Demystifying the Nuts & Bolts of Kubernetes Architecture

ReplicaSet101

docker

How can you ensure that there are 3 Pods instances which are always available and running at one point in time?

What is ReplicaSet all about?

Maintain a stable set of replica Pods running at any given time

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

  a. If there are access Pods, they get killed and vice versa
  b. New Pods are launched when they get failed, get deleted and terminated

- ReplicaSet & Pods are associated with “labels”

Replication Controller Vs ReplicaSet

- ReplicaSet is the next generation of Replication Controller
- Both serve the same purpose

$$\begin{tikzcd}
\text{ReplicaSet} \arrow[d] & \text{Replication Controller} \arrow[d] \\
\text{Set-based Selectors} & \text{Equality-based Selectors}
\end{tikzcd}$$

Labels & Selectors

When Pods are scaled, how are these Pods Managed at such large scale?

Pods

Controllers & Services

Labels

```
#Pod-Spec
apiVersion: v1
kind: pod
metadata:
  name: nginx-Pod
  labels:
    app: guestbook
    tier: frontend
    env: dev
spec:
  replicas: 5..
```

Selectors

Equality-based Selectors

Operators:

= and ==

Examples:

environment = production
tier! = frontend

Commandline:

```
$kubectl get pods -l environment=production
```

In Manifest:

```
..
selector:
  environment: production
  tier: frontend
..
```

Supports: Services, Replication Controller

Set-based Selectors

Operators:

in notin exists

Examples:

environment in (production, qa)
tier notin(frontend, backend)

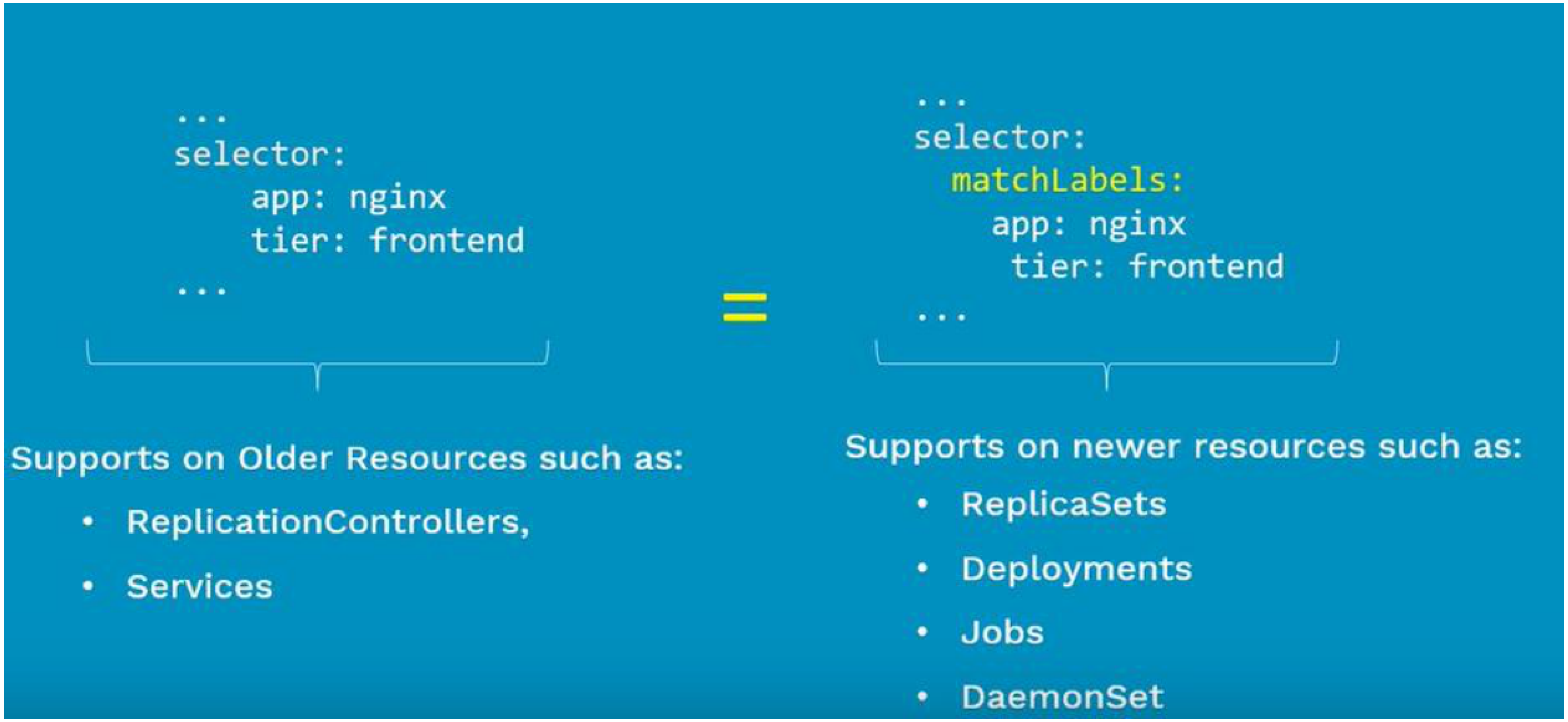
Commandline:

```
$kubectl get pods -l `enviornment in(production)
```

