accenture technology

Learning and Knowledge Management

Module 4: Working with Images and Containers

Module Objectives

At the end of this module, you will be able to:

- Describe Docker image and its attributes
- Define the container lifecycle
- Explain container processes and the key tasks associated with it
- Determine how to manage containers

Topic List

- Docker Image
- Container Lifecycle
- Container Processes
- Container Management

Topic List

Docker Image

Container Lifecycle

Container Processes

Container Management

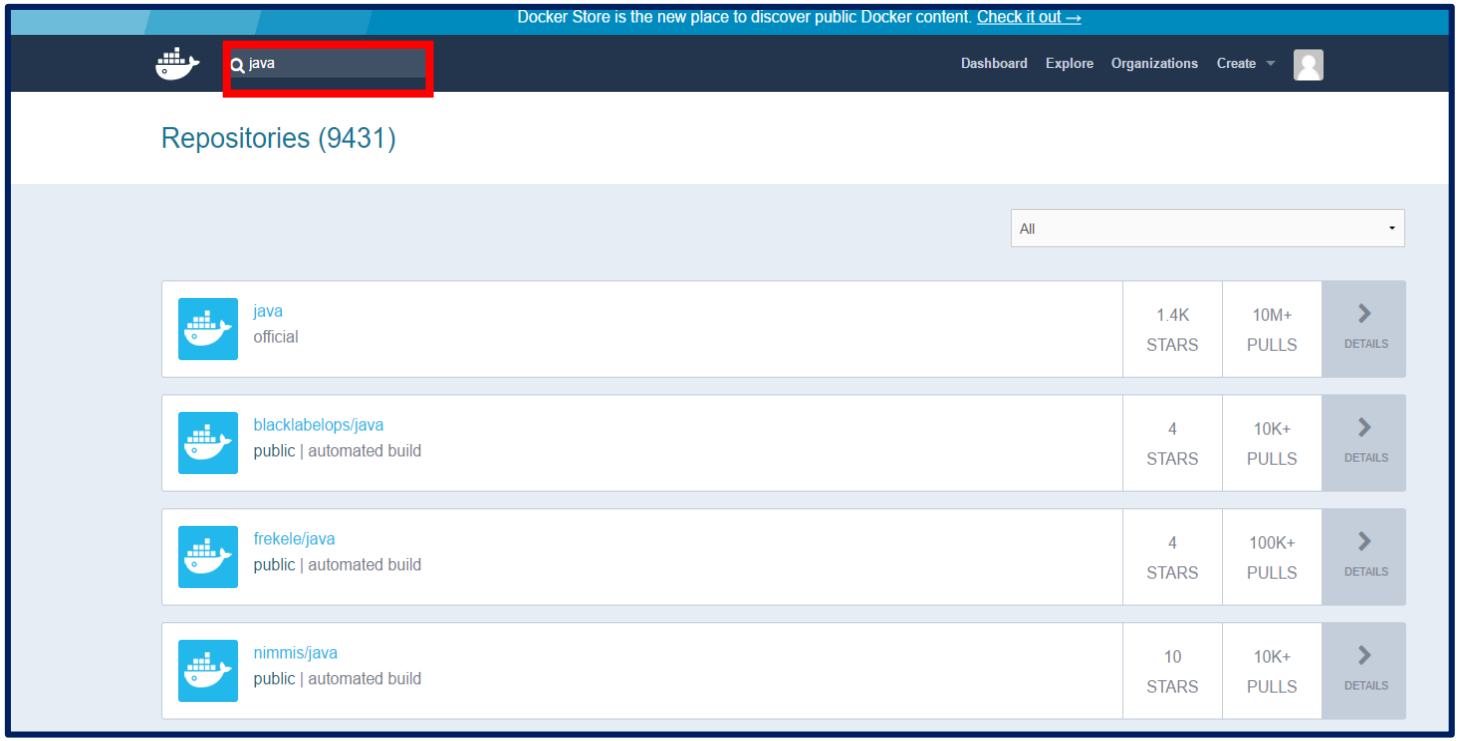
Docker Image (1)

What is a Docker image?

