Designing for deployment - Category microservice

In this lab we will look into an existing brown field category microservice and deploy it to the K8s cluster.

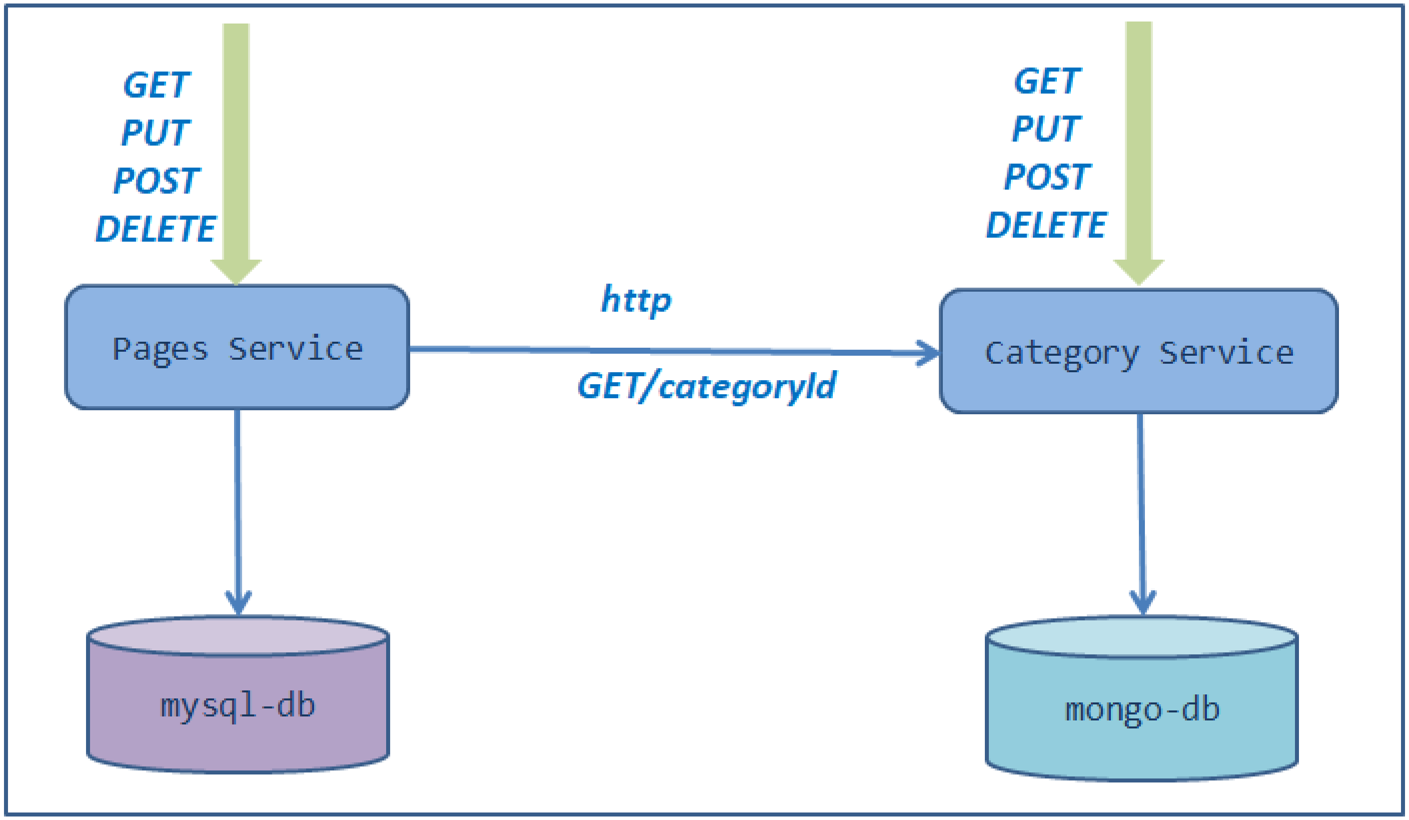
Learning Outcomes

After completing the lab, you will be able to:

