Kuberentes contd

Exercise 4: create a pod manifest for postgres where we need to pass environmental variables

- Solution: Refer Here for the changeset

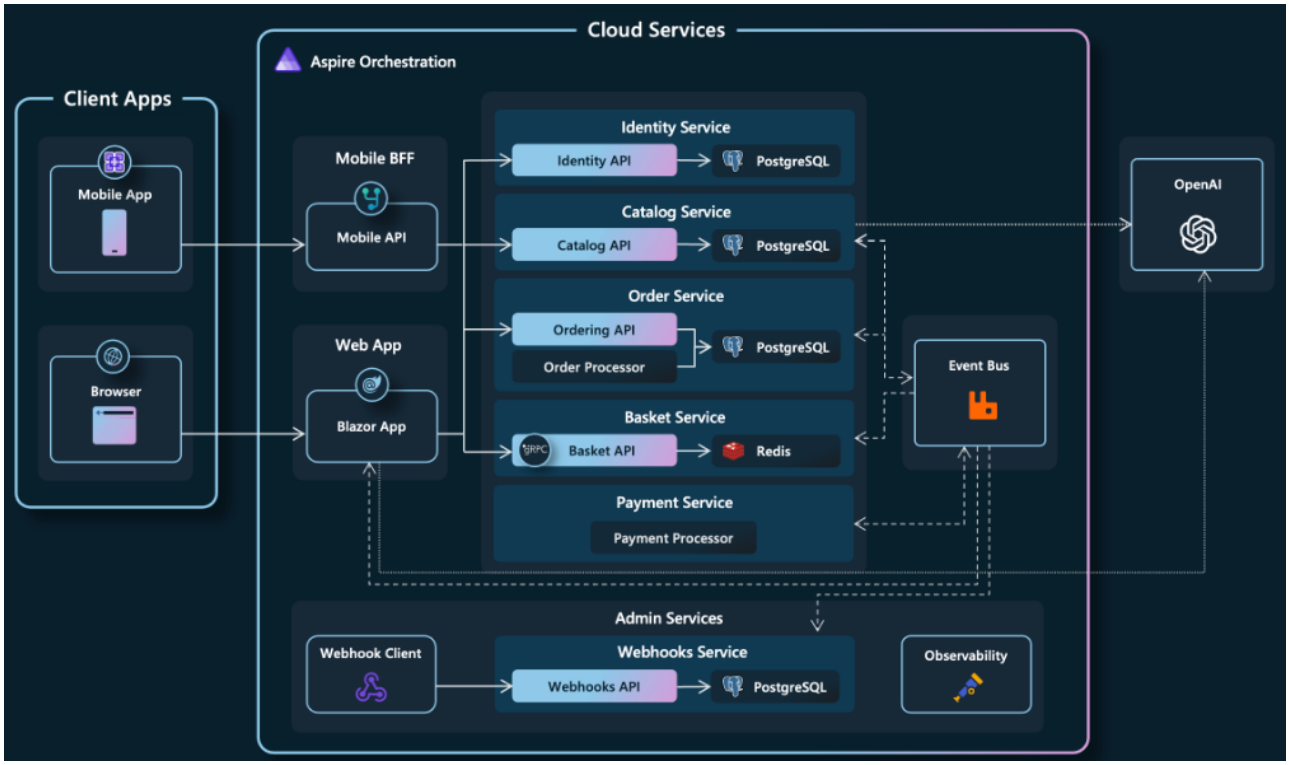
```
Initialising Kubernetes... done

controlplane $ cd manifests/
controlplane $ kubectl apply -f exercise4.yaml
pod/nop-db created
controlplane $ kubectl get po
NAME     READY   STATUS              RESTARTS   AGE
nop-db   0/1     ContainerCreating   0          5s
controlplane $ kubectl get po -w
NAME     READY   STATUS              RESTARTS   AGE
nop-db   0/1     ContainerCreating   0          9s
nop-db   1/1     Running             0          15s
```

Managed Kubernetes or Kubernetes as a service

- Managed k8s is offered by cloud providers
- Cloud providers manage
  - control plane (charged hourly)
  - no extra charges for nodes (they are charged as if you are running ec2 or azure vms)
  - network addons provided
    - load balancers
    - firewalls
    - dns
    - various cni plugins
  - storage
    - volumes on cloud
    - secrets in vaults
- Some cloud providers have a new model of pricing. Price per pod not nodes

Microservices & It architectures

- Ecommerce reference microservice architecture



- Here each service needs to be scaled. Generally there are two types of scaling
    - manual
    - automatic
- Whenever we get a new release it is not necessarily all micro-services, irrespective of features or services impacted, we should perform zero downtime deployment
- Generally scaling the pods indirectly demands more infra
- Lets see what Kuberentes can do
    - Replicasets can scale pods
    - Automatic scaling of
        - pods can be done horizontal pod autoscaler
        - nodes in managed kuberentes (aks, eks, gke) is possible via node auto scalers

Controllers

- Controllers control pods
- We have following controllers
    - Replicaset (Replication Controller)
    - Deployment
    - Daemoset
    - Job
    - Cronjob
    - Stateful set (will be discussed post persistent volumes)
- Generally when we deal with controllers, we will also need to understand
    - service

Replicaset

- Refer Here for official docs
- This controller is responsible for replicas of pods
- desired state
  - which pod
  - how many
- Replicaset will find the number of pods by matching labels
- Generally in all controller objects of k8s we have a field called as template
- In Replicaset we create pods, so the template will have pod specificiation.
- Lets write a replicaset of 2 httpd containers Refer Here for the manifest

```
controlplane $ kubectl get rs
No resources found in default namespace.
controlplane $ kubectl get po
No resources found in default namespace.
controlplane $ kubectl apply -f httpd-rs.yaml
replicaset.apps/httpd-rs created
controlplane $ kubectl get rs
NAME       DESIRED   CURRENT   READY   AGE
httpd-rs   2         2         0       5s
controlplane $ kubectl get po
NAME             READY   STATUS    RESTARTS   AGE
httpd-rs-66lln   1/1     Running   0          17s
httpd-rs-c2f2h   1/1     Running   0          17s
controlplane $
```

- We have tried (watch classroom video for this)
  - impact of labels on replicaset
  - deleting replicasets
  - changing the replicas
- We also have ReplicationController which does the same thing as replicaset, but there is one major reason and two differences
  - Major reason for Replicaset was created is to support deployments
  - two differences
    - replication controller uses only equality based selection of pods
    - replicaset internally supports persistence
- Exercise:
  - write a replicaset spec to create 3 springpetclinic pods with label `app=spc`