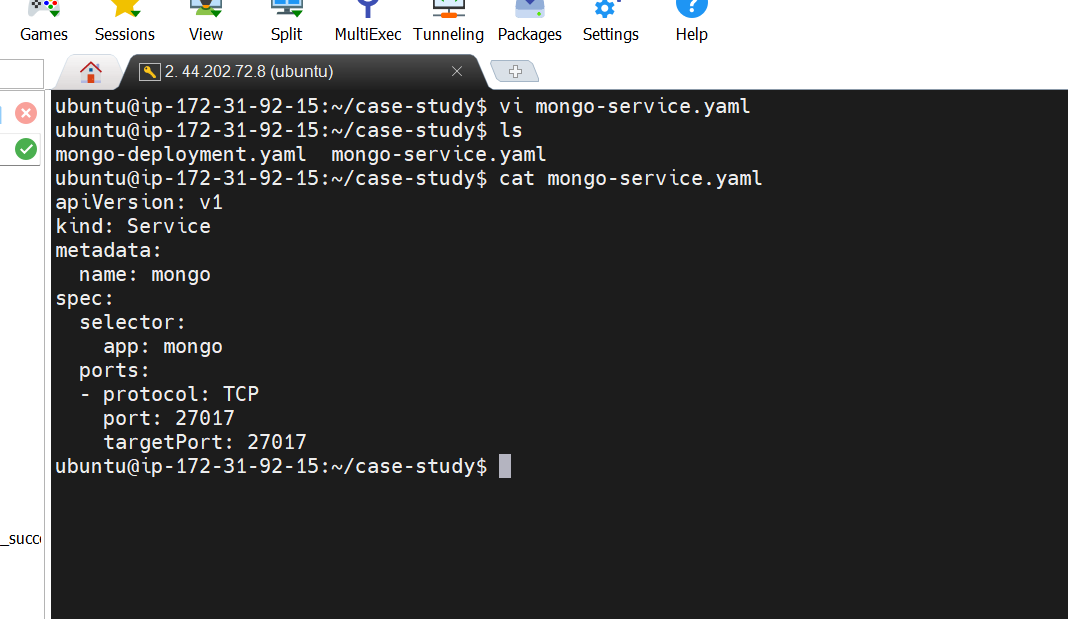
**Hint:Name the Mongo DB Service and deployment, specifically as “mongo”.**

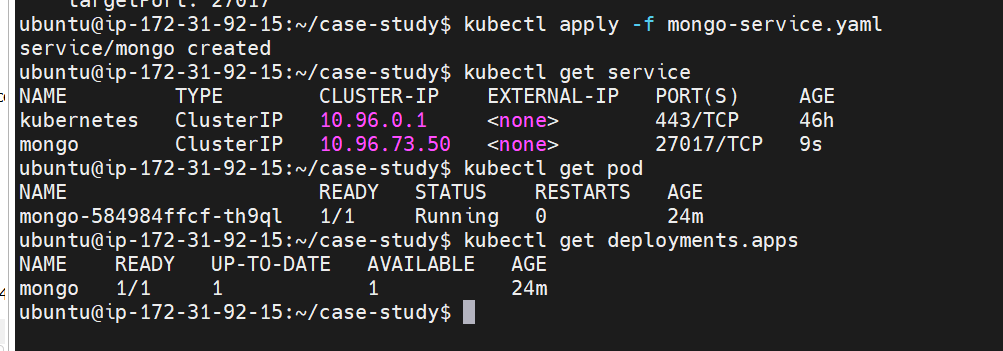
1. **Creating deployment and service for mongo DB**

