# 7. Three Tier web application using Kubernetes

### Step 1: Clone the Guestbook Repository

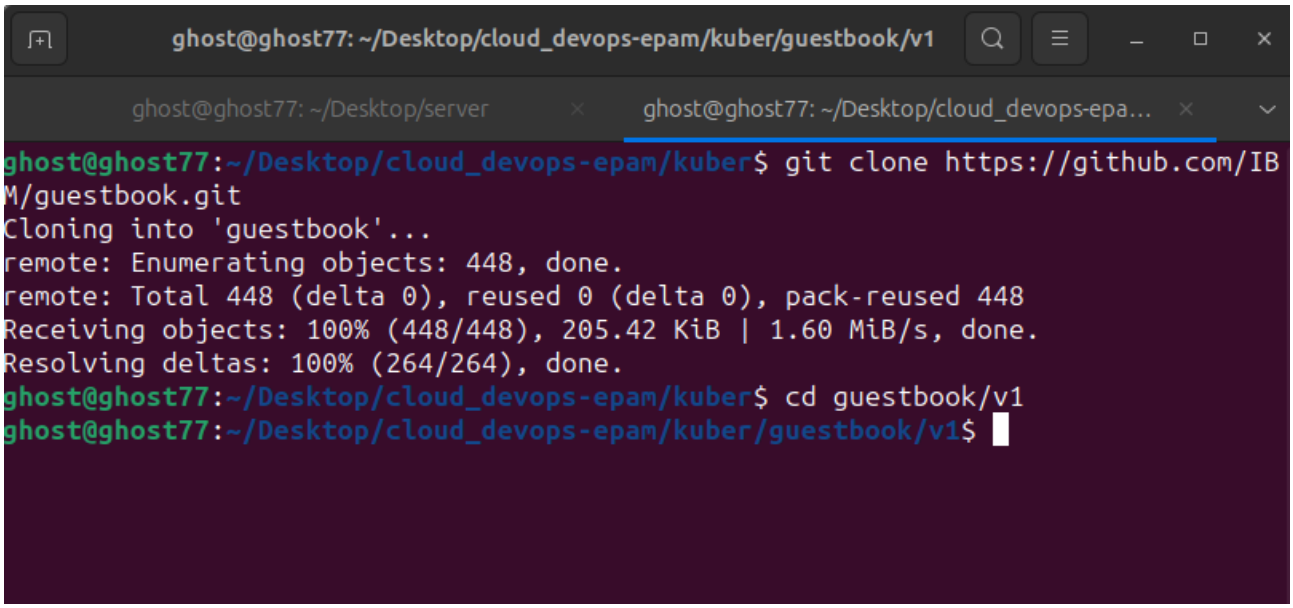
1. Open your terminal.
2. Clone the GitHub repository containing the guestbook application by running the following command:

   bash

- `git clone https://github.com/IBM/guestbook.git`

- Change directory to the version 1 of the guestbook application:

3. `cd guestbook/v1`



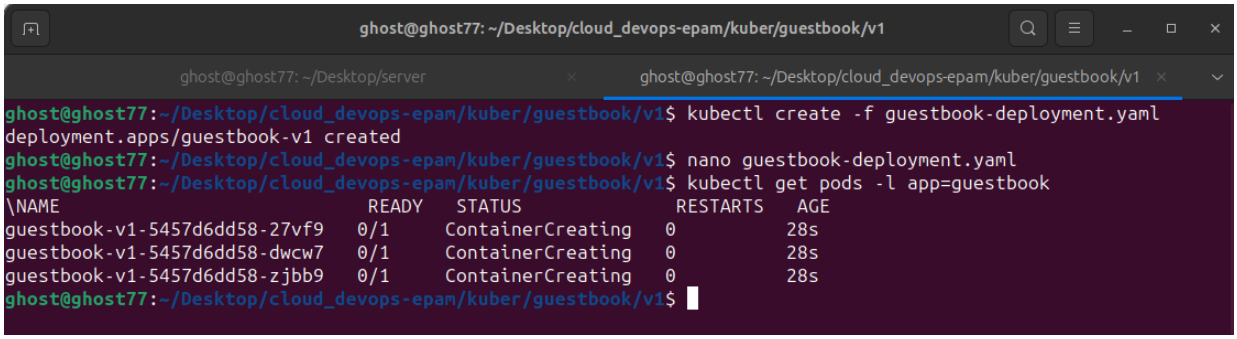

### Step 2: Deploy the Guestbook Application

1. Deploy the guestbook application by creating a Deployment using the provided configuration file:

- `kubectl create -f guestbook-deployment.yaml`

- Verify that the pods have been created by listing the pods with the label `app=guestbook`:

2. `kubectl get pods -l app=guestbook`



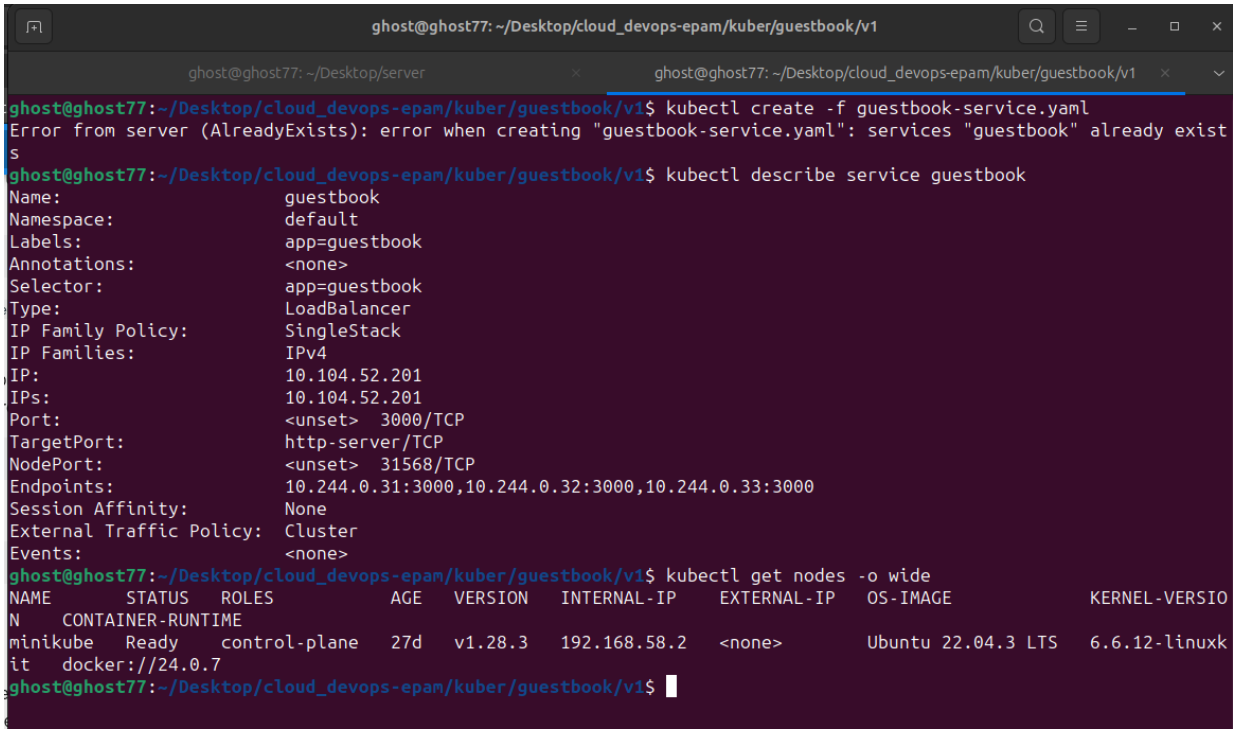
## Step 3: Expose the Guestbook Application

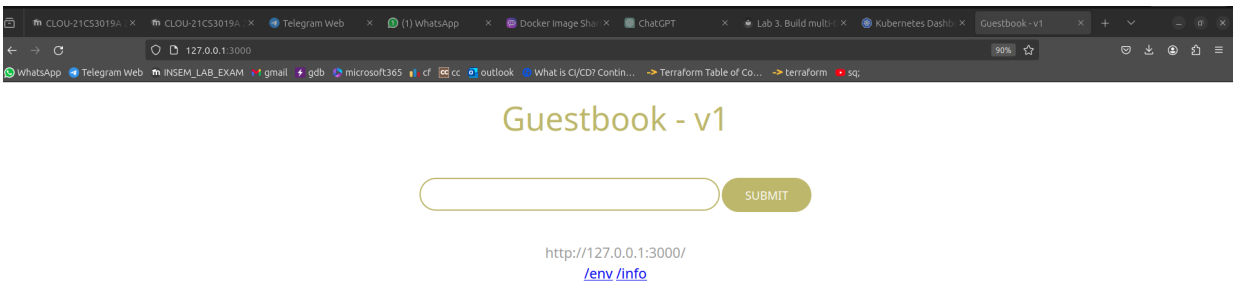
1. Create a Service to expose the guestbook application:

- `kubectl create -f guestbook-service.yaml`

- To access the guestbook application, find out the NodePort and Public IP by running the following commands:

2. 
```
kubectl describe service guestbook
kubectl get nodes -o wide
```




## Step 4: Connect to the Backend Service (Redis)

1. Deploy the Redis database by creating a Deployment using the provided configuration file:

```
lua
```

- `kubectl create -f redis-master-deployment.yaml`

- Verify that the Redis server pod is running:

- `kubectl get pods -l app=redis,role=master`

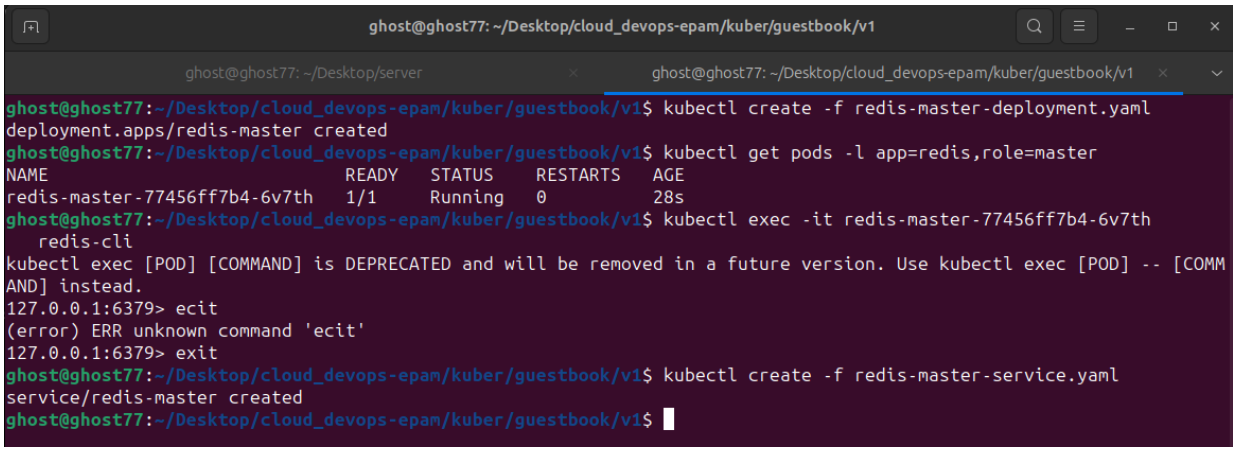
- Test the Redis standalone by connecting to the Redis server pod using the `redis-cli`:
- `kubectl exec -it <redis-pod-name> redis-cli`

You can exit the Redis CLI by typing `exit`.

- Create a Service to expose the Redis master:

`lua`

4. `kubectl create -f redis-master-service.yaml`

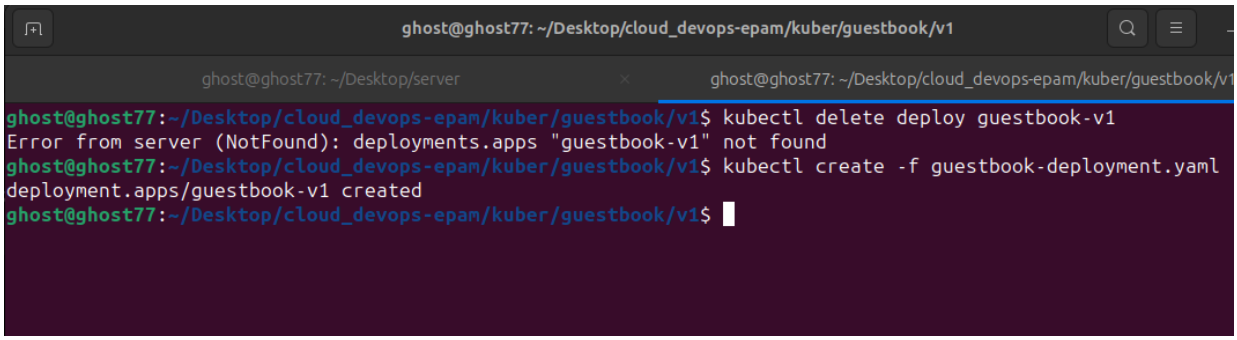



### Step 5: Update the Guestbook Application to Use Redis

1. Restart the guestbook application to make it use the Redis service:

1. `kubectl delete deploy guestbook-v1`
   `kubectl create -f guestbook-deployment.yaml`