In Manifest:

```
..
selector:
  matchExpressions:
    - {key:environment,operator:in,values:[prod,qa]}
    - {key:tier,operator:Notin,values:[frontend,backend]}
..
```

Supports: Job, Deployment, ReplicaSet, DaemonSet



```
...
selector:
    app: nginx
    tier: frontend
...
```

=

```
...
selector:
  matchLabels:
    app: nginx
     tier: frontend
...
```

Supports on Older Resources such as:

- ReplicationControllers,
- Services

Supports on newer resources such as:

- ReplicaSets
- Deployments
- Jobs
- DaemonSet

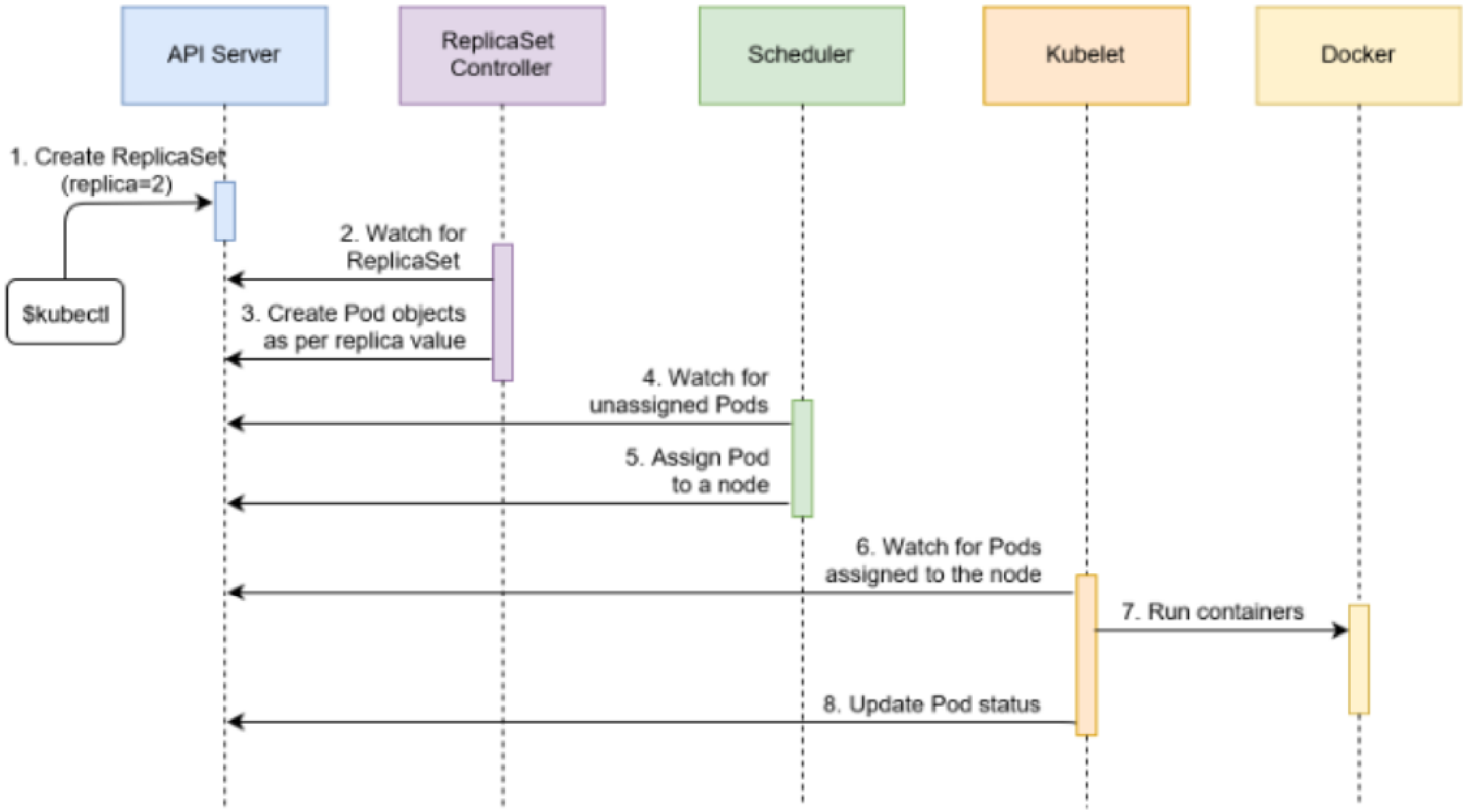
ReplicaSet Examples:

- Manifest file
- Deploy app using RS
- Display and validate RS
- Test – Node Fails
- Test – Scale Up
- Test – Scale Down

ReplicaSet Manifest File

```yaml
apiVersion: apps/v1
kind: ReplicaSet
metadata:
  name: nginx-rs
spec:
  replicas: 2
  selector:
    matchLabels:
      app: nginx-app
  template:
    metadata:
      name: nginx-pod
      labels:
        app: nginx-app
        tier: frontend
    spec:
      containers:
        - name: nginx
          image: nginx
          ports:
          - containerPort: 80
```

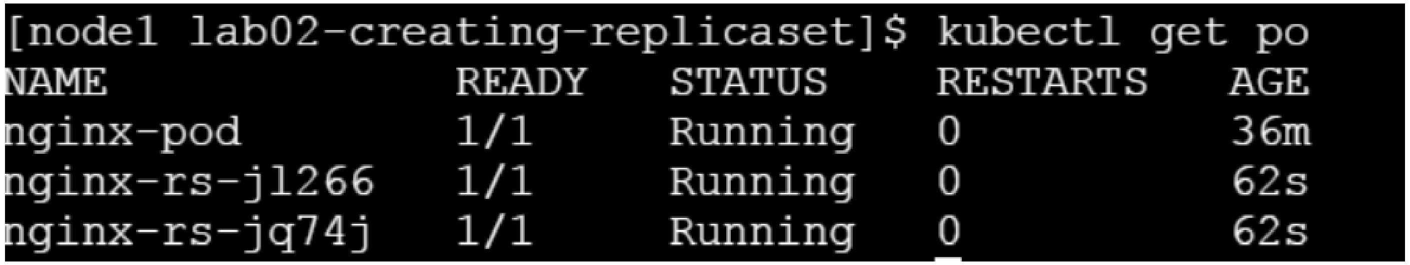
A Typical Replicaset Workflow



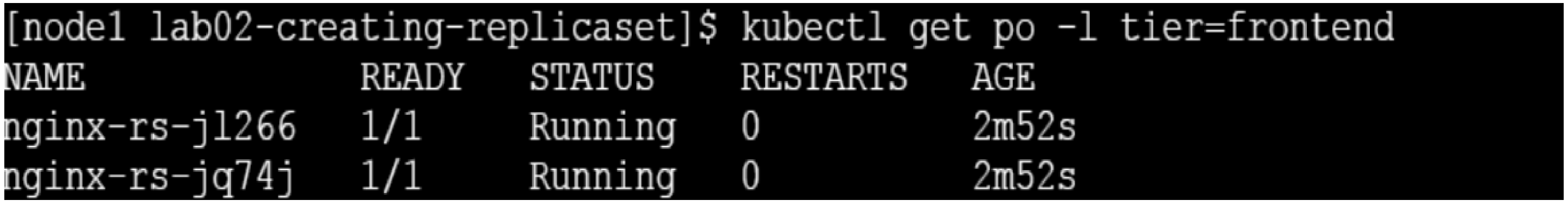
Credits" Viktor Farcic

