Starter Labs (Python)

WORKSHOP MODULES

Remote Access to Your Application

Containers are treated as immutable infrastructure and therefore it is generally not recommended to modify the content of a container through SSH or running custom commands inside the container. Nevertheless, in some use-cases, such as debugging an application, it might be beneficial to get into a container and inspect the application.

Exercise: Remote Shell Session to a Container Using the CLI

OpenShift allows establishing remote shell sessions to a container without the need to run an SSH service inside each container. In order to establish an interactive session inside a container, you can use the `oc rsh` command. First get the list of available pods:

```
oc get pods
```

You should an output similar to the following:

```
NAME                         READY   STATUS    RESTARTS   AGE
parksmap-65c4f8b676-fxcrq    1/1     Running   0          52m
```

Now you can establish a remote shell session into the pod by using the pod name:

```
oc rsh parksmap-65c4f8b676-fxcrq
```

You would see the following output:

```
sh-4.2$
```

> The default shell used by `oc rsh` is `/bin/sh`. If the deployed container does not have **sh** installed and uses another shell, (e.g. **A Shell**) the shell command can be specified after the pod name in the issued command.

Run the following command to list the files in the top folder:

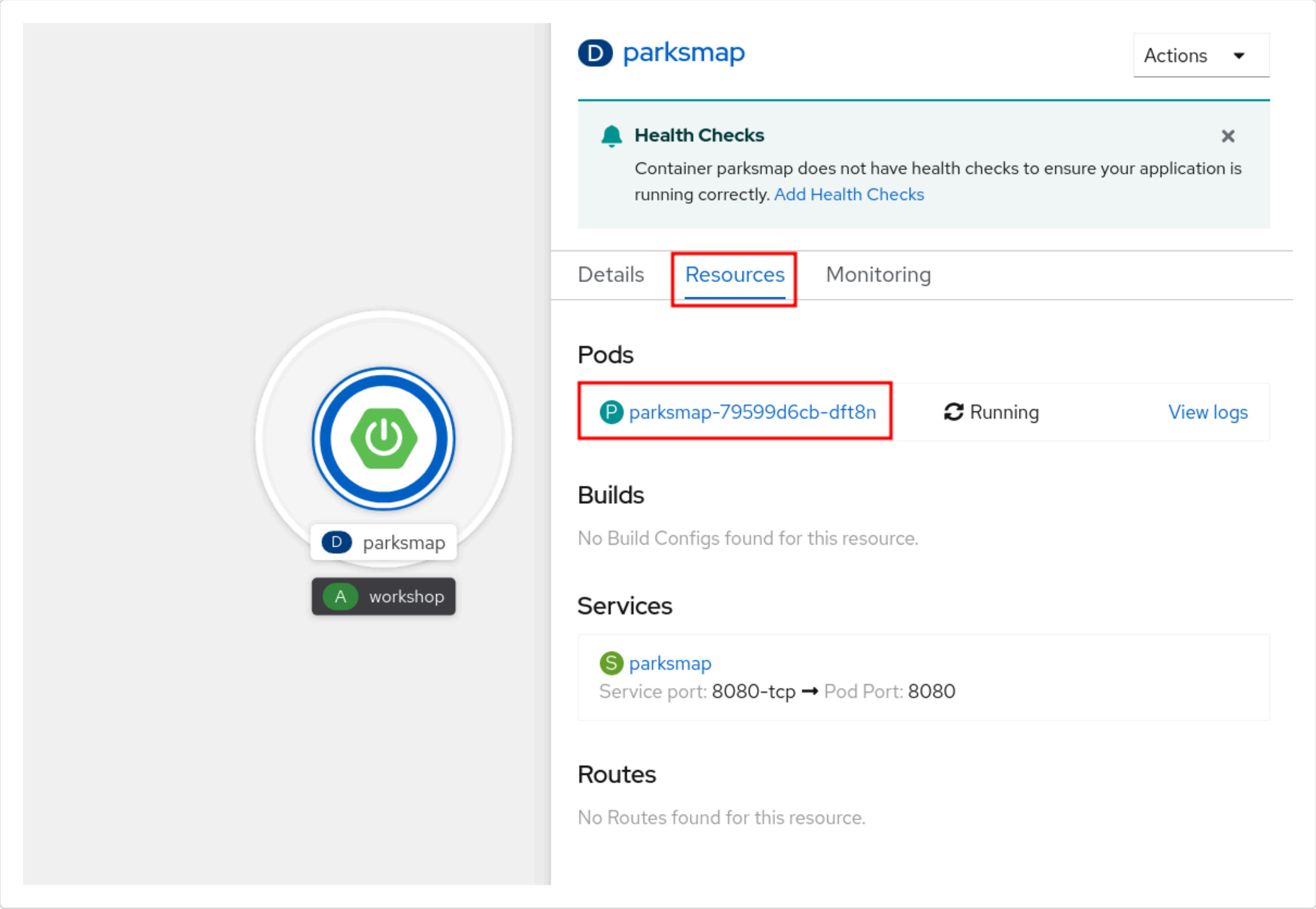
```
ls /
```

```
anaconda-post.log  bin  dev  etc  home  lib  lib64  lost+found  media  mnt  opt  parksmap.jar  proc
root  run  sbin  srv  sys  tmp  usr  var
```

Exercise: Remote Shell Session to a Container Using the Web Console

The OpenShift Web Console also provides a convenient way to access a terminal session on the container without having to use the CLI.

