A CLOUD GURU

Lesson Guide - Working with Container Images Using Podman and Skopeo - Part 1

Container images are a key part of Podman. In this lesson, we'll examine how to manage container images using Podman. Upon completion of this lesson, you will have a solid understanding of how and when to use Podman to manage your container images.

Resources

Building, Running, and Managing Linux Containers on Red Hat Enterprise Linux 8 - Working with Container Images

Container Registry Information:

What Is a Container Registry? - Red Hat

How to Implement a Simple Personal/Private Linux Container Image Registry for Internal Use - Red Hat

Instructions

We need to get our container images in order!

Before we start working with Podman containers, we're going to want some container images to work with. We're going to take a look at how we can use `podman` and `skopeo` to manage container images. We'll explore how we can use `podman` and `skopeo` to retrieve container images, display image information, and more.

Let's check it out!

Managing Container Images Using Podman

The `podman` command comes with quite a bit of container management functionality included. Let's take a look!

There are a couple of helpful `podman` commands that will help keep you grounded.

If you need to check the version of `podman` you're running, use:

```
podman --version
```

Everyone needs a little help from time to time, and you can get it with:

```
podman --help
```

You might want more information on a `podman` subcommand:

```
podman image --help
```

Let's grab an image to work with. We'll search for latest the Red Hat `ubi` image:

```
podman search ubi:latest | more
```

This will search the configured registries for the `ubi:latest` image.

We'll grab the latest `ubi8` image from the `registry.access.redhat.com` registry:

```
podman pull registry.access.redhat.com/ubi8/ubi:latest
```

Sometimes `podman` will have commands that do the same thing as another command, such as listing container images:

```
podman images
```

```
podman image list
```

Or, drilling down into a container image's metadata:

```
podman inspect ubi | more
```

```
podman image inspect ubi | more
```

I encourage you to use the `--help` switch to drill down into the commands and explore all the possibilities!

We may want to add a tag to a container image to give it another name, such as `ubi8` for our existing `ubi:latest` image:

```
podman tag ubi:latest ubi8
```

Taking a look at our images:

```
podman images
```

We can see the container image registry that's running on the server using `podman ps`:

```
sudo podman ps -a
```

The registry itself is a Podman container, running on the lab server!

Let's log in to the local registry, using `registryuser` and `registryuserpassword`:

```
podman login localhost:5000
```

We're going to push the `ubi:latest` image in our local container image storage to the registry running on our lab server. First we need to tag the image:

```
podman tag registry.access.redhat.com/ubi8/ubi:latest
localhost:5000/ubi8/ubi
```

Next, we'll push the image to our local registry:

```
podman push localhost:5000/ubi8/ubi:latest
```

Checking the contents of the local container registry, using `curl`:

```
curl -u registryuser:registryuserpassword
https://localhost:5000/v2/_catalog
```

We can see the repository for `ubi8/ubi`.

Let's remove the `ubi8` tag we assigned earlier:

```
podman untag ubi8
```

Checking our local container images:

```
podman image list
```

Notice that we now have an untagged image. Let's remove it by using the *imageID*:

```
podman rmi <imageID>
```

Checking our local container images:

```
podman image list
```

We're all cleaned up now! Let's log out of our local container image repository:

```
podman logout localhost:5000
```

Now that we've taken a high-level look at how we can use the `podman` command to manage container images, let's move on to `skopeo`. Again, I encourage you to use the `--help` switch to drill down into the `podman` commands and explore all the possibilities!

Notes

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

Environment Setup:

Create your Cloud Playground server and log in.

Install Podman and Configure a Container Registry:

Install, enable, and start firewall services:

```
sudo yum -y install firewall* skopeo
```

```
sudo systemctl enable --now firewalld
```

Set up a container registry:

How to Implement a Simple Personal/Private Linux Container Image Registry for Internal Use - Red Hat

I used `localhost` for the hostname, as I'm going to access the container registry using `localhost`.

You're ready to go!