Creating Nginx-rs Pods

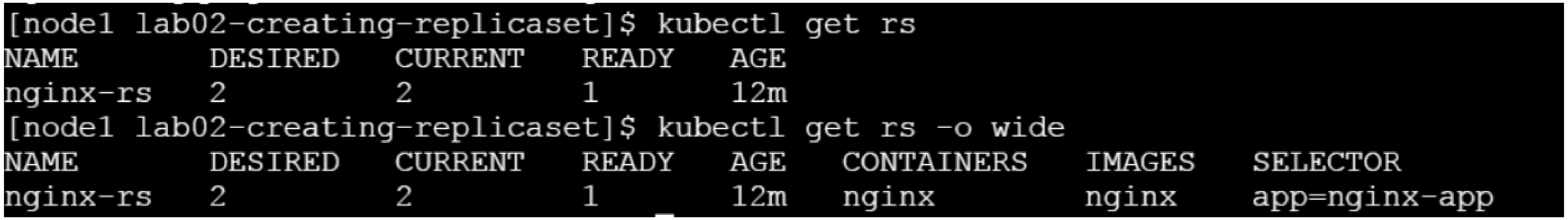
$kubectl create –f nginx-rs.yaml



```
[node1 lab02-creating-replicaset]$ kubectl get po
NAME             READY     STATUS    RESTARTS   AGE
nginx-pod        1/1       Running   0          36m
nginx-rs-jl266   1/1       Running   0          62s
nginx-rs-jq74j   1/1       Running   0          62s
```

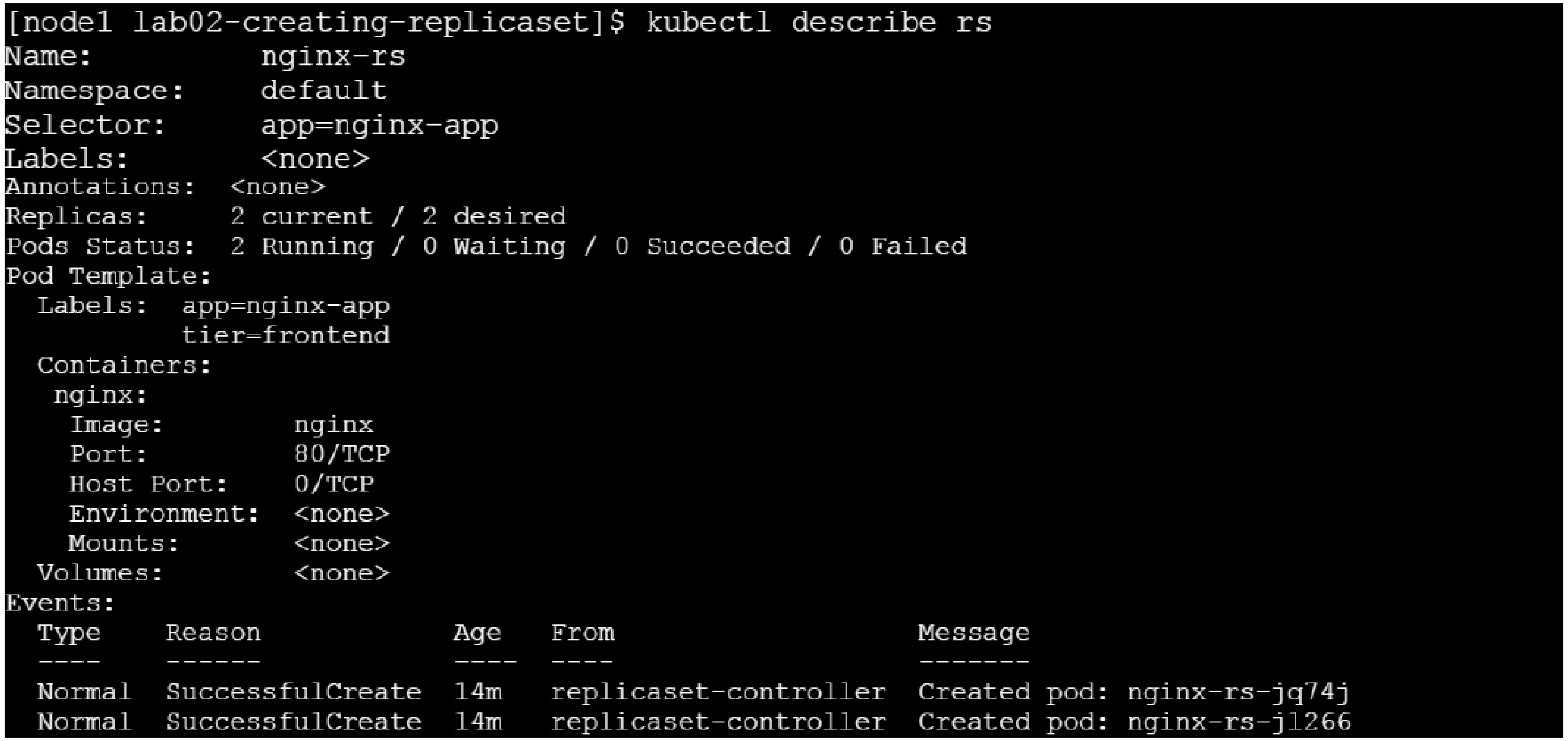


```
[node1 lab02-creating-replicaset]$ kubectl get po -l tier=frontend
NAME             READY     STATUS    RESTARTS   AGE
nginx-rs-jl266   1/1       Running   0          2m52s
nginx-rs-jq74j   1/1       Running   0          2m52s
```



