



Kubernetes Objects



Table of Contents



- ▶ Kubernetes objects
- ▶ PODs
- ▶ ReplicaSets
- ▶ Deployment
- ▶ Namespaces
- ▶ Object Model



1

Kubernetes Objects

Kubernetes Objects



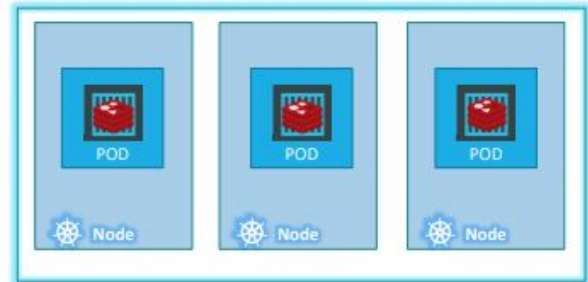
- Kubernetes objects are persistent entities in the Kubernetes system.
- Kubernetes uses these entities to manage the cluster.





2

PODs



PODs



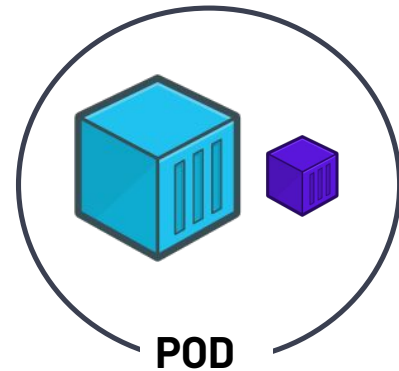
- Kubernetes doesn't deal with containers directly.
- PODs are Kubernetes objects that encapsulate the containers.
- Pods are the smallest deployable units of computing that you can create and manage in Kubernetes.



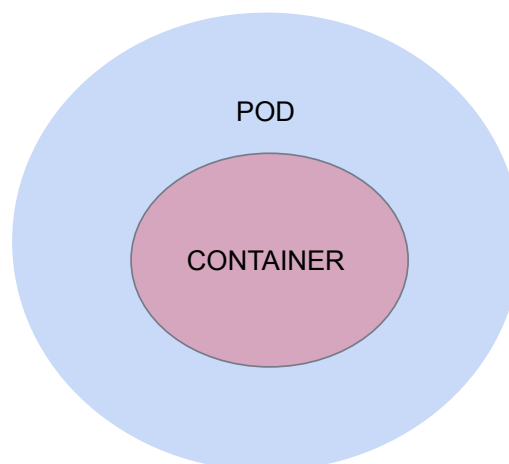
PODs



- A POD can have multiple containers.
- Sometimes an application need a helper container, such as logging, monitoring, etc.
- These helper containers should coexist with your application container.



