

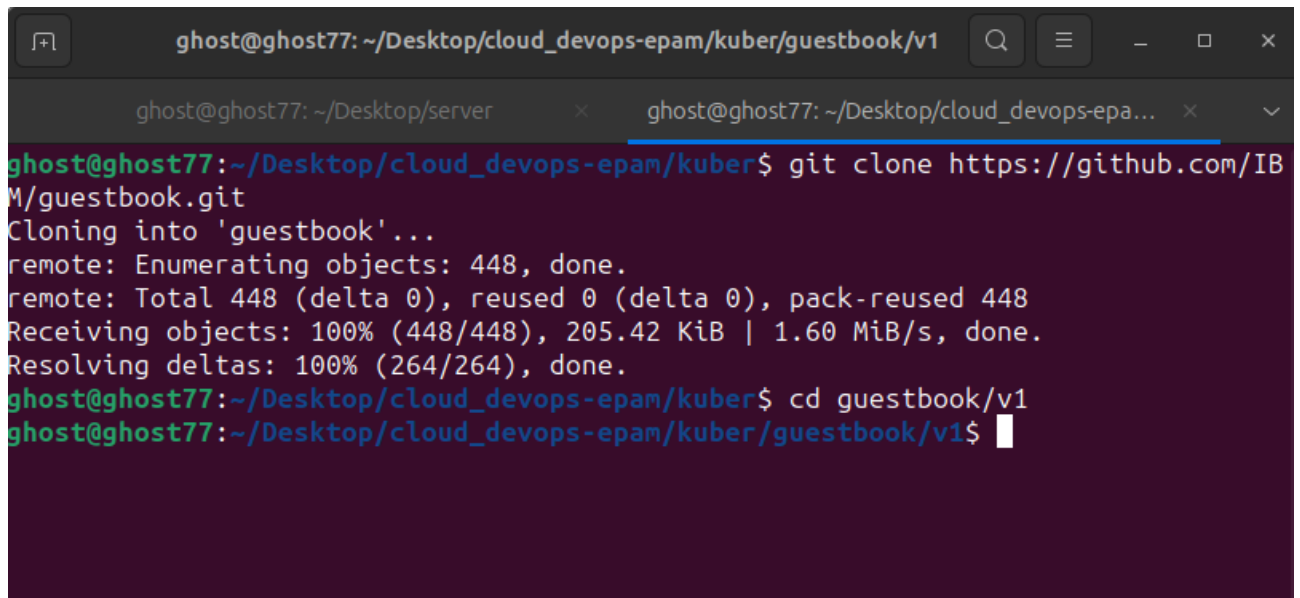
7. Three Tier web application using Kubernetes

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

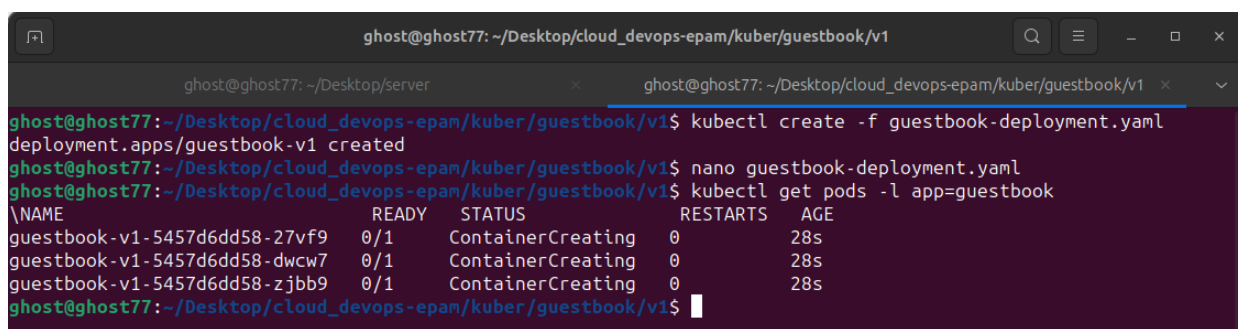
- `git clone https://github.com/IBM/guestbook.git`
 - Change directory to the version 1 of the guestbook application:
3. `cd guestbook/v1`



```
ghost@ghost77: ~/Desktop/cloud_devops-epam/kuber/guestbook/v1
ghost@ghost77: ~/Desktop/server x ghost@ghost77: ~/Desktop/cloud_devops-epa... x
ghost@ghost77:~/Desktop/cloud_devops-epam/kuber$ git clone https://github.com/IBM/guestbook.git
Cloning into 'guestbook'...
remote: Enumerating objects: 448, done.
remote: Total 448 (delta 0), reused 0 (delta 0), pack-reused 448
Receiving objects: 100% (448/448), 205.42 KiB | 1.60 MiB/s, done.
Resolving deltas: 100% (264/264), done.
ghost@ghost77:~/Desktop/cloud_devops-epam/kuber$ cd guestbook/v1
ghost@ghost77:~/Desktop/cloud_devops-epam/kuber/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`

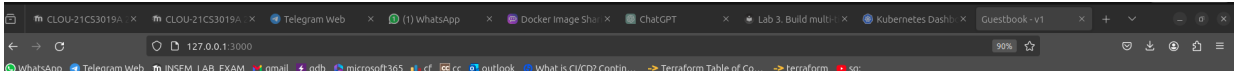


