

## # Day - 4

→ Various objects in kubernetes.

- 1) pod
- 2) Service
- 3) Replica Set
- 4) Deployment
- 5) Configmap
- 6) Secret
- 7) Volumes.

1) pod :-

- \* Smallest deployable units of computing that we can create and manage in k8s.
- \* Consists of one or more containers with shared storage and network resources.

Simple pod definition file :-

```
apiVersion: v1,  
kind: pod  
metadata:  
  name: nginx  
spec:  
  containers:  
    - name: nginx  
      image: nginx:1.14.2  
      ports:  
        - containerPort: 80
```

## 2) Service :-

- \* A service is a method for exposing a network application that is running as one/more pods in your cluster.
- \* Each Service object defines logical set of endpoints (usually these endpoints are pods) along with policy about how to make these pods available.

## 3) Replica Set :-

- \* A ReplicaSet's purpose is to maintain a stable set of pods running at any given time. i.e it is used to guarantee the availability of a specified number of identical pods.

## 4) Deployment :-

- \* Deployments provides declarative updates for pods & Replica Sets.
- \* You can describe state in deployment and the deployment controller changes actual state to desired state at controlled state. you can define deployments to create new replicasets or to remove existing deployments and adopt all their resources with new deployments.

### 5) Configmaps:-

- \* It is an API object used to store key-value pairs which are non-confidential.
- \* pods can consume Configmaps as environment variable, CLI arguments & Config files in Volumes.
- \* Name of the Configmap must be valid DNS Subdomain name.

### 6) Secrets:-

- \* A Secret is a object that contains a small amount of Sensitive data such as password, a token or a key.
- \* using a Secret means you don't need to use Confidential data in your application code.

### 7) Volumes:-

On disk files in a container are ephemeral, when container terminates / crashes we lost all data.

Multiple containers are running in a pod and needs to share files. It's challenging to establish shared file system among all of containers. K8s "Volume" abstraction solves both of these problems.

- Kubernetes supports several types of volumes.