



TURN IT UP

CISCO *Live!*

#CiscoLive



The bridge to possible

An Introduction to Kubernetes For DevNet Certifications

Robert Whitaker
BRKCRT-2007

CISCO *Live!*

#CiscoLive



A decorative graphic on the left side of the slide, consisting of several overlapping, teardrop-shaped elements in various colors (blue, green, yellow, red, dark blue) radiating from a central point, resembling a stylized flower or a burst of energy.

Agenda

- Understanding Orchestration
- Kubernetes High-level Architecture
- Kubernetes Pods
- Kubernetes ReplicaSets
- Kubernetes Deployments
- Where To Go and What To Do Next

Understanding Orchestration



“Docker is great to create and manage containers on a single guest OS.”

docker ps

docker images

docker commit

docker pull ubuntu:16.04

docker run

docker attach

docker start

“Google starts over 2 billion containers per week!”

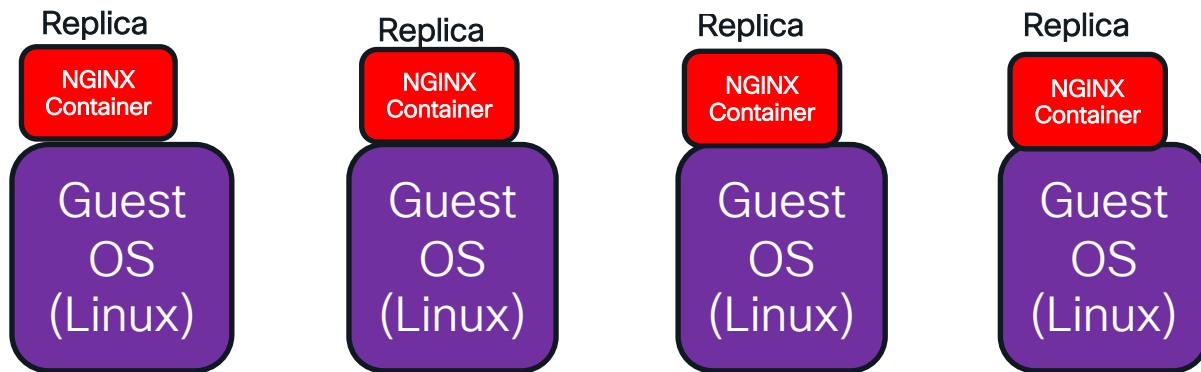
*“How many people would
Google have to hire to start 2
BILLION containers?”*

Orchestrator Can Deploy Containers

Orchestrator



kubernetes



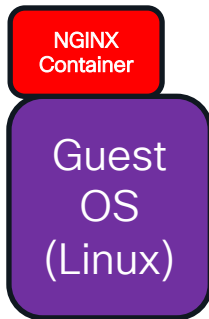
Orchestrator Can Detect Faulty Container

