

Final Examination DO180 for CKO 1C

Soal 1 (Create Image with Dockerfile)

Ridwan, seorang pengembang software, telah selesai membuat sebuah aplikasi Java bernama Myjava-app 8.5 yang tersimpan dalam 2 file berikut:

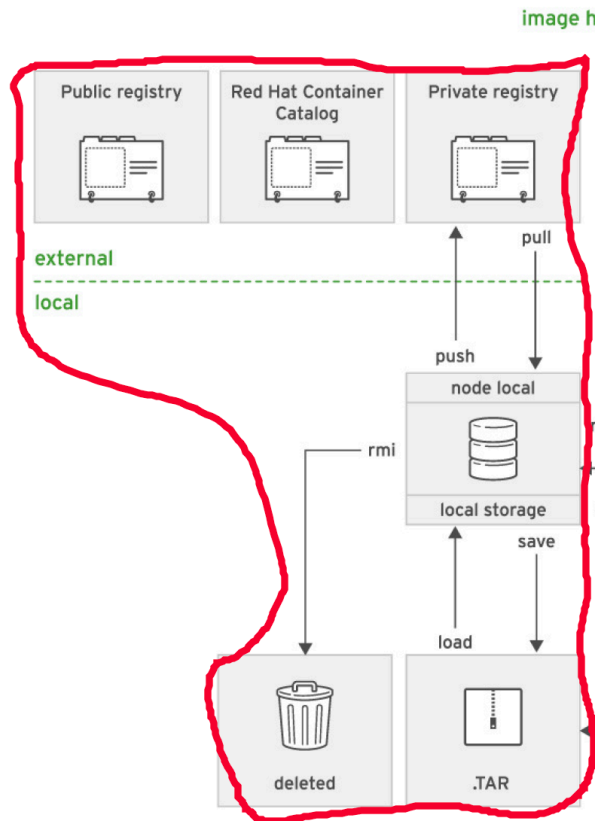
- myjava-app-8.5.0.zip
- myapp.war

Ridwan telah membuat sebuah Dockerfile agar aplikasi Myjava-app 8.5 dapat dijalankan di dalam Container (Dockerfile terlampir di Google Drive).

Pertanyaan:

- Jelaskan fungsi perintah-perintah berikut: FROM, MAINTAINER, RUN, WORKDIR, ADD, ENV, EXPOSE, USER dan CMD !
- Berapa jumlah **Layer** yang akan dibuat di dalam image aplikasi tersebut ?
- Tuliskan ulang Dockerfile tersebut** dengan penambahan **Description** di bagian "# Add your description here ..." untuk memberi penjelasan singkat apa yang dilakukan di baris tersebut dan lakukan perubahan yang diperlukan sehingga Dockerfile tersebut dapat menghasilkan **jumlah Layer yang lebih sedikit !**
- Tuliskan perintah untuk *mem-build* Image dari Dockerfile tersebut dengan Nama Image **localhost/myjava-app:8.5 !**

Soal 2 (Managing Images)



Presentasi diatas mengilustrasikan *sub-command (verbs)* dari perintah podman yang digunakan untuk bekerja dengan Image.

