Table of Contents

Kubernetes Basics

Version

Author	Version	Last update	Comment
luwenwu@cn.ibm.com	V1.0	03/06/18	Create the document

What is Kubernetes

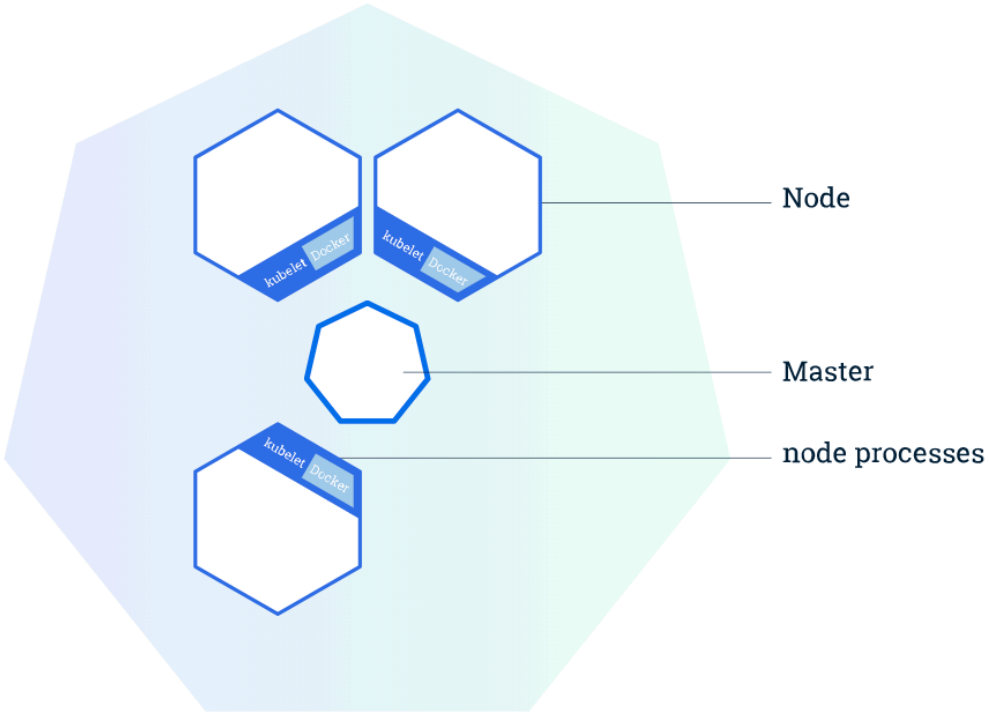
https://kubernetes.io/

Kubernetes is an open-source system for automating deployment, scaling, and management of containerized applications.
It groups containers that make up an application into logical units for easy management and discovery. Kubernetes builds upon 15 years of experience of running production workloads at Google, combined with best-of-breed ideas and practices from the community.

Creating a Cluster

Masters manage the cluster and the nodes are used to host the running applications. A Kubernetes cluster that handles production traffic should have a minimum of three nodes.

Cluster Diagram



Kubernetes cluster

Minikube Install & Run on Mac

With Virtualbox & docker installed on my MacOS, I just run:

$brew cask install minikube
$minikube version
$minikube start
$curl http://localhost:8001/
$minikube stop

Minikube Run on Kubernetes.io Tutorial

https://kubernetes.io/docs/tutorials/kubernetes-basics/cluster-interactive/

$minikube start

$ kubectl version

Client Version: version.Info{Major:"1", Minor:"9", GitVersion:"v1.9.0", GitCommit:"925c127ec6b946659ad0fd596fa959be43f0cc05", GitTreeState:"clean", BuildDate:"2017-12-15T21:07:38Z", GoVersion:"go1.9.2", Compiler:"gc", Platform:"linux/amd64"}
Server Version: version.Info{Major:"", Minor:"", GitVersion:"v1.9.0", GitCommit:"925c127ec6b946659ad0fd596fa959be43f0cc05", GitTreeState:"clean", BuildDate:"2018-01-26T19:04:38Z", GoVersion:"go1.9.1", Compiler:"gc", Platform:"linux/amd64"}

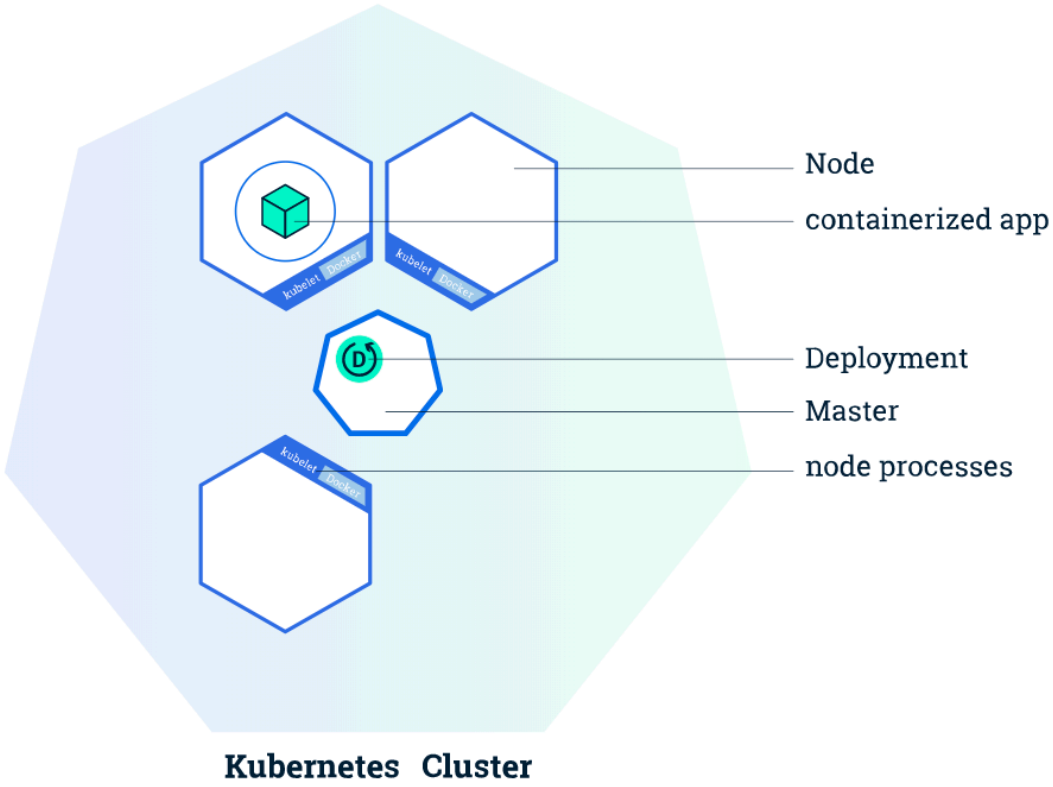
$kubectl cluster-info

$kubectl get nodes

$kubectl cluster-info dump

Deploy an App

Deploying your first app on Kubernetes



```
$ kubectl run kubernetes-bootcamp --image=gcr.io/google-samples/kubernetes-bootcamp:v1 --port=8080
deployment "kubernetes-bootcamp" created
```

```
$ kubectl get deployments
NAME                  DESIRED   CURRENT   UP-TO-DATE   AVAILABLE   AGE
kubernetes-bootcamp   1         1         1            1           40s
```

View our App

Pods that are running inside Kubernetes are running on a private, isolated network. By default they are visible from other pods and services within the same kubernetes cluster, but not outside that network. When we use kubectl, we're interacting through an API endpoint to communicate with our application.
The kubectl command can create a proxy that will forward communications into the cluster-wide, private network.

```
$ kubectl proxy
Starting to serve on 127.0.0.1:8001
```

We now have a connection between our host (the online terminal) and the Kubernetes cluster. The proxy enables direct access to the API from these terminals.

```
$kubectl version
Client Version: version.Info{Major:"1", Minor:"9", GitVersion:"v1.9.0", GitCommit:"925c127ec6b946659ad0fd596fa959be43f0cc05",
GitTreeState:"clean", BuildDate:"2017-12-15T21:07:38Z", GoVersion:"go1.9.2", Compiler:"gc", Platform:"linux/amd64"}
Server Version: version.Info{Major:"", Minor:"", GitVersion:"v1.9.0", GitCommit:"925c127ec6b946659ad0fd596fa959be43f0cc05",
GitTreeState:"clean", BuildDate:"2018-01-26T19:04:38Z", GoVersion:"go1.9.1", Compiler:"gc", Platform:"linux/amd64"}
```

```
$ curl http://localhost:8001/version
{
  "major": "",
  "minor": "",
```

```
 "gitVersion": "v1.9.0",
 "gitCommit": "925c127ec6b946659ad0fd596fa959be43f0cc05",
 "gitTreeState": "clean",
 "buildDate": "2018-01-26T19:04:38Z",
 "goVersion": "go1.9.1",
 "compiler": "gc",
 "platform": "linux/amd64"
}$
```

```
$ kubectl get pods
NAME                          READY    STATUS    RESTARTS   AGE
kubernetes-bootcamp-5dbf48f7d4-wvbbs   1/1      Running   0        19m
```