Orchestrator

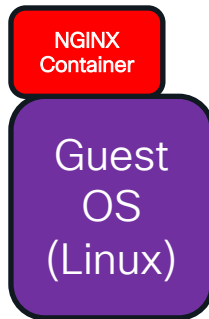


kubernetes

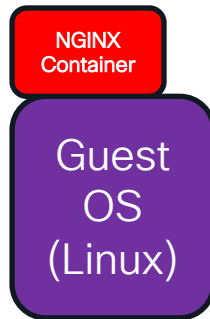
Replica



Replica



Replica



Replica



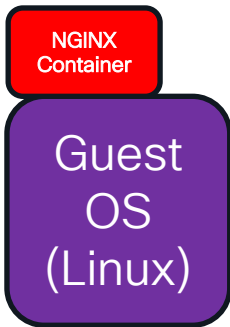
Orchestrator Can Scale Up

Orchestrator

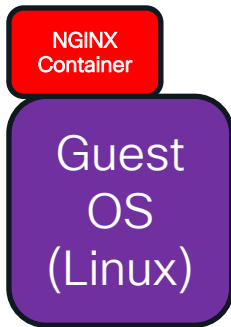


kubernetes

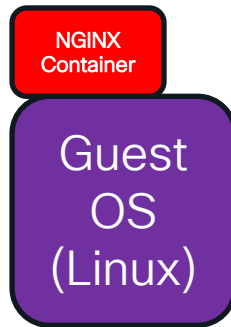
Replica



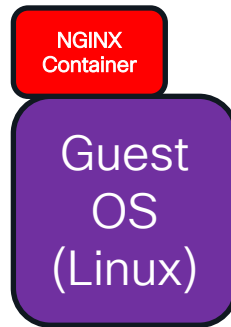
Replica



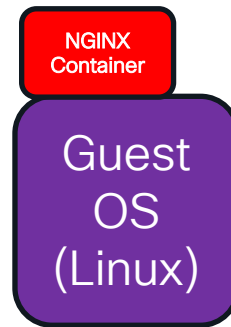
Replica



Replica



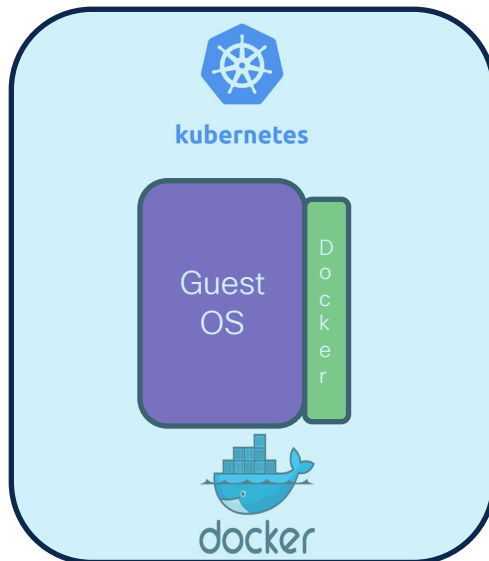
Replica



Kubernetes High-level Architecture

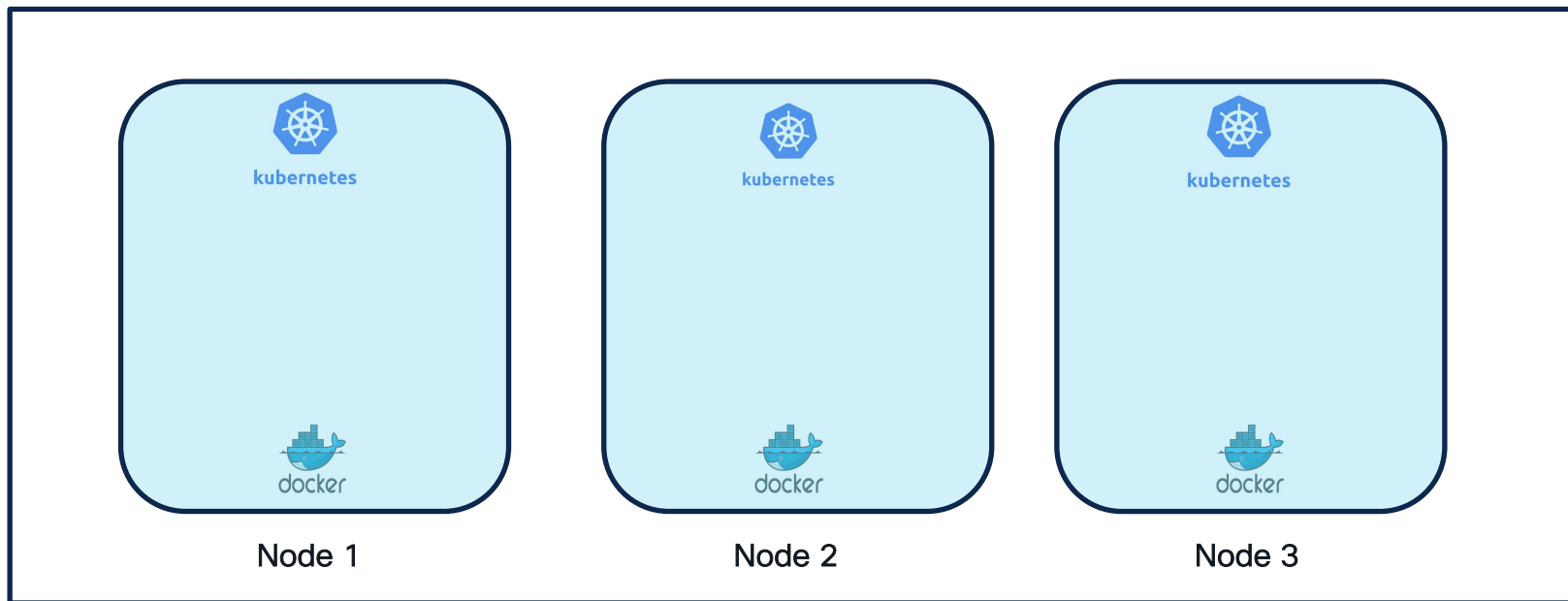


Kubernetes Worker Node



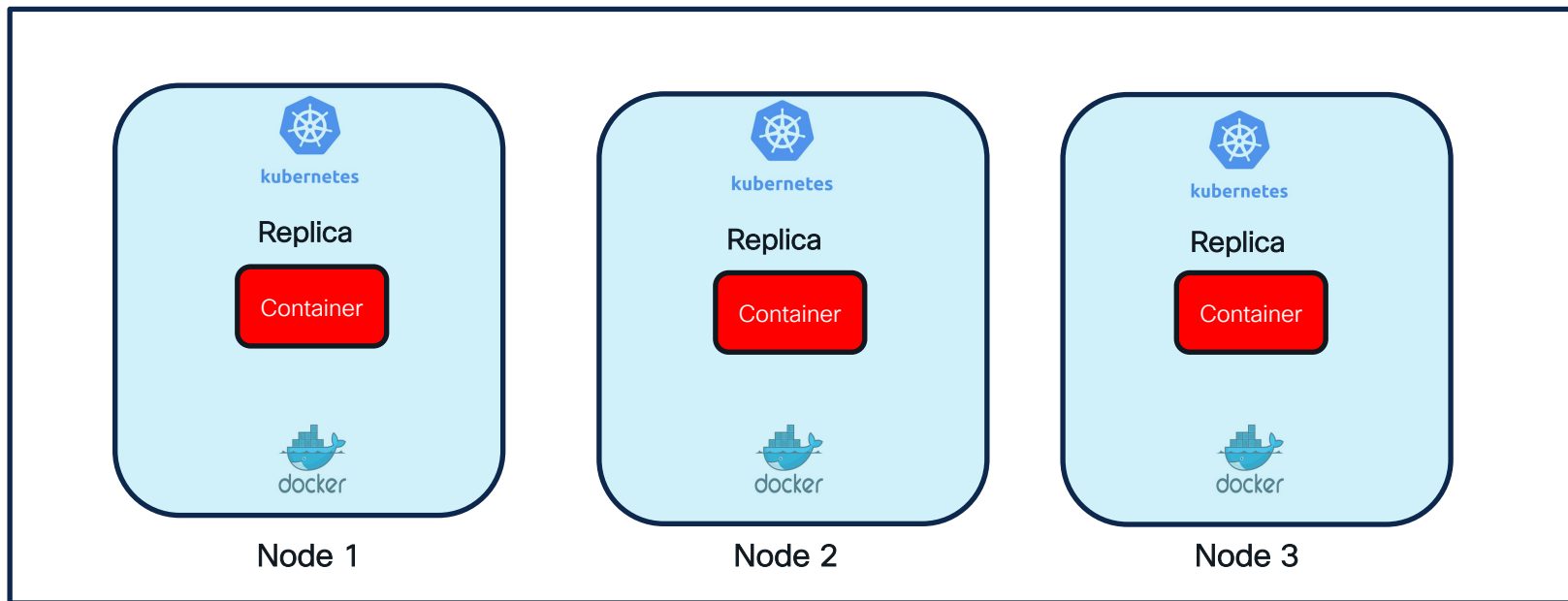
Nodes

Kubernetes Worker Nodes Are Clustered



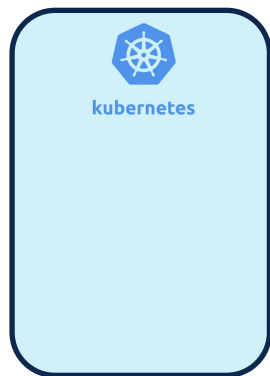
Cluster

Containers Will Run On Worker Nodes

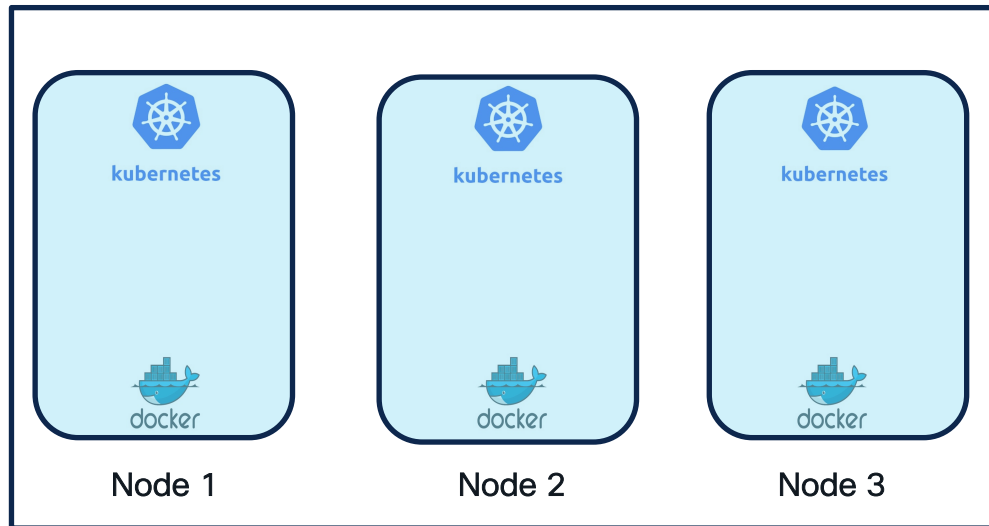


Cluster

Kubernetes Master

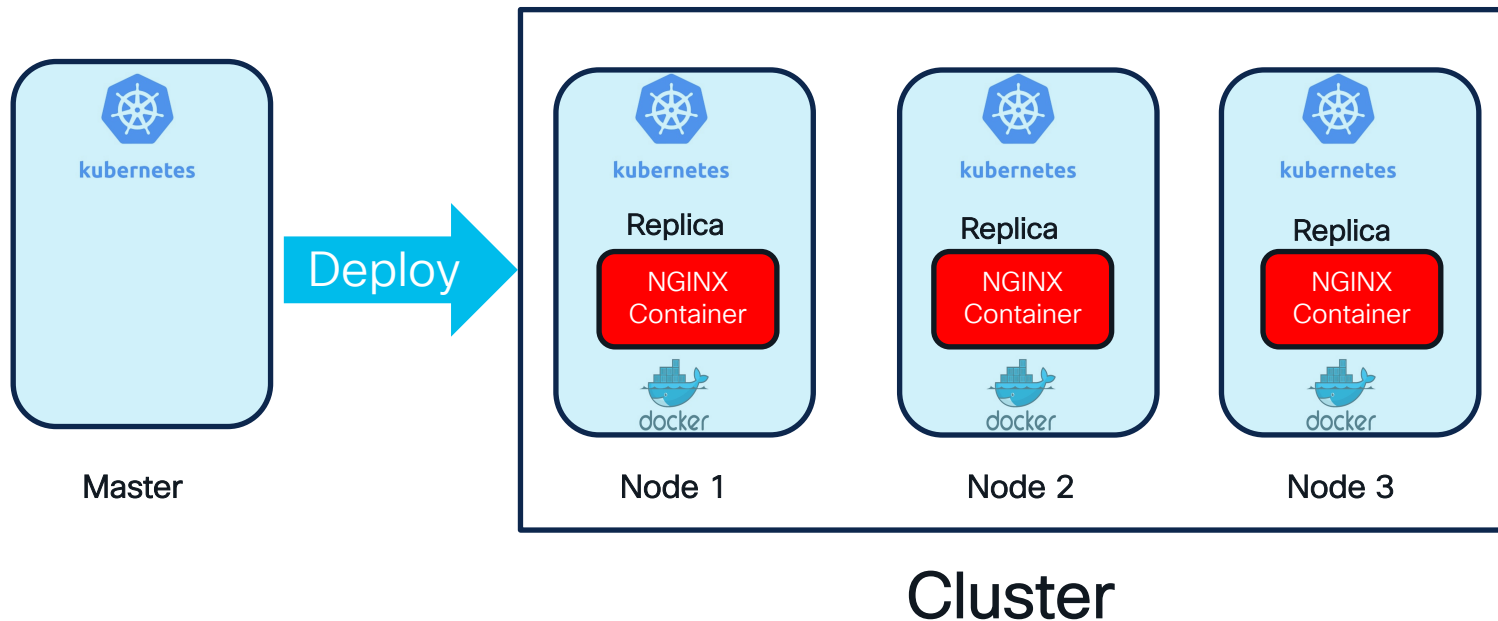


Master

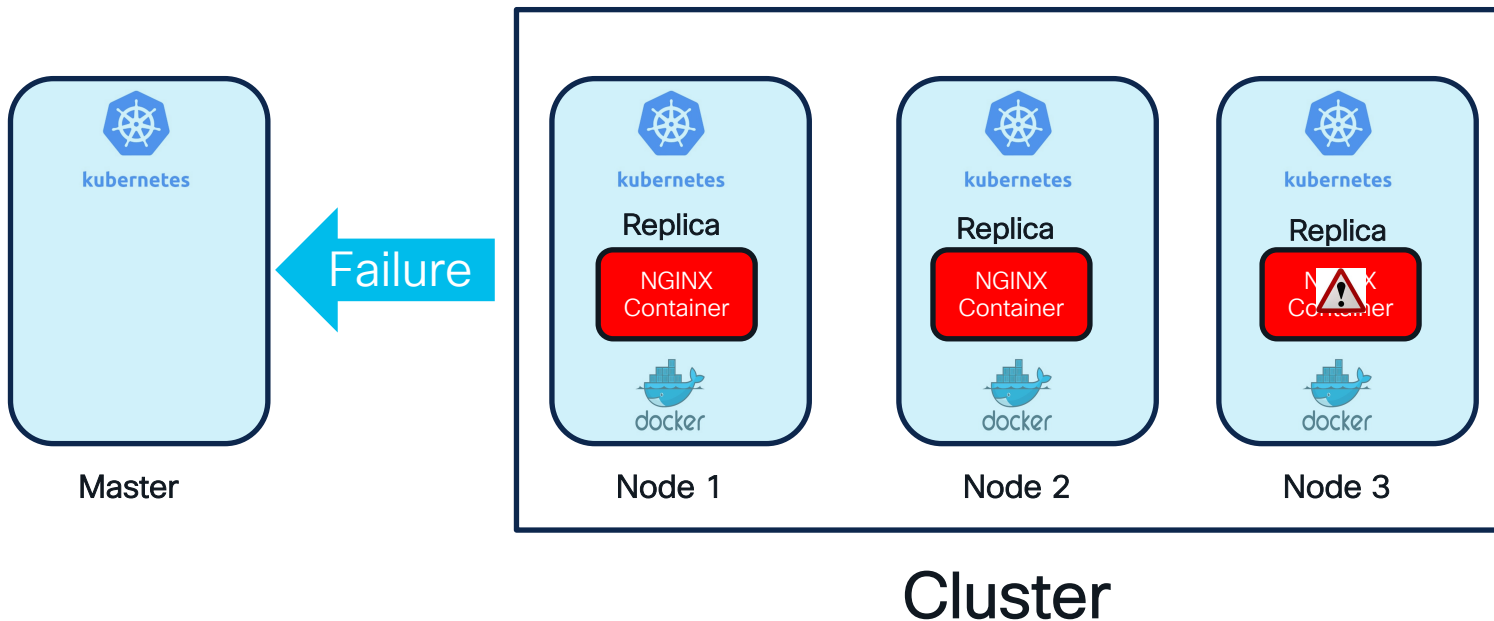


Cluster

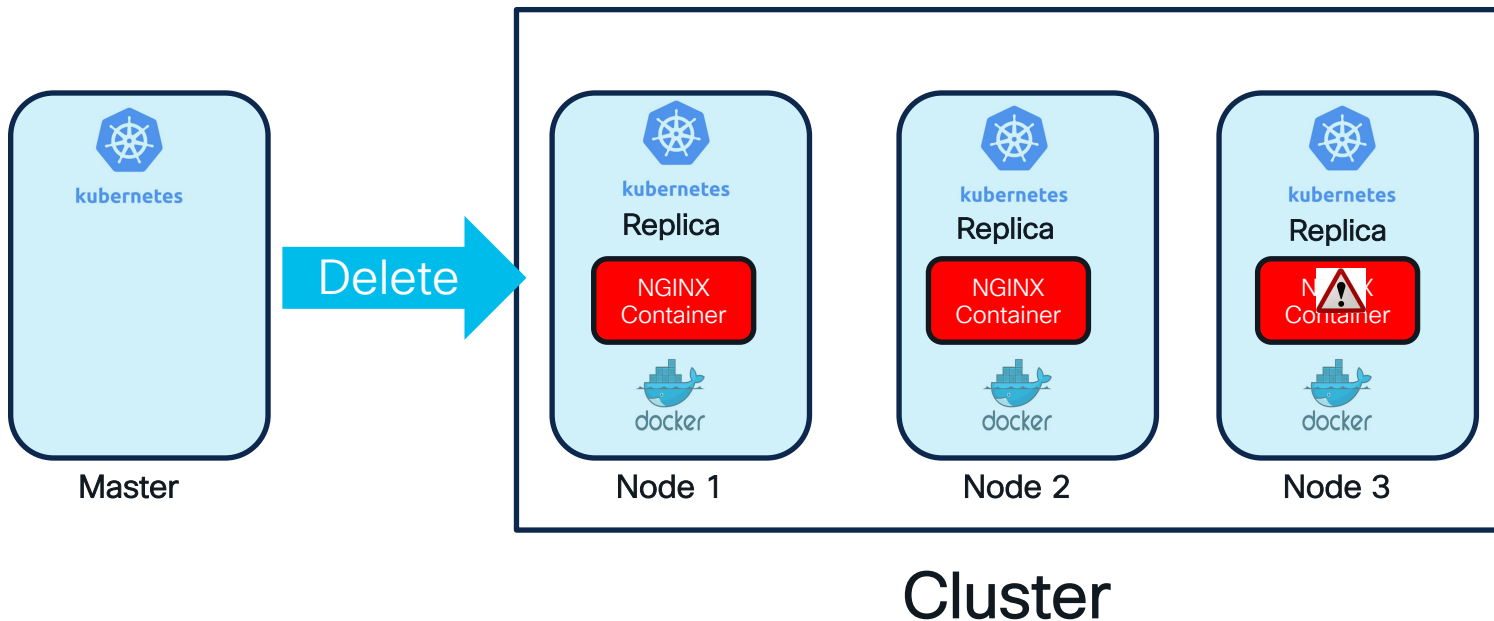
Kubernetes Master Can Deploy Containers



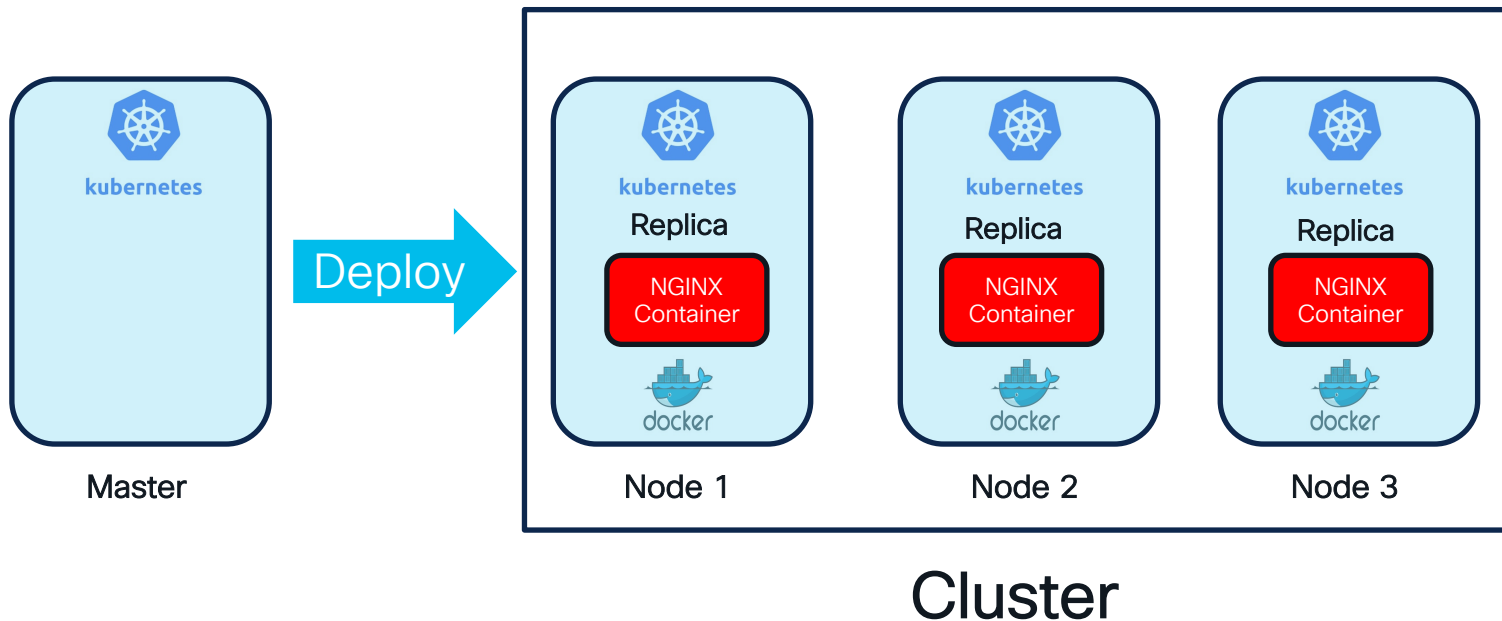
Kubernetes Master Can Detect A Faulty Container



Kubernetes Master Can Detect A Faulty Container

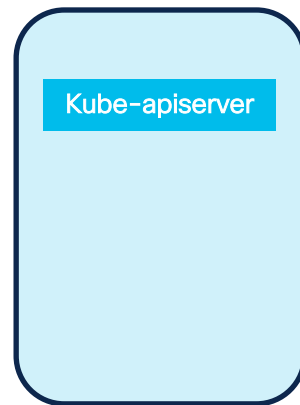