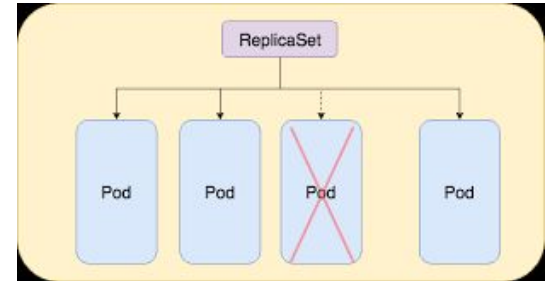
PODs





3

ReplicaSets

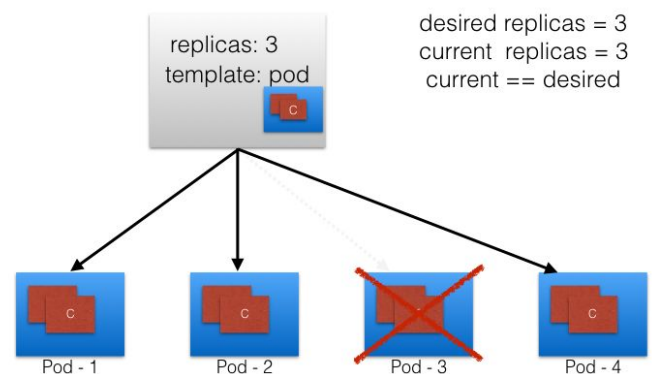


ReplicaSets

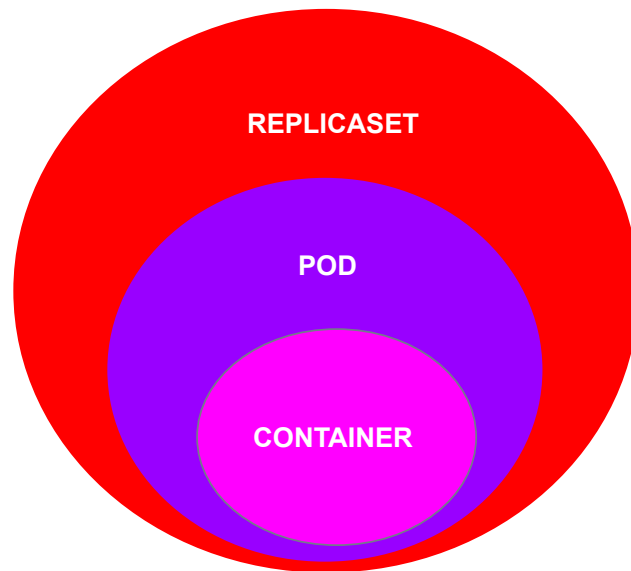


- A **ReplicaSet's** purpose is to maintain a stable set of replica Pods running at any given time.
- Even if you have a single POD, the ReplicaSet will bring up a new POD when the existing one fails.