****

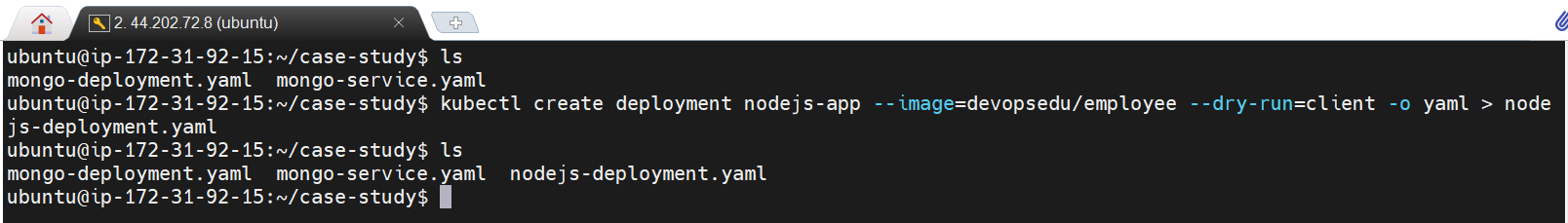
****

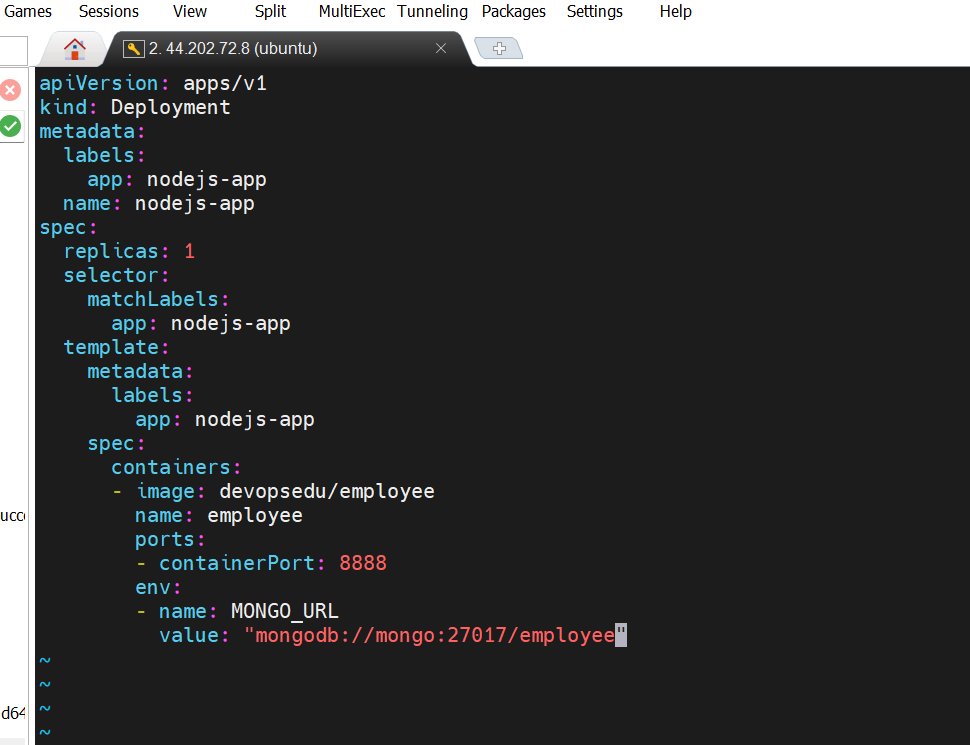
****

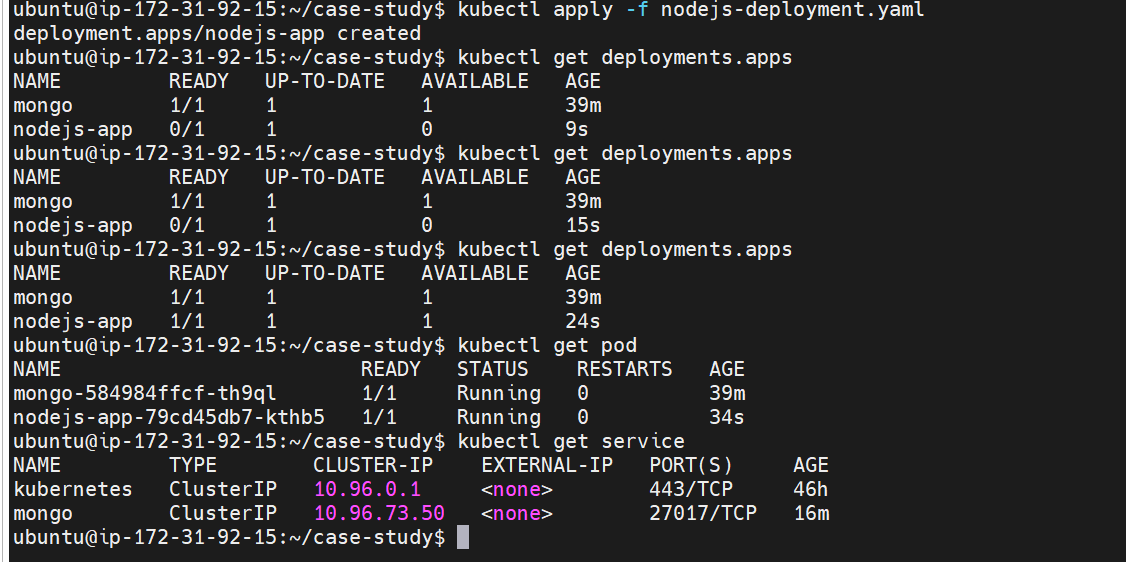
****

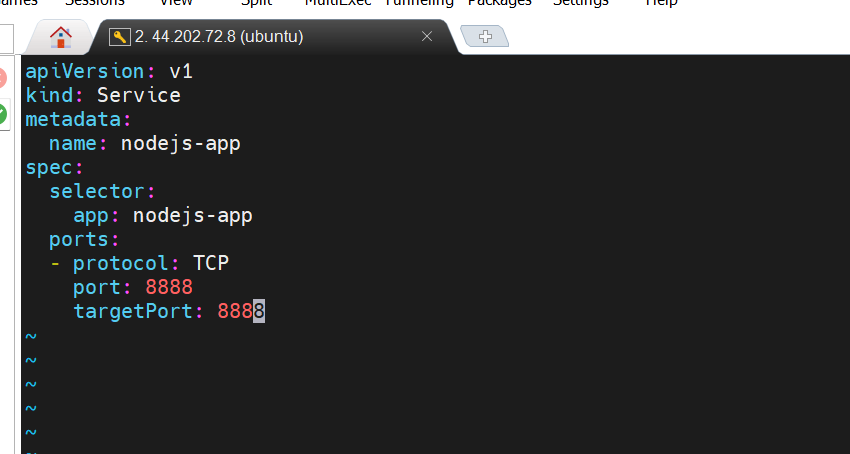