A docker image is a combination of a file system and parameters.

How do find images on Docker hub?

Images on the Docker hub can be found in official repositories which are certified repositories



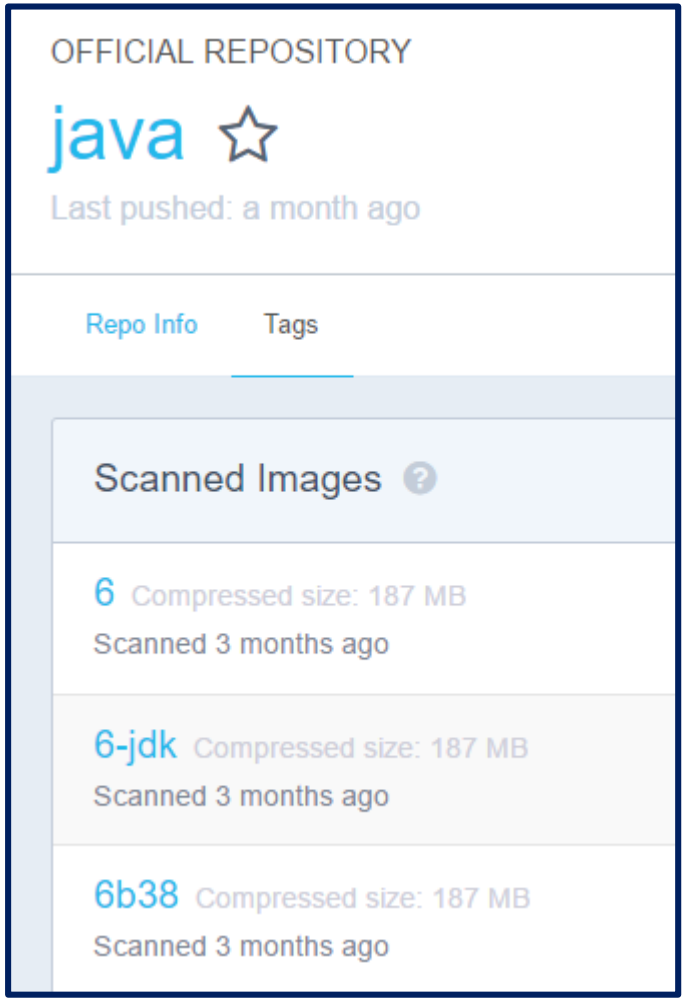
Docker Image (2)

Tagging an image

A tag is a label applied to a Docker image in a repository

Tags are how various images in a repository are distinguished from each other

The default tag is the latest



Docker Image (3)

Docker Command

A specific command which tells the Docker program on the operating system that an operation or a task needs to be carried out.
Example: docker run hello-world
sudo docker run centos –it /bin/bash

The run command helps to create an instance of an image, called a container.

"hello-world" represents the image from which the container is made

Docker Image (4)

Image Search

Searching an image refers to listing of Docker images. The default docker images will show all top level images, their repository and tags, and their size.

Usage

```
docker images [OPTIONS] [REPOSITORY[:TAG]]
```

Options

Name, shorthand	Description
--all , -a	Show all images (default hides intermediate images
--digests	Show digests
--filter , -f	Filter output based on conditions provided
--format	Pretty-print images using a Go template
--no-trunc	Don’t truncate output
--quiet , -q	Only show numeric IDs

Exercise 4

Let’s practice what we have learned so far!

Perform the following steps to search for images:

- Search for images on the docker hub/registry
- docker search java (search from client)

Exercise 5

Let’s practice what we have learned so far!

Perform the following steps to execute an image:

- sudo docker run hello-world

Exercise 6

Let’s practice what we have learned so far!