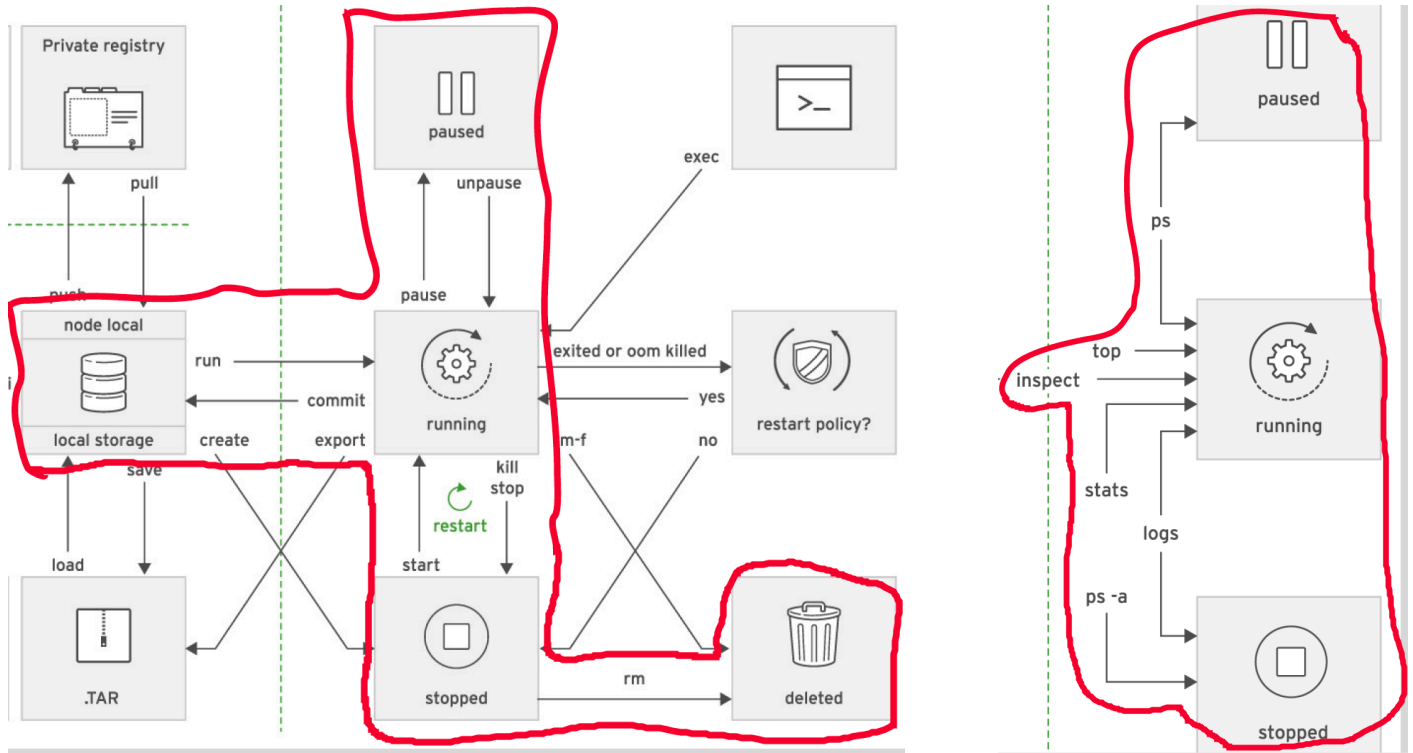
Pertanyaan:

- Jelaskan fungsi sub-comand** tersebut: pull, tag, push, load, save, rmi !
- Tuliskan perintah-perintah yang diperlukan** sehingga Image localhost/myjava-app:8.5 dapat disimpan di Image Registry **quay.io** !
- Tuliskan perintah-perintah yang diperlukan** sehingga Image localhost/myjava-app:8.5 dapat disimpan di Image Registry **docker.io** !
- Tuliskan perintah-perintah yang diperlukan** sehingga Image localhost/myjava-app:8.5 dapat disimpan di Image Registry **myregistry.mycompany.com** !
- Tuliskan perintah untuk mem-backup Image localhost/myjava-app:8.5 menjadi myjava-app-8.5.tar.gzip !

Catatan:

Di semua Image Registry, *quay.io*, *docker.io*, *myregistry.mycompany.com*, harus melakukan *authentication* sebagai furqanpr dengan password redhat123 dan image myjava-app:8.5 harus tersimpan di repository furqanpr/mycompany di setiap Image Registry-nya.

Soal 3 (Managing Container Life Cycle)