In order to access a pod’s terminal via the Web Console, go to the Topology view in the Developer Perspective, click the `parksmap` entry, and then click on the **Pod**.



Health Checks

Container parksmap does not have health checks to ensure your application is running correctly. Add Health Checks

Pods

P parksmap-79599d6cb-dft8n Running View logs

Builds

No Build Configs found for this resource.

Services

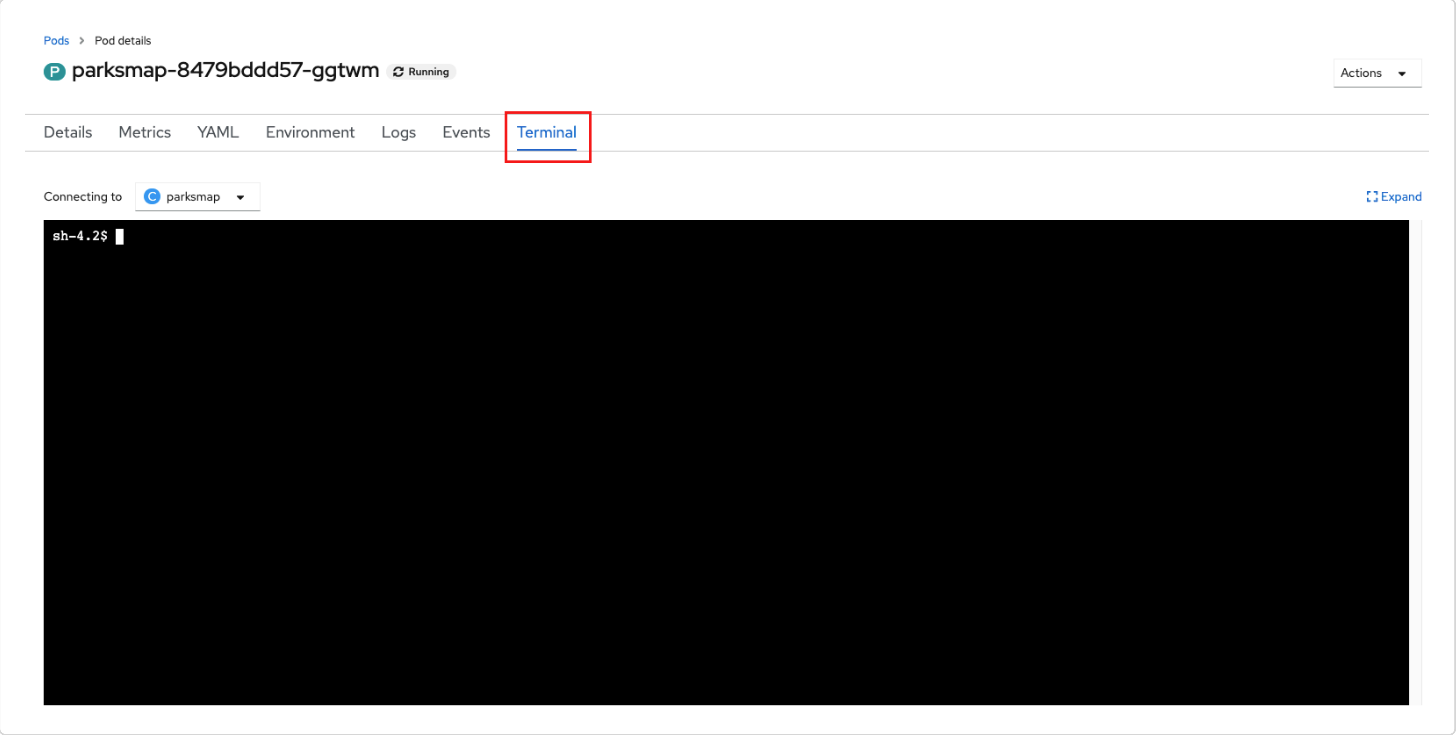
S parksmap

Service port: 8080-tcp ➞ Pod Port: 8080

Routes

No Routes found for this resource.

Once you are viewing the information for the selected pod, click on the **Terminal** tab to open up a shell session.



```
sh-4.2$
```

Go ahead and execute the same commands you did when using the CLI to see how the Web Console based terminal behaves.

Before proceeding, close the connection to the pod.

```
exit
```

Exercise: Execute a Command in a Container

In addition to remote shell, it is also possible to run a command remotely in an already running container using the `oc exec` command. This does not require that a shell is installed, but only that the desired command is present and in the executable path.

In order to show just the JAR file, run the following:

```
oc exec parksmap-2-mcjsw -- ls -l /parksmap.jar
```

You would see something like the following:

```
-rw-r--r--. 1 root root 39138901 Apr  1 16:54 /parksmap.jar
```

> The `--` syntax in the `oc exec` command delineates where exec's options end and where the actual command to execute begins. Take a look at `oc exec --help` for more details.

You can also specify the shell commands to run directly with the **oc rsh** command:

```
oc rsh parksmap-2-mcjsw whoami
```

You would see something like:

```
1000580000
```

> It is important to understand that, for security reasons, OpenShift does not run containers as the user specified in the Dockerfile by default. In fact, when OpenShift launches a container its user is actually randomized.
>
> If you want or need to allow OpenShift users to deploy container images that do expect to run as root (or any specific user), a small configuration change is needed. You can learn more about the container image guidelines for OpenShift.

Continue