

EX180

Total no. of questions: 8

1] Deploy and run a container using Dockerfile mentioned at `/path/to/directory` (Jboss-eap application image and container should run a standalone.sh command once it is started.)

- You will be asked to customize the dockerfile
- Do check the comments in the Dockerfile, they provide the hint what needs to be placed.
- Name for the container should be as mentioned in the question only.

2] Within the same `/path/to/directory`, we have multiple *.sh files, we are asked to create shell scripts:

- Write a shell script to run the container in background with specific name and port mapping.
(sudo podman ps -d -p <host-port>:<container-port> --name=<name of container> <image>)
- Write a shell script to stop and remove the container.
(sudo podman stop <container-id> && sudo podman rm <container-id>)
- Write a shell script to print last 10 lines of the logs for the container.
(sudo podman logs --tail 10 <container-id or name>)

3] Build and Publish an image to the private registry.

- Build an image from the contents mentioned in `/path/to/folder`.
- Tag the image with "version1.0:Snapshot".
- Publish it to the private registry.

```
(sudo podman build -t <tag> . )  
(sudo podman images // Check if tag was applied correctly)  
(sudo podman tag <local-image> <private-registry>:5000/<local-image>)  
(sudo podman push <private-registry>:5000/<local-image>)
```

4] Deploy a database container using podman.

- Container should run in background.
- Container should be provided with some environment variables
- Port mapping
- Naming container

```
(sudo podman run --pod <pod-name> --name <name> -d -p <host-port>:<container-port>  
-e key=value -e key=value <image>)
```

Note: Above question needs modification in Dockerfile and then deploying container using startcontainer.sh script.

[5] Deploy frontend application that connects to the database container created in question 4.

- Container should run in background.
- Container should be provided with some environment variables
- Port mapping
- Naming container.

```
(sudo podman run --pod <pod-name> --name <name> -d -p <host-port>:<container-port> -e  
key=value -e key=value <image>)
```

Note: Above question needs modification in Dockerfile and then deploying container using startcontainer.sh script.

[6] Deploy database application (postgresql) on OpenShift using Template

- Template file is given (but do not modify this file at all)
- Deploy application using OpenShift Console using postgresql template
- Set the parameters as mentioned in the question.
- Label all resources as key=value
- Label all resources as key2=value2

[7] Deploy frontend application and connect it to the database application in question no. 6

- Image for deploying frontend application is given.
- Set the parameters as mentioned in the question.
- Set labels to the resources at the time of deployment. (app=xyz)
- Expose svc to create a route with specific name

```
(oc new-app --name <name> --docker-image=<image> -e key=value -e key=value -l  
app=<xyz>)  
(oc expose svc <svc-name>)
```

[8] Troubleshooting openshift apps

- Shell script files are given.
- How to check logs of the containers deployed in Q 7 ? - Shell script 1
- How to get description of the pod - shell script2

Note for most of the questions, hyperlink for accessing the application is given below the questions, after deploying the containers/pod, please check if you can access it.