How can you ensure there are at least 3 Pod instances (example) are always available and running at point in time?

Replication Controller

Concept

Objectives

Concept

a. Replication Controller Overview

b. Advantages

c. Manifest file

Review Demo

a. Deploy application with Replication Controller

b. Display and validate

c. Scale up and Scale down the application

d. Clean up

Replication Controller

- Ensures that a specified number of pods are running at any time

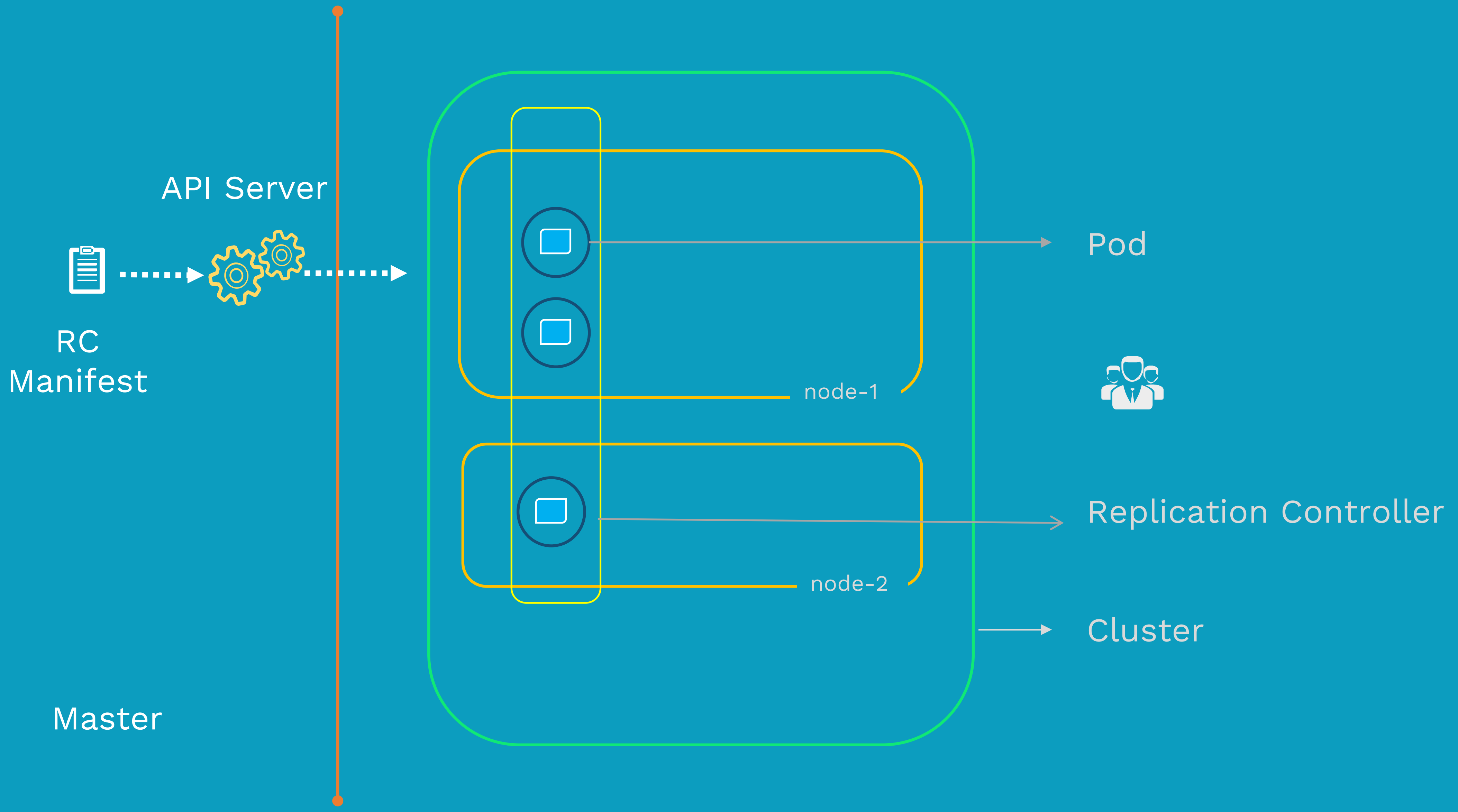
  a. If there are excess Pods, they get killed and vice versa

