OpenShift Commands Cheat Sheet

#login with admin user

oc login https://<master-server>:8443 -u admin -p redhat

#login as developer user

oc login https://<master-server>:8443 -u developer -p developer

#Logged in username

oc whoami

#Create a new app from a GitHub Repository

oc new-app https://github.com/chetantiwary/example-app

#New app from a different branch

oc new-app --name=web nginx:1.10~https://github.com/chetantiwaryxxxx#mybranch

#Create objects from a file:

oc create -f anyobject.yaml -n <project>

Eg. oc create -f service.yaml -n operation

#Delete objects contained in a file:

oc delete -f anyobject.yaml -n <project>

#Create or merge objects from file

oc apply -f anyobject.yaml -n <project>

#Monitor Pod status

watch oc get pods

#Gather information of a pod deployment with more details

oc get pods -o wide

#do not show inactive Pods

oc get pods --show-all=false

#show all resources

oc get all

oc get pods

oc get service

oc get route

oc get secrets

oc get configmap

oc get limitranges

oc get resourcequota

oc get hpa dc/dcname

oc get pv

oc get pvc

oc get nodes

oc get ingress

oc get networkpolicy

#Get Openshift Console Address

```
oc whoami --show-console

#Copy a local folder to app Pod under the folder /opt/jboss
oc cp ./file app:/opt/jboss

#Create a ConfigMap from file
oc create configmap my-config --from-file=config.properties

oc create secret generic my-secret --from-file=secret.key

#Create a ConfigMap/Secret from literals
oc create configmap my-config --from-literal=foo=bar --from-literal=app=blu
oc create secret generic my-secret --from-literal=secret.key=value

#Set a ConfigMap/Secret in a deployment
oc set env deployment/my-deployment --from configmap/my-config
oc set env deployment/my-deployment --from secret/my-secret

# Update deployment green with a new environment variable
oc set env dc/green STORAGE_DIR=/local

# List the environment variables defined on all pods
oc set env pods --all --list

# Import environment from a secret
oc set env --from=secret/mysecret dc/myapp
```

```
#Get Nodes list
oc get nodes

#Check on which Node your Pods are running
oc get pods -o wide

#List all pods which are running on a Node
oc adm manage-node node1.fqdn --list-pods

#Add a label to a Node
oc label node node1.fqdn label=value

#Remove a label from a Node
oc label node node1.fqdn label-

#create a PersistentVolumeClaim
oc set volume dc/<dcname> --add --name=shared-storage \
-t pvc --claim-mode=ReadWriteMany --claim-size=1Gi \
--claim-name=shared-storage --claim-class=ocs-storagecluster-cephfs \
--mount-path=/opt/app-root/src/uploaded \
-n shared-storage

#Manual deployment
oc rollout latest <dcname>
```

#Pause automatic deployment rollout

oc rollout pause dc <dcname>

Resume automatic deployment rollout

oc rollout resume dc <dcname>

#Define resource requests and limits in DeploymentConfig

oc set resources deployment nginx --limits=cpu=200m,memory=512Mi --requests=cpu=100m,memory=256

#Define livenessProve and readinessProve in DeploymentConfig

oc set probe dc/nginx --readiness --get-url=http://:8080/healthz --initial-delay-seconds=10

oc set probe dc/nginx --liveness --get-url=http://:8080/healthz --initial-delay-seconds=10

#Scale the number of Pods to 5

oc scale dc/nginx --replicas=5

#Define Horizontal Pod Autoscaler (hpa)

oc autoscale dc nginx --max=4 --min=2 --cpu-percent=60

#Create route with default hostname

oc expose service <servicename>

#Create Route and expose it through a custom Hostname

oc expose service <servicename> --hostname <hostname>

#Read the Route Host attribute

oc get route my-route -o jsonpath --template="{.spec.host}"

#Common Troubleshooting

oc delete all -l key=value

oc get all

oc describe pod <pod-name>

oc describe node <node-name>

oc get nodes -L <label>

oc label node <node-name> key=value

oc get endpoints -n <namespace/project>

oc describe node <node-name> | grep -i taint

oc get events

oc get dc <dc-name> -o yaml

oc rollout latest hello

oc logs <hello-2-abcd>

oc expose service hello --hostname=hello.apps.lab.example.com

oc debug pod <PODNAME>

oc edit service <service-name>

oc edit deployment <deploymentname>

oc edit ingress <ingressname>

oc edit route <routename>

oc adm cordon <nodename> (to mark the node unschedulable)