A CLOUD GURU

Lesson Guide - Creating Your First Podman Pod

Now that we have an understanding of what a pod is, we're going to put that knowledge to use. In this lesson we're going to create our first pod using Podman. Upon completion of this lesson, you will be able to create a pod using Podman.

Resources

Moving from docker-compose to Podman pods - Red Hat - Enable Sysadmin

Podman: Managing pods and containers in a local container runtime - Red Hat Developer

Instructions

Let's create our first Podman pod!

The best way to get experience with pods is to get hands-on and do the things! We're going to roll up our sleeves and take a look at how to launch our first pod.

Onward!

Commands Covered

- `podman pod`: manages Podman pods
- `podman run`: runs a command in a new container from the given image
- `podman ps`: displays information about containers

Creating Our First Podman Pod!

We're going to stand up a WordPress instance in a pod. Let's see how it's done!

Let's start by checking for exising containers and pods:

```
podman ps -a --pod
```

We can use the `--pod` option to show the ID and name of the pod the containers belong to.

We can display pods using:

```
podman pod ls
```

We'll start by creating our pod. Remember, we want to publish port `80` in the pod to `8080` on the host. We want to name the pod `wp-pod`.

To do this, we run:

```
podman pod create --name wp-pod -p 8080:80
```

Checking for pods now:

```
podman pod ls
```

We can see our pod!

Checking for containers:

```
podman ps -a --pod
```

We now see our Infra container! You can see that the pod's published ports are listed along with the container.

First, let's start the `mariadb` container:

```
podman run -d --restart=always --pod=wp-pod -e
MYSQL_ROOT_PASSWORD="dbpass" -e MYSQL_DATABASE="wp" -e
MYSQL_USER="wordpress" -e MYSQL_PASSWORD="wppass" --name=wp-db mariadb
```

Checking for containers again:

```
podman ps -a --pod
```

We can see our `mariadb` container!

Next, we'll start the WordPress container:

```
podman run -d --restart=always --pod=wp-pod -e WORDPRESS_DB_NAME="wp" -e
WORDPRESS_DB_USER="wordpress" -e WORDPRESS_DB_PASSWORD="wppass" -e
WORDPRESS_DB_HOST="127.0.0.1" --name wp-web wordpress
```

Checking for containers again:

```
podman ps -a --pod
```

We can see our WordPress container now. Our WordPress pod is fully running!

Checking with a `curl` command:

```
curl -s http://localhost:8080
```

```
echo $?
```

We can see that the WordPress login page is working! We can now log in with our web browser, using port 8080.

Congratulations! We just ran our first pod!

Notes

Recording - Environment used: Cloud Playground - Medium 3 unit RHEL 8 Cloud Server

Environment Setup:

Create your Cloud Playground server and log in.

Install the `container-tools` Application Stream:

```
sudo yum -y module install container-tools
```

You're ready to go!