****

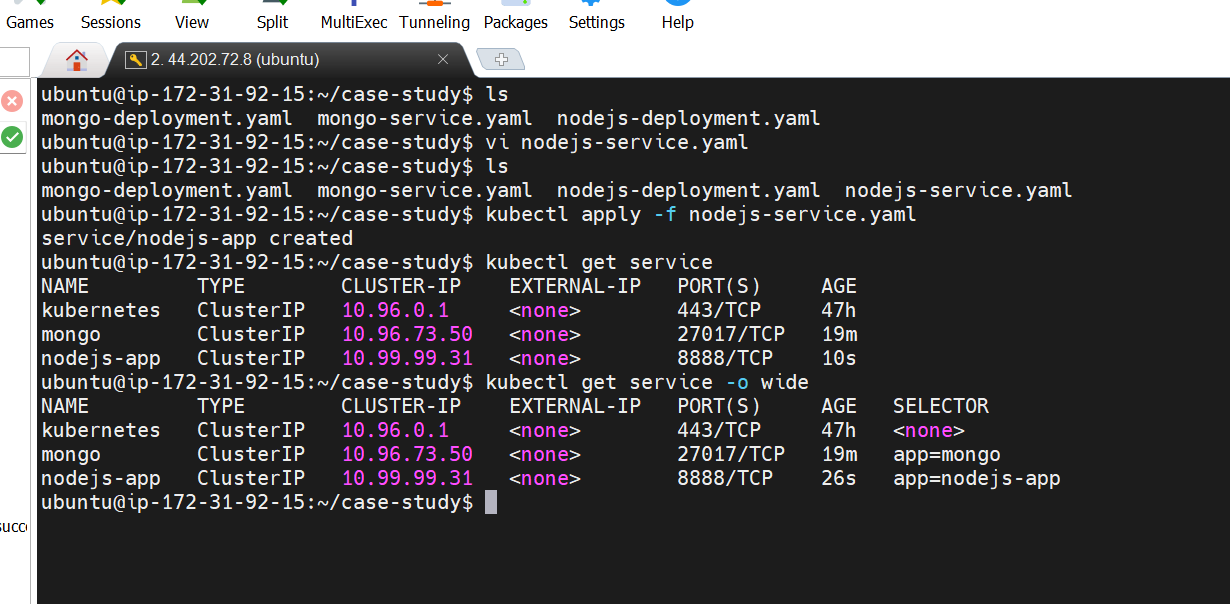
1. **Create a Node.js Deployment and Service**

****

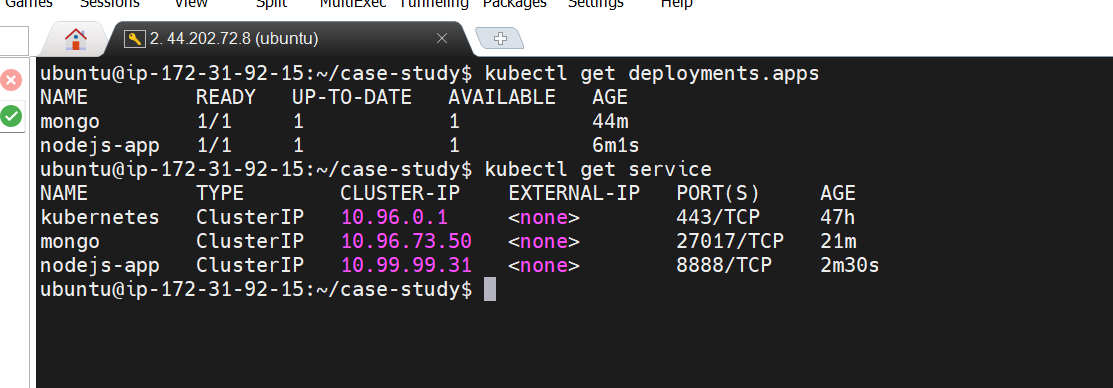
****

****

****

****

1. **Verification**

****