Kubernetes Master Can Detect A Faulty Container



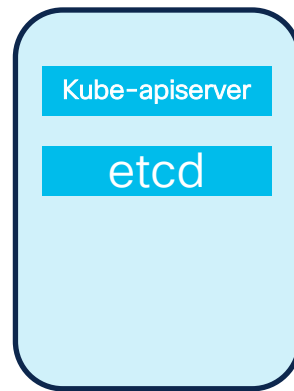
Software on Kubernetes Master – API

- API – Processes REST operations
- Allows users to interact with Kubernetes software
- Kubernetes commands are sent to the API
- Nodes and Master communicate via API



Software on Kubernetes Master – etcd

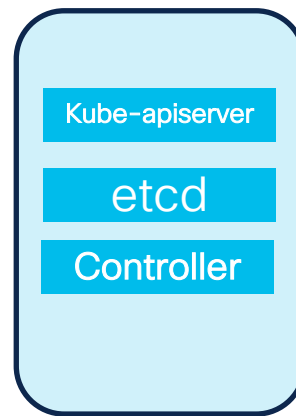
- **etcd** – File where master and node cluster information is stored
- All configuration information is stored in etcd (Pods, ReplicaSets, Deployments)
- Data is stored in key value pairs
- Kubectl get commands query etcd
- File should be backed up



Master

Software on Kubernetes Master – Controller Manager

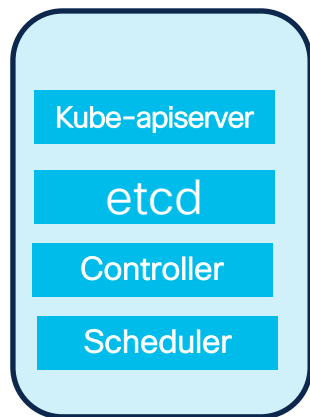
- **controller** – Monitors and manages cluster state
- Controller queries worker node every 5 seconds to determine health
- 40 seconds worker node will be defines as unreachable



Master

Software on Kubernetes Master – Scheduler

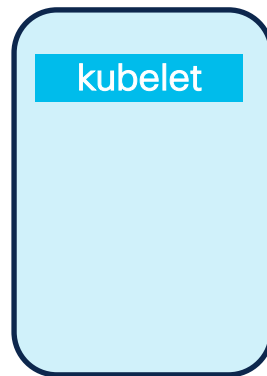
- **scheduler** – assigns containers to nodes



Master

Software on Kubernetes Nodes – kubelet

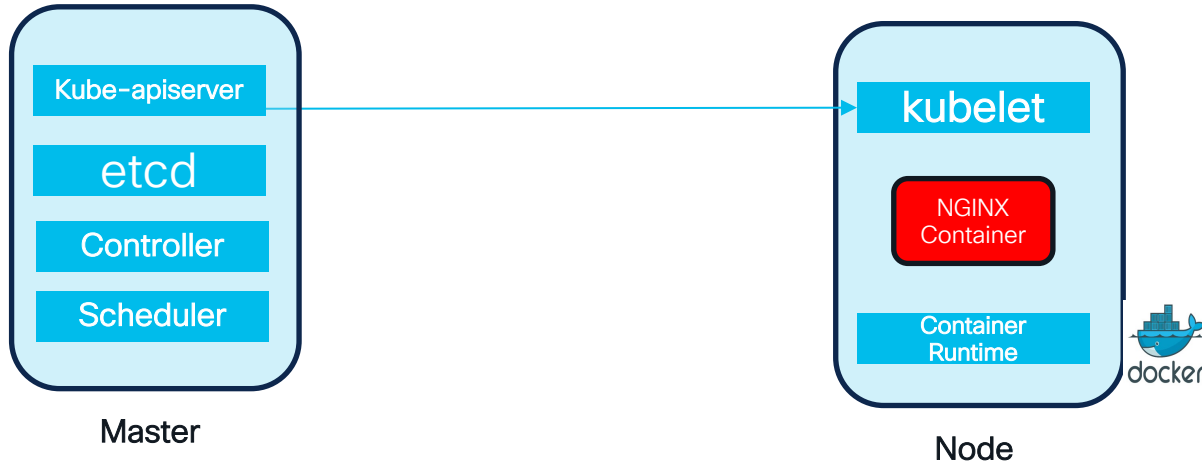
- **Kubelet** – An agent that runs on each node in the cluster.
- It makes sure that containers are running