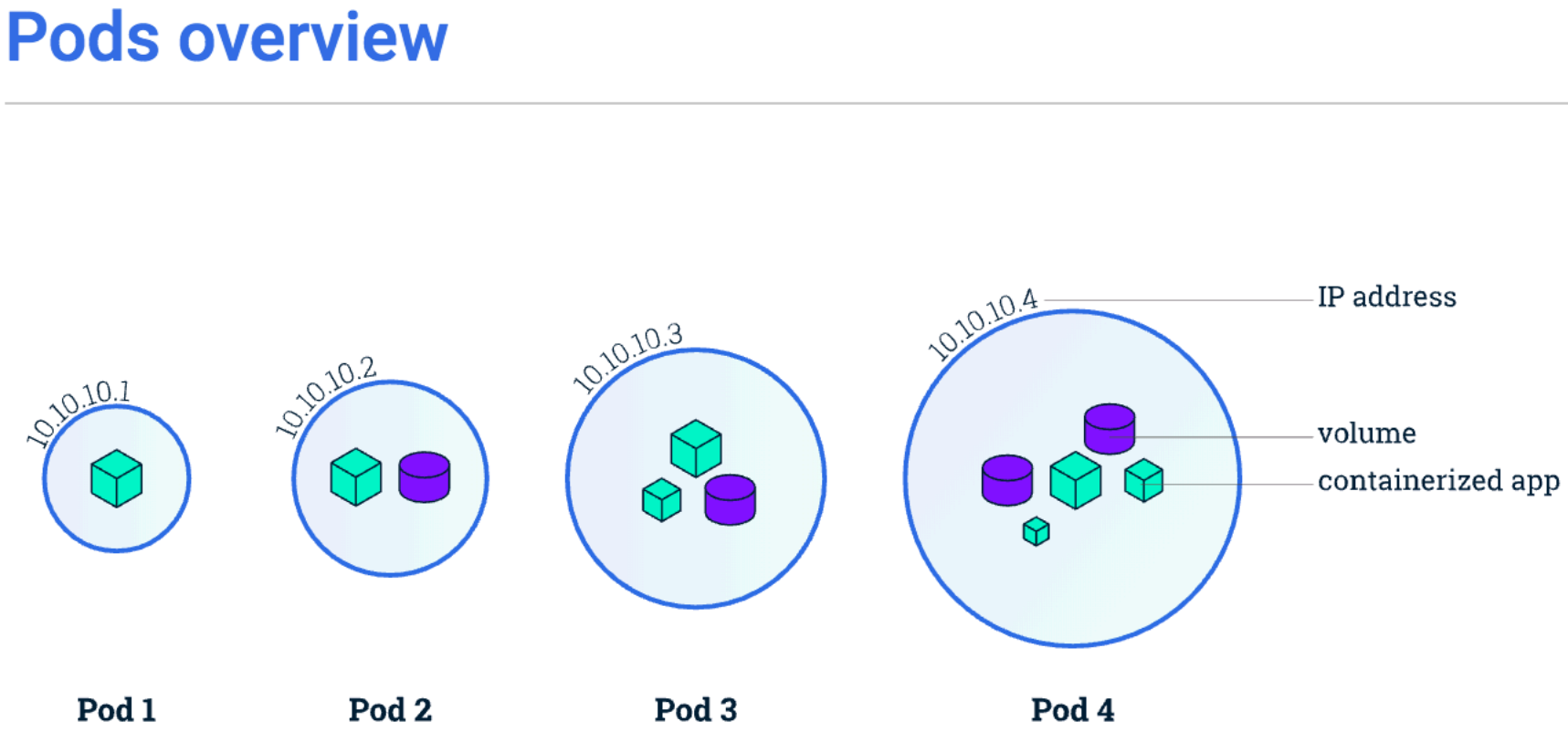
```
$export POD_NAME=$(kubectl get pods -o go-template --template '{{range .items}}{{.metadata.name}}{{"\n"}}{{end}}')
$echo Name of the Pod: $POD_NAME

$ curl http://localhost:8001/api/v1/proxy/namespaces/default/pods/$POD_NAME/
Hello Kubernetes bootcamp! | Running on: kubernetes-bootcamp-5dbf48f7d4-wvbbs | v=1
```

Explore your App

Pods overview

A Pod is a group of one or more application containers (such as Docker or rkt) and includes shared storage (volumes), IP address and information about how to run them.



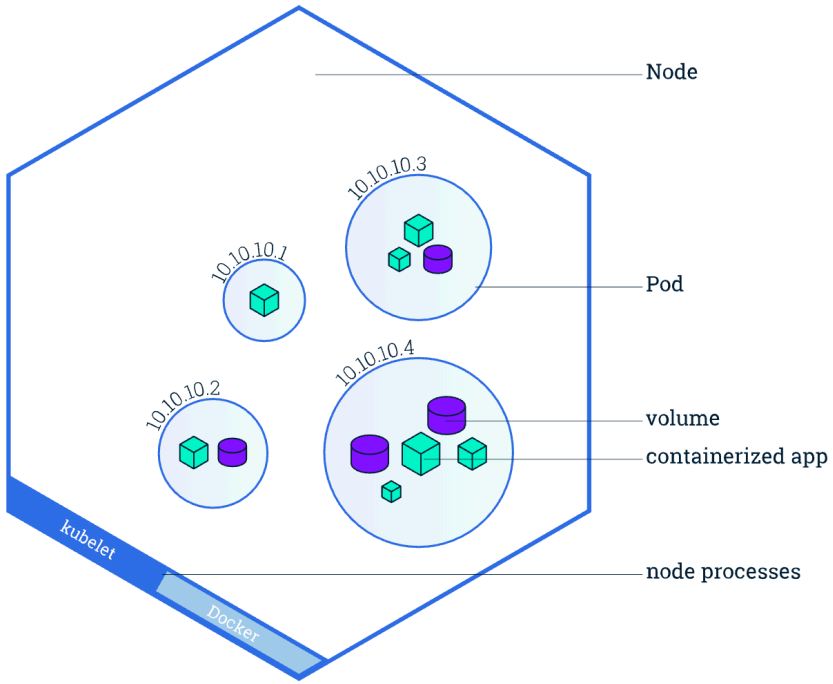
Nodes overview

A Pod always runs on a Node. A Node is a worker machine in Kubernetes and may be either a virtual or a physical machine, depending on the cluster. Each Node is managed by the Master. A Node can have multiple pods, and the Kubernetes master automatically handles scheduling the pods across the Nodes in the cluster. The Master's automatic scheduling takes into account the available resources on each Node.

Every Kubernetes Node runs at least:

Kubelet, a process responsible for communication between the Kubernetes Master and the Node; it manages the Pods and the containers running on a machine.
A container runtime (like Docker, rkt) responsible for pulling the container image from a registry, unpacking the container, and running the application.
Containers should only be scheduled together in a single Pod if they are tightly coupled and need to share resources such as disk.

Node overview



Get Pod Info

$ kubectl get pods

```
NAME                                READY   STATUS    RESTARTS   AGE
kubernetes-bootcamp-5dbf48f7d4-djr6t   1/1     Running   0          1m
```

Get Container Info in Pod

$ kubectl describe pods

```
Name:           kubernetes-bootcamp-5dbf48f7d4-djr6t
Namespace:      default
Node:           host01/172.17.0.41
Start Time:     Thu, 01 Mar 2018 03:55:36 +0000
Labels:         pod-template-hash=1869049380
                run=kubernetes-bootcamp
Annotations:    <none>
Status:         Running
IP:             172.18.0.4
Controlled By:  ReplicaSet/kubernetes-bootcamp-5dbf48f7d4
Containers:
  kubernetes-bootcamp:
    Container ID:   docker://6ef65802f6ece6cb135c02f7ed70c71b2ec76654e6807f3d58d1110b7a9ef155
    Image:          gcr.io/google-samples/kubernetes-bootcamp:v1
    Image ID:       docker-pullable://jocatalin/kubernetes-
bootcamp@sha256:0d6b8ee63bb57c5f5b6156f446b3bc3b3c143d233037f3a2f00e279c8fcc64af
    Port:           8080/TCP
    State:          Running
      Started:      Thu, 01 Mar 2018 03:55:37 +0000
    Ready:          True
    Restart Count:  0
    Environment:    <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from default-token-6vlgq (ro)
Conditions:
  Type          Status
  Initialized   True
  Ready         True
  PodScheduled  True
```

```
Volumes:
  default-token-6vlgq:
    Type:       Secret (a volume populated by a Secret)
    SecretName:  default-token-6vlgq
    Optional:    false
QoS Class:      BestEffort
Node-Selectors:  <none>
Tolerations:     <none>
Events:
  Type     Reason            Age          From            Message
  ----     ------            ----         ----            -------
  Warning  FailedScheduling  1m (x4 over 1m)  default-scheduler  0/1 nodes are available: 1 NodeNotReady.
  Normal   Scheduled         1m           default-scheduler  Successfully assigned kubernetes-bootcamp-5dbf48f7d4-djr6t to host01
  Normal   SuccessfulMountVolume  1m           kubelet, host01    MountVolume.SetUp succeeded for volume "default-token-6vlgq"
  Normal   Pulled            1m           kubelet, host01    Container image "gcr.io/google-samples/kubernetes-bootcamp:v1" already present
on machine
  Normal   Created           1m           kubelet, host01    Created container
  Normal   Started           1m           kubelet, host01    Started container
```

$ kubectl cluster-info

Kubernetes master is running at https://172.17.0.41:8443

Show App Output on Terminal

$kubectl proxy

Now again, we'll get the Pod name and query that pod directly through the proxy. To get the Pod name and store it in the POD_NAME environment variable:

$export POD_NAME=$(kubectl get pods -o go-template --template '{{range .items}}{{.metadata.name}}{{"\n"}}{{end}}')

$echo Name of the Pod: $POD_NAME

To see the output of our application, run a curl request.

$curl http://localhost:8001/api/v1/proxy/namespaces/default/pods/$POD_NAME/

View the container logs

Anything that the application would normally send to STDOUT becomes logs for the container within the Pod. We can retrieve these logs using the kubectl logs command:

```
$kubectl logs $POD_NAME
```

Note: We don't need to specify the container name, because we only have one container inside the pod.

Executing command on the container

We can execute commands directly on the container once the Pod is up and running. Let's list the environment variables:

```
$kubectl exec $POD_NAME env
```

Again, worth mentioning that the name of the container itself can be omitted since we only have a single container in the Pod.