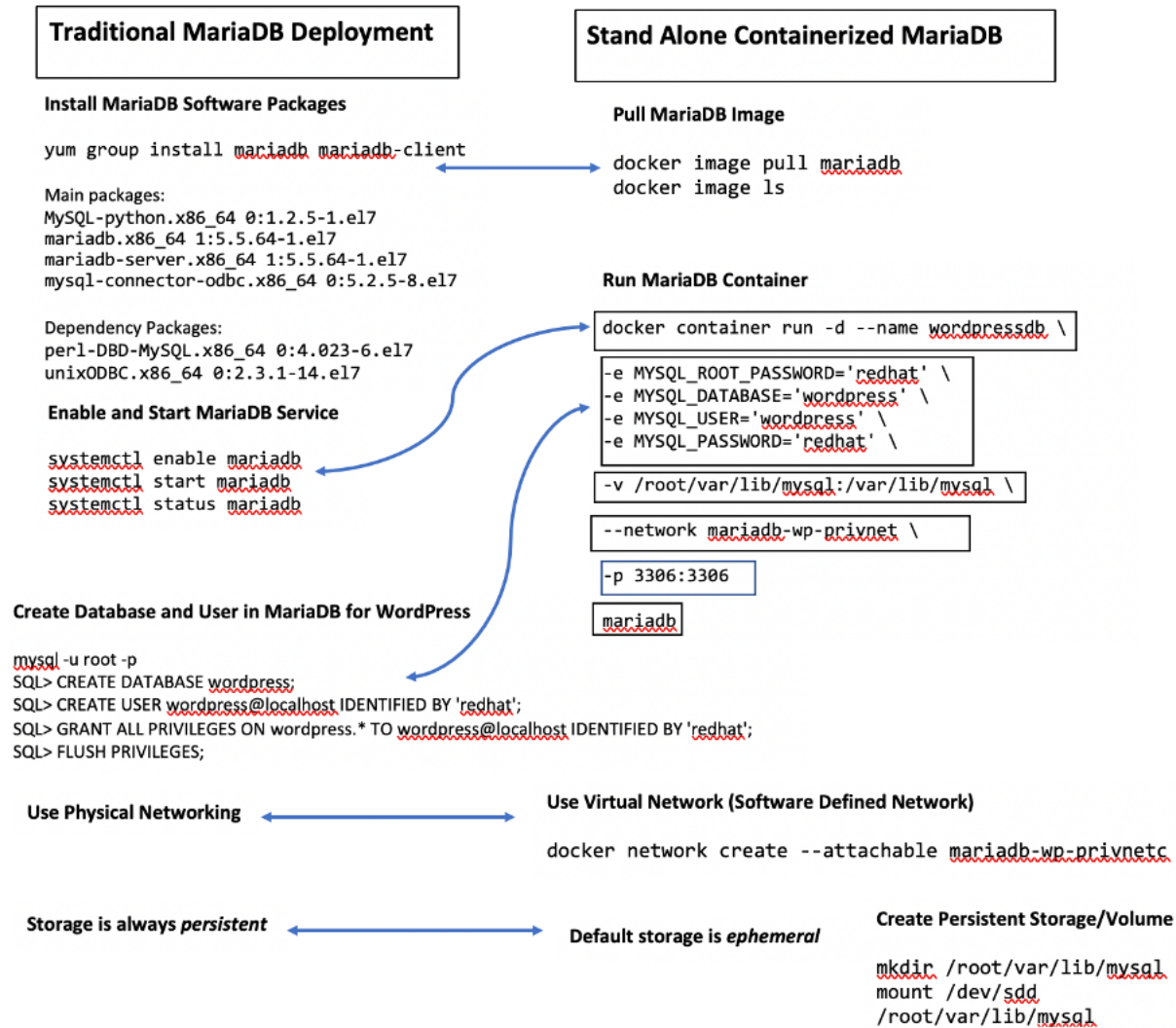


Presentasi diatas mengilustrasikan *sub-command (verbs)* dari perintah podman yang digunakan untuk bekerja dengan Container.

Pertanyaan:

Jelaskan fungsi semua *sub-command* yang tertulis di Presentasi diatas !

Soal 4 (Traditional App vs Docker Deployment)



Presentasi diatas mengilustrasikan perbedaan antara deployment traditional package-based application dengan containerized application.

Pertanyaan:

Berikan penjelasan anda tentang presentasi di atas berdasarkan pemahaman anda dengan menggunakan bahasa anda sendiri !

