Series of events

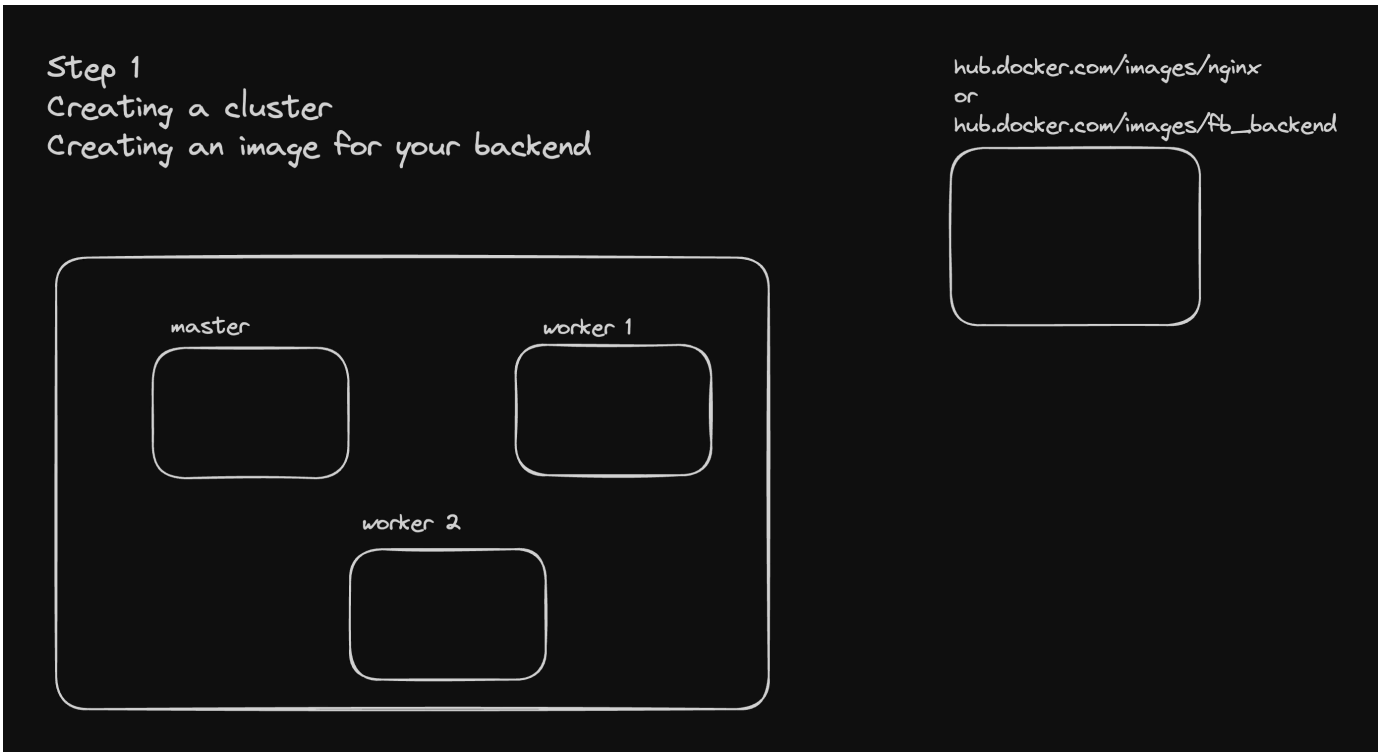
Step 1 - Create your cluster

- Create kind.yml

```
kind: Cluster
apiVersion: kind.x-k8s.io/v1alpha4
nodes:
- role: control-plane
  extraPortMappings:
  - containerPort: 30007
    hostPort: 30007
- role: worker
  extraPortMappings:
  - containerPort: 30007
    hostPort: 30008
- role: worker
```

