Everything contained in Kubernetes is represented by a RESTful resource. We refer to these resources as Kubernetes objects. Each Kubernetes object exists at a unique HTTP path; for example, https://your-k8s.com/api/v1/namespaces/default/pods/my-pod leads to the representation of a Pod in the default namespace named my-pod. The kubectl command makes HTTP requests to these URLs to access the Kubernetes objects that reside at these paths.

Multiple Objects
The most basic command for viewing Kubernetes objects via kubectl is get. If you run kubectl get <resource-name>, you will get a listing of all resources in the current namespace. If you want to get a specific resource, you can use kubectl get <resource-name> <obj-name>.

The following command renders all Pods in the default namespace:

kubectl get pods

By default, kubectl uses a human-readable printer for viewing the responses from the API server, but this human-readable printer removes many of the details of the objects to fit each object on one terminal line. One way to get slightly more information is to add the -o wide flag, which gives more details, on a longer line.

The following command lists all Pods in the default namespace including additional information:

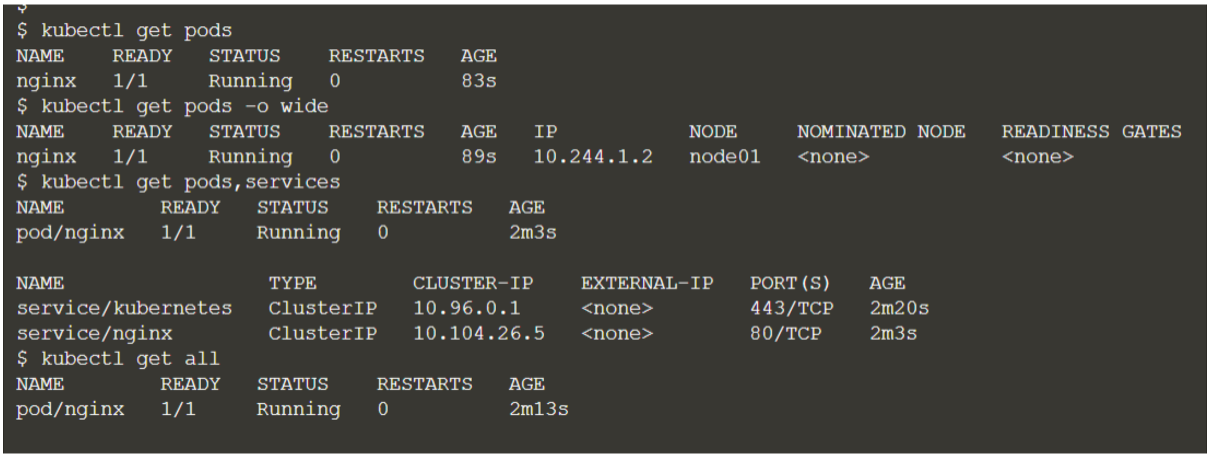
kubectl get pods -o wide

You can also view multiple objects of different types by using a comma-separated list of types—for example:

kubectl get pods,services

This will display all Pods and services for a given namespace. You can also use the wildcard notation all instead of listing individual API resource types.

kubectl get all



```
$ kubectl get pods
NAME    READY   STATUS    RESTARTS   AGE
nginx   1/1     Running   0          83s
$ kubectl get pods -o wide
NAME    READY   STATUS    RESTARTS   AGE   IP           NODE     NOMINATED NODE   READINESS GATES
nginx   1/1     Running   0          89s   10.244.1.2   node01   <none>           <none>
$ kubectl get pods,services
NAME        READY   STATUS    RESTARTS   AGE
pod/nginx   1/1     Running   0          2m3s

NAME                 TYPE        CLUSTER-IP    EXTERNAL-IP   PORT(S)   AGE
service/kubernetes   ClusterIP   10.96.0.1     <none>        443/TCP   2m20s
service/nginx        ClusterIP   10.104.26.5   <none>        80/TCP    2m3s
$ kubectl get all
NAME        READY   STATUS    RESTARTS   AGE
pod/nginx   1/1     Running   0          2m13s
```

Single Object
You can limit the information rendered by spelling out the object's name. Here, we are only viewing the Pod named nginx.

kubectl get pod nginx

If you want to view the complete object, you can also view the objects as raw JSON or YAML using the -o json or -o yaml flags, respectively.