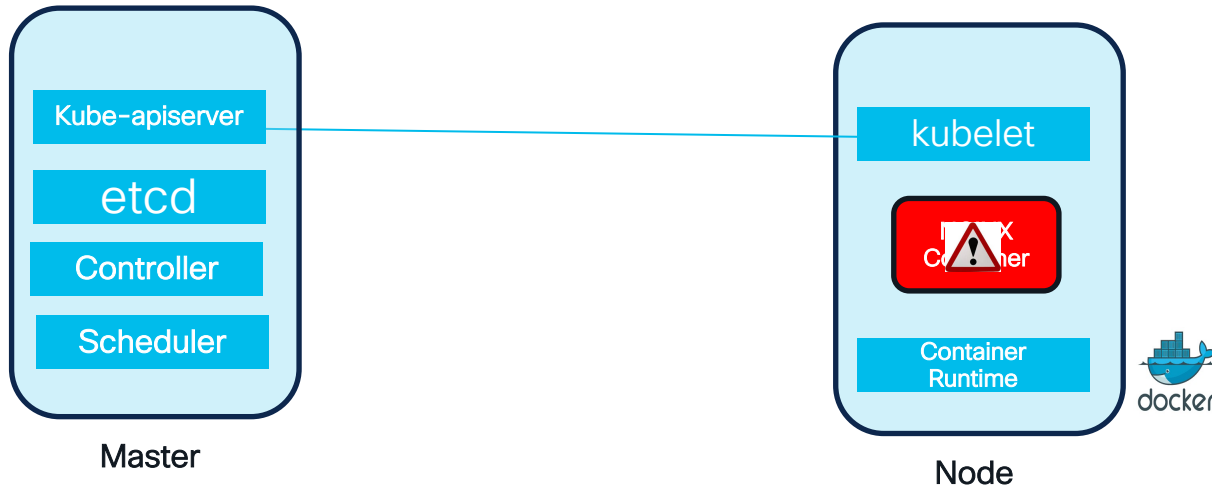
Node

Kubernetes Masters and Worker Nodes

– Create Container



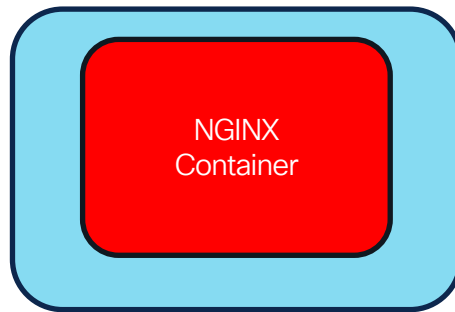
Kubernetes Masters and Nodes Detect Faulty Container



Kubernetes Pods

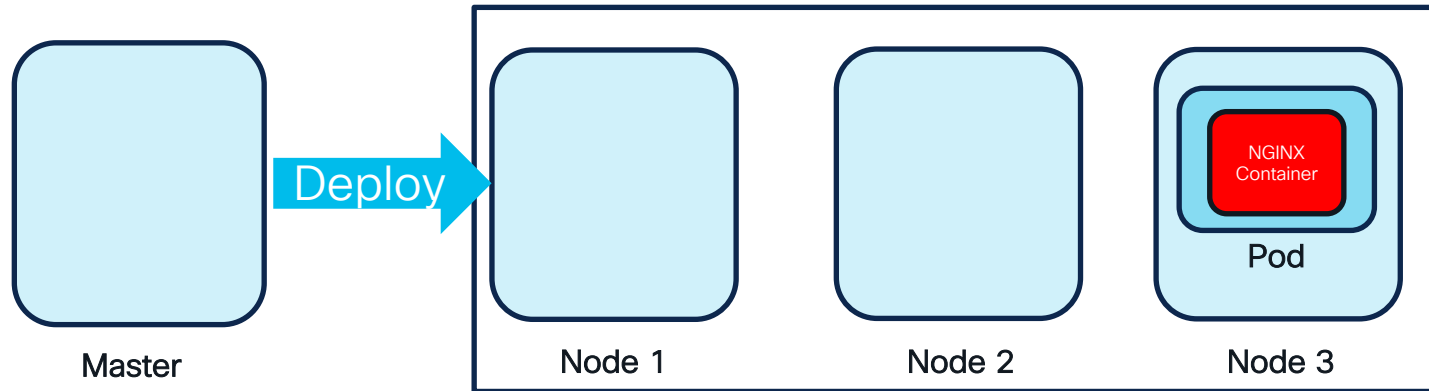


Pods



Pod

Pods

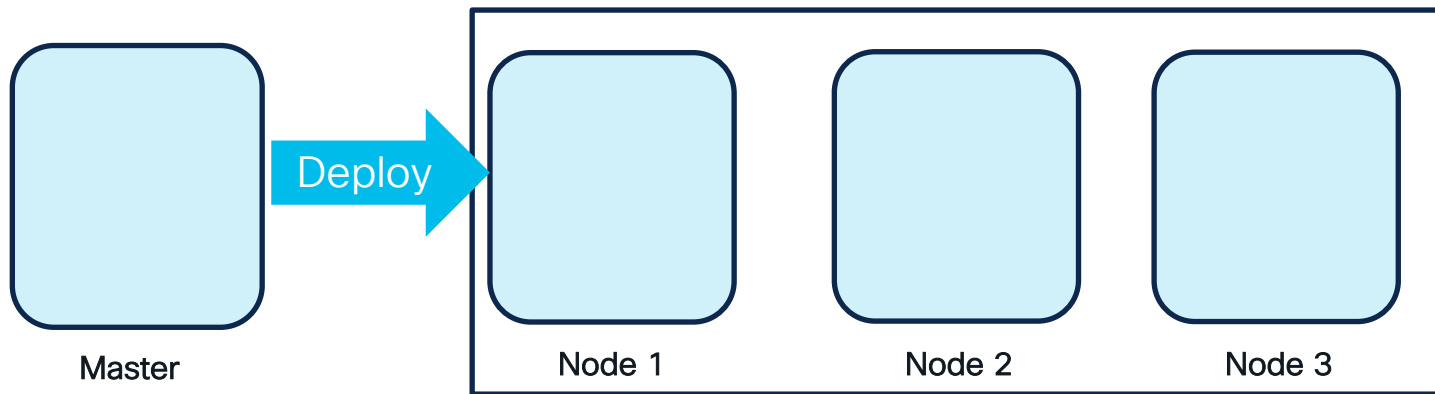


Command To Deploy A Pod

```
root@ubuntu-server:~# kubectl run my-nginx-app --image nginx
```

```
root@ubuntu-server:~# kubectl get pods
```

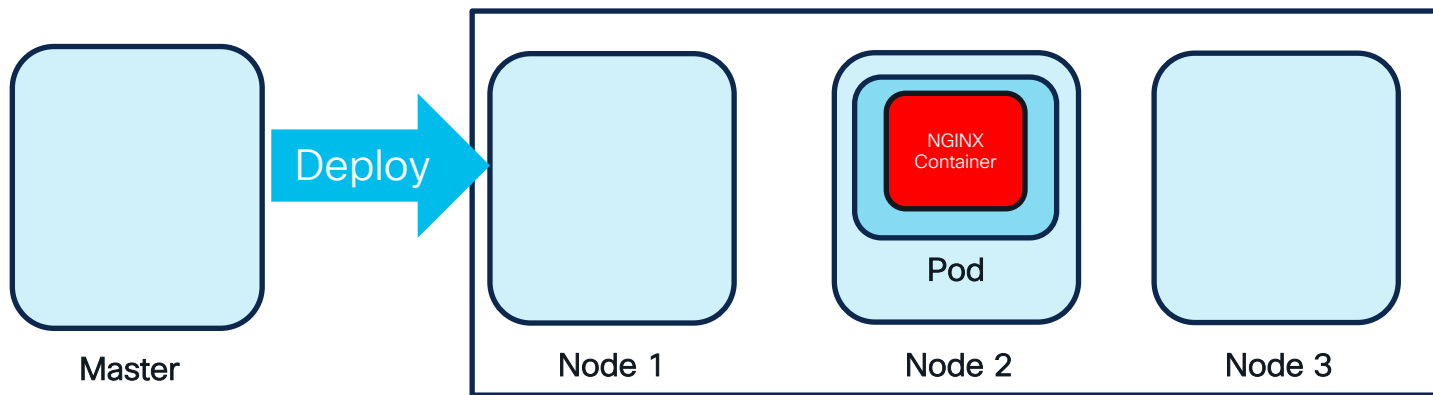
NAME	READY	STATUS	RESTARTS	AGE
my-nginx-app-6df95bb585-5bj5s	1/1	Running	0	3m59s



Command To Deploy A Pod

```
root@node-0:~# kubectl get pods -o wide
```

NAME	READY	STATUS	RESTARTS	AGE	IP	NODE	NOMINATED NODE	READINESS GATES
my-nginx-app-6df95bb585-w9mrn	1/1	Running	0	3m59s	10.244.2.2	node-2	<none>	<none>



“YAML files are used in Kubernetes to deploy several types of configurations.”

Kubernetes YAML File Required Objects

apiVersion:

kind:

metadata:

spec:

apiVersion

apiVersion: v1

Kind

apiVersion: v1
kind: Pod

Metadata

```
apiVersion: v1
kind: Pod
metadata:
  name: my-pod
  labels:
    type: front-end
```

Spec

```
apiVersion: v1
kind: Pod
metadata:
  name: my-pod
spec:
  containers:
    - name: nginx-container
      image: nginx
```

Understanding The Configuration

apiVersion: v1

kind: Pod

metadata:

name: my-pod

spec:

containers:

- name: nginx-container

image: nginx



my-pod

Understanding The Configuration

apiVersion: v1

kind: Pod

metadata:

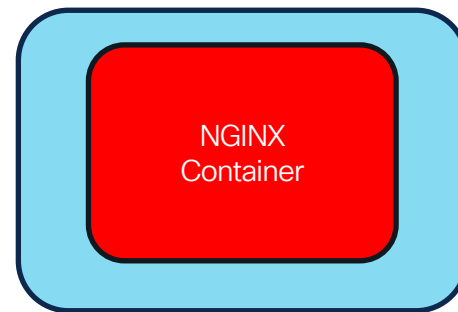
name: my-pod

spec:

containers:

- name: nginx-container

image: nginx



my-pod

Review: Configuring a Pod

apiVersion: v1

kind: Pod

metadata:

name: my-pod

labels:

type: front-end

spec:

containers:

- name:** nginx-container

image: nginx



Write



Pod.yaml

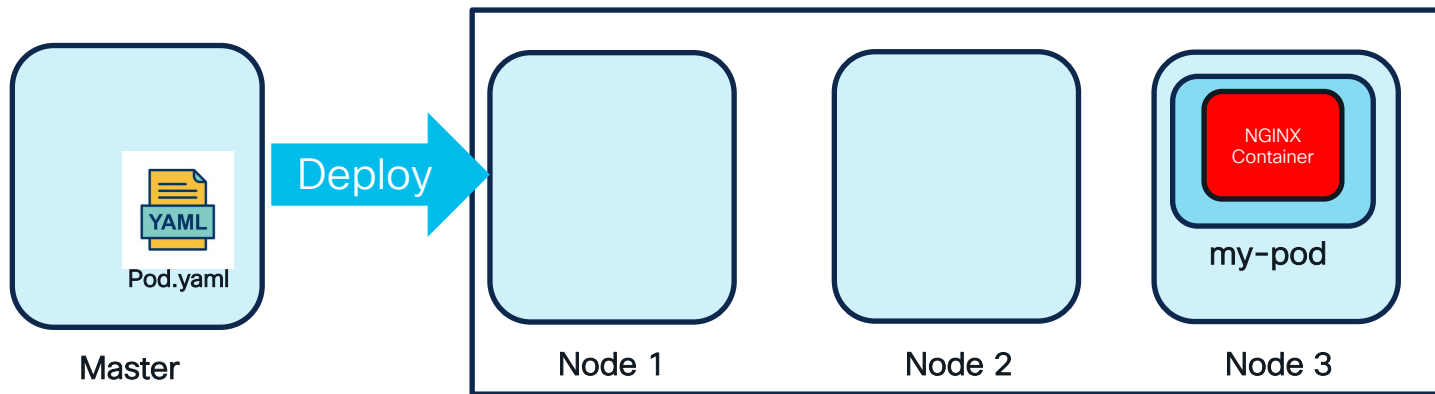
Yet Another Markup Language (YAML) Example

```
root@ubuntu-server:~# kubectl create -f pod.yaml
```

```
pod/my-pod created
```

```
root@ubuntu-server:~# kubectl get pods -o wide
```

NAME	READY	STATUS	RESTARTS	AGE	IP	NODE
my-pod	1/1	Running	0	23s	10.244.2.3	node-3



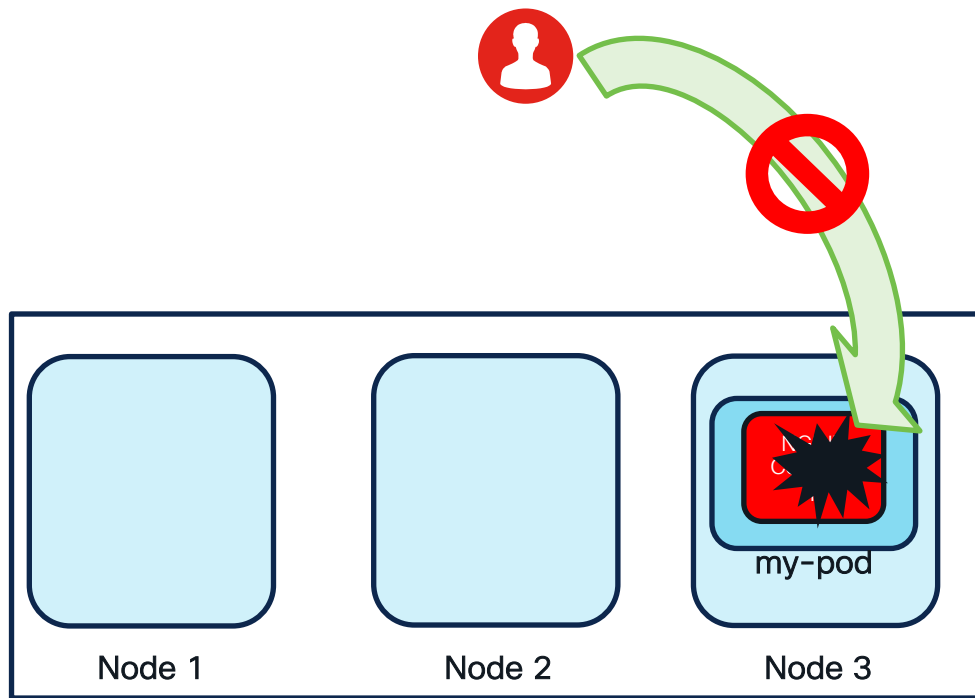
What Are Some Useful Fields?

```
root@ubuntu-server:~# kubectl describe pod my-pod
Name:          my-pod
Namespace:     default
Priority:       0
Node:          node-3/10.80.59.117
Start Time:    Sat, 10 Aug 2019 10:36:30 +0000
Labels:        type=front-end
Annotations:   <none>
Status:        Running
IP:            192.168.2.4
Containers:
  nginx-container:
    Container ID:  docker://9deeb55aa7011b73942d577eebc64a51adfa2f44a693e82d059b140f80828bf9
    Image:         nginx
    Image ID:      docker-pullable://nginx@sha256:eb3320e2f9ca409b7c0aa71aea3cf7ce7d018f03a372564dbdb023646958770b
    Port:          <none>
    Host Port:     <none>
    State:         Running
      Started:     Sat, 10 Aug 2019 10:36:37 +0000
    Ready:         True
    Restart Count: 0
    Environment:   <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from default-token-f2588 (ro)
Conditions:
  Type            Status
  Initialized     True
  Ready           True
  ContainersReady True
  PodScheduled    True
Volumes:
  default-token-f2588:
    Type:          Secret (a volume populated by a Secret)
    SecretName:     default-token-f2588
    Optional:       false
QoS Class:        BestEffort
Node-Selectors:   <none>
Tolerations:      node.kubernetes.io/not-ready:NoExecute for 300s
                  node.kubernetes.io/unreachable:NoExecute for 300s
Events:
  Type     Reason      Age   From              Message
  ----     -
  Normal   Scheduled   116s  default-scheduler Successfully assigned default/my-pod to node-3
  Normal   Pulling     113s  kubelet, node-3   Pulling image "nginx"
  Normal   Pulled      107s  kubelet, node-3   Successfully pulled image "nginx"
  Normal   Created     107s  kubelet, node-3   Created container nginx-container
  Normal   Started     107s  kubelet, node-3   Started container nginx-container
```