- Create cluster

```
kind create cluster --config kind.yml --name local
```



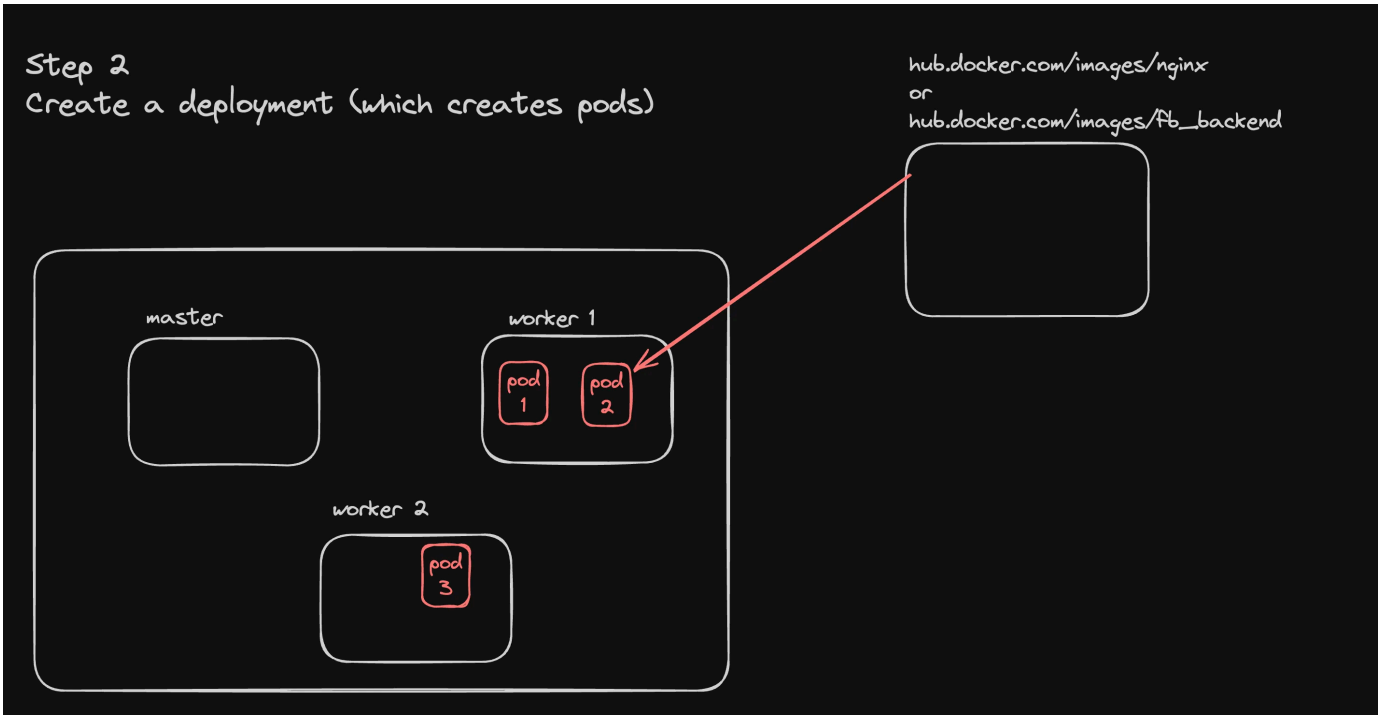
Step 2 - Deploy your pod

- Create deployment.yml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:latest
        ports:
```

- Apply the deployment

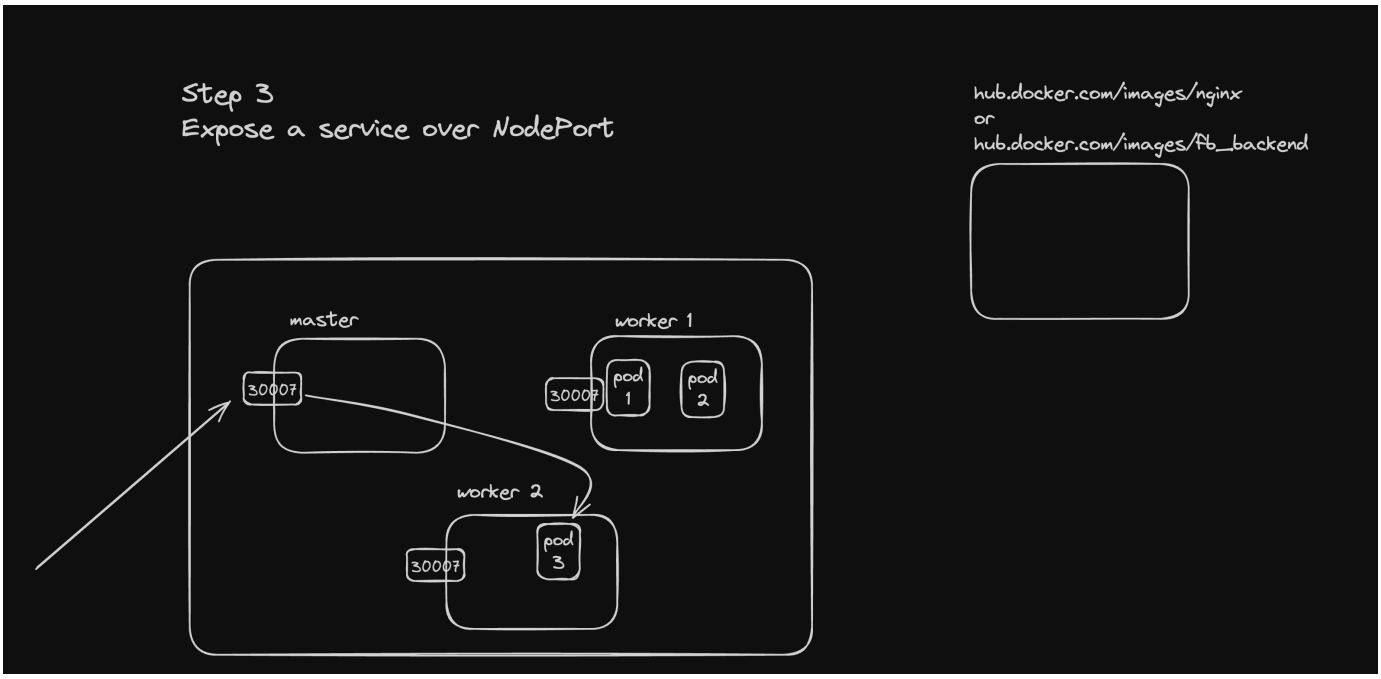
```
kubectl apply -f deployment.yml
```



Step 3 - Expose your app over a NodePort

- Create service.yml