```
ghost@ghost77: ~/Desktop/cloud_devops-epam/kuber/guestbook/v1
ghost@ghost77:~/Desktop/cloud_devops-epam/kuber/guestbook/v1$ kubectl create -f guestbook-deployment.yaml
deployment.apps/guestbook-v1 created
ghost@ghost77:~/Desktop/cloud_devops-epam/kuber/guestbook/v1$ nano guestbook-deployment.yaml
ghost@ghost77:~/Desktop/cloud_devops-epam/kuber/guestbook/v1$ kubectl get pods -l app=guestbook
NAME                                READY   STATUS             RESTARTS   AGE
guestbook-v1-5457d6dd58-27vf9       0/1     ContainerCreating   0           28s
guestbook-v1-5457d6dd58-dwcw7       0/1     ContainerCreating   0           28s
ghostbook-v1-5457d6dd58-zjbb9       0/1     ContainerCreating   0           28s
ghost@ghost77:~/Desktop/cloud_devops-epam/kuber/guestbook/v1$
```

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`

```
ghost@ghost77: ~/Desktop/cloud_devops-epam/kuber/guestbook/v1
ghost@ghost77: ~/Desktop/server
ghost@ghost77: ~/Desktop/cloud_devops-epam/kuber/guestbook/v1
ghost@ghost77:~/Desktop/cloud_devops-epam/kuber/guestbook/v1$ kubectl create -f guestbook-service.yaml
Error from server (AlreadyExists): error when creating "guestbook-service.yaml": services "guestbook" already exist
ghost@ghost77:~/Desktop/cloud_devops-epam/kuber/guestbook/v1$ kubectl describe service guestbook
Name:                guestbook
Namespace:            default
Labels:               app=guestbook
Annotations:          <none>
Selector:             app=guestbook
Type:                 LoadBalancer
IP Family Policy:     SingleStack
IP Families:          IPv4
IP:                   10.104.52.201
IPs:                  10.104.52.201
Port:                 <unset> 3000/TCP
TargetPort:           http-server/TCP
NodePort:             <unset> 31568/TCP
Endpoints:            10.244.0.31:3000,10.244.0.32:3000,10.244.0.33:3000
Session Affinity:     None
External Traffic Policy: Cluster
Events:               <none>
ghost@ghost77:~/Desktop/cloud_devops-epam/kuber/guestbook/v1$ kubectl get nodes -o wide
NAME                STATUS    ROLES          AGE   VERSION   INTERNAL-IP   EXTERNAL-IP   OS-IMAGE             KERNEL-VERSION
minikube            Ready    control-plane  27d   v1.28.3   192.168.58.2   <none>        Ubuntu 22.04.3 LTS   6.6.12-linuxk
it docker://24.0.7
ghost@ghost77:~/Desktop/cloud_devops-epam/kuber/guestbook/v1$
```



Guestbook - v1

SUBMIT

<http://127.0.0.1:3000/>
[/env /info](#)

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`