Next let's start a bash session in the Pod's container:

```
$kubectl exec -ti $POD_NAME bash
```

We have now an open console on the container where we run our NodeJS application. The source code of the app is in the server.js file:

```
$cat server.js
```

You can check that the application is up by running a curl command:

$curl localhost:8080

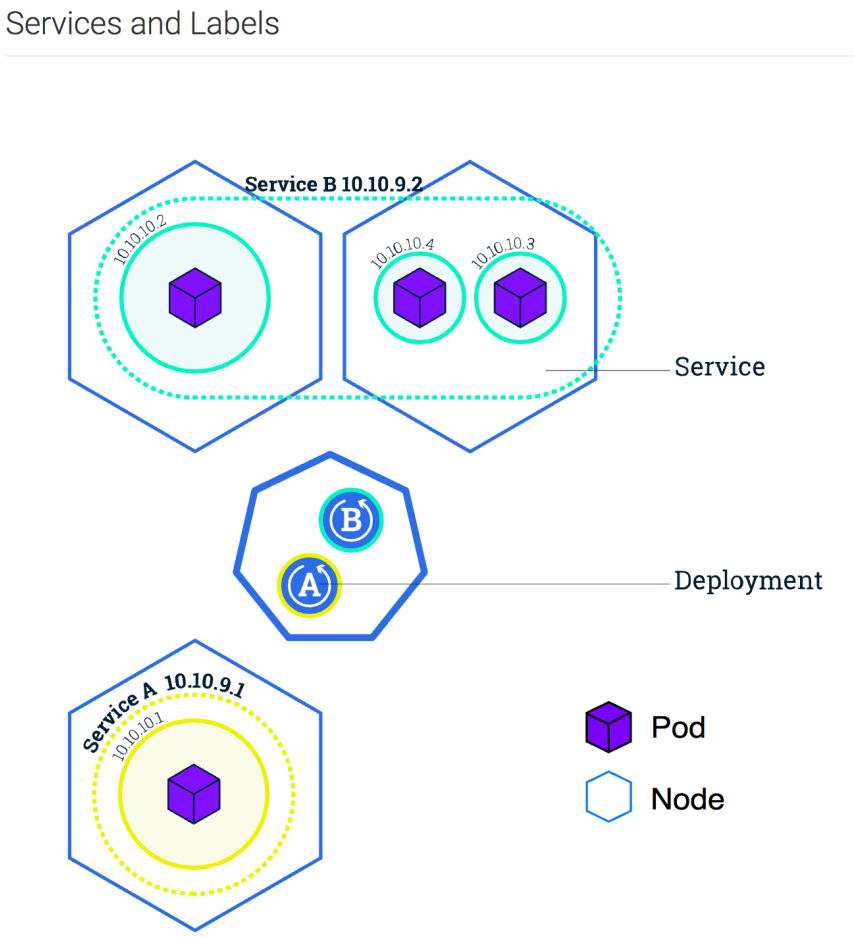
Note: here we used localhost because we executed the command inside the NodeJS container

To close your container connection type exit.

Using a Service to Expose App

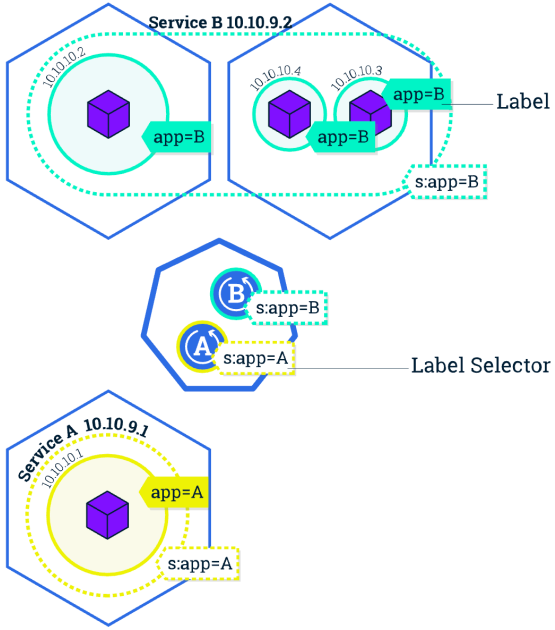
Services and Labels

A Service routes traffic across a set of Pods. Services are the abstraction that allow pods to die and replicate in Kubernetes without impacting your application. Discovery and routing among dependent Pods (such as the frontend and backend components in an application) is handled by Kubernetes Services.



Services match a set of Pods using labels and selectors, a grouping primitive that allows logical operation on objects in Kubernetes. Labels are key/value pairs attached to objects and can be used in any number of ways:

Designate objects for development, test, and production
Embed version tags
Classify an object using tags
You can create a Service at the same time you create a Deployment by using
--expose in kubectl.



Labels can be attached to objects at creation time or later on. They can be modified at any time. Let's expose our application now using a Service and apply some labels.

Create a New Service

We have a Service called kubernetes that is created by default when minikube starts the cluster. To create a new service and expose it to external traffic we'll use the expose command with NodePort as parameter (minikube does not support the LoadBalancer option yet)

```
$ kubectl get pods
NAME                         READY    STATUS    RESTARTS   AGE
kubernetes-bootcamp-5dbf48f7d4-wvl28   1/1      Running   0        7s

$ kubectl get services
NAME       TYPE       CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
kubernetes   ClusterIP   10.96.0.1    <none>       443/TCP   30s

# deployment/kubernetes-bootcamp is already there, could be checked by using kubectl get deployments

$ kubectl expose deployment/kubernetes-bootcamp --type="NodePort" --port 8080
service "kubernetes-bootcamp" exposed

$ kubectl get services
NAME              TYPE       CLUSTER-IP       EXTERNAL-IP   PORT(S)          AGE
kubernetes          ClusterIP   10.96.0.1       <none>       443/TCP          2m
kubernetes-bootcamp   NodePort    10.107.223.119   <none>       8080:30385/TCP   8s

$ kubectl describe services/kubernetes-bootcamp
Name:                 kubernetes-bootcamp
Namespace:            default
Labels:               run=kubernetes-bootcamp
Annotations:          <none>
Selector:             run=kubernetes-bootcamp
Type:                 NodePort
```

```
IP:                     10.107.223.119
Port:                   <unset>  8080/TCP
TargetPort:             8080/TCP
NodePort:               <unset>  30385/TCP
Endpoints:              172.18.0.2:8080
Session Affinity:       None
External Traffic Policy:  Cluster
Events:                 <none>
```

$export NODE_PORT=$(kubectl get services/kubernetes-bootcamp -o go-template='{{(index .spec.ports 0).nodePort}}')

$echo NODE_PORT=$NODE_PORT

$ curl $(minikube ip):$NODE_PORT

Hello Kubernetes bootcamp! | Running on: kubernetes-bootcamp-5dbf48f7d4-wvl28 | v=1

Using Labels

The Deployment created automatically a label for our Pod. With describe deployment command you can see the name of the label:

$kubectl get deployments

$kubectl describe deployment

```
Name:                   kubernetes-bootcamp
Namespace:              default
CreationTimestamp:      Thu, 01 Mar 2018 14:32:59 +0000
Labels:                 run=kubernetes-bootcamp
Annotations:            deployment.kubernetes.io/revision=1
Selector:               run=kubernetes-bootcamp
Replicas:               1 desired | 1 updated | 1 total | 1 available | 0 unavailable
StrategyType:           RollingUpdate
MinReadySeconds:        0
```

```
RollingUpdateStrategy:  1 max unavailable, 1 max surge
Pod Template:
  kubernetes-bootcamp:
   Image:       gcr.io/google-samples/kubernetes-bootcamp:v1
   Port:        8080/TCP
   Environment:  <none>
   Mounts:      <none>
 Volumes:       <none>
Conditions:
 Type         Status  Reason
 ----         ------  ------
 Available    True    MinimumReplicasAvailable
OldReplicaSets:  <none>
NewReplicaSet:   kubernetes-bootcamp-5dbf48f7d4 (1/1 replicas created)
Events:
 Type    Reason          Age   From                Message
 ----    ------          ----  ----                -------
 Normal  ScalingReplicaSet  6m    deployment-controller  Scaled up replica set kubernetes-bootcamp-5dbf48f7d4 to 1
```

Let's use this label to query our list of Pods. We'll use the kubectl get pods command with -l as a parameter, followed by the label values:

```
$kubectl get pods -l run=kubernetes-bootcamp
NAME                          READY    STATUS    RESTARTS   AGE
kubernetes-bootcamp-5dbf48f7d4-wvl28   1/1      Running   0        7m
```

You can do the same to list the existing services:

```
$kubectl get services -l run=kubernetes-bootcamp
NAME              TYPE      CLUSTER-IP      EXTERNAL-IP   PORT(S)        AGE
kubernetes-bootcamp   NodePort   10.107.223.119   <none>      8080:30385/TCP   6m
```

Get the name of the Pod and store it in the POD_NAME environment variable:

```
$export POD_NAME=$(kubectl get pods -o go-template --template '{{range .items}}{{.metadata.name}}{{"\n"}}{{end}}')
```
$echo Name of the Pod: $POD_NAME

To apply a new label we use the label command followed by the object type, object name and the new label:

$kubectl label pod $POD_NAME app=v1

This will apply a new label to our Pod (we pinned the application version to the Pod), and we can check it with the describe pod command:

```
$ kubectl label pod $POD_NAME app=v1
pod "kubernetes-bootcamp-5dbf48f7d4-wvl28" labeled

$ kubectl describe pods $POD_NAME
Name:           kubernetes-bootcamp-5dbf48f7d4-wvl28
Namespace:      default
Node:           host01/172.17.0.24
Start Time:     Thu, 01 Mar 2018 14:33:02 +0000
Labels:         app=v1
                pod-template-hash=1869049380
                run=kubernetes-bootcamp
Annotations:    <none>
Status:         Running
IP:             172.18.0.2
Controlled By:  ReplicaSet/kubernetes-bootcamp-5dbf48f7d4
Containers:
  kubernetes-bootcamp:
    Container ID:   docker://1e3406acf086001f680fb63e532ac9727899967adfd5064699fa39c0067adfab
    Image:          gcr.io/google-samples/kubernetes-bootcamp:v1
```

```
  Image ID:      docker-pullable://jocatalin/kubernetes-
bootcamp@sha256:0d6b8ee63bb57c5f5b6156f446b3bc3b3c143d233037f3a2f00e279c8fcc64af
  Port:          8080/TCP
  State:         Running
     Started:      Thu, 01 Mar 2018 14:33:03 +0000
    Ready:         True
    Restart Count:  0
    Environment:    <none>
    Mounts:
     /var/run/secrets/kubernetes.io/serviceaccount from default-token-nhqq2 (ro)
  Conditions:
   Type          Status
   Initialized   True
   Ready         True
   PodScheduled   True
  Volumes:
   default-token-nhqq2:
    Type:       Secret (a volume populated by a Secret)
    SecretName:  default-token-nhqq2
    Optional:    false
  QoS Class:      BestEffort
  Node-Selectors:  <none>
  Tolerations:     <none>
  Events:
   Type     Reason             Age            From            Message
   ----     ------             ----           ----            -------
   Warning  FailedScheduling      11m (x3 over 11m)  default-scheduler  0/1 nodes are available: 1 NodeNotReady.
   Normal   Scheduled           11m            default-scheduler  Successfully assigned kubernetes-bootcamp-5dbf48f7d4-wvl28 to host01
   Normal   SuccessfulMountVolume  11m            kubelet, host01    MountVolume.SetUp succeeded for volume "default-token-nhqq2"
   Normal   Pulled             11m            kubelet, host01    Container image "gcr.io/google-samples/kubernetes-bootcamp:v1" already
present on machine
```

```
  Normal  Created   11m   kubelet, host01  Created container
  Normal  Started   11m   kubelet, host01  Started container
```

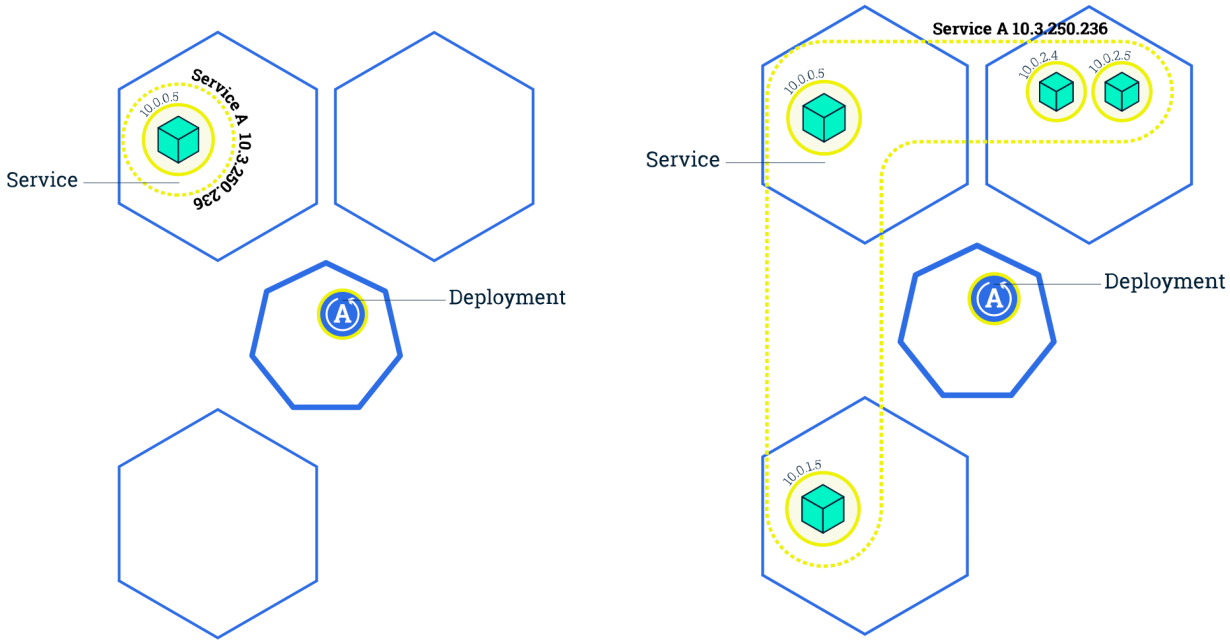
We see here that the label is attached now to our Pod. And we can query now the list of pods using the new label:

```
$kubectl get pods -l app=v1
NAME                               READY   STATUS    RESTARTS   AGE
kubernetes-bootcamp-5dbf48f7d4-wvl28   1/1     Running   0          12m
```

And we see the Pod.

Scale Your App

Scaling out a Deployment will ensure new Pods are **created** and scheduled to Nodes with available resources. **Scaling in** will **reduce** the number of Pods to the new desired state. Kubernetes also supports autoscaling of Pods, but it is outside of the scope of this tutorial. Scaling to zero is also possible, and it will terminate all Pods of the specified Deployment.



You can create from the start a Deployment with multiple instances using the --replicas parameter for the kubectl run command.

$ kubectl get deployments

NAME DESIRED CURRENT UP-TO-DATE AVAILABLE AGE

```
kubernetes-bootcamp   1       1       1         1        8m
```

$ kubectl scale deployments/kubernetes-bootcamp --replicas=4

```
deployment "kubernetes-bootcamp" scaled
```

$ kubectl get deployments

```
NAME                  DESIRED   CURRENT   UP-TO-DATE   AVAILABLE   AGE
kubernetes-bootcamp   4       4       4         4        9m
```

$ kubectl get pods -o wide

```
NAME                                  READY    STATUS    RESTARTS   AGE     IP          NODE
kubernetes-bootcamp-5dbf48f7d4-qjmxn   1/1      Running   0       16s     172.18.0.5   host01
kubernetes-bootcamp-5dbf48f7d4-tdjl6   1/1      Running   0       16s     172.18.0.6   host01
kubernetes-bootcamp-5dbf48f7d4-vsw5d   1/1      Running   0       16s     172.18.0.7   host01
kubernetes-bootcamp-5dbf48f7d4-wfxjw   1/1      Running   0       9m      172.18.0.2   host01
```

$ kubectl describe deployments/kubernetes-bootcamp

```
Name:                   kubernetes-bootcamp
Namespace:              default
CreationTimestamp:      Tue, 06 Mar 2018 19:16:14 +0000
Labels:                 run=kubernetes-bootcamp
Annotations:            deployment.kubernetes.io/revision=1
Selector:               run=kubernetes-bootcamp
Replicas:               4 desired | 4 updated | 4 total | 4 available | 0 unavailable
StrategyType:           RollingUpdate
MinReadySeconds:        0
RollingUpdateStrategy:  1 max unavailable, 1 max surge
Pod Template:
  Labels:  run=kubernetes-bootcamp
  Containers:
   kubernetes-bootcamp:
```

```
    Image:       gcr.io/google-samples/kubernetes-bootcamp:v1
    Port:        8080/TCP
    Environment:  <none>
    Mounts:      <none>
  Volumes:       <none>
Conditions:
  Type         Status  Reason
  ----         ------  ------
  Available    True    MinimumReplicasAvailable
OldReplicaSets:  <none>
NewReplicaSet:   kubernetes-bootcamp-5dbf48f7d4 (4/4 replicas created)
Events:
  Type    Reason             Age   From                   Message
  ----    ------             ----  ----                   -------
  Normal  ScalingReplicaSet  9m    deployment-controller  Scaled up replica set kubernetes-bootcamp-5dbf48f7d4 to 1
  Normal  ScalingReplicaSet  23s   deployment-controller  Scaled up replica set kubernetes-bootcamp-5dbf48f7d4 to 4
$
```

$ kubectl get services

```
NAME                  TYPE        CLUSTER-IP     EXTERNAL-IP   PORT(S)          AGE
kubernetes            ClusterIP   10.96.0.1      <none>        443/TCP          12m
kubernetes-bootcamp   NodePort    10.106.20.92   <none>        8080:30307/TCP   11m
```

$ kubectl get deployments

```
NAME                  DESIRED   CURRENT   UP-TO-DATE   AVAILABLE   AGE
kubernetes-bootcamp   4         4         4            4           12m
```

$ kubectl describe services/kubernetes-bootcamp

```
Name:                   kubernetes-bootcamp
Namespace:              default
Labels:                 run=kubernetes-bootcamp
```

```
Annotations:          <none>
Selector:             run=kubernetes-bootcamp
Type:                 NodePort
IP:                   10.106.20.92
Port:                 <unset>  8080/TCP
TargetPort:           8080/TCP
NodePort:             <unset>  30307/TCPEndpoints:          172.18.0.2:8080,172.18.0.5:8080,172.18.0.6:8080 + 1 more...
Session Affinity:     NoneExternal Traffic Policy:  Cluster
Events:               <none>
```

```
$ export NODE_PORT=$(kubectl get services/kubernetes-bootcamp -o go-template='{{(index .spec.ports 0).nodePort}}')
$ echo NODE_PORT=$NODE_PORT
NODE_PORT=30307

# Run curl for several times to hit a different pod
$ curl $(minikube ip):$NODE_PORT
Hello Kubernetes bootcamp! | Running on: kubernetes-bootcamp-5dbf48f7d4-qjmxn | v=1

$ curl $(minikube ip):$NODE_PORT
Hello Kubernetes bootcamp! | Running on: kubernetes-bootcamp-5dbf48f7d4-wfxjw | v=1
$ curl $(minikube ip):$NODE_PORT
Hello Kubernetes bootcamp! | Running on: kubernetes-bootcamp-5dbf48f7d4-qjmxn | v=1
$ curl $(minikube ip):$NODE_PORT
Hello Kubernetes bootcamp! | Running on: kubernetes-bootcamp-5dbf48f7d4-tdjl6 | v=1
$


$ kubectl scale deployments/kubernetes-bootcamp --replicas=2
deployment "kubernetes-bootcamp" scaled
```

