

When building applications in Kubernetes, it is often necessary to provide some configuration for your containers. In this lesson we will discuss some frequently-used container configuration options: `command`, `args`, and `containerPort`. After completing this lesson, you will have a basic understanding of some the ways in which Kubernetes allows you to customize how your containers are run within a pod.

Relevant Documentation

- <https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/>

Lesson Reference

You can specify custom commands for your containers.

```
apiVersion: v1
kind: Pod
metadata:
  name: my-command-pod
  labels:
    app: myapp
spec:
  containers:
  - name: myapp-container
    image: busybox
    command: ['echo']
    restartPolicy: Never
```

You can also add custom arguments like so:

```
apiVersion: v1
kind: Pod
metadata:
  name: my-args-pod
  labels:
    app: myapp
spec:
  containers:
  - name: myapp-container
    image: busybox
    command: ['echo']
    args: ['This is my custom argument']
    restartPolicy: Never
```

Here is a pod with a `containerPort`:

```
apiVersion: v1
kind: Pod
metadata:
  name: my-containerport-pod
  labels:
    app: myapp
spec:
  containers:
  - name: myapp-container
    image: nginx
    ports:
    - containerPort: 80
```

You can check the status of your pods at any time with `kubectl get pods`.