# ProLUG – Docker and K3s

**Required Materials**

Rocky 9 or equivalent

Root or sudo command access

**EXERCISES (Warmup and quick review)**

1. which podman
2. dnf whatprovides podman
3. rpm -qi podman
   When was this installed?
   What version is it?
   Why might this be important to know?
4. podman images
5. podman ps
   What do you learn from those two commands?
   Why might it be important to know on a system?

**Building and running containers**

Your tasks in this lab are designed to get you thinking about how container deployments interact with our Linux systems that we support.

1. Pull and run a container

```
podman run -dt -p 8080:80/tcp docker.io/library/httpd
```

What do you see on your screen as this happens?

2. Check your images again (from your earlier exercises)

```
podman images
```

Is there a new image, and if so, what do you notice about it?

3. Check your podman running containers

```
podman ps
```

What appears to be happening? Can you validate this with your Linux knowledge?

```
ss -ntulp
curl 127.0.0.1:8080
```

4. Inspect the running pod

```
podman inspect -l
```

What format is the output in?
What important information might you want from this in the future?

```
podman logs -l
```

What info do you see in the logs?
Do you see your connection attempt from earlier? What is the return code and why is that important for troubleshooting?

```
podman top -l
```

What processes is the pod running?
What other useful information might you find here?
Why might it be good to know the user being run within the pod?

5. Stop the pod by its name

```
podman stop <podname>
```

Can you verify it is stopped from your previous commands?

```
podman ps
ss -ntulp
curl 127.0.0.1:8080
```

Does the container still exist? Why might you want to know this?

```
podman image
```

**Build an application in a container**

The ProLUG lab will already have a version of this setup for you to copy and run. If you are in a different environment, follow https://docs.docker.com/build/concepts/dockerfile/ for the general same steps.

1. Setup your lab environment

```
[root@rocky11 stream]# cd /lab_work/
[root@rocky11 lab_work]# ls
[root@rocky11 lab_work]# mkdir scott_lab9
[root@rocky11 lab_work]# cd scott_lab9/
[root@rocky11 scott_lab9]# ls
[root@rocky11 scott_lab9]# cp /labs/lab9.tar.gz .
[root@rocky11 scott_lab9]# tar -xzvf lab9.tar.gz
```

```
lab9/
lab9/Dockerfile
lab9/hello.py
[root@rocky11 scott_lab9]# ls
lab9  lab9.tar.gz
[root@rocky11 scott_lab9]# cd lab9
[root@rocky11 lab9]# pwd
/lab_work/scott_lab9/lab9
[root@rocky11 lab9]# ls
Dockerfile  hello.py
```

2. Create a docker image from the docker file:

```
time podman build -t scott_hello .
#Use your name
```

What output to your screen do you see as you build this?
Approximately how long did it take?

If this breaks in the lab, how might you fix it? What do you suspect?

3. Verify that you have built the container

```
podman images
```

4. Run the container as a daemon

```
podman run -dt localhost/scott_example
```

5. Verify the name and that it is running

```
podman ps
```

6. Exec into the pod and see that you are on the ubuntu container

```
podman exec -it festive_pascal sh
cat /etc/*release
exit
```

**Conclusion**

There are a lot of ways to use these tools. There are a lot of ways you will support them. At the end of the day you're a Linux System Administrator, you're expected to understand everything that goes on in your system. To this end, we want to know the build process and run processes so we can help the engineers we support keep working in a Linux environment.

Notes and resources used:

https://podman.io/docs
https://docs.docker.com/build/concepts/dockerfile/
https://docs.podman.io/en/latest/markdown/podman-exec.1.html