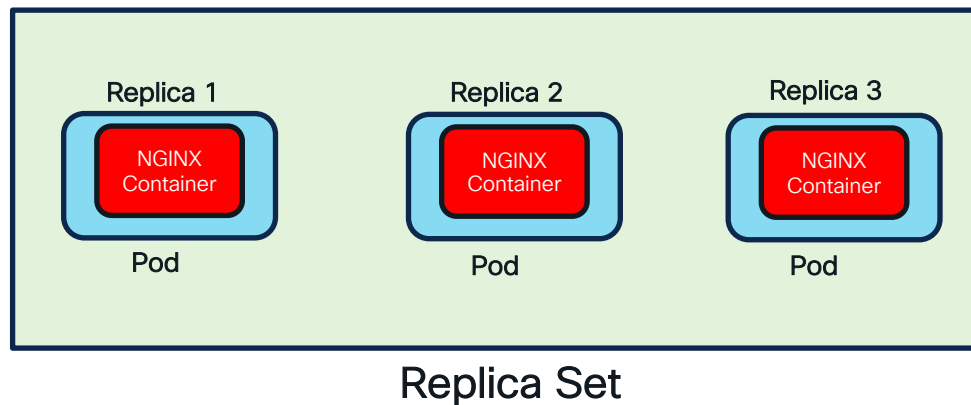
Kubernetes ReplicaSets



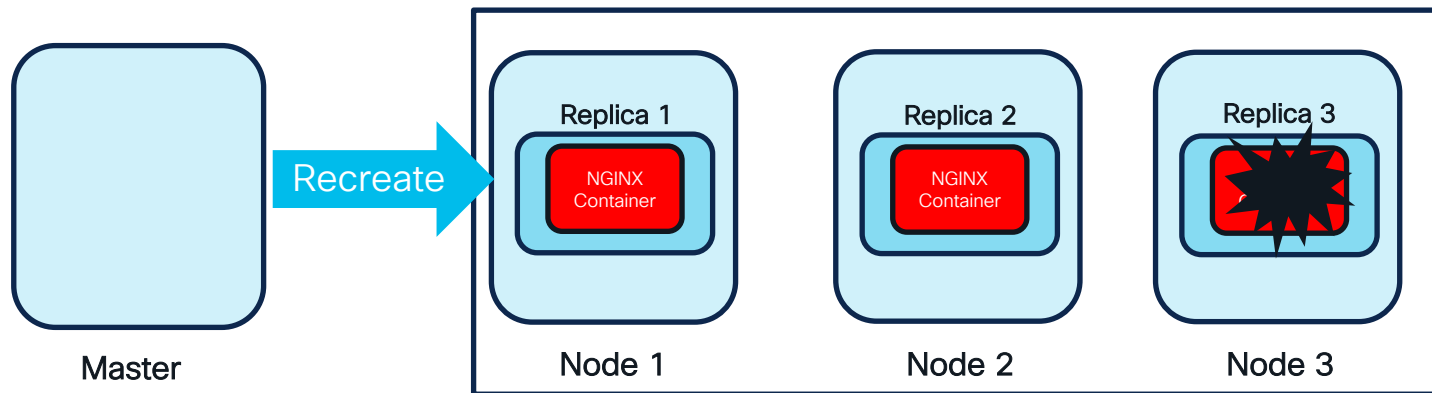
Single Pod = Single Point Of Failure



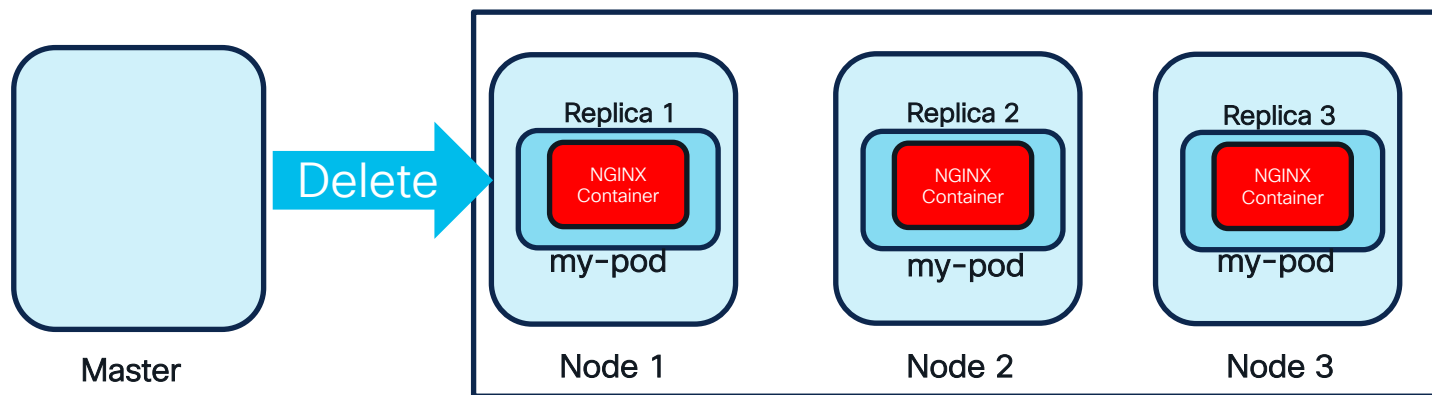
Replica Set – Contain Pods



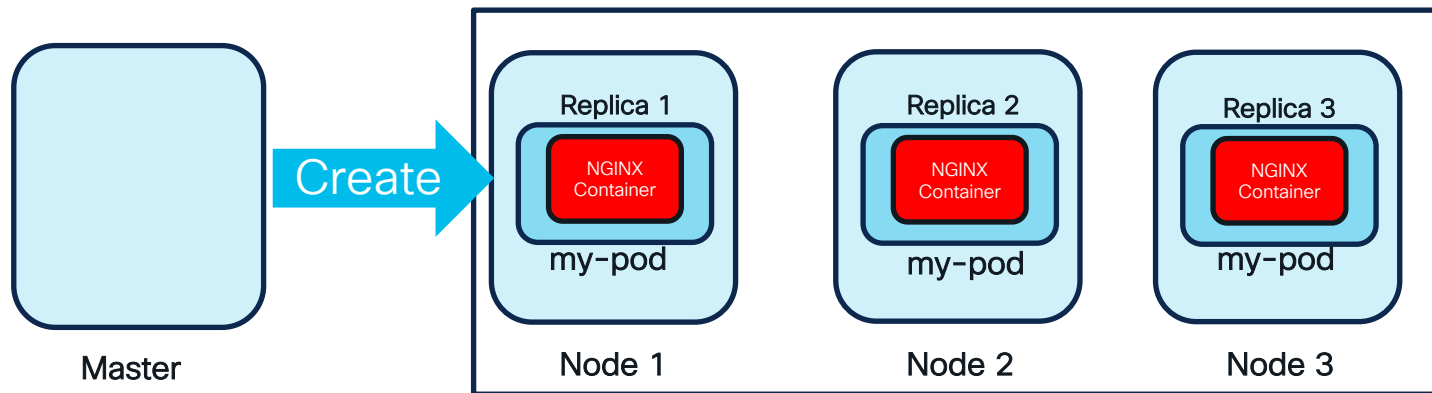
Replica Set – High Availability



Replica Set – Scale Down



Replica Set – Scale Up



Kubernetes ReplicaSet YAML File Required Objects

apiVersion:

kind:

metadata:

spec:

Understanding The Configuration

apiVersion: apps/v1

kind: ReplicaSet

metadata:

name: my-replicaset

spec:

template:

metadata:

labels:

type: front-end

spec:

containers:

- name: nginx-container

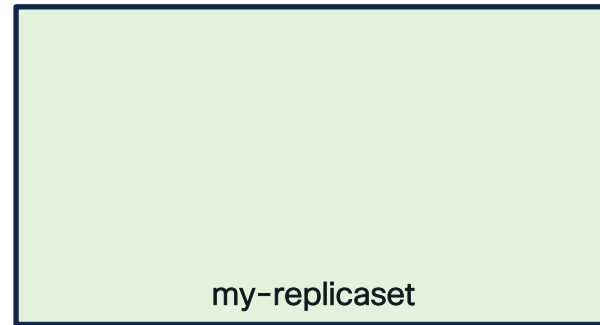
image: nginx

replicas: 3

selector:

matchLabels:

type: front-end



Understanding The Configuration

apiVersion: apps/v1

kind: ReplicaSet

metadata:

name: my-replicaset

spec:

template:

metadata:

labels:

type: front-end

spec:

containers:

- **name:** nginx-container

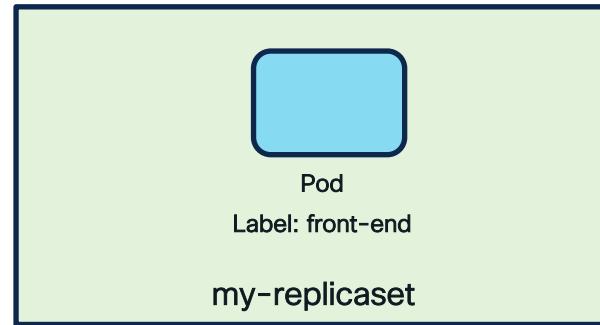
image: nginx

replicas: 3

selector:

matchLabels:

type: front-end



Understanding The Configuration

apiVersion: apps/v1

kind: ReplicaSet

metadata:

name: my-replicaset

spec:

template:

metadata:

labels:

type: front-end

spec:

containers:

- **name:** nginx-container

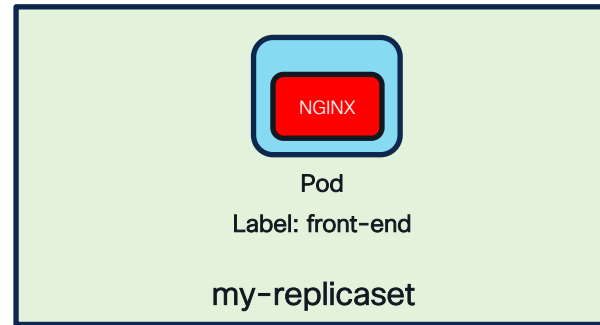
image: nginx

replicas: 3

selector:

matchLabels:

type: front-end



Understanding The Configuration

apiVersion: apps/v1

kind: ReplicaSet

metadata:

name: my-replicaset

spec:

template:

metadata:

labels:

type: front-end

spec:

containers:

- name: nginx-container

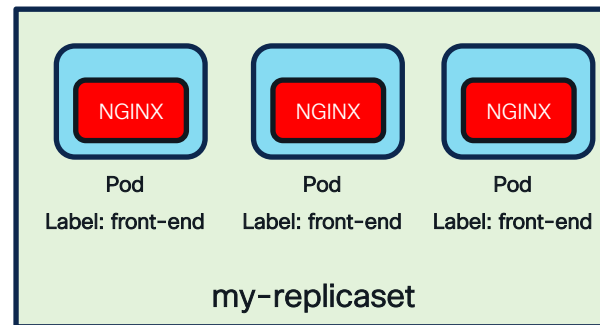
image: nginx

replicas: 3

selector:

matchLabels:

type: front-end



Configuring a ReplicaSet

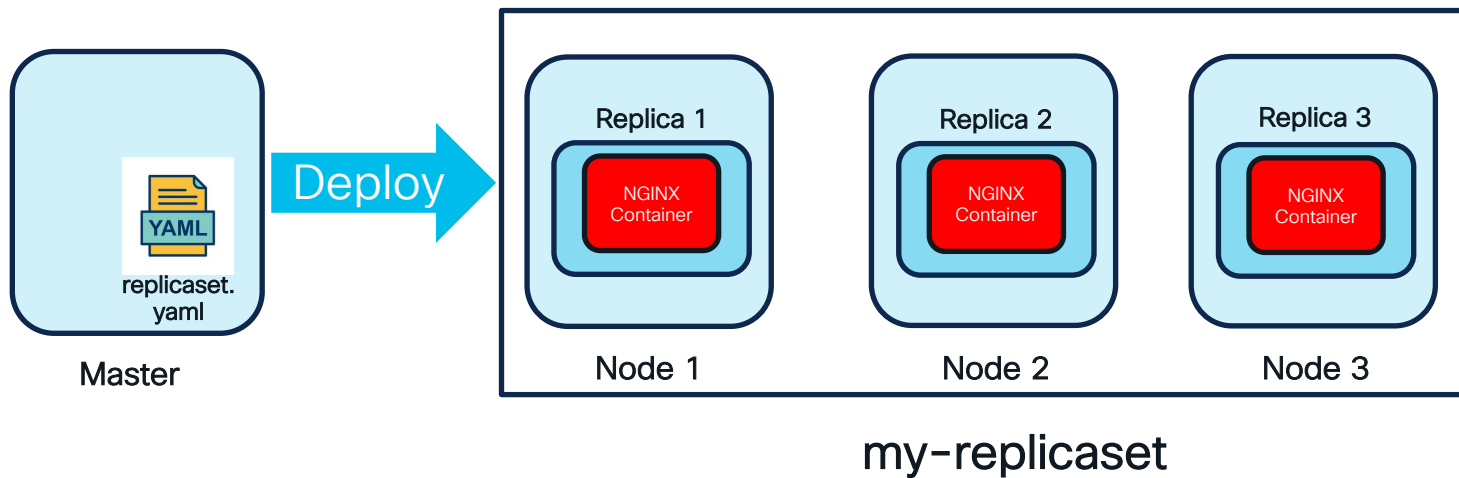
```
apiVersion: apps/v1
kind: ReplicaSet
metadata:
  name: my-replicaset
spec:
  template:
    metadata:
      labels:
        type: front-end
    spec:
      containers:
        - name: nginx-container
          image: nginx
  replicas: 3
  selector:
    matchLabels:
      type: front-end
```



replicaset.yaml

Configuring a ReplicaSet

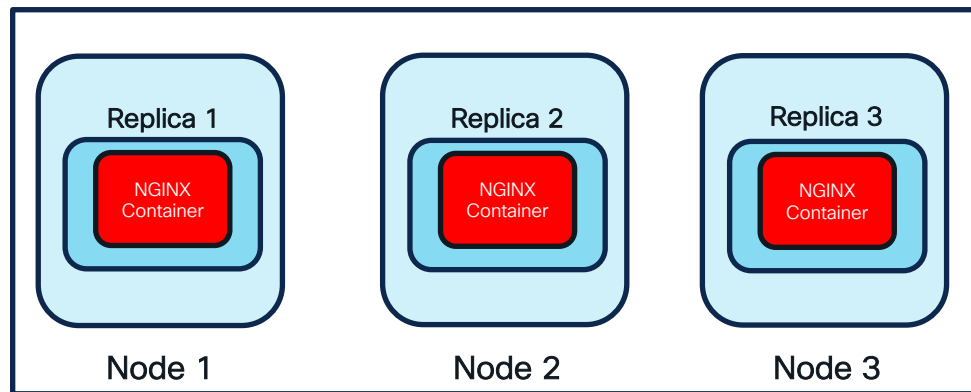
```
root@ubuntu-server:~# kubectl create -f replicaset.yaml  
replicaset.apps/my-replicaset created
```



