## Exercise 2.5: Create a Simple Deployment

Creating a pod does not take advantage of orchestration abilities of Kubernetes. We will now create a `Deployment` which gives us scalability, reliability, and updates.

1. Now run a containerized webserver **nginx**. Use **kubectl create** to create a simple, single replica deployment running the nginx web server. It will create a single pod as we did previously but with new controllers to ensure it runs as well as other features.

```
student@ckad-1:~$  kubectl create deployment firstpod --image=nginx
```

```
deployment.apps/firstpod created
```

2. Verify the new deployment exists and the desired number of pods matches the current number. Using a comma, you can request two resource types at once. The **Tab** key can be helpful. Type enough of the word to be unique and press the **Tab** key, it should complete the word. The deployment should show a number `1` for each value, such that the desired number of pods matches the up-to-date and running number. The pod should show zero restarts.

```
student@ckad-1:~$ kubectl get deployment,pod
```

```
NAME                       READY UP-TO-DATE AVAILABLE AGE
deployment.apps/firstpod   1/1   1          1         2m42s

NAME                           READY STATUS  RESTARTS AGE
pod/firstpod-7d88d7b6cf-lrsbk  1/1   Running 0        2m42s
```

3. View the details of the deployment, then the pod. Work through the output slowly. Knowing what a healthy deployment and looks like can be helpful when troubleshooting issues. Again the **Tab** key can be helpful when using long auto-generated object names. You should be able to type firstpod**Tab** and the name will complete when viewing the pod.

```
student@ckad-1:~$ kubectl describe deployment firstpod
```

```
Name:               firstpod
Namespace:          default
CreationTimestamp:  Wed, 15 Arp 2020 17:17:25 +0000
Labels:             app=firstpod
Annotations:        deployment.kubernetes.io/revision=1
Selector:           app=firstpod
Replicas:           1 desired | 1 updated | 1 total | 1 available....
StrategyType:       RollingUpdate
MinReadySeconds:    0
<output_omitted>
```

```
student@ckad-1:~$ kubectl describe pod firstpod-6bb4574d94-rqk76
```

```
Name:              firstpod-6bb4574d94-rqk76
Namespace:         default
Priority:          0
PriorityClassName: <none>
Node:              ckad-1/10.128.0.2
Start Time:        Wed, 15 Apr 2020 17:17:25 +0000
Labels:            pod-template-hash=2660130850
                   app=firstpod
```

```
Annotations:        cni.projectcalico.org/podIP: 192.168.200.65/32
Status:             Running
IP:                 192.168.200.65
Controlled By:      ReplicaSet/firstpod-6bb4574d94

<output_omitted>
```

4. Note that the resources are in the default namespace. Get a list of available namespaces.

```
student@ckad-1:~$ kubectl get namespaces
```

```
NAME              STATUS     AGE
default           Active     20m
kube-node-lease   Active     20m
kube-public       Active     20m
kube-system       Active     20m
```

5. There are four default namespaces. Look at the pods in the `kube-system` namespace.

```
student@ckad-1:~$ kubectl get pod -n kube-system
```

```
NAME                                   READY    STATUS     RESTARTS    AGE
calico-node-5ftrr                      2/2      Running    0           24m
calico-node-f7zrw                      2/2      Running    0           21m
coredns-fb8b8dccf-cmkds                1/1      Running    0           24m
coredns-fb8b8dccf-grltk                1/1      Running    0           24m
etcd-v141-r24p                         1/1      Running    0           23m
<output_omitted>
```

6. Now look at the pods in a namespace that does not exist. Note you do not receive an error.

```
student@ckad-1:~$ kubectl get pod -n fakenamespace
```

```
No resources found in fakenamespaces namespace.
```

7. You can also view resources in all namespaces at once. Use the `--all-namespaces` options to select objects in all namespaces at once.

```
student@ckad-1:~$ kubectl get pod --all-namespaces
```

```
NAMESPACE     NAME                           READY    STATUS     RESTARTS    AGE
default       firstpod-69cfdfd8d9-kj6ql      1/1      Running    0           44m
kube-system   calico-node-5ftrr              2/2      Running    0           92m
kube-system   calico-node-f7zrw              2/2      Running    0           89m
kube-system   coredns-fb8b8dccf-cmkds        1/1      Running    0           92m
<output_omitted>
```

8. View several resources at once. Note that most resources have a short name such as `rs` for ReplicaSet, `po` for Pod, `svc` for Service, and `ep` for endpoint. Note the endpoint still exists after we deleted the pod.

```
student@ckad-1:~$ kubectl get deploy,rs,po,svc,ep
```

```
NAME                       READY  UP-TO-DATE  AVAILABLE  AGE
deployment.apps/firstpod   1/1    1           1          4m

NAME                                    DESIRED    CURRENT    READY....
replicaset.apps/firstpod-6bb4574d94-rqk76   1          1          1 ....

NAME                           READY  STATUS     RESTARTS    AGE
pod/firstpod-6bb4574d94-rqk76  1/1    Running    0           4m
```

```

NAME                 TYPE       CLUSTER-IP    EXTERNAL-IP PORT(S)      AGE
service/basicservice NodePort   10.108.147.76 <none>      80:31601/TCP 21m
service/kubernetes   ClusterIP  10.96.0.1     <none>      443/TCP      21m

NAME                   ENDPOINTS         AGE
endpoints/basicservice <none>            21m
endpoints/kubernetes   10.128.0.3:6443   21m
```

9. Delete the `ReplicaSet` and view the resources again. Note that the age on the `ReplicaSet` and the pod it controls is now less than a minute of age. The deployment operator started a new `ReplicaSet` operator when we deleted the existing one. The new `ReplicaSet` started another pod when the desired spec did not match the current status.

```
student@ckad-1:~$ kubectl delete rs firstpod-6bb4574d94-rqk76
```

```
replicaset.apps "firstpod-6bb4574d94-rqk76" deleted
```

```
student@ckad-1:~$ kubectl get deployment,rs,po,svc,ep
```

```
NAME                            READY  UP-TO-DATE AVAILABLE AGE
deployment.apps/firstpod  1/1    1          1         7m

NAME                                          DESIRED   CURRENT....
replicaset.apps/firstpod-6bb4574d94-rqk76    1         1     ....

NAME                           READY     STATUS    RESTARTS    AGE
pod/firstpod-7d99ffc75-p9hbw   1/1       Running   0           12s

NAME                 TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
service/kubernetes   ClusterIP   10.96.0.1    <none>        443/TCP   24m

NAME                   ENDPOINTS         AGE
endpoints/kubernetes   10.128.0.2:6443   80m
endpoints/basicservice   <none>            21m
```

10. This time delete the top-level controller. After about 30 seconds for everything to shut down you should only see the cluster service and endpoint remain for the cluster and the service we created.

```
student@ckad-1:~$ kubectl delete deployment firstpod
```

```
deployment.apps "firstpod" deleted
```

```
student@ckad-1:~$ kubectl get deployment,rs,po,svc,ep
```

```
NAME                 TYPE      CLUSTER-IP    EXTERNAL-IP PORT(S)      AGE
service/basicservice NodePort  10.108.147.76 <none>      80:31601/TCP 35m
kubernetes           ClusterIP 10.96.0.1     <none>      443/TCP      24m

NAME                   ENDPOINTS         AGE
endpoints/basicservice <none>            21m
kubernetes             10.128.0.3:6443   24m
```

11. As we won't need it for a while, delete the `basicservice` service as well.

```
student@ckad-1:~$ kubectl delete svc basicservice
```

```
service "basicservice" deleted
```