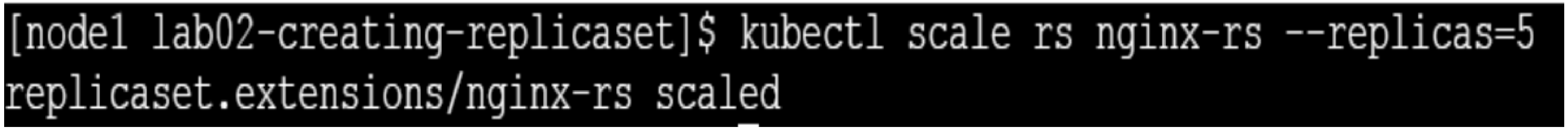
```
[node1 lab02-creating-replicaset]$ kubectl get rs
NAME       DESIRED   CURRENT   READY   AGE
nginx-rs   2         2         1       12m
[node1 lab02-creating-replicaset]$ kubectl get rs -o wide
NAME       DESIRED   CURRENT   READY   AGE   CONTAINERS   IMAGES   SELECTOR
nginx-rs   2         2         1       12m   nginx        nginx    app=nginx-app
```

Checking the state of ReplicaSet



```
[node1 lab02-creating-replicaset]$ kubectl describe rs
Name:           nginx-rs
Namespace:      default
Selector:       app=nginx-app
Labels:         <none>
Annotations:   <none>
Replicas:      2 current / 2 desired
Pods Status:   2 Running / 0 Waiting / 0 Succeeded / 0 Failed
Pod Template:
  Labels:  app=nginx-app
           tier=frontend
  Containers:
   nginx:
    Image:         nginx
    Port:          80/TCP
    Host Port:     0/TCP
    Environment:   <none>
    Mounts:        <none>
  Volumes:         <none>
Events:
  Type    Reason            Age   From                   Message
  ----    ------            ----  ----                   -------
  Normal  SuccessfulCreate  14m   replicaset-controller  Created pod: nginx-rs-jq74j
  Normal  SuccessfulCreate  14m   replicaset-controller  Created pod: nginx-rs-jl266
```

Scaling the Nginx Service



```
[node1 lab02-creating-replicaset]$ kubectl scale rs nginx-rs --replicas=5
replicaset.extensions/nginx-rs scaled
```

Thank You