#034 Kubernetes - Replicasets

Introduction

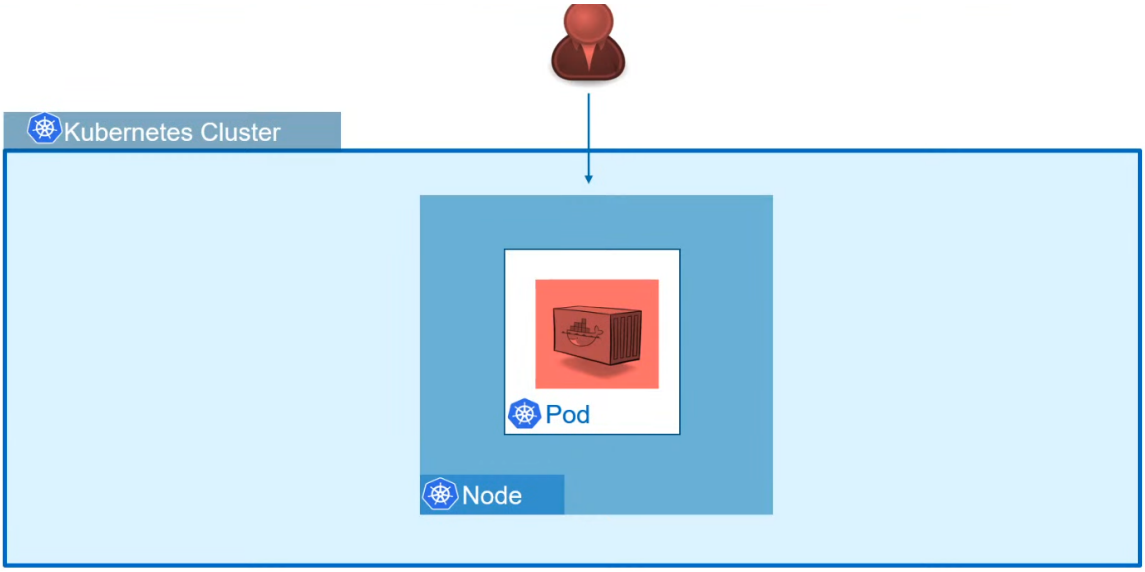
this is part 34 from the journey it's a long journey(360 day) so go please check previous parts , and if you need to walk in the journey with me please make sure to follow because I may post more than once in 1 Day but surely I will post daily at least one 😍.

And I will cover lot of tools as we move on.

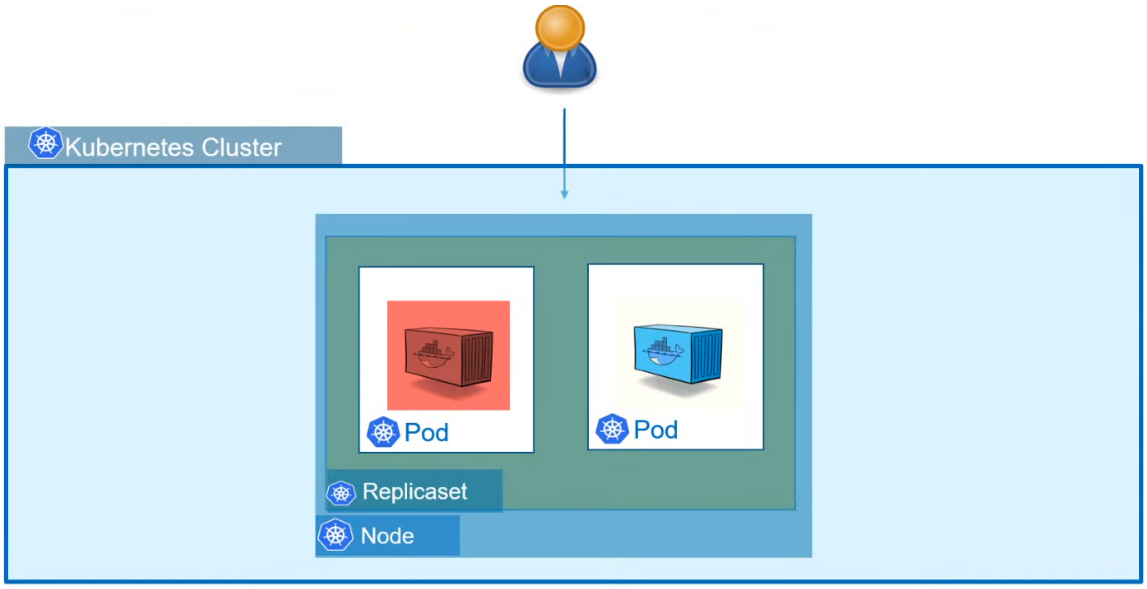
What replicasets mean?