Perform the following steps to display local images:

- docker images//Display local images verify that the image present
- docker pull busybox (OR any other image)//Pull busybox image from docker hub $ docker pull busybox
- Display your local images and verify your image is present $ docker images

Docker Image (5)

Pull and Push Image

Pulling an image from Docker Trusted Registry (DTR) is similar to pulling an image from Docker Hub or any other registry. Authentication is a must since DTR is secure by default. Pushing an image to DTR requires the creation of a repository to store the image. Authentication is needed for pushing an image as well.

Usage

docker login <dtr-url>: authenticates you on DTR
docker pull <image>:<tag>: pulls an image from DTR
docker push <image>:<tag>: pushes an image to DTR

Topic List

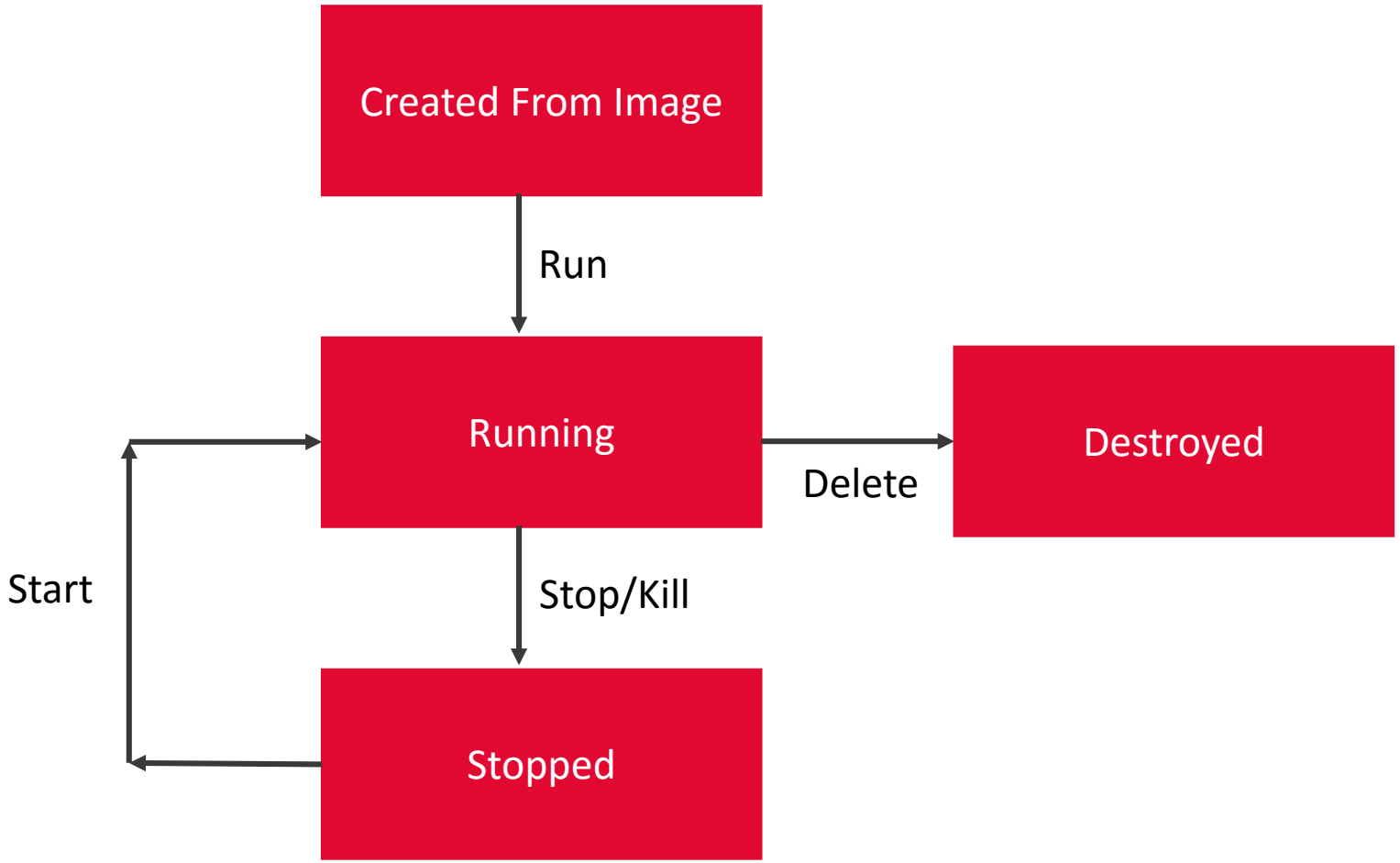
Docker Image

Container Lifecycle

Container Processes

Container Management

Container Lifecycle



Topic List

Docker Image

Container Lifecycle

Container Processes

Container Management

Container Processes (1)

