

RHCA - EX180/EX280

Red Hat Certified Specialist in Containers and Kubernetes

Notes, tips and tricks for the EX180 and EX280 exams.

Podman host setup

```
dnf module install container-tools
```

```
dnf install -y buildah
```

Podman basics

Registry file: */etc/containers/registries.conf*

Login to a registry

```
podman login registry.access.redhat.com
```

Search for images

```
podman search mariadb
```

Inspect images without downloading

```
skopeo inspect docker://registry.access.redhat.com/rhsc1/mariadb-102-rhel7
```

Download images

```
podman pull registry.access.redhat.com/rhsc1/mariadb-102-rhel7
```

List images

```
podman images
```

Inspect images. *Useful for locating config.user

```
podman inspect registry.access.redhat.com/rhsc1/mariadb-102-rhel7:latest
```

Configure container volume path

```
sudo mkdir /srv/mariadb
sudo chown -R 27:27 /srv/mariadb # UID found in podman inspect
sudo semanage fcontext -a -t container_file_t "/srv/mariadb(/.*)"
sudo restorecon -Rv /srv/mariadb
```

Run image

-d detached

-e per variable

-p local_port:container_port

-v local/path:/path/in/pod

```
podman run -d -e MYSQL_USER=user \
-e MYSQL_PASSWORD=pass -e MYSQL_DATABASE=db \
-p 33306:3306 \
-v /srv/mariadb:/var/lib/mysql:Z \ # :Z isn't needed if SELinux manually configured
rhsc1/mariadb-102-rhel7
```

Basic pod status and names

```
podman ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
c917919d6745	registry.access.redhat.com/rhsc1/mariadb-102-rhel7:latest	run-mysqld	5 minutes ago	Up 5 minutes ago	0.0.0.0:3306->3306/tcp	heuristic stonebraker

Enter container in interactive shell

```
podman exec -it container-name /bin/bash
```

Commit changes to running image

```
podman commit container-name image-name
```

Export image

```
podman save image-name > /path/to/image.tar
```

Remove images

```
podman rmi image-name --force
```

Restore/Load image

```
podman load -i /path/to/image.tar
```

Dockerfile Basics

Container for scanning a network.

```
# Start with a base image
FROM registry.access.redhat.com/ubi8

# Maintainer information
MAINTAINER ClusterApps <mail@clusterapps.com>

# Run commands to build the container
# Do as much with the fewest RUN lines
RUN yum --assumeyes update && \
    yum --assumeyes install \
    nmap iproute procps-ng && \
    bash && \
    yum clean all

# Entrypoint is the command that run when the container starts
ENTRYPOINT ["/usr/bin/nmap"]

# The arguments for the entrypoint
CMD ["-sn", "192.168.252.0/24"]
```

Build the image with a tag

```
podman build -t notes180 .
```

List images

```
podman images
```

Run the image

```
podman run localhost/notes180
```

OpenShift Basics

Setup oc completion

```
source <(oc completion bash)
```

Create a new project

```
oc new-project nginx
```

Create a new application based on an image

```
oc new-app bitnami/nginx
```

Basic information about the project

```
oc status
```

```
[p1@rhelcvm01 ~]$ oc status
In project nginx on server https://api.crc.testing:6443

svc/nginx - 10.217.5.45 ports 8080, 8443
  deployment/nginx deploys istag/nginx:latest
    deployment #2 running for 20 seconds - 1 pod
    deployment #1 deployed 20 seconds ago
```

Get more detailed information about the project

```
oc get all
```

```
NAME                                READY   STATUS    RESTARTS   AGE
pod/nginx-84654f9574-9gpnt          1/1     Running   0           3m13s

NAME                                TYPE          CLUSTER-IP   EXTERNAL-IP   PORT(S)          AGE
service/nginx                        ClusterIP     10.217.5.45  <none>        8080/TCP,8443/TCP 3m13s

NAME                                READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/nginx               1/1     1             1           3m13s

NAME                                DESIRED   CURRENT   READY   AGE
replicaset.apps/nginx-65fbb4c68d    0         0         0       3m13s
replicaset.apps/nginx-84654f9574    1         1         1       3m13s

NAME                                IMAGE REPOSITORY                                  TAGS   UPDATED
imagestream.image.openshift.io/nginx default-route-openshift-image-registry.apps-crc.testing/nginx/nginx latest 3 minutes ago
```

Get full details about a pod

```
oc describe pod/nginx-84654f9574-9gpnt
```

Get full details about a deployment

```
oc describe deployment.apps/nginx
```

Application deployments

Generate a YAML file to use as a base.