1. Deploy category microservice
2. Approaching the deployment scenario for a distributed microservice architecture

Understanding the high-level distributed architecture

Distributed Application Architecture

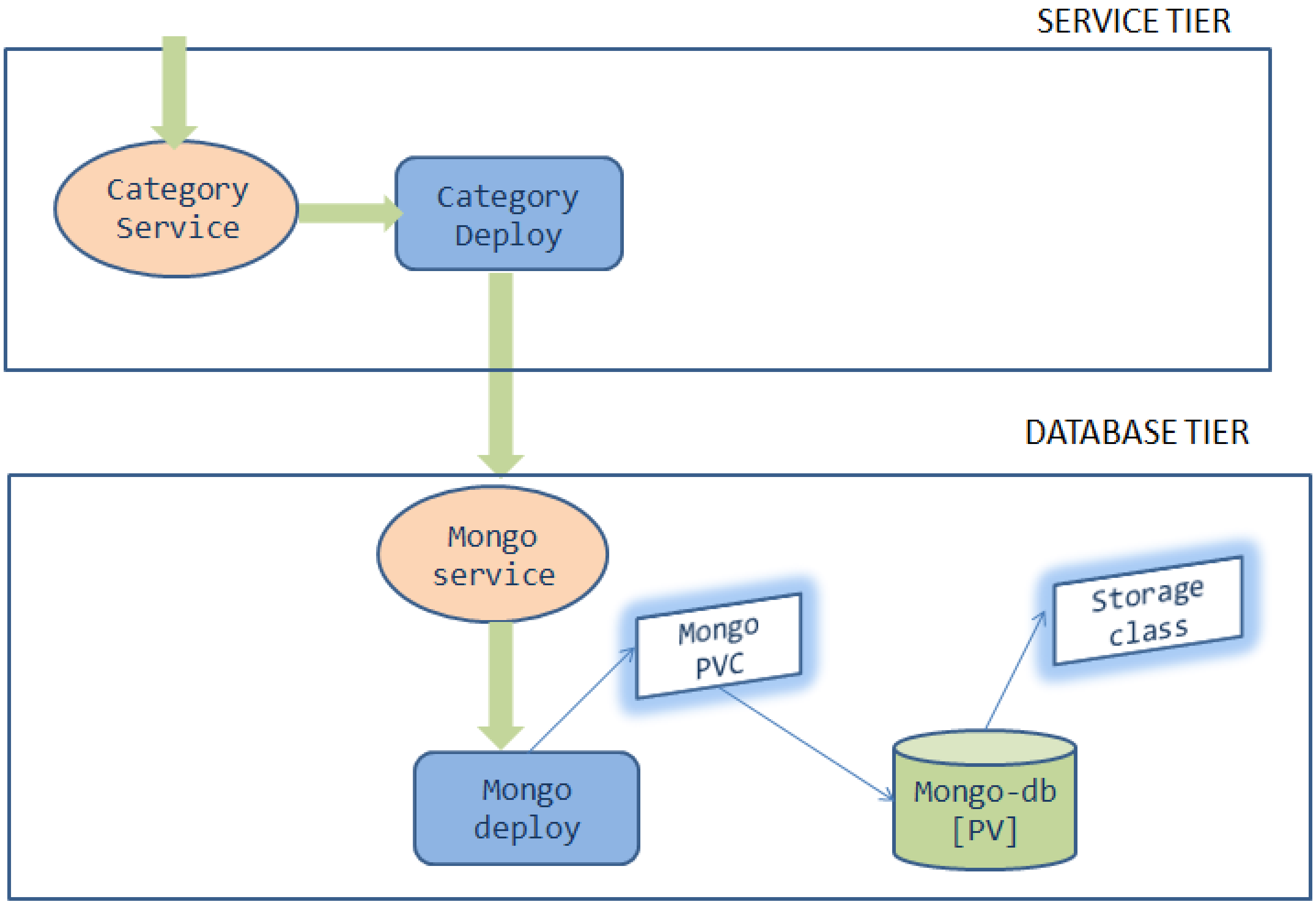


We will be focusing on the category microservice in this lab. In the next lab we will deploy the pages microservice

Deploying category microservice to K8s

1. Delete the contents of `~/workspace/kubernetes-manifests` directory
2. Download category microservice manifest files and extract to `~/workspace/kubernetes-manifests/category`
3. Download mongodb manifest files and extract to `~/workspace/kubernetes-manifests/mongo`
4. Walkthrough the manifest files & understand the solution to the deployment architecture.

5. Before we start deploying, replace `[student-name]` with your namespace in all the manifest files.



Category Service - Deployment Architecture

Deployment Guide

1. Verify the kubectl context `kubectl config get-contexts` is set to minikube. If not, set it to minikube `kubectl config use-context minikube`

2. Set up `[student-name]` namespace to point to the current context. If the namespace is not created, the deployments will not work.

```
kubectl config set-context --current --namespace=[student-name]
```

3. Create the Database tier

```
cd ~/workspace/kubernetes-manifests/mongo
kubectl apply -f storage-class.yaml
kubectl apply -f pv.yaml
kubectl apply -f pvc.yaml
kubectl apply -f service.yaml
kubectl apply -f deployment.yaml
```

4. Verify the deployment of database tier

```
kubectl get deployment mongo
kubectl get service mongo
kubectl get pvc
```

5. Proceed further if there are no errors, otherwise troubleshoot and fix them.

6. Create the service tier

```
cd ~/workspace/kubernetes-manifests/category
kubectl apply -f service.yaml
kubectl apply -f deployment.yaml
```

7. Verify the deployment of service tier

```
kubectl get deployment category
kubectl get service category
```

8. Access the category application

```
kubectl port-forward svc/category 8080:8080
```

9. Refer Curl Guide for testing and proceed with the next steps

Task Accomplished

We successfully deployed a 2 tier category microservice application to K8s cluster.