$ kubectl get pods

```
NAME                          READY    STATUS       RESTARTS   AGE
kubernetes-bootcamp-5dbf48f7d4-qjmxn   1/1      Terminating   0        10m
kubernetes-bootcamp-5dbf48f7d4-tdjl6   1/1      Running       0        10m
kubernetes-bootcamp-5dbf48f7d4-vsw5d   1/1      Terminating   0        10m
kubernetes-bootcamp-5dbf48f7d4-wfxjw   1/1      Running       0        19m
```

$ kubectl get pods -o wide

```
NAME                          READY    STATUS    RESTARTS   AGE     IP          NODE
kubernetes-bootcamp-5dbf48f7d4-tdjl6   1/1      Running   0        11m     172.18.0.6   host01
kubernetes-bootcamp-5dbf48f7d4-wfxjw   1/1      Running   0        20m     172.18.0.2   host01
$
```

Update Your App

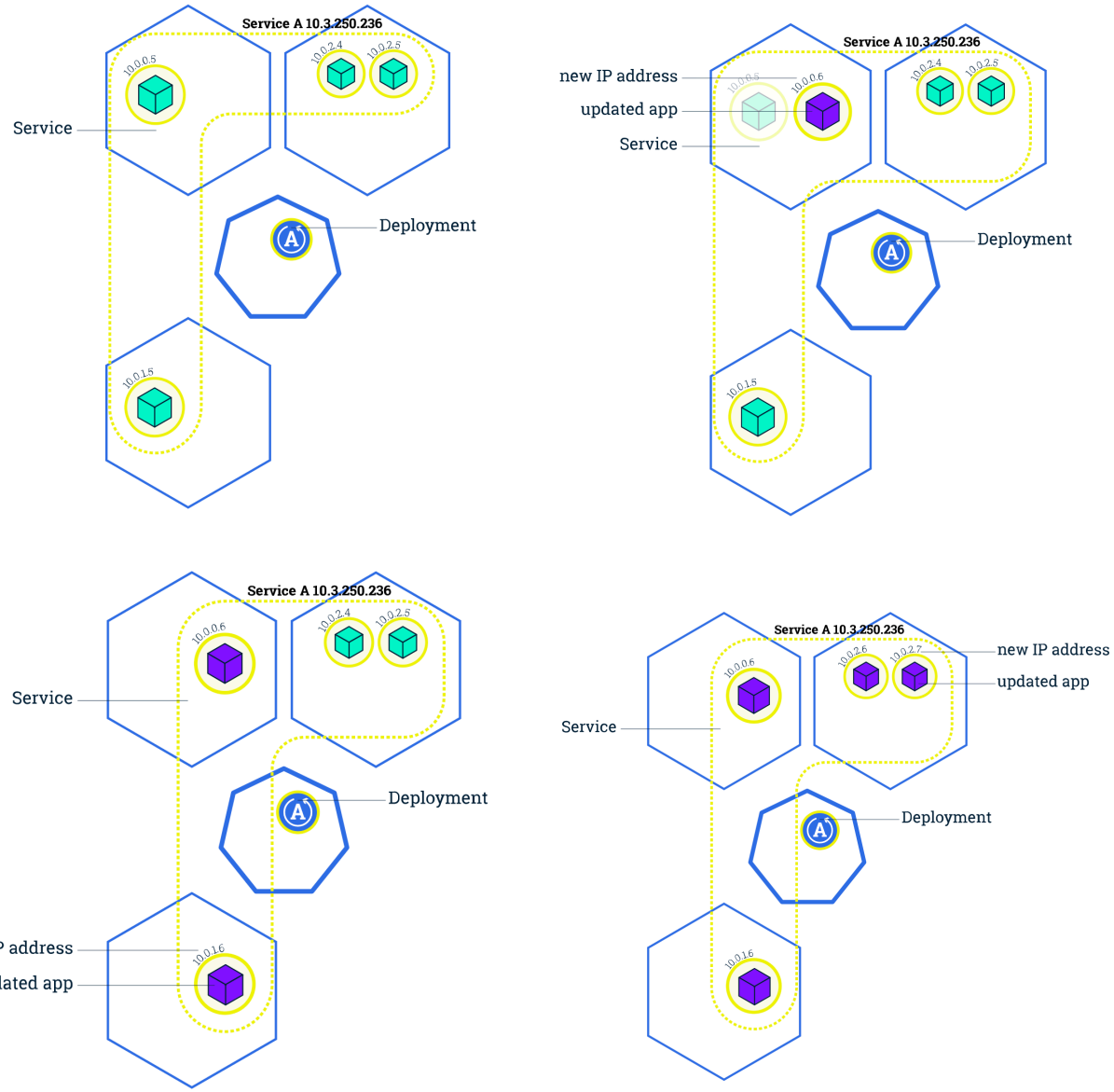
Performing a Rolling Update

Users expect applications to be available all the time and developers are expected to deploy new versions of them several times a day. In Kubernetes this is done with rolling updates. Rolling updates allow Deployments' update to take place with zero downtime by incrementally updating Pods instances with new ones. The new Pods will be scheduled on Nodes with available resources.

In the previous module we scaled our application to run multiple instances. This is a requirement for performing updates without affecting application availability. By default, the maximum number of Pods that can be unavailable during the update and the maximum number of new Pods that can be created, is one. Both options can be configured to either numbers or percentages (of Pods). In Kubernetes, updates are versioned and any Deployment update can be reverted to previous (stable) version.

Summary:
Updating an app
Rolling updates allow Deployments' update to take place with zero downtime by incrementally updating Pods instances with new ones.