```
oc create deployment nginx --image=bitnami/nginx --dry-run=client -o yaml > newapp.yml
```

Create a temporary deployment using the YAML file,

```
oc create -f newapp.yml
```

Create the service by exposing the deployment

```
# dry run to add to YAML file
oc expose deployment --port=8080 --dry-run=client nginx -o yaml >> newapp.yml

# run to create the service
oc expose deployment --port=8080 nginx
```

Expose the service

```
oc expose svc nginx --dry-run=client -o yaml >> newapp.yml
```

New app with parameters, as a deployment-config, and with labels (app=database)

```
oc new-app --name mdb -l app=database -e MYSQL_USER=dbuser -e
MYSQL_PASSWORD=SuperAwesomePassword -e MYSQL_DATABASE=dbname bitnami/mysql --as-deployment-
config
```

Delete the temporary application

```
oc delete all --all
```

Edit the YAML file and break up the sections into proper YAML.

Add `---` at the beginning of a section

Add `...` at the end of a section.

The new nginx app YAML file.

```
---
apiVersion: apps/v1
kind: Deployment
metadata:
  creationTimestamp: null
  labels:
    app: nginx
  name: nginx
spec:
  replicas: 1
  selector:
    matchLabels:
      app: nginx
  strategy: {}
  template:
    metadata:
      creationTimestamp: null
      labels:
        app: nginx
    spec:
      containers:
      - image: bitnami/nginx
        name: nginx
        resources: {}
status: {}
...
```

```
apiVersion: v1
kind: Service
metadata:
  creationTimestamp: null
  labels:
    app: nginx
  name: nginx
spec:
  ports:
```

- port: 8080
protocol: TCP
targetPort: 8080
selector:
app: nginx
status:
loadBalancer: {}
...

```
apiVersion: route.openshift.io/v1
kind: Route
metadata:
creationTimestamp: null
labels:
app: nginx
name: nginx
spec:
port:
targetPort: 8080
to:
kind: ""
name: nginx
weight: null
status: {}
...
```

Delete API resources defined by a YAML file.

```
oc delete -f newapp.yml
```

Using Templates

Get a list of templates

```
oc get templates -n openshift
```

Get template details

Review the parameters section for a list of environment variables.

```
oc describe template -n openshift mariadb-persistent
```

Create a new application from a template

```
oc new-app --template=mariadb-persistent \  
-p MYSQL_USER=jack -p MYSQL_PASSWORD=password \  
-p MYSQL_DATABASE=jack
```

Check the status of the deployment.

```
oc get all  
oc describe pod/mariadb-1-qlvrj
```

Source 2 Image

Get a list of templates and streams

```
oc new-app -L
```

Deploy a new app from a git repo

```
oc new-app php~https://github.com/clusterapps/simpleapp.git
```

Watch the app get built

```
oc logs -f buildconfig/simpleapp
```

Review the deployment with `oc get all`

NAME	READY	STATUS	RESTARTS	AGE
pod/mariadb-1-deploy	0/1	Completed	0	16m
pod/mariadb-1-qlvrj	1/1	Running	0	16m
pod/simpleapp-1-build	0/1	Completed	0	2m10s
pod/simpleapp-fb5554fd9-4kgnb	1/1	Running	0	89s

NAME	DESIRED	CURRENT	READY	AGE
replicationcontroller/mariadb-1	1	1	1	16m

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
service/mariadb	ClusterIP	10.217.5.246	<none>	3306/TCP	16m
service/simpleapp	ClusterIP	10.217.4.16	<none>	8080/TCP,8443/TCP	2m11s

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
deployment.apps/simpleapp	1/1	1	1	2m11s

NAME	DESIRED	CURRENT	READY	AGE
replicaset.apps/simpleapp-75c686cbb8	0	0	0	2m11s
replicaset.apps/simpleapp-fb5554fd9	1	1	1	89s

NAME	REVISION	DESIRED	CURRENT	TRIGGERED BY
deploymentconfig.apps.openshift.io/mariadb	1	1	1	config,image(mariadb:10.3-el8)

NAME	TYPE	FROM	LATEST
buildconfig.build.openshift.io/simpleapp	Source	Git	1

NAME	TYPE	FROM	STATUS	STARTED	DURATION
build.build.openshift.io/simpleapp-1	Source	Git@26e2f16	Complete	2 minutes ago	41s

NAME	IMAGE REPOSITORY	TAGS	UPDATED
imagestream.image.openshift.io/simpleapp	default-route-openshift-image-registry.apps-		
crc.testing/nginx2/simpleapp	latest	About a minute ago	

Revision #12

Created 11 February 2022 00:48:01 by Michael Cleary

Updated 13 March 2022 23:48:42 by Michael Cleary