```yaml
apiVersion: v1
kind: Service
metadata:
  name: nginx-service
spec:
  selector:
    app: nginx
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
      nodePort: 30007  # This port can be any valid port within the NodePort rang
  type: NodePort
```



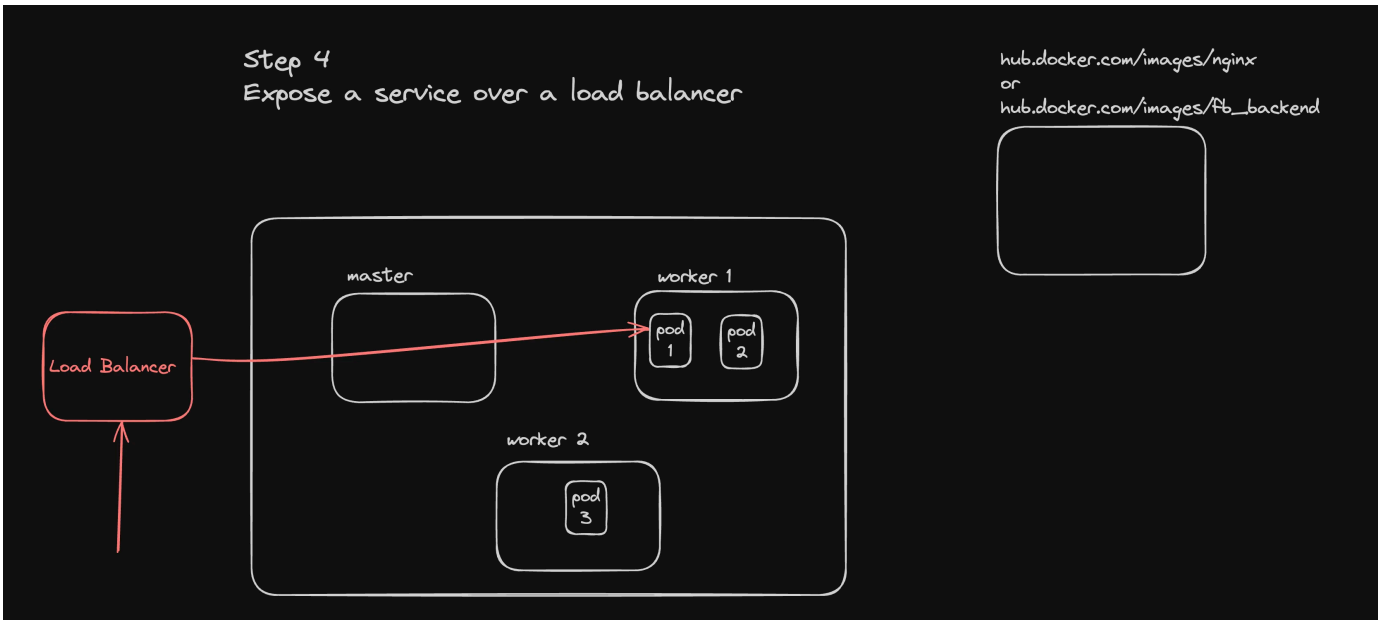
Step 4 - Expose it over a LoadBalancer

- Create a load balancer service (service-lb.yml)

```yaml
apiVersion: v1
kind: Service
metadata:
  name: nginx-service
spec:
  selector:
    app: nginx
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
  type: LoadBalancer
```

- Apply the configuration

```
kubectl apply service-lb.yml
```



Check the cloud dashboard