Rolling update overview



$ kubectl describe pods

```
Name:           kubernetes-bootcamp-7689dc585d-c4fnp
Namespace:      default
Node:           host01/172.17.0.97
Start Time:     Tue, 06 Mar 2018 19:58:09 +0000
Labels:         pod-template-hash=3245871418
                run=kubernetes-bootcamp
Annotations:    <none>
Status:         Running
IP:             172.18.0.9
Controlled By:  ReplicaSet/kubernetes-bootcamp-7689dc585d
Containers:
  kubernetes-bootcamp:
    Container ID:   docker://d6c1a0efdbb1810a009f3fd2c830dc89ccdf1217db24e721ae299845b026bbfe
    Image:          jocatalin/kubernetes-bootcamp:v2
    Image ID:       docker-pullable://jocatalin/kubernetes-bootcamp@sha256:fb1a3ced00cecfc1f83f18ab5cd14199e30adc1b49aa4244f5d65ad3f5feb2a5
    Port:           8080/TCP
    State:          Running
      Started:      Tue, 06 Mar 2018 19:58:10 +0000
    Ready:          True
    Restart Count:  0
    Environment:    <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from default-token-tgjdb (ro)
Conditions:
  Type          Status
  Initialized   True
  Ready         True
  PodScheduled  True
Volumes:
  default-token-tgjdb:
    Type:       Secret (a volume populated by a Secret)
    SecretName:  default-token-tgjdb
```

```
    Optional:    false
QoS Class:       BestEffort
Node-Selectors:  <none>
Tolerations:     <none>
Events:
  Type    Reason                  Age   From               Message
  ----    ------                  ----  ----               -------
  Normal  Scheduled               6m    default-scheduler  Successfully assigned kubernetes-bootcamp-7689dc585d-c4fnp to host01
  Normal  SuccessfulMountVolume   6m    kubelet, host01    MountVolume.SetUp succeeded for volume "default-token-tgjdb"
  Normal  Pulled                  6m    kubelet, host01    Container image "jocatalin/kubernetes-bootcamp:v2" already present on machine
  Normal  Created                 6m    kubelet, host01    Created container
  Normal  Started                 6m    kubelet, host01    Started container


Name:           kubernetes-bootcamp-7689dc585d-flfw6
Namespace:      default
Node:           host01/172.17.0.97
Start Time:     Tue, 06 Mar 2018 19:58:09 +0000
Labels:         pod-template-hash=3245871418
                run=kubernetes-bootcamp
Annotations:    <none>
Status:         Running
IP:             172.18.0.8
Controlled By:  ReplicaSet/kubernetes-bootcamp-7689dc585d
Containers:
  kubernetes-bootcamp:
    Container ID:   docker://33c17569e7a8cf85d2a9c3f7ef7a1817972868a253478239b578b371f6c25839
    Image:          jocatalin/kubernetes-bootcamp:v2
    Image ID:       docker-pullable://jocatalin/kubernetes-
bootcamp@sha256:fb1a3ced00cecfc1f83f18ab5cd14199e30adc1b49aa4244f5d65ad3f5feb2a5
    Port:           8080/TCP
```

```
    State:          Running
      Started:      Tue, 06 Mar 2018 19:58:10 +0000
    Ready:          True
    Restart Count:  0
    Environment:    <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from default-token-tgjdb (ro)
Conditions:
  Type          Status
  Initialized   True
  Ready         True
  PodScheduled  True
Volumes:
  default-token-tgjdb:
    Type:       Secret (a volume populated by a Secret)
    SecretName:  default-token-tgjdb
    Optional:   false
QoS Class:      BestEffort
Node-Selectors:  <none>
Tolerations:    <none>
Events:
  Type    Reason                  Age   From               Message
  ----    ------                  ----  ----               -------
  Normal  Scheduled               6m    default-scheduler  Successfully assigned kubernetes-bootcamp-7689dc585d-flfw6 to host01
  Normal  SuccessfulMountVolume  6m    kubelet, host01    MountVolume.SetUp succeeded for volume "default-token-tgjdb"
  Normal  Pulled                  6m    kubelet, host01    Container image "jocatalin/kubernetes-bootcamp:v2" already present on machine
  Normal  Created                 6m    kubelet, host01    Created container
  Normal  Started                 6m    kubelet, host01    Started container


Name:           kubernetes-bootcamp-7689dc585d-j74hc
```

```
Namespace:      default
Node:           host01/172.17.0.97
Start Time:     Tue, 06 Mar 2018 20:03:05 +0000
Labels:         pod-template-hash=3245871418
                run=kubernetes-bootcamp
Annotations:    <none>
Status:         Running
IP:             172.18.0.4
Controlled By:  ReplicaSet/kubernetes-bootcamp-7689dc585d
Containers:
  kubernetes-bootcamp:
    Container ID:   docker://8ea443cd5a7876fb5e201acebec70a2704a791e7d190a91434a5e860b6111c34
    Image:          jocatalin/kubernetes-bootcamp:v2
    Image ID:       docker-pullable://jocatalin/kubernetes-
bootcamp@sha256:fb1a3ced00cecfc1f83f18ab5cd14199e30adc1b49aa4244f5d65ad3f5feb2a5
    Port:           8080/TCP
    State:          Running
      Started:      Tue, 06 Mar 2018 20:03:06 +0000
    Ready:          True
    Restart Count:  0
    Environment:    <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from default-token-tgjdb (ro)
Conditions:
  Type          Status
  Initialized   True
  Ready         True
  PodScheduled  True
Volumes:
  default-token-tgjdb:
    Type:       Secret (a volume populated by a Secret)
```

```
    SecretName:  default-token-tgjdb
    Optional:    false
QoS Class:       BestEffort
Node-Selectors:  <none>
Tolerations:     <none>
Events:
  Type    Reason                  Age   From               Message
  ----    ------                  ----  ----               -------
  Normal  Scheduled               1m    default-scheduler  Successfully assigned kubernetes-bootcamp-7689dc585d-j74hc to host01
  Normal  SuccessfulMountVolume  1m    kubelet, host01    MountVolume.SetUp succeeded for volume "default-token-tgjdb"
  Normal  Pulled                  1m    kubelet, host01    Container image "jocatalin/kubernetes-bootcamp:v2" already present on machine
  Normal  Created                 1m    kubelet, host01    Created container
  Normal  Started                 1m    kubelet, host01    Started container


Name:           kubernetes-bootcamp-7689dc585d-k9kbh
Namespace:      default
Node:           host01/172.17.0.97
Start Time:     Tue, 06 Mar 2018 19:58:10 +0000
Labels:         pod-template-hash=3245871418
                run=kubernetes-bootcamp
Annotations:    <none>
Status:         Running
IP:             172.18.0.10
Controlled By:  ReplicaSet/kubernetes-bootcamp-7689dc585d
Containers:
  kubernetes-bootcamp:
    Container ID:   docker://c28adc5e033d061f4a4dfcaf3238776fcd2990b7c628cce4534568c630b40bdc
    Image:          jocatalin/kubernetes-bootcamp:v2
    Image ID:       docker-pullable://jocatalin/kubernetes-
bootcamp@sha256:fb1a3ced00cecfc1f83f18ab5cd14199e30adc1b49aa4244f5d65ad3f5feb2a5
```

```
    Port:           8080/TCP
    State:          Running
      Started:      Tue, 06 Mar 2018 19:58:12 +0000
    Ready:          True
    Restart Count:  0
    Environment:    <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from default-token-tgjdb (ro)
Conditions:
  Type          Status
  Initialized   True
  Ready         True
  PodScheduled  True
Volumes:
  default-token-tgjdb:
    Type:       Secret (a volume populated by a Secret)
    SecretName:  default-token-tgjdb
    Optional:   false
QoS Class:      BestEffort
Node-Selectors:  <none>
Tolerations:    <none>
Events:
  Type    Reason                  Age   From               Message
  ----    ------                  ----  ----               -------
  Normal  Scheduled               6m    default-scheduler  Successfully assigned kubernetes-bootcamp-7689dc585d-k9kbh to host01
  Normal  SuccessfulMountVolume  6m    kubelet, host01    MountVolume.SetUp succeeded for volume "default-token-tgjdb"
  Normal  Pulled                  6m    kubelet, host01    Container image "jocatalin/kubernetes-bootcamp:v2" already present on machine
  Normal  Created                 6m    kubelet, host01    Created container
  Normal  Started                 6m    kubelet, host01    Started container
```

```
$ kubectl set image deployments/kubernetes-bootcamp kubernetes-bootcamp=jocatalin/kubernetes-bootcamp:v2

$ kubectl get pods
NAME                                  READY     STATUS    RESTARTS   AGE
kubernetes-bootcamp-7689dc585d-c4fnp   1/1       Running   0          6m
kubernetes-bootcamp-7689dc585d-flfw6   1/1       Running   0          6m