```
ghost@ghost77: ~/Desktop/cloud_devops-epam/kuber/guestbook/v1
ghost@ghost77: ~/Desktop/server
ghost@ghost77: ~/Desktop/cloud_devops-epam/kuber/guestbook/v1
ghost@ghost77:~/Desktop/cloud_devops-epam/kuber/guestbook/v1$ kubectl create -f redis-master-deployment.yaml
deployment.apps/redis-master created
ghost@ghost77:~/Desktop/cloud_devops-epam/kuber/guestbook/v1$ kubectl get pods -l app=redis,role=master
NAME                                READY   STATUS    RESTARTS   AGE
redis-master-77456ff7b4-6v7th       1/1     Running   0           28s
ghost@ghost77:~/Desktop/cloud_devops-epam/kuber/guestbook/v1$ kubectl exec -it redis-master-77456ff7b4-6v7th
redis-cli
kubectl exec [POD] [COMMAND] is DEPRECATED and will be removed in a future version. Use kubectl exec [POD] -- [COMMAND] instead.
127.0.0.1:6379> ecit
(error) ERR unknown command 'ecit'
127.0.0.1:6379> exit
ghost@ghost77:~/Desktop/cloud_devops-epam/kuber/guestbook/v1$ kubectl create -f redis-master-service.yaml
service/redis-master created
ghost@ghost77:~/Desktop/cloud_devops-epam/kuber/guestbook/v1$
```

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.

```
ghost@ghost77: ~/Desktop/cloud_devops-epam/kuber/guestbook/v1
ghost@ghost77: ~/Desktop/server
ghost@ghost77: ~/Desktop/cloud_devops-epam/kuber/guestbook/v1
ghost@ghost77:~/Desktop/cloud_devops-epam/kuber/guestbook/v1$ kubectl delete deploy guestbook-v1
Error from server (NotFound): deployments.apps "guestbook-v1" not found
ghost@ghost77:~/Desktop/cloud_devops-epam/kuber/guestbook/v1$ kubectl create -f guestbook-deployment.yaml
deployment.apps/guestbook-v1 created
ghost@ghost77:~/Desktop/cloud_devops-epam/kuber/guestbook/v1$
```

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

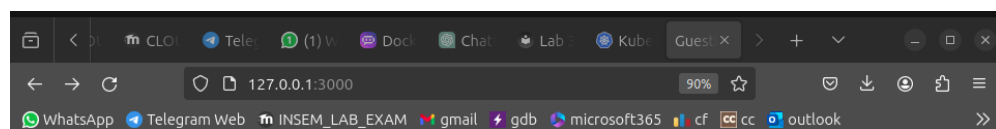
1. Deploy the Redis slave by creating a Deployment using the provided configuration file:

- `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.

```
ghost@ghost77: ~/Desktop/cloud_devops-epam/kuber/guestbook/v1
ghost@ghost77: ~/Desktop/server
ghost@ghost77: ~/Desktop/cloud_devops-epam/kuber/guestbook/v1
redis-slave-5698767898-st9rt 0/1 ContainerCreating 0 11s
ghost@ghost77:~/Desktop/cloud_devops-epam/kuber/guestbook/v1$ kubectl create -f redis-slave-service.yaml
service/redis-slave created
ghost@ghost77:~/Desktop/cloud_devops-epam/kuber/guestbook/v1$ kubectl delete deploy guestbook-v1
deployment.apps "guestbook-v1" deleted
ghost@ghost77:~/Desktop/cloud_devops-epam/kuber/guestbook/v1$ kubectl create -f guestbook-deployment.yaml
deployment.apps/guestbook-v1 created
ghost@ghost77:~/Desktop/cloud_devops-epam/kuber/guestbook/v1$ kubectl describe service guestbook
Name:         guestbook
Namespace:    default
Labels:       app=guestbook
Annotations:  <none>
Selector:     app=guestbook
Type:         LoadBalancer
IP Family Policy: SingleStack
IP Families:  IPv4
IP:           10.104.52.201
IPs:          10.104.52.201
Port:         <unset> 3000/TCP
TargetPort:   http-server/TCP
NodePort:     <unset> 31568/TCP
Endpoints:    10.244.0.40:3000,10.244.0.41:3000,10.244.0.42:3000
Session Affinity: None
External Traffic Policy: Cluster
Events:       <none>
ghost@ghost77:~/Desktop/cloud_devops-epam/kuber/guestbook/v1$ kubectl port-forward service/guestbook 3000:3000
Forwarding from 127.0.0.1:3000 -> 3000
Forwarding from [::1]:3000 -> 3000
Handling connection for 3000
```

S



Guestbook - v1

wew
weve
hello
i am moksh

<http://127.0.0.1:3000/>
[/env /info](#)