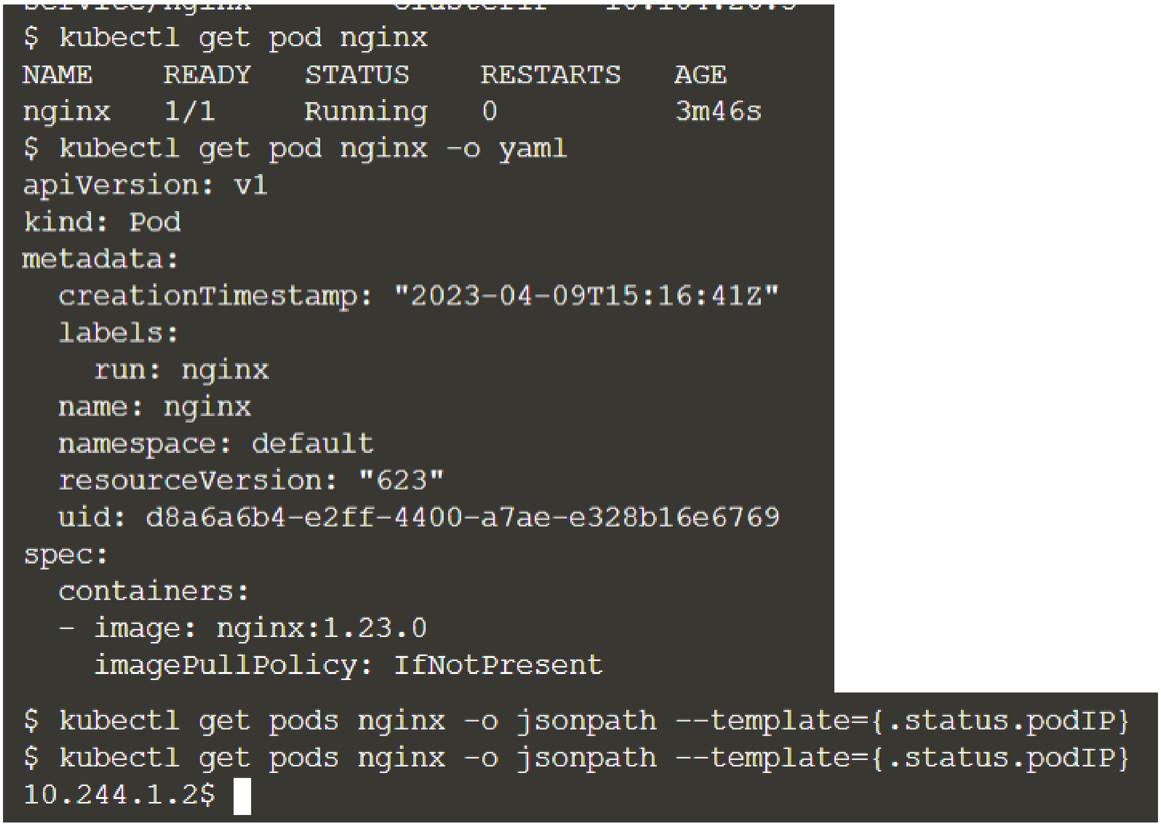
The following command renders the YAML representation of the Pod named nginx:

kubectl get pod nginx -o yaml

A common option for manipulating the output of kubectl is to remove the headers, which is often useful when combining kubectl with Unix pipes (e.g., kubectl ... | awk ...). If you specify the --no-headers flag, kubectl will skip the headers at the top of the human-readable table.