  b. New Pods are launched when they get fail, get deleted or terminated

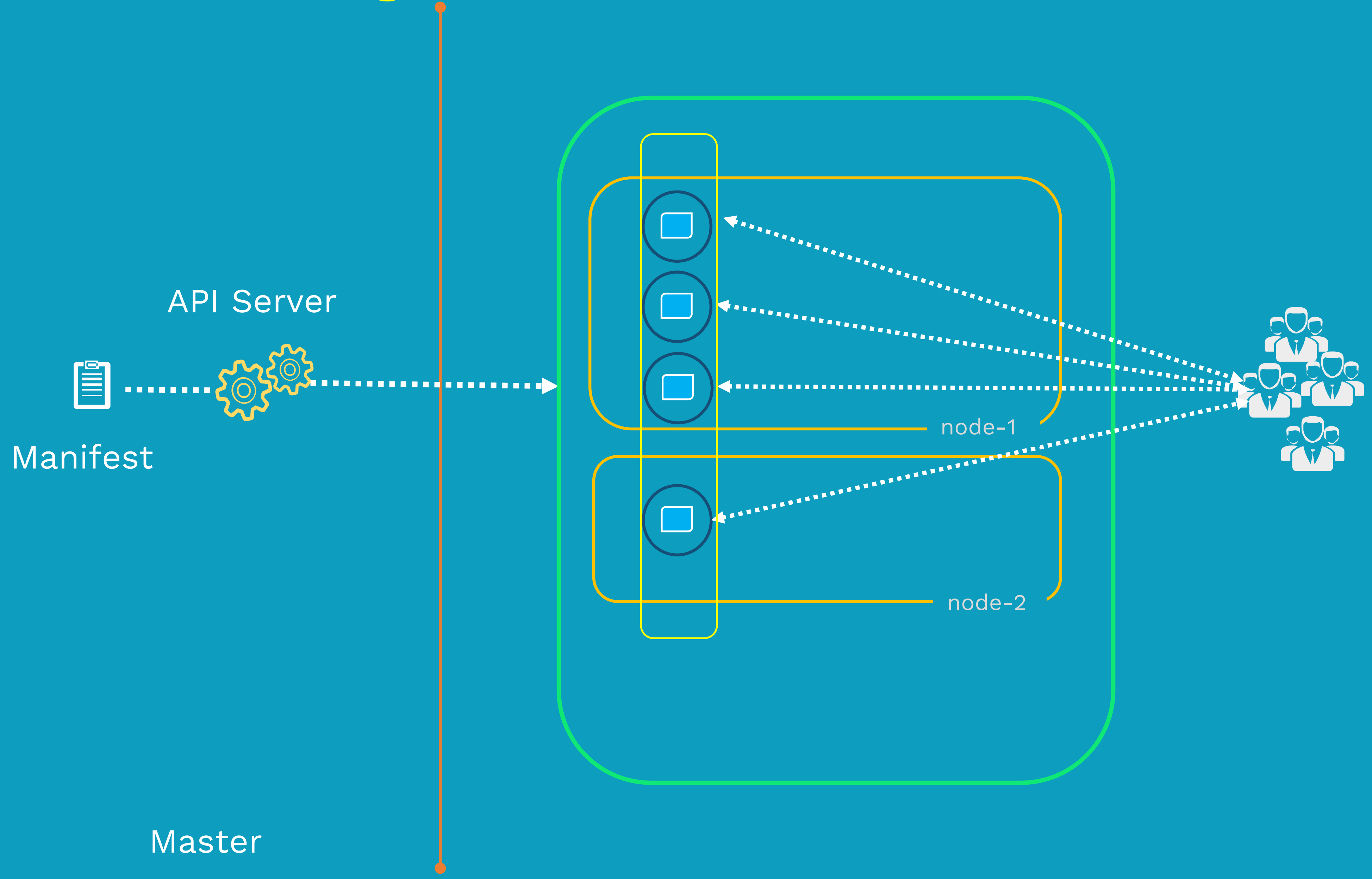
- Replication Controllers and Pods are associated with “ labels ”

- Creating a “rc” with count of 1 ensure that a pod is always available

High Availability



Load Balancing



srinathchalla@outlook.com

One last thing

Replication Controller - OLD

↓

ReplicaSet - NEW

Review

Demo

a. Manifest file

b. Deploy app using RC

c. Display and validate RC

d. Test – Node fails

e. Test – Scale up

f. Test – Scale down

Replication Controller – Config

```yaml
# nginx-rc.yaml
apiVersion: v1
kind: ReplicationController
metadata:
  name: nginx-rc
spec:
  replicas: 3
  selector:
    app: nginx-app
  template:
    metadata:
      name: nginx-pod
      labels:
        app: nginx-app
    spec:
      containers:
      - name: nginx-container
        image: nginx
        ports:
        - containerPort: 80
```

Pod def

Replication Controller – Create & Display

```
 [srinath@master ~]$ kubectl create -f nginx-rc.yaml
replicationcontroller/nginx-rc created
```

```
[srinath@master ~]$ kubectl get pods
NAME              READY     STATUS     RESTARTS    AGE
nginx-rc-62vpv    1/1       Running    0           6m
nginx-rc-fk67w    1/1       Running    0           6m
nginx-rc-qk4ph    1/1       Running    0           6m
```

```
[srinath@master ~]$ kubectl get po -l app=nginx-app
NAME              READY     STATUS     RESTARTS    AGE
nginx-rc-62vpv    1/1       Running    0           4m
nginx-rc-fk67w    1/1       Running    0           4m
nginx-rc-qk4ph    1/1       Running    0           4m
```