Soal 5 (Application Deployment Docker vs OpenShift)

Stand Alone Containerized MariaDB

1. Create OS Path for Persistent Volume

```
# mkdir /root/var/lib/mysql
# mount /dev/sdd /root/var/lib/mysql
```

2. Run MariaDB Container

```
$ docker container run -d --name wordpressdb \
-e MYSQL_ROOT_PASSWORD='redhat' \
-e MYSQL_DATABASE='wordpress' \
-e MYSQL_USER='wordpress' \
-e MYSQL_PASSWORD='redhat' \
-v /root/var/lib/mysql:/var/lib/mysql \
-p 3306:3306 \
docker.io/mariadb:latest
```

Clustered Containerized MariaDB with OpenShift

1. Create PersistentVolume

```
$ vi pv-mariadb.yaml
```

```
$ oc create -f pv-mariadb.yaml
```

2. Create PersistentVolumeClaim

```
$ vi pvc-mariadb.yaml api
```

```
$ oc create -f pvc-mariadb.yaml
```

3. Run oc new-app

```
$ oc new-app --name wordpressdb \
-e MYSQL_ROOT_PASSWORD='redhat' \
-e MYSQL_DATABASE='wordpress' \
-e MYSQL_USER='wordpress' \
-e MYSQL_PASSWORD='redhat' \
```

```
docker.io/mariadb:latest
```

4. Run oc expose

```
$ oc expose svc/mymariadb
```

5. Scale Pod

```
$ oc scale replicas=3 wordpressdb
```

Presentasi diatas mengilustrasikan perbedaan antara deployment aplikasi MariaDB dengan Docker di stand-alone server dengan OpenShift di clustered server.

Pertanyaan:

Berikan penjelasan anda tentang presentasi ini !

Soal 6 (MultiContainer Application Deployment)

Step 1

```
# mkdir -p /root/var/lib/mysql
# mkdir -p /root/var/www/html
```

Step 2

```
# docker network create --attachable mariadb-wp-privnet
```

Step 3

Missing commands are here

Step 4

```
# vi myscrip1.sh
#!/bin/bash docker container run -d \
--name mymariadb \
-e MYSQL_ROOT_PASSWORD='redhat' \
-e MYSQL_DATABASE='wpappdb' \
-e MYSQL_USER='wpapp' \
-e MYSQL_PASSWORD='redhat123' \
-v /root/var/lib/mysql:/var/lib/mysql \
--network mariadb-wp-privnet \
mariadb

# vi myscript2.sh
#!/bin/bash docker container run -d \
--name wordpress \
-e WORDPRESS_DB_HOST=YOUR-ANSWER-HERE \
-e WORDPRESS_DB_USER=YOUR-ANSWER-HERE \
-e WORDPRESS_DB_PASSWORD=YOUR-ANSWER-HERE \
-e WORDPRESS_DB_NAME=YOUR-ANSWER-HERE \
-v /root/var/www/html:/var/www/html \
--network YOUR-ANSWER-HERE \
-p 80:80 \
wordpress
```