if you work with Databases before you may be familiar with this concept . If not replicasets is the duplication of a thing , if we take example of database if a copy of database down you will have another replication of the database. And how you can synchronize between them so if one got updated all others also got updated.

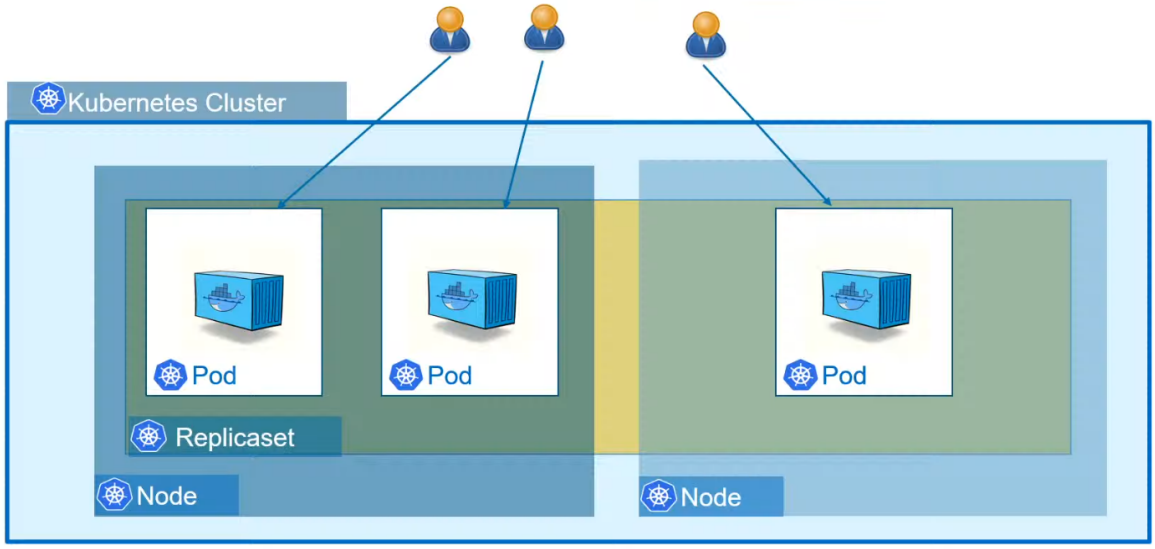
Theory and Advantages



let's assume that my application is now on the production phase and users can access it , so let's assume that a container fail for some reason , so what is the solution? the solution is do many copy of it so we can feature high availability to the users.



they are ordered as we see inside the node(not any node we will mention it) we have replicaset and inside we have the pod then container.



We insure that we have load balancing , also as we see we have replicaset (in yellow) cover 2 nodes and 3 pods.

yml of replicaset

```yaml
apiVersion: apps/v1
kind: ReplicaSet
metadata:
  name: frontend
  labels:
    app: guestbook
    tier: frontend
spec:
  # modify replicas according to your case
  replicas: 3
  selector:
    matchLabels:
      tier: frontend
  template:
    metadata:
      labels:
        tier: frontend
    spec:
      containers:
      - name: php-redis
        image: gcr.io/google_samples/gb-frontend:v3
```

this how a replicaset look like , don't worry the next part we will explain it piece by piece. What I need from it now selector . We can run all the pods that have type of frontend only and do replica for them.
so in our last example we use restapi

```
1 app_033.yml
1   apiVersion: v1
 1
 2 kind : Pod
 3
 4 metadata:
 5   name: first-api-dec
 6   labels:
 7     app: myapp
 8     type: restapi
 9
10 spec:
11
12   containers:
13     - name: simple-api
14       image: emondek/simple-api:latest
```

we use the type as restapi so we can do replicas for this type for example.
Don't worry next lecture will make it all clear :)