|  |  |
| --- | --- |
| POD |  |
| Replication set / Replica controller | Replicate set  the replication controller can help by automatically bringing up a new POD when the existing one fails. Thus the replication |
| Namespace |  |
| deployment | Recreate & rolling update |
| service |  |
| Daemon set |  |
| Node affinity, Taint and tolerant |  |
|  |  |

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |

Yaml

apiVersion:

kind:

metadata:

name:

labels:

spec:

template:

labels:

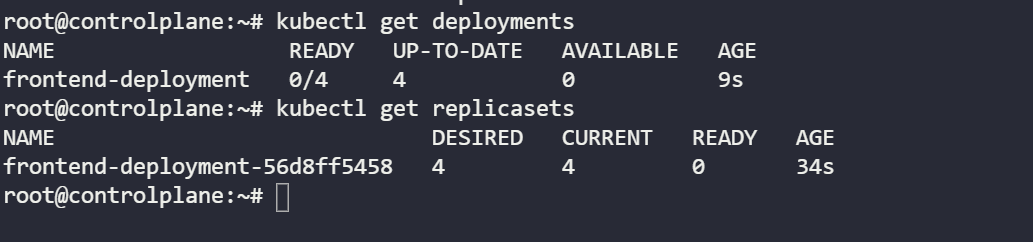
spec:

containers:

-name:

Image:

|  |  |
| --- | --- |
| POD | Apiversion: v1, kind, metadata(name, labels ), spec[template(labels), spec[container]] |
| Service | V1 |
| ReplicaSet | Apps/v1, kind, metadata, spec , selector, replicas |
| Deployment | Apps/v1, kind, metadata, spec , selector, replicas |
| Replication controller |  |



**Why containers**

* Compatibility/Dependency
* Long setup time
* Different Dev/Test/Prod environments

So what are containers? Containers are completely isolated environments, as in they can have their own processes or services, their own network interfaces, their own mounts, just like Virtual machines, except that they all share the same OS kernel.

**Container/Docker VS Virtual Machine**

1. Low Utilization , memory and resources
2. Fast booting
3. Less isolation between services is disadvantage

VM has it’s own OS and hence more isolation

**Container Vs Image**

An image is a package or a template, just like a VM template that you might have worked with in the virtualization world. It is used to create one or more containers.

Containers are running instances off images that are isolated and have their own environments and set of processes

**Kub Architecture - Mater node**

Api- server

ETCD – key value data store

Scheduler - assign container/pods to nodes

Controller - responsible for managing nodes/ container brings up if it goes down

**Worker node**

Kublet – Agent runs on each node to check the container

Container run time like docker

**POD**

Application container and helper container