**DaemonSet**

Create a daemonset with

name : my-ds

image: nginx

| **apiVersion**: apps/v1  **kind**: DaemonSet  **metadata**:  **name**: my-ds  **spec**:  **template**:  **metadata**:  **name**: pod-for-ds  **labels**:  **app**: ds  **spec**:  **containers**:  **- image**: nginx  **name**: nginx  **selector**:  **matchLabels**:  **app**: ds |
| --- |

**Service**

Expose your deployment with the following command:

kubectl expose deployment “Your-deployment-name” --name mysvc --port 80

Get the IP of the svc

kubectl get svc -o wide

Access the application running behind the svc

curl <ClusterIP> command

Create Nodeport svc

kubectl expose deployment mydep --name mysvc --port 80 --type NodePort

Create loadbalancer svc

kubectl expose deployment mydep --name mysvc-lb --port 80 --type LoadBalancer

**Storage:**

**Volumes**

| **apiVersion**: v1  **kind**: Pod  **metadata**:  **name**: random-number-generator  **spec**:  **containers** :  **- image**: alpine  **name**: alpine  **command**: ["/bin/sh","-c","shuf -i 0-100 -n 1 >> /opt/number.out && sleep 3600"]  **volumeMounts**:  **- mountPath**: /opt  **name**: mydata  **volumes**:  **- name**: mydata  **hostPath**:  **path**: /tmp  **type**: Directory |
| --- |

Check the random number generated

kubectl exec random-number-generator -- cat /opt/number.out

Delete the pod, and recreate it.

kubectl delete pod random-number-generator

Check the random number generated (you should see the previous number preserved)

kubectl exec random-number-generator -- cat /opt/number.out

**PV - PVC and Pods**

1. Create a PV with 1Gi storage

| **apiVersion**: v1  **kind**: PersistentVolume  **metadata**:  **name**: pv1  **spec**:  **capacity**:  **storage**: 1Gi  **accessModes**:  - ReadWriteOnce  **hostPath**:  **path**: /tmp/data |
| --- |

1. Create a PVC with 500Mi of storage

| **apiVersion**: v1  **kind**: PersistentVolumeClaim  **metadata**:  **name**: claim1  **spec**:  **accessModes**:  - ReadWriteOnce  **resources**:  **requests**:  **storage**: 500Mi |
| --- |

1. Check with PVC is bound or not

kubectl get pvc

1. Mount the storage in Pod

| **apiVersion**: v1  **kind**: Pod  **metadata**:  **name**: mypod  **spec**:  **containers**:  **- name**: myfrontend  **image**: nginx  **volumeMounts**:  **- mountPath**: "/tmp/html"  **name**: my-pvc  **volumes**:  **- name**: my-pvc  **persistentVolumeClaim**:  **claimName**: claim1 |
| --- |