

***Docker***

**Training Assignments**

|  |  |
| --- | --- |
| Document Code | 25e-BM/HR/HDCV/FSOFT |
| Version | 1.1 |
| Effective Date | 25/02/2022 |

**Hanoi, 02/2022**

RECORD OF CHANGES

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Effective Date | Change Description | Reason | Reviewer | Approver |
| 1 | 25/03/2022 | Create a new Assignment | Create new | ThuongDD | VinhNV |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

Contents

[Day 1. Unit 1: Introduction & Set up Docker 4](#_Toc111452810)

[Day 2. Unit 2: Dockerfile 6](#_Toc111452811)

[Day 3. Unit 3: Docker network 7](#_Toc111452812)

[Day 4. Unit 4: Docker Volume 8](#_Toc111452813)

[Day 5. Unit 5: Docker-compose 10](#_Toc111452814)

[Day 6. Unit 6: Docker swarm 12](#_Toc111452815)

[Day 7. Unit 7: Deploy app on Docker Swarm 13](#_Toc111452816)

|  |  |
| --- | --- |
| D:\FSOFT\SERVER GST\86.WIP\2.Template\6_Logo\Logo FA\Logo FA New-02.png | **CODE : Docker & Kubernetes\_Assignments/Labs**  **TYPE : Lab**  **LOC : N/A**  **DURATION : 12 days** |

# Day 1. Unit 1: Introduction & Set up Docker

Assignment :

* Set up docker, docker-compose.
* Create dockerhub.
* Write Dockerfile.
* Build, run, pull, push docker images.

Installation reference:

Linux: <https://docs.docker.com/engine/install/ubuntu/>

Window: <https://docs.docker.com/desktop/windows/install/>

MacOS: https://docs.docker.com/desktop/mac/install/

**-** Check docker version

$ docker version

- Search docker images on Docker Hub

$ docker search node

- Pull nginx latest image from Docker Hub

$ docker pull nginx:latest

- Show list image on local

$ docker images ls

- Run docker container in detach mode and mapping to port 8080 of local machine

$ docker run -d –p 8080:80 --name nginx nginx:latest

- Check running containers

$ docker ps

- Check all containers (include not running containers)

$ docker ps –a

- Stop a container

$ docker stop <container\_name> or <container\_id>

- Start a container

$ docker start <container\_name> or <container\_id>

- Remote to a running container or run a command in a running container

$ docker exec -it <container\_name> <command>

e.g. docker exec -it nginx sh

docker exec -it nginx “echo hello Docker”

- Tagging a docker image

$ docker tag <image\_id> <target\_image\_name>:<tag>

- Create your own Docker Hub account

- Create repository

- Push nginx image to your repository

- Delete nginx image in your local machine

- Try to pull the nginx image from your repository

Objectives:

* Set up lab.
* Know about docker, docker-compose, modify with docker image.

Problem Descriptions:

Assumptions:

Technical Requirements:

* Bash scripts.

Questions to answer:

Estimated Time to complete: 240 mins

# Day 2. Unit 2: Dockerfile

Assignment :

* Write Dockerfile build and run nginx, apache, nodejs, …

Task 1: Build nginx image

- Write Dockerfile to build a nginx image from Ubuntu/centos/redhat/oraclelinux/alpine base image

- Create a html file on your local machine and copy to your image

- Push image to your repositoty

- Run nginx container from your image

- Test curl from localhost

**Task 2: Build tomcat image**

- Write Dockerfile to build a tomcat image from Ubuntu/centos/redhat/oraclelinux/alpine base image

- Push image to your repositoty

- Run tomcat container from your image

- Test curl from localhost

Task 3: Build nodejs image

- Write Dockerfile to build a nodejs image from Ubuntu/centos/redhat/oraclelinux/alpine base image

- Push image to your repositoty

- Run nodejs container from your image

- Test curl from localhost

* Write and run docker-compose

Objectives:

* Know about docker-compose file

Problem Descriptions:

Assumptions:

Technical Requirements:

* Know bash scripts

Questions to answer:

Estimated Time to complete: 240 mins

# Day 3. Unit 3: Docker network

Assignment :

* Build bright, host, overlay, macvlan network.

Task 1: Using bridge networking

- Reference: https://docs.docker.com/network/network-tutorial-standalone/

- Practice using multiple bridge network:

+ Create 2 bridges

+ Create 2 containers on each bridge and test the connection between 2 containers

Task 2: Using host networking

- Reference: <https://docs.docker.com/network/network-tutorial-host/>

Task 3: Using overlay networking

- Reference: <https://docs.docker.com/network/network-tutorial-overlay/>

Task 4: Using macvlan networking

- Reference: <https://docs.docker.com/network/network-tutorial-macvlan/>

Objectives:

* Know about docker network.

Problem Descriptions:

Assumptions:

Technical Requirements:

* Have knownledge about networking.
* Network infrastructure.

