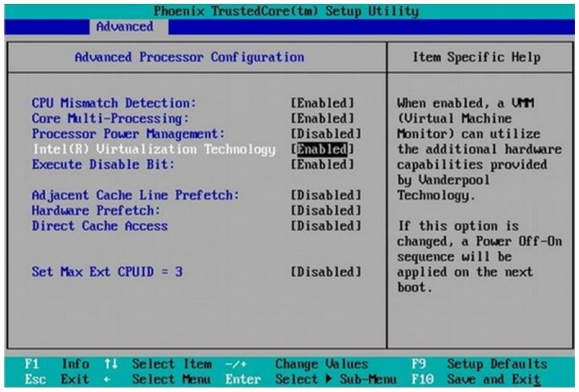
Docker Examples

Outline

- Installing Docker on Windows 10
- Example 1: Hello-world from Docker
- Example 2: Creating a MySQL database server from a container image
- Example 3: Creating a Docker image using a DockerFile

Installing Docker on Windows 10

- Enable virtualization
- Enable Windows Subsystem for Linux (WSL 2):
  - By entering the following command in an administrator PowerShell or Windows Command Prompt and then restarting your machine.
  - C:\> wsl --install
- Download and install **Docker Desktop** for Windows



Example 1: Hello-world from Docker

- $ docker run hello-world

```
C:\Windows\system32>docker run hello-world

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/
```

Example 2: Creating a MySQL database server from a container image

- To create and start a process within a new container, use the **docker run** command.
- To start a container image as a background process, pass the **-d** to the **docker run** command:
- To assign a name to the created container, the **--name** option may be passed to the **docker run** command
- To start the MySQL server with different user credentials, use the following parameters

- $ docker run mysql
- $ docker run -d mysql:5.5
- $ docker run --name mysql-container mysql:5.5
- $ docker run --name mysql-container -e MYSQL_USER=user1 –e MYSQL_PASSWORD=mypass \ -d mysql:5.5

Example 2: Creating a MySQL database server from a container image

1. Start a container from the Docker Hub MySQL image.
2. Access the container terminal by running the following command:
3. Add data to the database.
   i. Log in to MySQL as the database administrator user (root).

1. $ docker run --name mysql-container -e MYSQL_USER=user1 -e MYSQL_PASSWORD=mypass -e MYSQL_DATABASE=testDB -e MYSQL_ROOT_PASSWORD=my-secret-pw –d mysql:5.6
2. $ docker exec -it mysql-basic bash root@13568029202d:/#
3.
   i. root@13568029202d:/# mysql –pmy-secret-pw

Example 2: Creating a MySQL database server from a container image

ii. Create a new table in the items database. From the MySQL prompt, run the following commands

iii. Run the following command to verify that the table was created:

iv. Insert a row in the table and verify that it was added to the table by running the following commands:

v. Exit from the MySQL prompt

ii. mysql> show databases;
mysql> use testDB;
mysql> CREATE TABLE Courses (id int NOT NULL, name varchar(100) NOT NULL, PRIMARY KEY ((id));

iii. mysql> show tables;

iv. mysql> insert into Courses values (1, 'Docker');
mysql> select * from Courses;

v. mysql> exit
root@13568029202d:/#

Example 3: Creating a Docker image using a Dockerfile

- Building an image from a Dockerfile is a three-step process:
  1. Create a working directory that contains the files needed to build the image.
  2. Write the Dockerfile specification.
  3. Build the image with the docker command.

Example 3: Creating a Docker image using a Dockerfile

- The contents of the Dockerfile is shown in the figure.
- To create an image, use the following docker build command:

  $ docker build –t mypython-image:v1 .

```
FROM python:3

WORKDIR /usr/src/app

COPY requirements.txt ./
RUN pip install --no-cache-dir -r requirements.txt

COPY . .

CMD [ "python", "./your-daemon-or-script.py" ]
```

Thank you