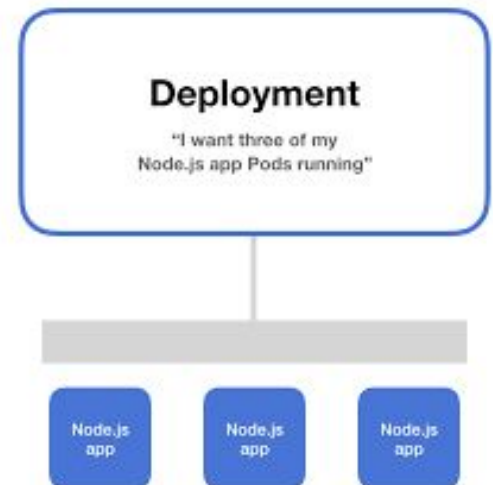
Replica Set



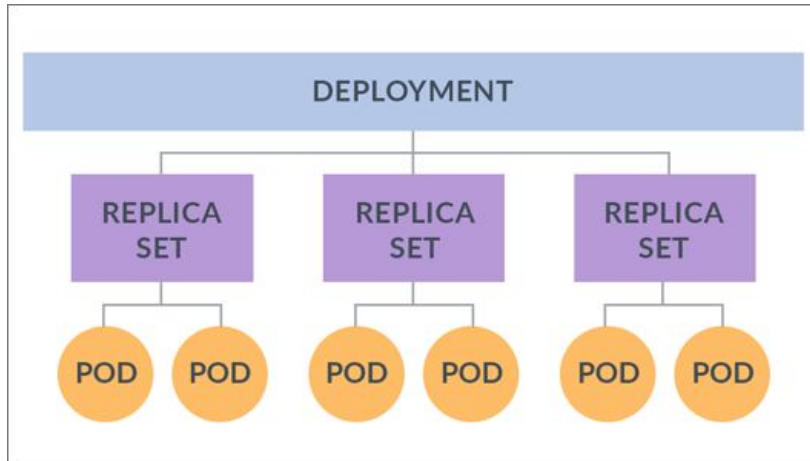
ReplicaSets



4 Deployment

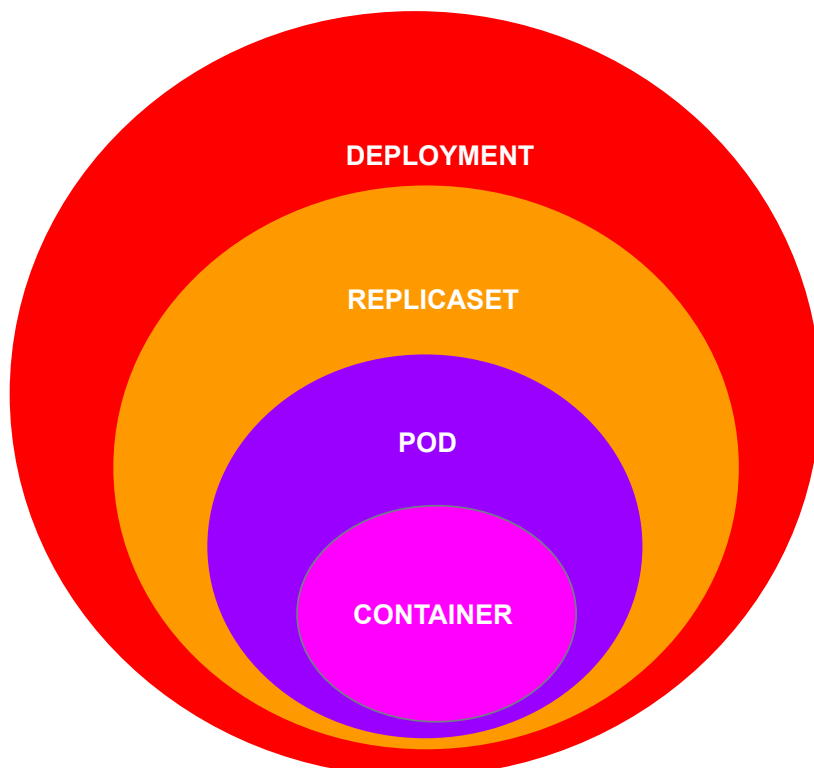


Deployment



- One step higher in the hierarchy, deployments provides declarative updates for Pods and ReplicaSets.

Deployment





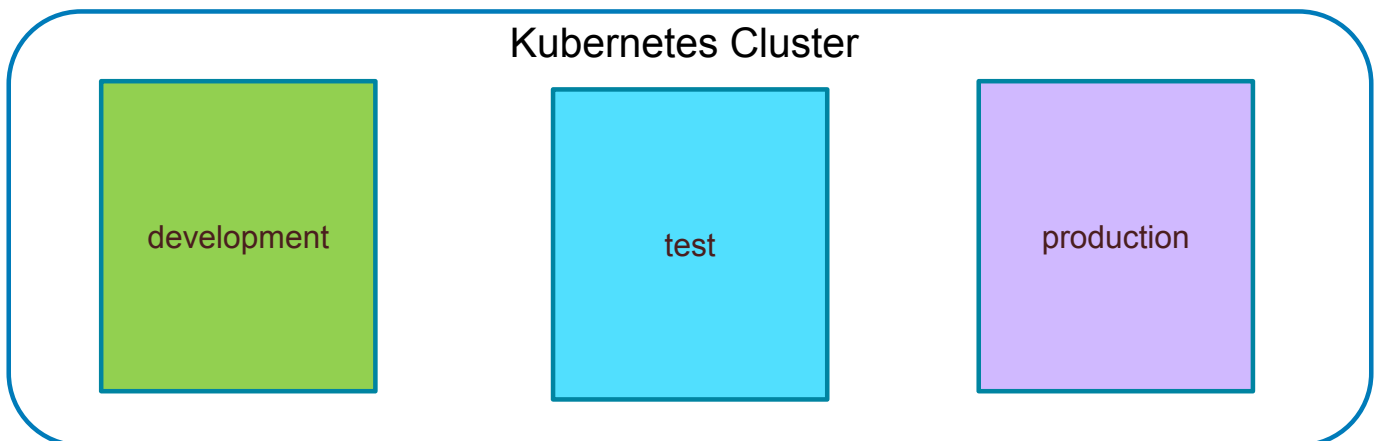
5

Namespaces

Namespaces



- Kubernetes supports multiple virtual clusters backed by the same physical cluster. These virtual clusters are called **namespaces**.
- Namespaces are intended for use in environments with many users spread across multiple teams, or projects.





6 Object Model

Object Model



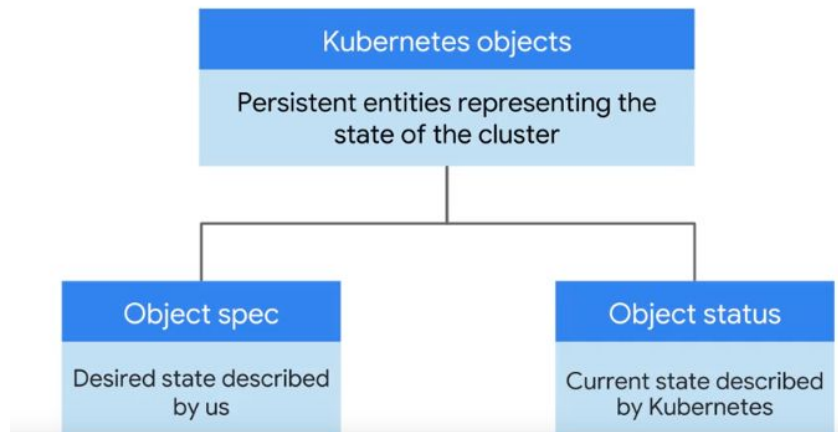
```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
spec:
  selector:
    matchLabels:
      app: nginx
  replicas: 2
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
        - name: nginx
          image: nginx:1.14.2
          ports:
            - containerPort: 80
```

All objects must have **apiVersion**, **kind**, **metadata** and **spec** fields.

