**Getting a shell to a Container**

In this exercise, you create a Pod that has one Container. The Container runs the nginx image. Here is the configuration file for the Pod:

| [**application/shell-demo.yaml**](https://raw.githubusercontent.com/kubernetes/website/master/content/en/examples/application/shell-demo.yaml) |
| --- |
| apiVersion: v1  kind: Pod  metadata:  name: shell-demo  spec:  volumes:  - name: shared-data  emptyDir: {}  containers:  - name: nginx  image: nginx  volumeMounts:  - name: shared-data  mountPath: /usr/share/nginx/html |

Create the Pod:

kubectl apply -f shell-demo.yaml

Verify that the Container is running:

kubectl get pod shell-demo

Get a shell to the running Container:

kubectl exec -it shell-demo -- /bin/bash

**Note:** The double dash symbol “–” is used to separate the arguments you want to pass to the command from the kubectl arguments.

In your shell, list the root directory:

root@shell-demo:/# ls /

In your shell, experiment with other commands. Here are some examples:

root@shell-demo:/# ls /

root@shell-demo:/# cat /proc/mounts

root@shell-demo:/# cat /proc/1/maps

root@shell-demo:/# apt-get update

root@shell-demo:/# apt-get install -y tcpdump

root@shell-demo:/# tcpdump

root@shell-demo:/# apt-get install -y lsof

root@shell-demo:/# lsof

root@shell-demo:/# apt-get install -y procps

root@shell-demo:/# ps aux

root@shell-demo:/# ps aux | grep nginx

**Writing the root page for nginx**

Look again at the configuration file for your Pod. The Pod has an emptyDir volume, and the Container mounts the volume at /usr/share/nginx/html.

In your shell, create an index.html file in the /usr/share/nginx/html directory:

root@shell-demo:/# echo Hello shell demo > /usr/share/nginx/html/index.html

In your shell, send a GET request to the nginx server:

root@shell-demo:/# apt-get update

root@shell-demo:/# apt-get install curl

root@shell-demo:/# curl localhost

The output shows the text that you wrote to the index.html file:

Hello shell demo

When you are finished with your shell, enter exit.

**Running individual commands in a Container**

In an ordinary command window, not your shell, list the environment variables in the running Container:

kubectl exec shell-demo env

Experiment running other commands. Here are some examples:

kubectl exec shell-demo ps aux

kubectl exec shell-demo ls /

kubectl exec shell-demo cat /proc/1/mounts

**Opening a shell when a Pod has more than one Container**

If a Pod has more than one Container, use --container or -c to specify a Container in the kubectl exec command. For example, suppose you have a Pod named my-pod, and the Pod has two containers named main-app and helper-app. The following command would open a shell to the main-app Container.

kubectl exec -it my-pod --container main-app -- /bin/bash