2. Test the guestbook app again using a browser.




### Step 6: Scale the Backend Service (Redis) for Read Operations

1. Deploy the Redis slave by creating a Deployment using the provided configuration file:

- `kubectl create -f redis-slave-deployment.yaml`

- Verify that the Redis slave pods are running:

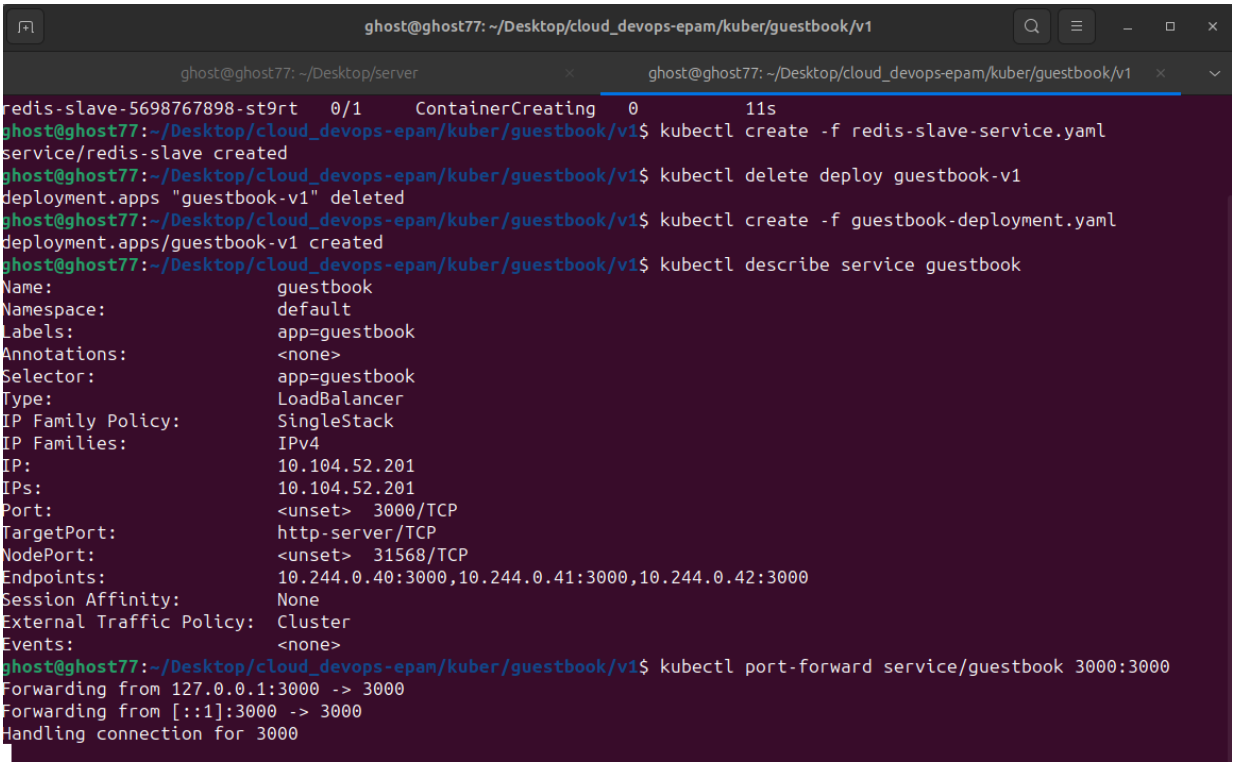
- `kubectl get pods -l app=redis,role=slave`

- Test the Redis slave by connecting to one of the pods using the `redis-cli`.
- Create a Service to expose the Redis slave:
- `kubectl create -f redis-slave-service.yaml`

- Restart the guestbook application to make it use the Redis slave service:

5. `kubectl delete deploy guestbook-v1`
   `kubectl create -f guestbook-deployment.yaml`

6. Test the guestbook app again using a browser.




S