

IERG 4330 / ESTR 4316

Tutorial 1

Content

- Introduction Docker
- Play with Docker

Docker

Wikipedia defines Docker as:

an open-source project that automates the deployment of software applications inside containers by providing an additional layer of abstraction and automation of OS-level virtualization on Linux.

In simpler words, Docker is a tool that allows developers, sys-admins etc. to easily deploy their applications in a sandbox (called containers) to run on the host operating system i.e. Linux.

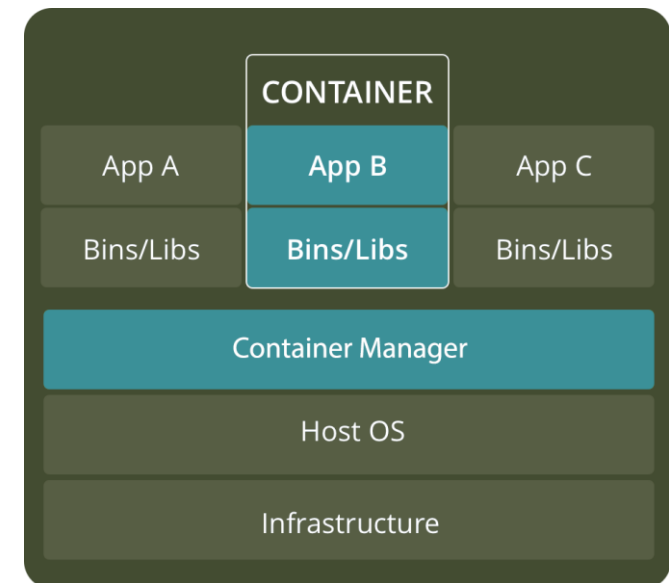
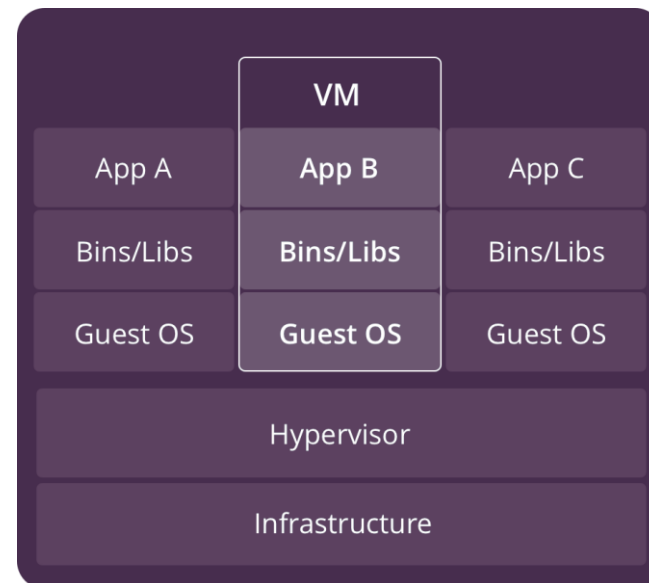
VM vs Docker

Virtual machine

- Take up a lot of system resources.
- Each VM runs not just a full copy of an operating system, but a virtual copy of all the hardware that the operating system needs to run.

Docker

- Sharing OS resources
- Portable



Terminology

- Images - The blueprints of our application which form the basis of containers. In the demo later, we will use the `docker pull` command to download the busybox image.
- Containers - Created from Docker images and run the actual application. A list of running containers can be seen using the `docker ps` command.
- Docker Daemon - The background service running on the host that manages building, running and distributing Docker containers. The daemon is the process that runs in the operation system to which clients talk to.
- Docker Client - The command line tool that allows the user to interact with the daemon. More generally, there can be other forms of clients too - such as Kitematic which provide a GUI to the users.
- Docker Hub - A registry of Docker images. You can think of the registry as a directory of all available Docker images. If required, one can host their own Docker registries and can use them for pulling images.

Install Docker

- **Install using the repository** Before you install Docker CE for the first time on a new host machine, you need to set up the Docker repository. Afterward, you can install and update Docker from the repository.
- 1). Set up the repository

Install packages to allow apt to use a repository over HTTPS:

```
ubuntu@master:~$ sudo apt-get update
ubuntu@master:~$ sudo apt-get install apt-transport-https ca-certificates curl
ubuntu@master:~$ sudo apt-get install gnupg2 software-properties-common
```

Add and verify Docker's official GPG key:

```
ubuntu@master:~$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key
ubuntu@master:~$ sudo apt-key fingerprint 0EBFCD88
```

Add the stable repository:

```
ubuntu@master:~$ sudo add-apt-repository \
    "deb [arch=amd64] https://download.docker.com/linux/ubuntu \
    $(lsb_release -cs) \
    stable"
```

Install Docker

2). Install Docker CE

```
ubuntu@master:~$ sudo apt-get update  
ubuntu@master:~$ sudo apt-get install docker-ce
```

3). Test Docker installation

```
ubuntu@master:~$ sudo docker run hello-world
```

You can also follow the installation guide in the official website:

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

Play with Busybox

To get started, let's run the following in our terminal:

```
ubuntu@master:~$ sudo docker pull busybox
```

The pull command fetches the busybox image from the Docker registry and saves it to our system.

You can use the docker images command to see a list of all images on your system.

```
ubuntu@master:~$ sudo docker images
```


Play with Busybox

Let's now run a Docker container based on this image. To do that we are going to use the almighty `docker run`

```
ubuntu@master:~$ sudo docker run busybox
```

But there is nothing happened.

When you call `run`, the Docker client finds the image (`busybox` in this case), loads up the container and then runs a command in that container. When we run `docker run busybox`, we didn't provide a command, so the container booted up, ran an empty command and then exited. Well, yeah - kind of a bummer. Let's try something more exciting.

```
ubuntu@master:~$ sudo docker run busybox echo "hello from busybox"
```

Play with Busybox

Check the status of your container

- The `docker ps` command shows you all containers that are currently running. Use parameter “-a” to show all the containers.

```
ubuntu@master:~$ sudo docker ps
```

```
ubuntu@master:~$ sudo docker ps -a
```

Play with Busybox

- Run on interactive mode

```
ubuntu@master:~$ sudo docker run -it busybox sh
```

- Restart your created container

Each time use “docker run”, you will create a new container. To restart your old container and go to the command line , use “docker restart” command and “docker exec” command.

End

- Next tutorial: Use Docker file to create your own images
- Homework 1 will release this Friday

Thank you!