Replication Controller – Describe

```
[srinath@master ~]$  kubectl describe rc nginx-rc
Name:          nginx-rc
Namespace:     default
Selector:      app=nginx-app
Labels:        app=nginx-app
Annotations:   <none>
Replicas:      3 current / 3 desired
Pods Status:   3 Running / 0 Waiting / 0 Succeeded / 0 Failed
Pod Template:
  Labels:  app=nginx-app
  Containers:
   nginx-container:
    Image:        nginx
    Port:         80/TCP
    Host Port:    0/TCP
    Environment:  <none>
    Mounts:       <none>
  Volumes:        <none>
Events:
  Type    Reason             Age   From                    Message
  ----    ------             ----  ----                    -------
  Normal  SuccessfulCreate   25m   replication-controller  Created pod: nginx-rc-62vpv
  Normal  SuccessfulCreate   25m   replication-controller  Created pod: nginx-rc-fk67w
  Normal  SuccessfulCreate   25m   replication-controller  Created pod: nginx-rc-qk4ph
```

Replication Controller – Node Fail

```
[srinath@master ~]$ kubectl get po -o wide
NAME              READY     STATUS     RESTARTS    AGE        IP              NODE      NOMINATED NODE
nginx-rc-5lb6b    1/1       Running    0           16s        10.240.1.32     node1     <none>
nginx-rc-l64js    1/1       Running    0           16s        10.240.1.33     node1     <none>
nginx-rc-qrqv7    1/1       Running    0           16s        10.240.2.28     node2     <none>
```

```
[srinath@master ~]$ kubectl get nodes
NAME      STATUS      ROLES     AGE       VERSION
master    Ready       master    1d        v1.11.2
node1     Ready       <none>    1d        v1.11.2
node2     NotReady    <none>    1d        v1.11.2
```

```
[srinath@master ~]$ kubectl get po -o wide
NAME              READY     STATUS     RESTARTS    AGE        IP              NODE     NOMINATED NODE
nginx-rc-5lb6b    1/1       Running    0           11m        10.240.1.32     node1    <none>
nginx-rc-l64js    1/1       Running    0           11m        10.240.1.33     node1    <none>
nginx-rc-mtwzs    1/1       Running    0           2m         10.240.1.34     node1    <none>
nginx-rc-qrqv7    1/1       Unknown    0           11m        10.240.2.28     node2    <none>
```

Replication Controller – Scaling up

```
[srinath@master ~]$ kubectl scale rc nginx-rc --replicas=5
replicationcontroller/nginx-rc scaled
```

```
[srinath@master ~]$ kubectl get rc nginx-rc
NAME        DESIRED    CURRENT    READY      AGE
nginx-rc    5          5          5          9m
```

```
[srinath@master ~]$ kubectl get po -o wide
NAME             READY     STATUS     RESTARTS    AGE       IP             NODE   NOMINATED NODE
nginx-rc-2x2kg   1/1       Running    0           36m       10.240.1.173   node1  <none>
nginx-rc-7kvhl   1/1       Running    0           36m       10.240.2.3     node2  <none>
nginx-rc-g7mwq   1/1       Running    0           33m       10.240.2.42    node2  <none>
nginx-rc-jgt28   1/1       Running    0           33m       10.240.1.3     node1  <none>
nginx-rc-wvmrx   1/1       Running    0           36m       10.240.2.4     node2  <none>
```

Replication Controller – Scaling down

```
[srinath@master ~]$ kubectl scale rc nginx-rc --replicas=3
replicationcontroller/nginx-rc scaled
```

```
[srinath@master ~]$ kubectl get rc nginx-rc
NAME        DESIRED    CURRENT    READY     AGE
nginx-rc    3          3          3         45m
```

```
[srinath@master ~]$ kubectl get po -o wide
NAME              READY     STATUS     RESTARTS    AGE       IP            NODE   NOMINATED NODE
nginx-rc-2x2kg    1/1       Running    0           36m       10.240.1.173  node1  <none>
nginx-rc-jgt28    1/1       Running    0           33m       10.240.1.3    node1  <none>
nginx-rc-wvmrx    1/1       Running    0           36m       10.240.2.4    node2  <none>
```

Replication Controller – Delete

```
[srinath@master ~]$ kubectl delete -f nginx-rc.yaml  #kubectl delete rc nginx-rc
replicationcontroller "nginx-rc" deleted
```

```
[srinath@master ~]$ kubectl get rc
No resources found.
```

```
[srinath@master ~]$ kubectl get po -l app=nginx-app
No resources found.
```

Summary

Concept

a. Ensures that a specified number of pods are running at any time

b. Advantages

Review Demo

a. Manifest file

b. Deploy application with Replication Controller

c. Display and validate

d. Scale up and Scale down the application

e. Clean up

Coming up...

Demo

Replication Controller