- Every container is executed as a process in OS
- Linux command to list the processes: **ps -ef**
- Container starts a process to run the application with PID as 1

Container Processes (2)

Key Container Tasks: Running a Container

Command to spun-up a container

Usage

```
docker run [OPTIONS] IMAGE [COMMAND] [ARG...]
```

Examples

```
docker run hello-world
docker run ubuntu:14.04 echo “hello world”
```

Exercise 7

Let’s practice what we have learned so far!

Perform the following steps to run a container:

- docker run ubuntu:14.04 echo “hello world”
- docker images (ubuntu will be shown)
- docker run ubuntu:14.04 ls
- docker run ubuntu:14.04 ps -ef (display full info of all processes)

Container Processes (3)

Key Container Tasks: Listing Containers

Command to list containers

Usage

- List running containers
  docker ps
- List all containers
  docker ps -a (all including stopped)
- List container IDs
  docker ps – q (just ID)
- List last container
  docker ps -l (last)

Exercise 8

Let’s practice what we have learned so far!

Perform the following steps to list a container:

- docker ps
- docker ps -a (all including stopped)
- docker ps – q (just ID)
- docker ps -l (last)
- docker ps -aq
- docker ps -lq
- docker ps -a --filter “status = exited”

Container Processes (4)

Key Container Tasks: Container with Terminal

Command to connect containers with terminals

Usage

- **Option –i**
  To connect STDIN on the container
- **Option -t**
  To get a pseudo terminal
- Exit from container
  exit: to shutdown the container and exit
  ctrl + p + q: exit without shutting down the container

Example

```
docker run -i -t ubuntu:14.04 bash
```

Exercise 9

Let’s practice what we have learned so far!

Perform the following steps to use a container with terminal:

- docker run -i -t ubuntu:14.04 bash
- -i → connect to STDIN on container
- -t → to get a pseudo terminal
- Try some commands like touch / vi / cat / ls
- exit to shutdown and exit a terminal
- ctrl + p + q → to exit without shutdown
- docker run -i -t ubuntu:14.04 bash
- ls (Earlier file is lost)

Container Processes (5)

Key Container Tasks: Detached Mode

Command to run container in detached mode

Usage

- **Option –d**
  To run container in detached mode

Example

```
docker run -d ubuntu:14.04 ping localhost -c 50
```

Exercise 10

Let’s practice what we have learned so far!

Perform the following steps to run a container in detached mode:

- docker run -d ubuntu:14.04 ping localhost -c 50
- docker ps (will show the container)
- docker ps (will not show the container) after some time

Container Processes (6)

Key Container Tasks: Web App Container

Command to map container ports to host ports

Usage

- **Option –P**
  To map container ports to host ports

Example

nginx container
docker run -d -P nginx

Exercise 11

Let’s practice what we have learned so far!

Perform the following steps to run a web application container:

- Docker run -d -P nginx (P to map container ports to host ports)
- Check the mapped port using docker ps
- Open browser and type localhost:<port>

Container Processes (7)

Key Container Tasks: Stop a Container

Command to stop running containers
docker stop [options] <containerID>

Usage

- **Option –t**
  Seconds to wait for stop before killing it
  Default 10

Example

```
docker run -d tomcat:8
docker stop <containerID>
```

Container Processes (8)

Key Container Tasks: Kill a Container

Command to kill running containers
docker kill [options] <containerID>

Usage

- **Option –s**
  Signal to send to the container
  Default KILL

Example

```
docker run -d tomcat:8
docker kill <containerID>
```

Container Processes (9)

Key Container Tasks: Restart a Container

Command to start stopped containers
docker kill [options] <containerID>

Usage

- **Option –s**
  Signal to send to the container
  Default KILL

Example

```
docker run -d tomcat:8
docker kill <containerID>
```

Exercise 12

Let’s practice what we have learned so far!