Another common task is extracting specific fields from the object. kubectl uses the JSONPath query language to select fields in the returned object. The complete details of JSONPath are beyond the scope of this lab, but as an example, this command will extract and print the IP address of the specified Pod:

kubectl get pods nginx -o jsonpath --template={.status.podIP}



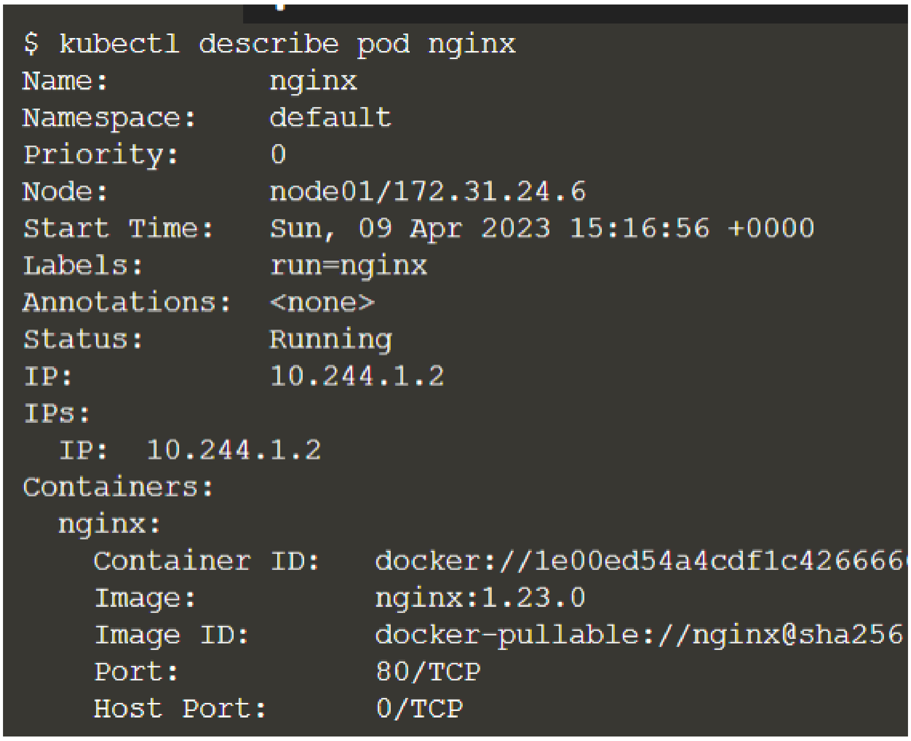
```
$ kubectl get pod nginx
NAME    READY   STATUS    RESTARTS   AGE
nginx   1/1     Running   0          3m46s
$ kubectl get pod nginx -o yaml
apiVersion: v1
kind: Pod
metadata:
  creationTimestamp: "2023-04-09T15:16:41Z"
  labels:
    run: nginx
  name: nginx
  namespace: default
  resourceVersion: "623"
  uid: d8a6a6b4-e2ff-4400-a7ae-e328b16e6769
spec:
  containers:
  - image: nginx:1.23.0
    imagePullPolicy: IfNotPresent
$ kubectl get pods nginx -o jsonpath --template={.status.podIP}
$ kubectl get pods nginx -o jsonpath --template={.status.podIP}
10.244.1.2$
```

Human-Readable Details
If you are interested in more detailed information about a particular object, use the describe command. The format looks as follows: kubectl describe <resource-name> <obj-name>.

The following command renders the details of the Pod named nginx:

kubectl describe pod nginx

This will provide a rich multiline human-readable description of the object as well as any other relevant, related objects and events in the Kubernetes cluster.



```
$ kubectl describe pod nginx
Name:          nginx
Namespace:     default
Priority:      0
Node:          node01/172.31.24.6
Start Time:    Sun, 09 Apr 2023 15:16:56 +0000
Labels:        run=nginx
Annotations:   <none>
Status:        Running
IP:            10.244.1.2
IPs:
  IP:  10.244.1.2
Containers:
  nginx:
    Container ID:   docker://1e00ed54a4cdf1c426666
    Image:          nginx:1.23.0
    Image ID:       docker-pullable://nginx@sha256
    Port:           80/TCP
    Host Port:      0/TCP
```

Supported Fields of an API Resource
If you would like to see a list of supported fields for each supported type of Kubernetes object, you can use the explain command:

kubectl explain pods

Sometimes you want to continually observe the state of a particular Kubernetes resource to see changes to the resource when they occur. For example, you might be waiting for your application to restart. The --watch flag enables this. You can add this flag to any kubectl get command to continuously monitor the state of a particular resource.

kubectl offers a variety of ways to list multiple objects or view their details. Details of each object can be viewed as a YAML, JSON, or textual representation.