kubernetes-bootcamp-7689dc585d-j74hc   1/1       Running   0          1m
kubernetes-bootcamp-7689dc585d-k9kbh   1/1       Running   0          6m

$ export NODE_PORT=$(kubectl get services/kubernetes-bootcamp -o go-template='{{(index .spec.ports 0).nodePort}}')
$ echo NODE_PORT=$NODE_PORT
NODE_PORT=32738

$ curl $(minikube ip):$NODE_PORT
Hello Kubernetes bootcamp! | Running on: kubernetes-bootcamp-7689dc585d-flfw6 | v=2

$ kubectl rollout status deployments/kubernetes-bootcamp
deployment "kubernetes-bootcamp" successfully rolled out

$ kubectl describe pods
Name:           kubernetes-bootcamp-7689dc585d-c4fnp
Namespace:      default
Node:           host01/172.17.0.97
Start Time:     Tue, 06 Mar 2018 19:58:09 +0000
Labels:         pod-template-hash=3245871418
                run=kubernetes-bootcamp
Annotations:    <none>
Status:         Running
IP:             172.18.0.9
Controlled By:  ReplicaSet/kubernetes-bootcamp-7689dc585d
Containers:
```

```
  kubernetes-bootcamp:
    Container ID:   docker://d6c1a0efdbb1810a009f3fd2c830dc89ccdf1217db24e721ae299845b026bbfe
    Image:          jocatalin/kubernetes-bootcamp:v2
    Image ID:       docker-pullable://jocatalin/kubernetes-
bootcamp@sha256:fb1a3ced00cecfc1f83f18ab5cd14199e30adc1b49aa4244f5d65ad3f5feb2a5
    Port:           8080/TCP
    State:          Running
      Started:      Tue, 06 Mar 2018 19:58:10 +0000
    Ready:          True
    Restart Count:  0
    Environment:    <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from default-token-tgjdb (ro)
Conditions:
  Type          Status
  Initialized   True
  Ready         True
  PodScheduled  True
Volumes:
  default-token-tgjdb:
    Type:       Secret (a volume populated by a Secret)
    SecretName:  default-token-tgjdb
    Optional:   false
QoS Class:      BestEffort
Node-Selectors:  <none>
Tolerations:     <none>
Events:
  Type    Reason                  Age   From               Message
  ----    ------                  ----  ----               -------
  Normal  Scheduled               7m    default-scheduler  Successfully assigned kubernetes-bootcamp-7689dc585d-c4fnp to host01
  Normal  SuccessfulMountVolume  7m    kubelet, host01    MountVolume.SetUp succeeded for volume "default-token-tgjdb"
```

```
  Normal  Pulled              7m    kubelet, host01    Container image "jocatalin/kubernetes-bootcamp:v2" already present on machine
  Normal  Created             7m    kubelet, host01    Created container
  Normal  Started             7m    kubelet, host01    Started container


Name:          kubernetes-bootcamp-7689dc585d-flfw6
Namespace:     default
Node:          host01/172.17.0.97
Start Time:    Tue, 06 Mar 2018 19:58:09 +0000
Labels:        pod-template-hash=3245871418
               run=kubernetes-bootcamp
Annotations:   <none>
Status:        Running
IP:            172.18.0.8
Controlled By:  ReplicaSet/kubernetes-bootcamp-7689dc585d
Containers:
  kubernetes-bootcamp:
    Container ID:   docker://33c17569e7a8cf85d2a9c3f7ef7a1817972868a253478239b578b371f6c25839
    Image:          jocatalin/kubernetes-bootcamp:v2
    Image ID:       docker-pullable://jocatalin/kubernetes-
bootcamp@sha256:fb1a3ced00cecfc1f83f18ab5cd14199e30adc1b49aa4244f5d65ad3f5feb2a5
    Port:           8080/TCP
    State:          Running
      Started:      Tue, 06 Mar 2018 19:58:10 +0000
    Ready:          True
    Restart Count:  0
    Environment:    <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from default-token-tgjdb (ro)
Conditions:
  Type          Status
```

```
  Initialized   True
  Ready         True
  PodScheduled  True
Volumes:
  default-token-tgjdb:
    Type:       Secret (a volume populated by a Secret)
    SecretName:  default-token-tgjdb
    Optional:   false
QoS Class:      BestEffort
Node-Selectors:  <none>
Tolerations:    <none>
Events:
  Type    Reason                 Age   From               Message
  ----    ------                 ----  ----               -------
  Normal  Scheduled              7m    default-scheduler  Successfully assigned kubernetes-bootcamp-7689dc585d-flfw6 to host01
  Normal  SuccessfulMountVolume  7m    kubelet, host01    MountVolume.SetUp succeeded for volume "default-token-tgjdb"
  Normal  Pulled                 7m    kubelet, host01    Container image "jocatalin/kubernetes-bootcamp:v2" already present on machine
  Normal  Created                7m    kubelet, host01    Created container
  Normal  Started                7m    kubelet, host01    Started container


Name:           kubernetes-bootcamp-7689dc585d-j74hc
Namespace:      default
Node:           host01/172.17.0.97
Start Time:     Tue, 06 Mar 2018 20:03:05 +0000
Labels:         pod-template-hash=3245871418
                run=kubernetes-bootcamp
Annotations:    <none>
Status:         Running
IP:             172.18.0.4
Controlled By:  ReplicaSet/kubernetes-bootcamp-7689dc585d
```

```
Containers:
  kubernetes-bootcamp:
    Container ID:   docker://8ea443cd5a7876fb5e201acebec70a2704a791e7d190a91434a5e860b6111c34
    Image:          jocatalin/kubernetes-bootcamp:v2
    Image ID:       docker-pullable://jocatalin/kubernetes-
bootcamp@sha256:fb1a3ced00cecfc1f83f18ab5cd14199e30adc1b49aa4244f5d65ad3f5feb2a5
    Port:           8080/TCP
    State:          Running
      Started:      Tue, 06 Mar 2018 20:03:06 +0000
    Ready:          True
    Restart Count:  0
    Environment:    <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from default-token-tgjdb (ro)
Conditions:
  Type          Status
  Initialized   True
  Ready         True
  PodScheduled  True
Volumes:
  default-token-tgjdb:
    Type:       Secret (a volume populated by a Secret)
    SecretName:  default-token-tgjdb
    Optional:   false
QoS Class:      BestEffort
Node-Selectors:  <none>
Tolerations:     <none>
Events:
  Type    Reason            Age   From            Message
  ----    ------            ----  ----            -------
  Normal  Scheduled         2m    default-scheduler  Successfully assigned kubernetes-bootcamp-7689dc585d-j74hc to host01
```

```
  Normal  SuccessfulMountVolume  2m    kubelet, host01    MountVolume.SetUp succeeded for volume "default-token-tgjdb"
  Normal  Pulled              2m    kubelet, host01    Container image "jocatalin/kubernetes-bootcamp:v2" already present on machine
  Normal  Created             2m    kubelet, host01    Created container
  Normal  Started             2m    kubelet, host01    Started container


Name:          kubernetes-bootcamp-7689dc585d-k9kbh
Namespace:     default
Node:          host01/172.17.0.97
Start Time:    Tue, 06 Mar 2018 19:58:10 +0000
Labels:        pod-template-hash=3245871418
               run=kubernetes-bootcamp
Annotations:   <none>
Status:        Running
IP:            172.18.0.10
Controlled By:  ReplicaSet/kubernetes-bootcamp-7689dc585d
Containers:
  kubernetes-bootcamp:
    Container ID:   docker://c28adc5e033d061f4a4dfcaf3238776fcd2990b7c628cce4534568c630b40bdc
    Image:          jocatalin/kubernetes-bootcamp:v2
    Image ID:       docker-pullable://jocatalin/kubernetes-
bootcamp@sha256:fb1a3ced00cecfc1f83f18ab5cd14199e30adc1b49aa4244f5d65ad3f5feb2a5
    Port:           8080/TCP
    State:          Running
      Started:      Tue, 06 Mar 2018 19:58:12 +0000
    Ready:          True
    Restart Count:  0
    Environment:    <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from default-token-tgjdb (ro)
Conditions:
```

```
  Type         Status
  Initialized   True
  Ready         True
  PodScheduled   True
Volumes:
  default-token-tgjdb:
    Type:       Secret (a volume populated by a Secret)
    SecretName:  default-token-tgjdb
    Optional:    false
QoS Class:      BestEffort
Node-Selectors:  <none>
Tolerations:     <none>
Events:
  Type    Reason             Age   From            Message
  ----    ------             ----  ----            -------
  Normal  Scheduled          7m    default-scheduler  Successfully assigned kubernetes-bootcamp-7689dc585d-k9kbh to host01
  Normal  SuccessfulMountVolume  7m    kubelet, host01    MountVolume.SetUp succeeded for volume "default-token-tgjdb"
  Normal  Pulled             7m    kubelet, host01    Container image "jocatalin/kubernetes-bootcamp:v2" already present on machine
  Normal  Created            7m    kubelet, host01    Created container
  Normal  Started            7m    kubelet, host01    Started container
```