Perform the following steps to stop, start, and kill a container:

- docker stop <containerID>
- docker start -a <containerID> (start and attach)
- docker kill <containerID>
- docker start <containerID> (start with same option as of initial run)
- ctrl + c

Topic List

Docker Image

Container Lifecycle

Container Processes

Container Management

Container Management (1)

Attaching Client to Container

Command to bring the container running in background to the foreground

Usage

ctrl+c to exit
ctrl+p+q to detach

Example

```
docker run -d -it ubuntu:14.04 ping localhost -c 50
docker attach <containerID>
```

Exercise 13

Let’s practice what we have learned so far!

Perform the following steps to attach/detach a container:

- docker run -d ubuntu:14.04 ping localhost -c 50
- docker attach <containerID>
- ctrl + c → for exit, ctrl + p + q → for detach
- docker run -d -it ubuntu:14.04 ping localhost -c 50 (a terminal is associated with container)
- docker attach <containerID>
- ctrl + c → for exit, ctrl + p + q → for detach

Exercise 13.1 (another example)

Let’s practice what we have learned so far!

Perform the following steps to attach/detach a container:

- Check Container processes (ps command to check the processes)
- docker run -i -t ubuntu:14.04 bash
- ps -ef (will show the bash process with Pid as 1)
- ctrl + p + q
- ps –ef

Container Management (2)

Execute Container

Command to execute additional process inside a container

Usage

- **docker exec**
  To execute additional process inside a container

Example

```
docker exec -i -t <container id> bash
```

Exercise 14

Let’s practice what we have learned so far!

Perform the following steps to execute a container:

- docker exec -i -t <container id> bash
- exit

Container Management (3)

Container Log

- Process running inside a container (PID 1) might write output to stdout and stderr
- This output can be seen using container log

Example

```
docker run -d ubuntu:14.04 ping localhost -c 100
docker logs <containerID>
docker logs -f (or --follow) <containerID>
docker logs --tail 10  <containerID>
```

Exercise 15

Let’s practice what we have learned so far!

Perform the following steps to create a container log:

- docker run -d ubuntu:14.04 ping localhost -c 100
- docker logs <containerID>
- docker logs -f <containerID>
- ctrl + c → to stop following the log
- docker logs --tail 10 <containerID>

Container Management (4)

Container Inspection

Command to get low-level information on Docker objects

Usage

docker inspect [OPTIONS] container Name (or ID)

Example

```
docker inspect <containerID>
docker inspect –format= ‘{{.Config.Cmd}}’<containerID>
```

Exercise 16

Let’s practice what we have learned so far!

Perform the following steps to inspect a container:

- docker inspect <containerID>
- docker inspect --format = ‘{{.Config.Cmd}}’ <containerID>
- docker inspect --format = ‘{{.NetworkSettings.IPAddress}}’ <containerID>
- docker inspect --format = ‘{{.Config}}’ <containerID>
- docker inspect --format = ‘{{json .Config}}’ <containerID>

Container Management (5)

Remove Container

Command to remove one or more containers

Usage

```
docker rm
```

Example

```
docker rm <containerID>
docker rm $(docker ps -ql)
```

Exercise 17

Let’s practice what we have learned so far!

Perform the following steps to remove a container:

- docker ps --filter = ‘status=exited’
- docker rm <containerID>
- docker rm $(docker ps -ql)
- docker rm $(docker ps -aq)

Module Summary

Now, you should be able to:

- Describe Docker image and its attributes
- Define the container lifecycle
- Explain container processes and the key tasks associated with it
- Determine how to manage containers

Thank You