Questions to answer:

Estimated Time to complete: 240 mins

# Day 4. Unit 4: Docker Volume

Assignment :

Task 1: VOLUME instruction on Dockerfile

Create an image with VOLUME instruction

Dockerfile

FROM nginx:alpine

LABEL maintainer="Collabnix"

VOLUME /myvol

CMD [ "nginx","-g","daemon off;" ]

Building Docker image

$ docker build -t volume:v1 .

Create a container based on volume:v1 image

$ docker container run --rm -d --name volume-test volume:v1

Finding the volume created on the host

Checking the volume name of the container

$ docker container inspect -f '' volume-test

ed09456a448934218f03acbdaa31f465ebbb92e0d45e8284527a2c538cc6b016

Listout Volume in the host

$ docker volume ls

DRIVER VOLUME NAME

local ed09456a448934218f03acbdaa31f465ebbb92e0d45e8284527a2c538cc6b016

You will see the volume has been created.

Volume mount path in host

$ docker container inspect -f '' volume-test

/var/lib/docker/volumes/ed09456a448934218f03acbdaa31f465ebbb92e0d45e8284527a2c538cc6b016/\_data

Testing mount working as exepected

Create a file in this folder

$ touch /var/lib/docker/volumes/ed09456a448934218f03acbdaa31f465ebbb92e0d45e8284527a2c538cc6b016/\_data/mytestfile.txt

Checking file is there in run container

$ docker container exec -it volume-test ls myvol

References:

- <https://training.play-with-docker.com/docker-volumes/>

- <https://www.tutorialspoint.com/docker/docker_storage.htm>

Task 2: Use the Device mapper storage device

* Reference: <https://docs.docker.com/storage/storagedriver/device-mapper-driver/>

Objectives:

* Know volume intruction on Docker

Problem Descriptions:

Assumptions:

Technical Requirements:

* Dockerfile

Questions to answer:

Estimated Time to complete: 300 mins

# Day 5. Unit 5: Docker-compose

Assignment :

Task 1: Install docker-compose

- Install docker-compose following this reference:

https://docs.docker.com/compose/install/

- Test the installation

Run command docker-compose version to check version of docker-compose

Run command docker-compose –help to see the cli references

Task 2: docker-compose quickstart

- Quickstart: Compose and WordPress

Using Docker Compose to easily run WordPress in an isolated environment built with Docker containers

References: <https://docs.docker.com/samples/wordpress/>

- Dockerize PostgreSQL

Create PostgreSQL container image with dockerfile, then run container PostgreSQL, test the DB connection and using container volume for PostgreSQL db container

References: <https://docs.docker.com/samples/postgresql_service/>

Task 3: Config custom network for docker-compose

- Create custom networks

- Define network when creating compose service

References: https://runnable.com/docker/docker-compose-networking

Task 4: Doing a real project including nodejs and mysql using docker-compose

- Create dockerfile for building nodejs and mysql container image

- Create docker-compose file to running nodejs and mysql service, connect to mysql service from nodejs service

- Display web page on your browser

Objectives:

* Making a real project with docker-compose

Problem Descriptions:

Assumptions:

Technical Requirements:

* SQL, wordpress, nodejs

Questions to answer:

Estimated Time to complete: 300 mins

# Day 6. Unit 6: Docker swarm

Assignment :

Task 1: Create a swarm with 1 manager node and 1 worker node

- Set up 2 virtual machine with installed Docker Engine

- Initial a swarm cluster

- Join other node as worker to the swarm

- Run docker node ls to check the status of nodes on the swarm

Task 2: Manage nodes in a swarm

- List nodes

- Inspect node

- Update node

- Promote node

- Leave the swarm

References: <https://docs.docker.com/engine/swarm/manage-nodes/>

Objectives:

* Set up DockerSwarm

Problem Descriptions:

Assumptions:

Technical Requirements:

* Docker

Questions to answer:

Estimated Time to complete: 300 mins

# Day 7. Unit 7: Deploy app on Docker Swarm

Assignment :

Task 1: Deploy a service

- Create nginx service

- Update nginx service: publish port, change image tag

- Try to run replicated service and global service

- Roll back to previous version

- (Optional) Create services using templates

References: <https://docs.docker.com/engine/swarm/services/>

Task 2: Manage sensitive data with Docker secrets

- Create a secret

- Run a service with secret

References:

- <https://docs.docker.com/engine/swarm/secrets/#simple-example-get-started-with-secrets>

- <https://docs.docker.com/engine/swarm/secrets/#defining-and-using-secrets-in-compose-files>

Task 3: Deploy a stack to swarm

- Using docker-compose file of previous lab to deploy stack

Example command:

$ docker stack deploy --compose-file docker-compose.yml stackdemo

- Check running service using command:

$ docker stack services stackdemo

References: https://docs.docker.com/engine/swarm/stack-deploy/#deploy-the-stack-to-the-swarm

(Optional) Task 6: Configure an external load balancer

References: <https://docs.docker.com/engine/swarm/ingress/#configure-an-external-load-balancer>

Objectives:

* Deploy app on Docker Swarm

Problem Descriptions:

Assumptions:

Technical Requirements:

* Docker, docker-compose, …

Questions to answer:

Estimated Time to complete: 480 mins