Confirming a ReplicaSet

```
root@ubuntu-server:~# kubectl get rs
```

NAME	DESIRED	CURRENT	READY	AGE
my-replicaset	3	3	3	3m

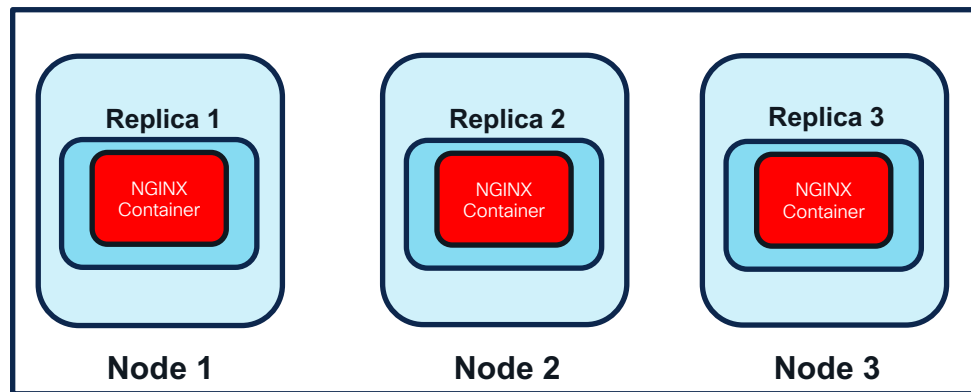


my-replicaset

Confirming Pods Within A ReplicaSet

```
root@ubuntu-server:~# kubectl get pods -o wide
```

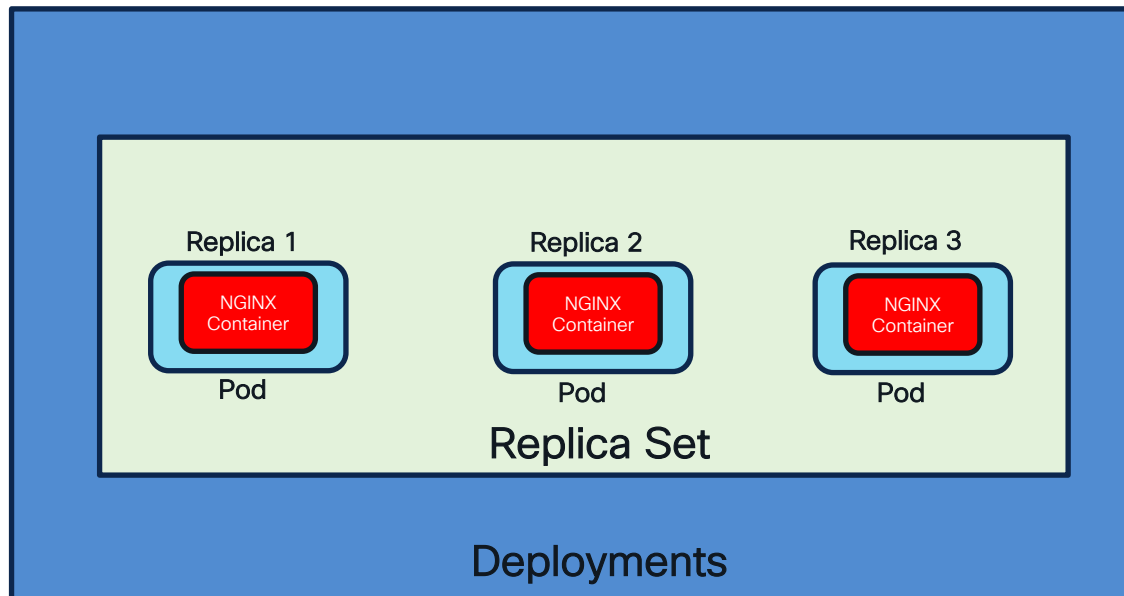
NAME	READY	STATUS	RESTARTS	AGE	IP	NODE	NOMINATED	NODE	READINESS
GATES									
my-replicaset-fgd4h	1/1	Running	0	8m28s	100.244.3.10	node-1	<none>		<none>
my-replicaset-tn2jf	1/1	Running	0	8m28s	100.244.5.11	node-2	<none>		<none>
my-replicaset-xd5zk	1/1	Running	0	8m28s	100.244.2.13	node-3	<none>		<none>



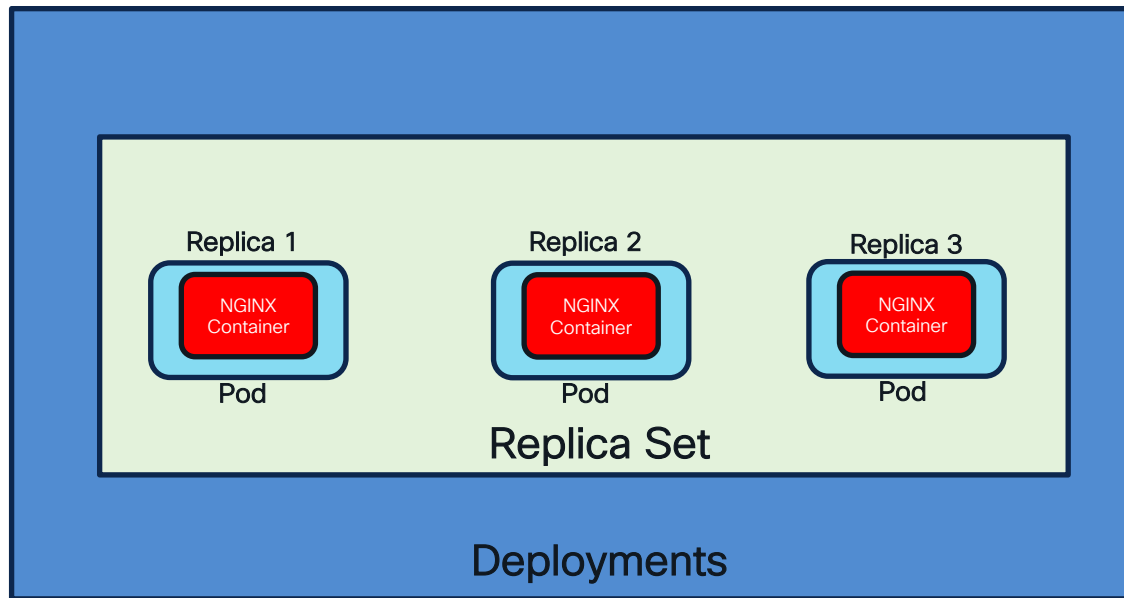
Kubernetes Deployments



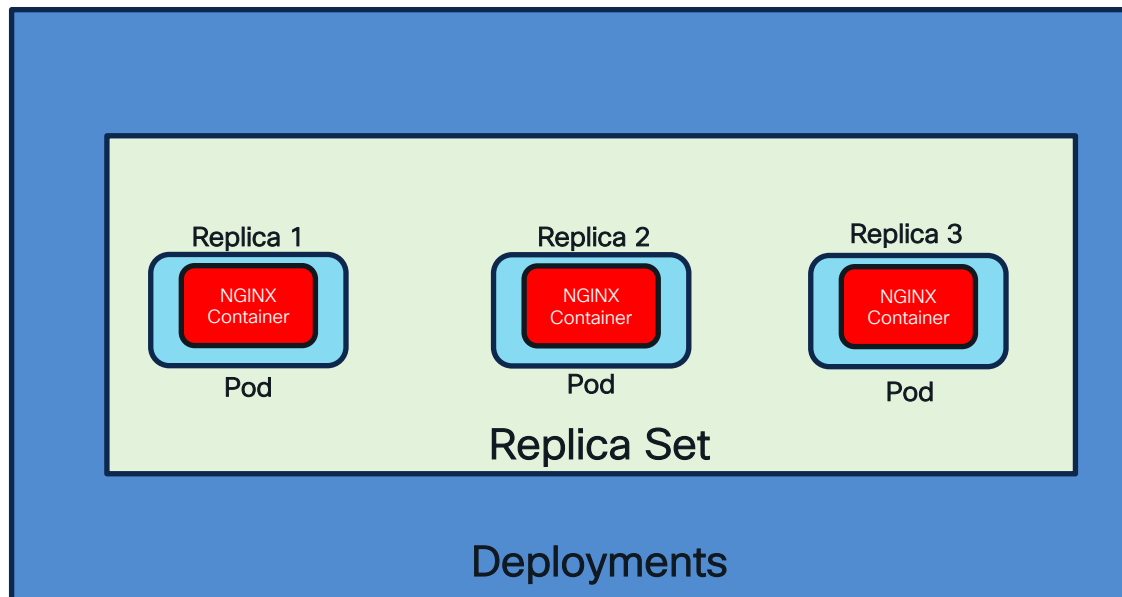
Deployments Contain Replica Sets



Deployment Benefits = Rolling Updates



Deployment Benefits = Rollback



Understanding The Configuration

apiVersion: apps/v1

kind: Deployment

metadata:

name: my-deployment

spec:

template:

metadata:

labels:

type: front-end

spec:

containers:

- name: nginx-container

image: nginx

replicas: 3

selector:

matchLabels:

type: front-end



Understanding The Configuration

apiVersion: apps/v1

kind: Deployment

metadata:

name: my-deployment

spec:

template:

metadata:

labels:

type: front-end

spec:

containers:

- name: nginx-container

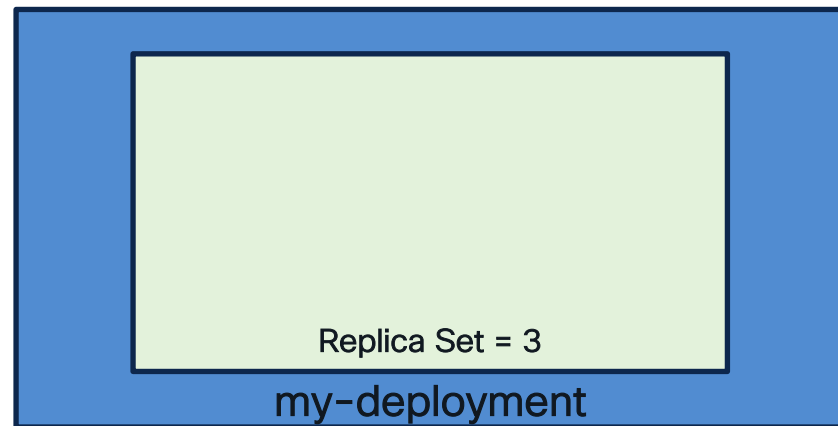
image: nginx

replicas: 3

selector:

matchLabels:

type: front-end



Understanding The Configuration

apiVersion: apps/v1

kind: Deployment

metadata:

name: my-deployment

spec:

template:

metadata:

labels:

type: front-end

spec:

containers:

- name: nginx-container

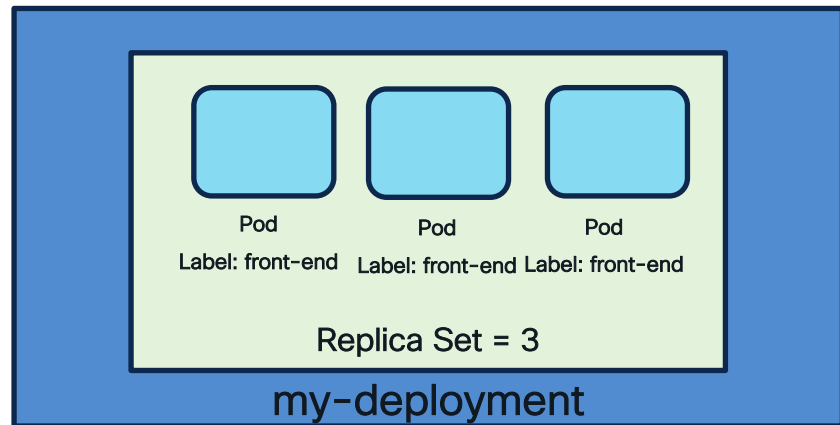
image: nginx

replicas: 3

selector:

matchLabels:

type: front-end



Understanding The Configuration

apiVersion: apps/v1

kind: Deployment

metadata:

name: my-deployment

spec:

template:

metadata:

labels:

type: front-end

spec:

containers:

- name: nginx-container

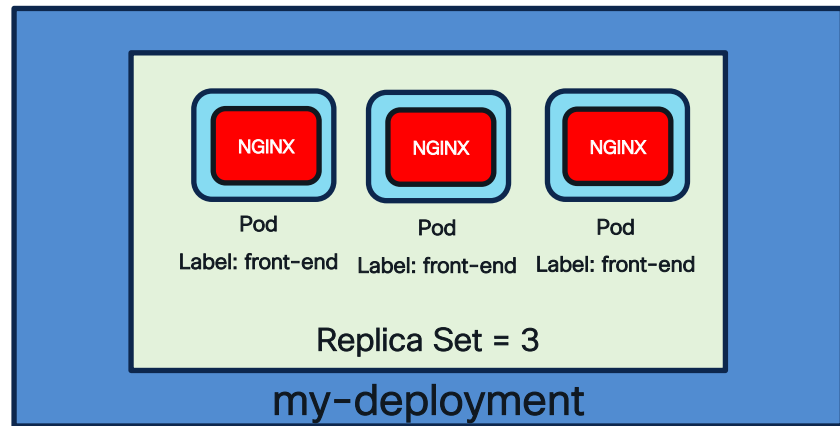
image: nginx

replicas: 3

selector:

matchLabels:

type: front-end



Configuring a Deployment

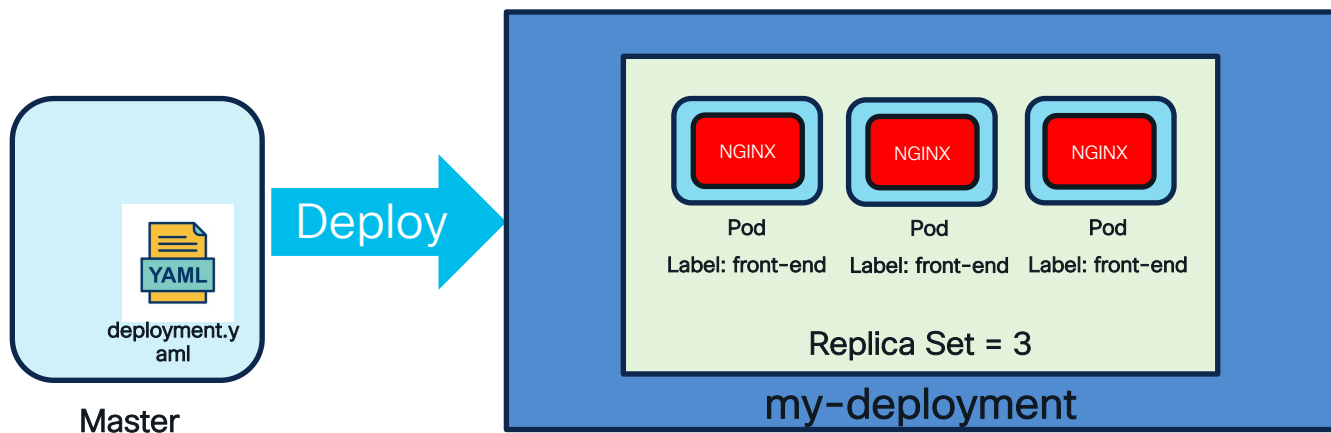
```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-deployment
spec:
  template:
    metadata:
      labels:
        type: front-end
    spec:
      containers:
        - name: nginx-container
          image: nginx
      replicas: 3
      selector:
        matchLabels:
          type: front-end
```



deployment.yaml

Configuring a Deployment

```
root@ubuntu-server:~# kubectl create -f deployment.yaml  
deployment.apps/my-deployment created
```



Confirming a Deployment

```
root@ubuntu-server:~# kubectl get deployments
```

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
------	-------	------------	-----------	-----

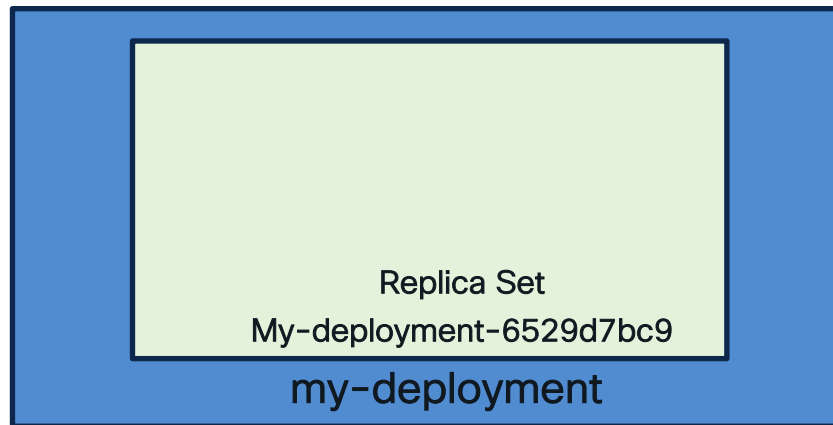
my-deployment	3/3	3	3	119s
---------------	-----	---	---	------



Confirming a ReplicaSet

```
root@ubuntu-server:~# kubectl get rs
```

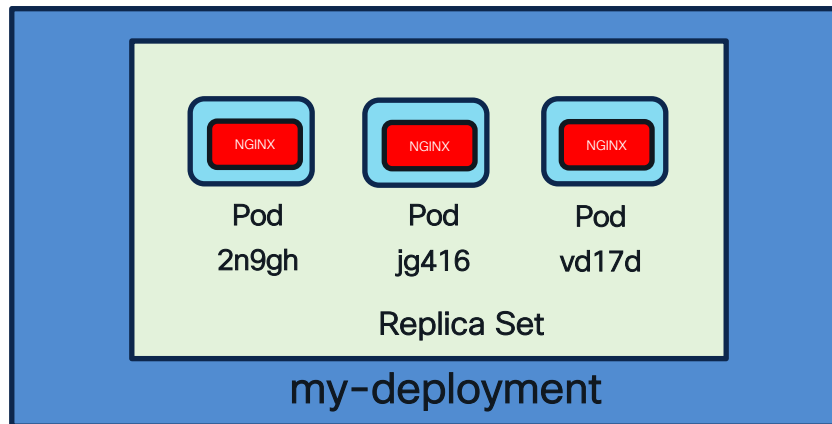
NAME	DESIRED	CURRENT	READY	AGE
my-deployment-65749d7bc9	3	3	3	3m46s



Confirming Pods Within A Pod

```
root@ubuntu-server:~# kubectl get pods
```

NAME	READY	STATUS	RESTARTS	AGE
my-deployment-65749d7bc9-2n9gh	1/1	Running	0	6m22s
my-deployment-65749d7bc9-jg4l6	1/1	Running	0	6m22s
my-deployment-65749d7bc9-vd17d	1/1	Running	0	6m22s



Viewing Deployments

```
root@ubuntu-server:~# kubectl describe deployment my-deployment
Name: my-deployment
Namespace: default
CreationTimestamp: Tue, 13 Aug 2019 18:18:51 +0000
Labels: <none>
Annotations: deployment.kubernetes.io/revision: 1
Selector: type=front-end
Replicas: 3 desired | 3 updated | 3 total | 3 available | 0 unavailable
StrategyType: RollingUpdate
MinReadySeconds: 0
RollingUpdateStrategy: 25% max unavailable, 25% max surge
Pod Template:
  Labels: type=front-end
  Containers:
    nginx-container:
      Image: nginx
      Port: <none>
      Host Port: <none>
      Environment: <none>
      Mounts: <none>
      Volumes: <none>
  Conditions:
    Type           Status    Reason
    ----           -
    Available      True     MinimumReplicasAvailable
    Progressing    True     NewReplicaSetAvailable
  OldReplicaSets: <none>
  NewReplicaSet: my-deployment-65749d7bc9 (3/3 replicas created)
Events:
  Type           Reason             Age   From              Message
  ----           -
  Normal         ScalingReplicaSet   20m   deployment-controller   Scaled up replica set my-deployment-65749d7bc9 to 3
```


Where to go from here?

Online Training:

- Cisco Platinum Learning Library: <https://digital-learning.cisco.com/#/login>
- Kubernetes for the Absolute Beginner on Udemy (Labs included):
<https://learning.oreilly.com/videos/kubernetes-for-absolute/9781838555962>
- Certified Kubernetes Administrator on Udemy (Labs included):
<https://www.udemy.com/course/certified-kubernetes-administrator-with-practice-tests/>



Where to go from here?

Lab Access:

- DevNet Learning Labs:
<https://developer.cisco.com/learning/>
- Install Minikube:
<https://kubernetes.io/docs/tasks/tools/install-minikube/>
- Setup Docker and Kubernetes Ubuntu VM:
<https://www.nakivo.com/blog/install-kubernetes-ubuntu/>





The bridge to possible

Thank you

CISCO *Live!*

#CiscoLive





TURN IT UP

CISCO *Live!*

#CiscoLive