Kubctl commands

```
 apiVersion: v1
kind: Namespace
metadata:
  name: auth
kubectl create -f ./ns.yaml
 --deployment file
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
  labels:
    app: nginx
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:1.14.2
        ports:
        - containerPort: 80kubectl apply -f ./dep.yaml
---service file
apiVersion: v1
kind: Service
metadata:
   name: nodejs-service
   namespace: auth
   labels:
     app: nodejs
spec:
   ports:
    - port: 80
      protocol: TCP
   type: LoadBalancer
   selector:
      app: nodejs-server
```

kubectl apply -f ./service.yaml
kubectl get pods
kubectl get svc -n test
kubectl expose deployment nodejs-deployment --type="LoadBalancer"

https://<external-ip>:<port>

kubectl apply -f deployment.yaml -n auth

Using charts:
.charts/charts/authorization-implementation-reactor
helm template .|kubectl apply -f - -n auth

kubectl edit deployment authorization-implementation-reactor -n auth
k describe pods <podname> -n auth

Commands:

kubectl get namespaces
kubectl get pods -n auth
kubectl edit deployment authorization-implementation-reactor -n auth
k describe pods <podname> -n auth
kubectl get pods --all-namespaces

Bash into pod :

- kubectl exec -it authorization-implementation-reactor-c4bd466f7-fztpl sh

To increase containers: (scaling):

kubectl scale deployment business-events-jobs-keeper --replicas=2
kubectl scale deployment business-events-producer-service --replicas=2
kubectl scale deployment identity-token-service --replicas=2

Only last 100 logs:

k logs -f <pod name> --tail=100

Eg: k logs -f business-events-jobs-keeper-85f6cfd665-4txsp --tail=100