- **apiVersion**: Which version of the Kubernetes API you're using to create this object
- **kind**: What kind of object you want to create
- **metadata**: Data that helps uniquely identify the object, including a **name** string, **labels**, and optional **namespace**
- **spec**: What state you desire for the object

Object Model

- Once the Deployment object is created, the Kubernetes system attaches the **status** field to the object.
- **status** is managed by Kubernetes and describes the **actual state** of the object and its history.



19

Object Model

Pod to ReplicaSet

```
apiVersion: v1
kind: Pod
metadata:
  name: nginx-pod
  labels:
    app: nginx
spec:
  containers:
  - name: mynginx
    image: nginx:1.19
    ports:
    - containerPort: 80
```

```
apiVersion: apps/v1
kind: ReplicaSet
metadata:
  name: nginx-rs
  labels:
    environment: dev
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: mynginx
        image: nginx:1.19
        ports:
        - containerPort: 80
```

20



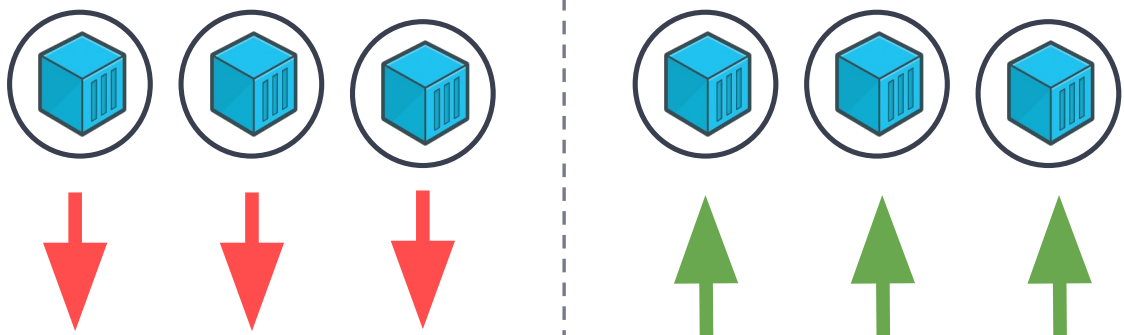
Pod Selector

```
apiVersion: apps/v1
kind: ReplicaSet
metadata:
  name: nginx-rs
  labels:
    environment: dev
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
        - name: mynginx
          image: nginx:1.19
          ports:
            - containerPort: 80
```

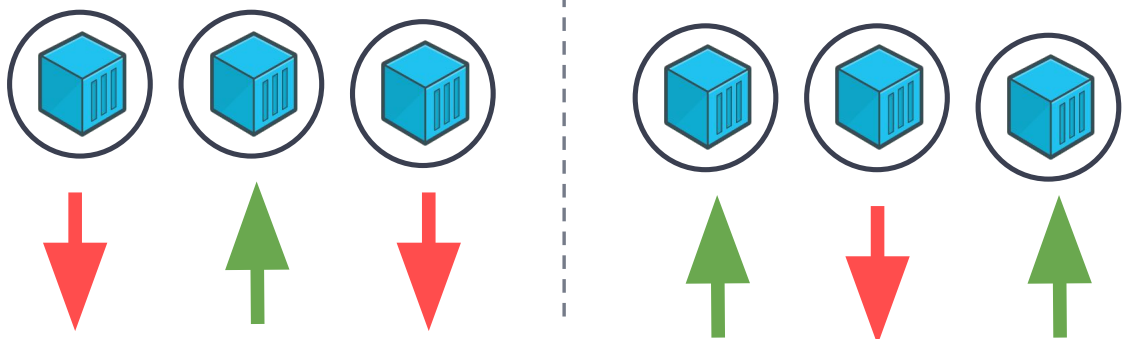
Deployment Strategy



Recreate



Rolling Update





THANKS!

Any questions?

You can find me at:

- ▶ James@clarusway.com

