**Kubernetes Notes**

|  |  |
| --- | --- |
| Kind | Version |
| POD | v1 |
| Service | v1 |
| ReplicaSet | apps/v1 |
| Deployment | apps/v1 |

YAML in Kubernetes

pod-definition.yml

apiVersion:

kind: <Type of Object>

metadata:

spec:

* Kubernetes uses YAML files as inputs for the creation of objects such as POD's, replicas, deployment services, etc.
* All of these follow similar structure Kubernetes definition file always contains 4 top level fields. The API version, kind, metadata and spec.
* These are the top level or root level properties.
* These are also required fields so you must have them in your configuration file.

**Create an NGINX Pod**

Create single pod of nginx-🡪 kubectl run nginx --image=nginx

Or

apiVersion: v1

kind: Pod

metadata:

name: myapp-pod

labels:

app: myapp

spec:

containers:

- name: nginx

image: nginx

**POD**

* Kubernetes runs workload by placing containers into Pods to run on *Nodes*. A node may be a virtual or physical machine, depending on the cluster. Each node contains the services necessary to run [Pods](https://kubernetes.io/docs/concepts/workloads/pods/), managed by the [control plane](https://kubernetes.io/docs/reference/glossary/?all=true#term-control-plane).
* *Pods* are the smallest deployable units of computing that we can create and manage in Kubernetes.
* In terms of Docker concepts, a Pod is similar to a group of Docker containers with shared namespaces and shared filesystem volumes.

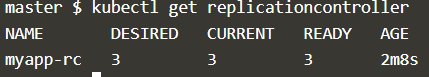
**Replication Controller and Replica Set**

* The only difference between replica set and replication controller is the selector types.
* The replication controller supports equality based selectors whereas the replica set supports equality based as well as set based selectors.

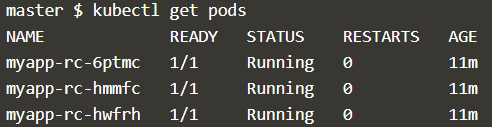
kubectl create -f rc-defintion.yml



kubectl get replicationcontroller



kubectl get pods

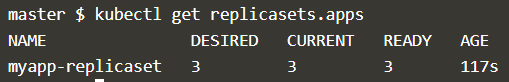


kubectl describe pod <pod-name>

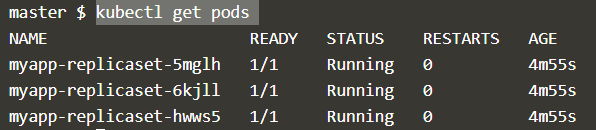
**kubectl create -f replicaset-definition.yml**



**kubectl get replicasets.apps**



**kubectl get pods**

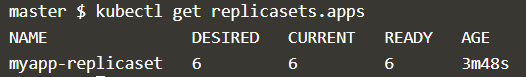


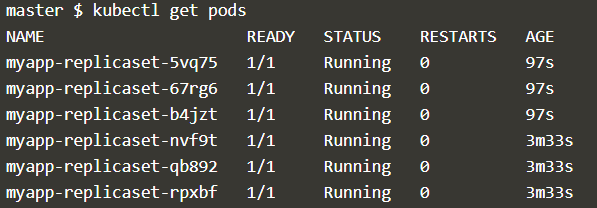
**Scale**

Change the number of replicas in the yml then run the below command

kubectl replace -f replicaset-definition.yml











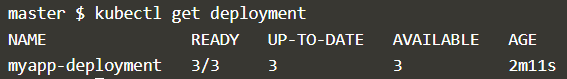
**Deployment**

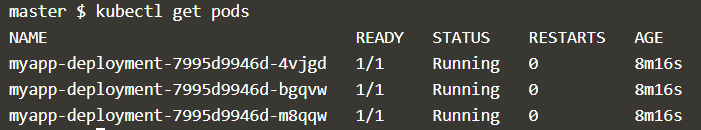
*Deployments* represent a set of multiple, identical [Pods](https://cloud.google.com/kubernetes-engine/docs/concepts/pod) with no unique identities. A Deployment runs multiple replicas of your application and automatically replaces any instances that fail or become unresponsive.

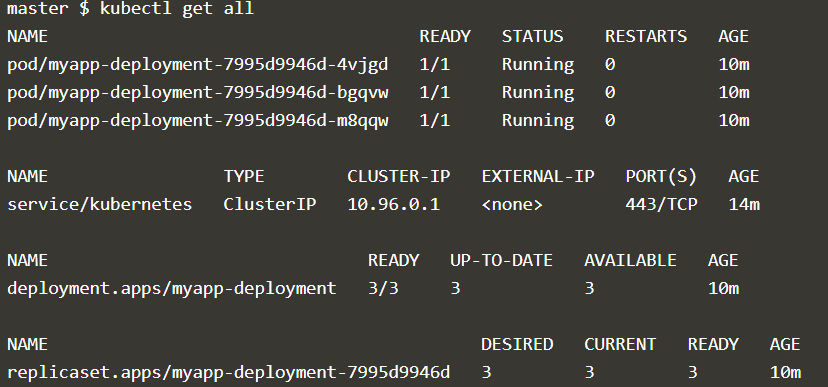
kubectl create -f deployment-definition.yml



kubectl get deployment







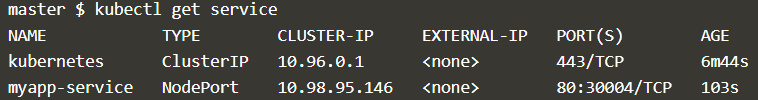
**Service**

* A service is responsible for enabling network access to a set of pods.
* A service can be defined as a logical set of pods. It can be defined as an abstraction on the top of the pod which provides a single IP address and DNS name by which pods can be accessed.

kubectl create -f service-defition.yml



kubectl get service



kubectl describe pod newpods-bqkbm | grep -i image

kubectl edit pods redis