$ kubectl set image deployments/kubernetes-bootcamp kubernetes-bootcamp=gcr.io/google-samples/kubernetes-bootcamp:v10

```
deployment "kubernetes-bootcamp" image updated
```

$ kubectl get deployments

```
NAME              DESIRED   CURRENT   UP-TO-DATE   AVAILABLE   AGE
kubernetes-bootcamp   4         5         2            3           15m
```

$ kubectl get pods

```
NAME                              READY     STATUS         RESTARTS   AGE
kubernetes-bootcamp-5569c6b8d6-7zzvm   0/1       ErrImagePull   0          6s
```

```
kubernetes-bootcamp-5569c6b8d6-8bm9r   0/1      ErrImagePull   0         6s
kubernetes-bootcamp-7689dc585d-c4fnp   1/1      Running        0         7m
kubernetes-bootcamp-7689dc585d-flfw6   1/1      Running        0         7m
kubernetes-bootcamp-7689dc585d-j74hc   1/1      Terminating    0         2m
kubernetes-bootcamp-7689dc585d-k9kbh   1/1      Running        0         7m
```

$ kubectl describe pods

```
Name:           kubernetes-bootcamp-5569c6b8d6-7zzvm
Namespace:      default
Node:           host01/172.17.0.97
Start Time:     Tue, 06 Mar 2018 20:05:23 +0000
Labels:         pod-template-hash=1125726482
                run=kubernetes-bootcamp
Annotations:    <none>
Status:         Pending
IP:             172.18.0.2
Controlled By:  ReplicaSet/kubernetes-bootcamp-5569c6b8d6
Containers:
  kubernetes-bootcamp:
    Container ID:
    Image:          gcr.io/google-samples/kubernetes-bootcamp:v10
    Image ID:
    Port:           8080/TCP
    State:          Waiting
      Reason:       ImagePullBackOff
    Ready:          False
    Restart Count:  0
    Environment:    <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from default-token-tgjdb (ro)
Conditions:
```

```
  Type          Status
  Initialized   True
  Ready         False
  PodScheduled  True
Volumes:
  default-token-tgjdb:
    Type:       Secret (a volume populated by a Secret)
    SecretName:  default-token-tgjdb
    Optional:   false
QoS Class:      BestEffort
Node-Selectors:  <none>
Tolerations:    <none>
Events:
  Type     Reason                 Age                From               Message
  ----     ------                 ----               ----               -------
  Normal   Scheduled              18s                default-scheduler  Successfully assigned kubernetes-bootcamp-5569c6b8d6-7zzvm to host01
  Normal   SuccessfulMountVolume  18s                kubelet, host01    MountVolume.SetUp succeeded for volume "default-token-tgjdb"
  Normal   BackOff                15s                kubelet, host01    Back-off pulling image "gcr.io/google-samples/kubernetes-bootcamp:v10"
  Warning  Failed                 15s                kubelet, host01    Error: ImagePullBackOff
  Normal   Pulling                2s (x2 over 17s)   kubelet, host01    pulling image "gcr.io/google-samples/kubernetes-bootcamp:v10"
  Warning  Failed                 1s (x2 over 16s)   kubelet, host01    Failed to pull image "gcr.io/google-samples/kubernetes-bootcamp:v10": rpc
error: code = Unknown desc = Error response from daemon: manifest for gcr.io/google-samples/kubernetes-bootcamp:v10 not found
  Warning  Failed                 1s (x2 over 16s)   kubelet, host01    Error: ErrImagePull


Name:           kubernetes-bootcamp-5569c6b8d6-8bm9r
Namespace:      default
Node:           host01/172.17.0.97
Start Time:     Tue, 06 Mar 2018 20:05:23 +0000
Labels:         pod-template-hash=1125726482
                run=kubernetes-bootcamp
```

```
Annotations:    <none>
Status:         Pending
IP:             172.18.0.3
Controlled By:  ReplicaSet/kubernetes-bootcamp-5569c6b8d6
Containers:
  kubernetes-bootcamp:
    Container ID:
    Image:          gcr.io/google-samples/kubernetes-bootcamp:v10
    Image ID:
    Port:           8080/TCP
    State:          Waiting
      Reason:       ErrImagePull
    Ready:          False
    Restart Count:  0
    Environment:    <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from default-token-tgjdb (ro)
Conditions:
  Type          Status
  Initialized   True
  Ready         False
  PodScheduled  True
Volumes:
  default-token-tgjdb:
    Type:       Secret (a volume populated by a Secret)
    SecretName:  default-token-tgjdb
    Optional:   false
QoS Class:      BestEffort
Node-Selectors:  <none>
Tolerations:     <none>
Events:
```

```
  Type     Reason                  Age   From               Message
  ----     ------                  ----  ----               -------
  Normal   Scheduled               18s   default-scheduler  Successfully assigned kubernetes-bootcamp-5569c6b8d6-8bm9r to host01
  Normal   SuccessfulMountVolume   17s   kubelet, host01    MountVolume.SetUp succeeded for volume "default-token-tgjdb"
  Normal   Pulling                 17s   kubelet, host01    pulling image "gcr.io/google-samples/kubernetes-bootcamp:v10"
  Warning  Failed                  15s   kubelet, host01    Failed to pull image "gcr.io/google-samples/kubernetes-bootcamp:v10": rpc error: code =
Unknown desc = Error response from daemon: manifest for gcr.io/google-samples/kubernetes-bootcamp:v10 not found
  Warning  Failed                  15s   kubelet, host01    Error: ErrImagePull
  Normal   BackOff                 14s   kubelet, host01    Back-off pulling image "gcr.io/google-samples/kubernetes-bootcamp:v10"
  Warning  Failed                  14s   kubelet, host01    Error: ImagePullBackOff


Name:           kubernetes-bootcamp-7689dc585d-c4fnp
Namespace:      default
Node:           host01/172.17.0.97
Start Time:     Tue, 06 Mar 2018 19:58:09 +0000
Labels:         pod-template-hash=3245871418
                run=kubernetes-bootcamp
Annotations:    <none>
Status:         Running
IP:             172.18.0.9
Controlled By:  ReplicaSet/kubernetes-bootcamp-7689dc585d
Containers:
  kubernetes-bootcamp:
    Container ID:   docker://d6c1a0efdbb1810a009f3fd2c830dc89ccdf1217db24e721ae299845b026bbfe
    Image:          jocatalin/kubernetes-bootcamp:v2
    Image ID:       docker-pullable://jocatalin/kubernetes-
bootcamp@sha256:fb1a3ced00cecfc1f83f18ab5cd14199e30adc1b49aa4244f5d65ad3f5feb2a5
    Port:           8080/TCP
    State:          Running
      Started:      Tue, 06 Mar 2018 19:58:10 +0000
```

```
    Ready:          True
    Restart Count:  0
    Environment:    <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from default-token-tgjdb (ro)
Conditions:
  Type          Status
  Initialized   True
  Ready         True
  PodScheduled  True
Volumes:
  default-token-tgjdb:
    Type:       Secret (a volume populated by a Secret)
    SecretName:  default-token-tgjdb
    Optional:   false
QoS Class:      BestEffort
Node-Selectors:  <none>
Tolerations:    <none>
Events:
  Type    Reason                  Age   From               Message
  ----    ------                  ----  ----               -------
  Normal  Scheduled               7m    default-scheduler  Successfully assigned kubernetes-bootcamp-7689dc585d-c4fnp to host01
  Normal  SuccessfulMountVolume  7m    kubelet, host01    MountVolume.SetUp succeeded for volume "default-token-tgjdb"
  Normal  Pulled                  7m    kubelet, host01    Container image "jocatalin/kubernetes-bootcamp:v2" already present on machine
  Normal  Created                 7m    kubelet, host01    Created container
  Normal  Started                 7m    kubelet, host01    Started container


Name:           kubernetes-bootcamp-7689dc585d-flfw6
Namespace:      default
Node:           host01/172.17.0.97
```

```
Start Time:     Tue, 06 Mar 2018 19:58:09 +0000
Labels:         pod-template-hash=3245871418
                run=kubernetes-bootcamp
Annotations:    <none>
Status:         Running
IP:             172.18.0.8
Controlled By:  ReplicaSet/kubernetes-bootcamp-7689dc585d
Containers:
  kubernetes-bootcamp:
    Container ID:   docker://33c17569e7a8cf85d2a9c3f7ef7a1817972868a253478239b578b371f6c25839
    Image:          jocatalin/kubernetes-bootcamp:v2
    Image ID:       docker-pullable://jocatalin/kubernetes-
bootcamp@sha256:fb1a3ced00cecfc1f83f18ab5cd14199e30adc1b49aa4244f5d65ad3f5feb2a5
    Port:           8080/TCP
    State:          Running
      Started:      Tue, 06 Mar 2018 19:58:10 +0000
    Ready:          True
    Restart Count:  0
    Environment:    <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from default-token-tgjdb (ro)
Conditions:
  Type          Status
  Initialized   True
  Ready         True
  PodScheduled  True
Volumes:
  default-token-tgjdb:
    Type:       Secret (a volume populated by a Secret)
    SecretName:  default-token-tgjdb
    Optional:   false
```

```
QoS Class:      BestEffort
Node-Selectors:  <none>
Tolerations:    <none>
Events:
  Type    Reason             Age   From            Message
  ----    ------             ----  ----            -------
  Normal  Scheduled          7m    default-scheduler  Successfully assigned kubernetes-bootcamp-7689dc585d-flfw6 to host01
  Normal  SuccessfulMountVolume  7m    kubelet, host01    MountVolume.SetUp succeeded for volume "default-token-tgjdb"
  Normal  Pulled             7m    kubelet, host01    Container image "jocatalin/kubernetes-bootcamp:v2" already present on machine
  Normal  Created            7m    kubelet, host01    Created container
  Normal  Started            7m    kubelet, host01    Started container


Name:                 kubernetes-bootcamp-7689dc585d-j74hc
Namespace:            default
Node:                 host01/172.17.0.97
Start Time:           Tue, 06 Mar 2018 20:03:05 +0000
Labels:               pod-template-hash=3245871418
                      run=kubernetes-bootcamp
Annotations:          <none>
Status:               Terminating (expires Tue, 06 Mar 2018 20:05:53 +0000)
Termination Grace Period:  30s
IP:                   172.18.0.4
Controlled By:        ReplicaSet/kubernetes-bootcamp-7689dc585d
Containers:
  kubernetes-bootcamp:
    Container ID:   docker://8ea443cd5a7876fb5e201acebec70a2704a791e7d190a91434a5e860b6111c34
    Image:          jocatalin/kubernetes-bootcamp:v2
    Image ID:       docker-pullable://jocatalin/kubernetes-
bootcamp@sha256:fb1a3ced00cecfc1f83f18ab5cd14199e30adc1b49aa4244f5d65ad3f5feb2a5
    Port:           8080/TCP
```

```
    State:          Running
      Started:      Tue, 06 Mar 2018 20:03:06 +0000
    Ready:          True
    Restart Count:  0
    Environment:    <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from default-token-tgjdb (ro)
Conditions:
  Type          Status
  Initialized   True
  Ready         True
  PodScheduled  True
Volumes:
  default-token-tgjdb:
    Type:       Secret (a volume populated by a Secret)
    SecretName:  default-token-tgjdb
    Optional:   false
QoS Class:      BestEffort
Node-Selectors:  <none>
Tolerations:    <none>
Events:
  Type    Reason                  Age   From               Message
  ----    ------                  ----  ----               -------
  Normal  Scheduled               2m    default-scheduler  Successfully assigned kubernetes-bootcamp-7689dc585d-j74hc to host01
  Normal  SuccessfulMountVolume  2m    kubelet, host01    MountVolume.SetUp succeeded for volume "default-token-tgjdb"
  Normal  Pulled                  2m    kubelet, host01    Container image "jocatalin/kubernetes-bootcamp:v2" already present on machine
  Normal  Created                 2m    kubelet, host01    Created container
  Normal  Started                 2m    kubelet, host01    Started container


Name:           kubernetes-bootcamp-7689dc585d-k9kbh
```

```
Namespace:      default
Node:           host01/172.17.0.97
Start Time:     Tue, 06 Mar 2018 19:58:10 +0000
Labels:         pod-template-hash=3245871418
                run=kubernetes-bootcamp
Annotations:    <none>
Status:         Running
IP:             172.18.0.10
Controlled By:  ReplicaSet/kubernetes-bootcamp-7689dc585d
Containers:
  kubernetes-bootcamp:
    Container ID:   docker://c28adc5e033d061f4a4dfcaf3238776fcd2990b7c628cce4534568c630b40bdc
    Image:          jocatalin/kubernetes-bootcamp:v2
    Image ID:       docker-pullable://jocatalin/kubernetes-
bootcamp@sha256:fb1a3ced00cecfc1f83f18ab5cd14199e30adc1b49aa4244f5d65ad3f5feb2a5
    Port:           8080/TCP
    State:          Running
      Started:      Tue, 06 Mar 2018 19:58:12 +0000
    Ready:          True
    Restart Count:  0
    Environment:    <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from default-token-tgjdb (ro)
Conditions:
  Type          Status
  Initialized   True
  Ready         True
  PodScheduled  True
Volumes:
  default-token-tgjdb:
    Type:       Secret (a volume populated by a Secret)
```

```
    SecretName:  default-token-tgjdb
    Optional:    false
QoS Class:       BestEffort
Node-Selectors:  <none>
Tolerations:     <none>
Events:
  Type    Reason                 Age   From               Message
  ----    ------                 ----  ----               -------
  Normal  Scheduled              7m    default-scheduler  Successfully assigned kubernetes-bootcamp-7689dc585d-k9kbh to host01
  Normal  SuccessfulMountVolume  7m    kubelet, host01    MountVolume.SetUp succeeded for volume "default-token-tgjdb"
  Normal  Pulled                 7m    kubelet, host01    Container image "jocatalin/kubernetes-bootcamp:v2" already present on machine
  Normal  Created                7m    kubelet, host01    Created container
  Normal  Started                7m    kubelet, host01    Started container
```

$ kubectl rollout undo deployments/kubernetes-bootcamp

```
deployment "kubernetes-bootcamp"
```

$ kubectl get pods

```
NAME                                  READY     STATUS        RESTARTS   AGE
kubernetes-bootcamp-5569c6b8d6-7zzvm   0/1       Terminating   0          24s
kubernetes-bootcamp-5569c6b8d6-8bm9r   0/1       Terminating   0          24s
kubernetes-bootcamp-7689dc585d-c4fnp   1/1       Running       0          7m
kubernetes-bootcamp-7689dc585d-flfw6   1/1       Running       0          7m
kubernetes-bootcamp-7689dc585d-j74hc   1/1       Terminating   0          2m
kubernetes-bootcamp-7689dc585d-k9kbh   1/1       Running       0          7m
kubernetes-bootcamp-7689dc585d-x8xks   1/1       Running       0          4s
```

$ kubectl get pods

```
NAME                                  READY     STATUS    RESTARTS   AGE
kubernetes-bootcamp-7689dc585d-c4fnp   1/1       Running   0          8m
kubernetes-bootcamp-7689dc585d-flfw6   1/1       Running   0          8m
```

```
kubernetes-bootcamp-7689dc585d-k9kbh   1/1      Running   0         8m
kubernetes-bootcamp-7689dc585d-x8xks   1/1      Running   0         28s
$
```

We see that the deployment is using a stable version of the app (v2). The Rollback was successful.