Step 5

```
# chmod +x myscript1.sh
# chmod +x myscript2.sh
```

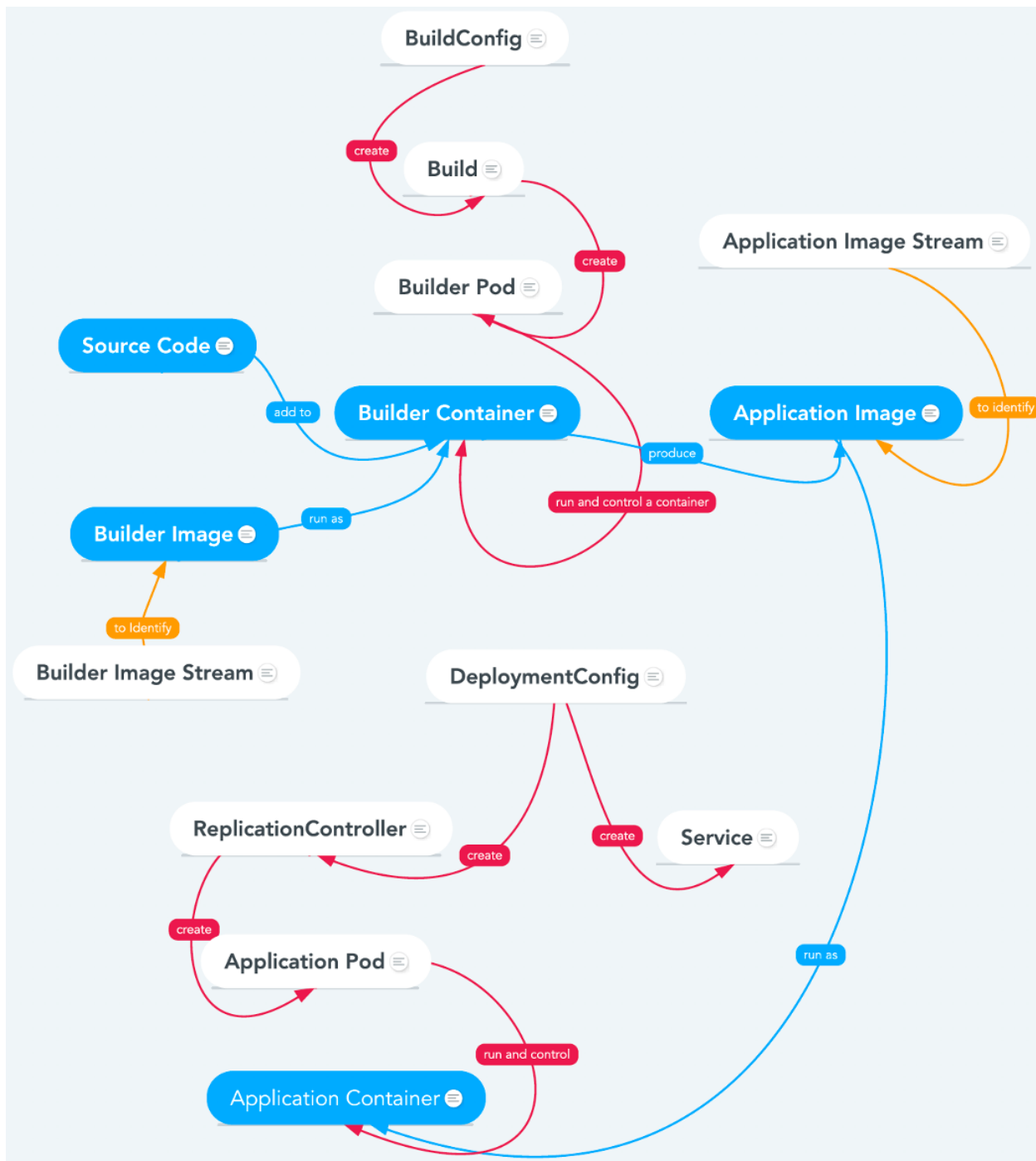
Step 6

```
# ./myscript1.sh
# ./myscript1.sh
```

Pertanyaan:

- A. Jika deployment MariaDB tersebut dilakukan di RHEL 8 dengan **SELinux Mode** yang diset **Enforcing**, **tuliskan perintah-perintah yang harus di jalan di Step 3 !**
- B. **Tuliskan ulang** `myscript2.sh` agar *Container Wordpress* dapat mengakses *Database* di *Container MariaDB !*

Soal 7 (OpenShift S2I Process)



Presentasi diatas mengilustrasikan tahapan-tahapan (*phases/steps*) dan sumber daya (*resources/objects*) yang dibuat selama proses build & deploy aplikasi dari source code di OpenShift yang disebut dengan S2I Process.

Pertanyaan:

Berikan penjelasan anda tentang Presentasi di atas berdasarkan pemahaman anda dengan menggunakan bahasa anda sendiri !

Soal 8 (OpenShift Template)

Terlampir di Google Drive adalah sebuah OpenShift Template untuk melakukan deployment Progesql.

Pertanyaan:

- A. Sebuah Template terdiri dari 5 bagian utama yaitu: apiVersion, kind, labels, objects dan parameters. **Jelaskan fungsi ke-5 bagian ini !**
- B. **Resources apa saja perlu didefinisikan** dalam sebuah Template ?
- C. Resource mana yang dapat dianggap sebagai **Resource Utama** ?
- D. **Mengapa tidak ada resource bertipe POD** di dalam Template ?
- E. **Informasi apa yang sebaiknya disimpan** dalam Resource bertipe **SECRET**?
- F. **Di bagian mana kita menuliskan Nama Image Postgresql** yang akan di-*pull*?

Soal 9 (Troubleshooting Application Deployment)

Buatlah ringkasan *cara/methodologies/techniques* dan perintah-perintah yang dapat digunakan untuk melakukan ***Troubleshooting Containerized Applications*** !