

Experiment No. 2

Name: Hardik Thakkar

Roll No.: 70

Aim: Study and implement OS Level Virtualization

Tasks:

1) What is OS virtualization

In OS virtualization, the operating system is altered so that it operates like several different, individual systems. The virtualized environment accepts commands from different users running different applications on the same machine. The users and their requests are handled separately by the virtualized operating system.

2) What are containers

Containers are a solution to the problem of how to get software to run reliably when moved from one computing environment to another. This could be from a developer's laptop to a test environment, from a staging environment into production, and perhaps from a physical machine in a data center to a virtual machine in a private or public cloud.

Containers are a method of operating system virtualization that allow you to run an application and its dependencies in resource-isolated processes

3) Difference between hypervisor and linux containers

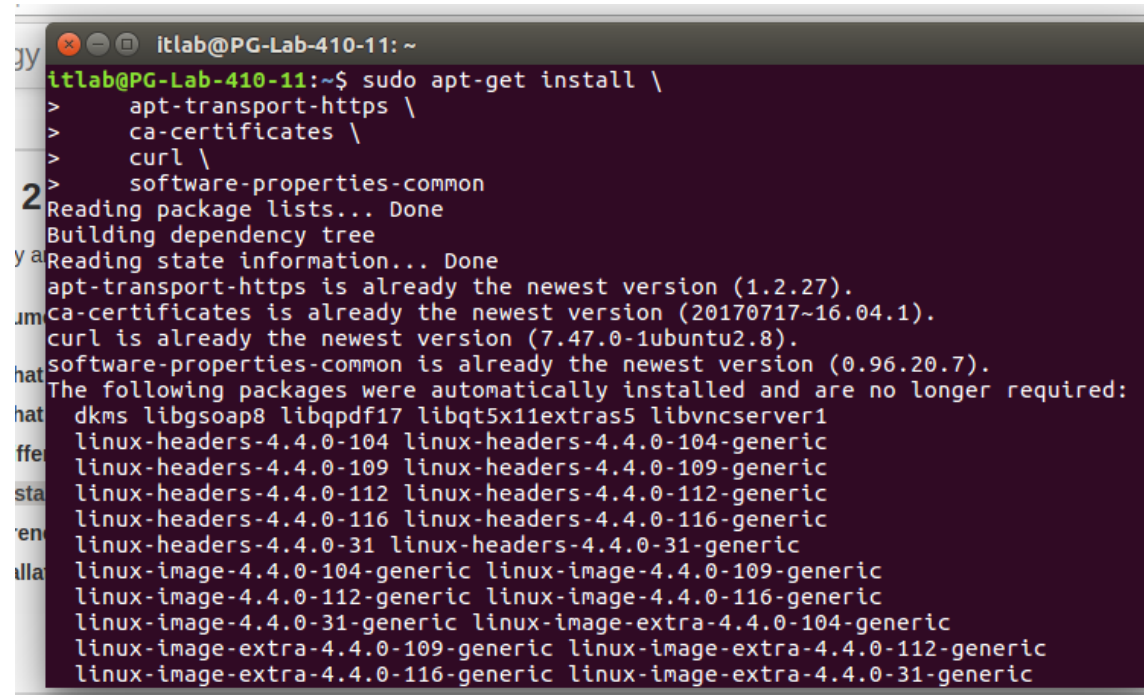
Primary difference is that containers provide a way to virtualize an OS so that multiple workloads can run on a single OS instance. With Hypervisors, the hardware is being virtualized to run multiple OS instances.

Hypervisors can run guest OSs' with different kernel than host OS, Containers cannot do this.

4) Installation and usage of docker (with screenshots)

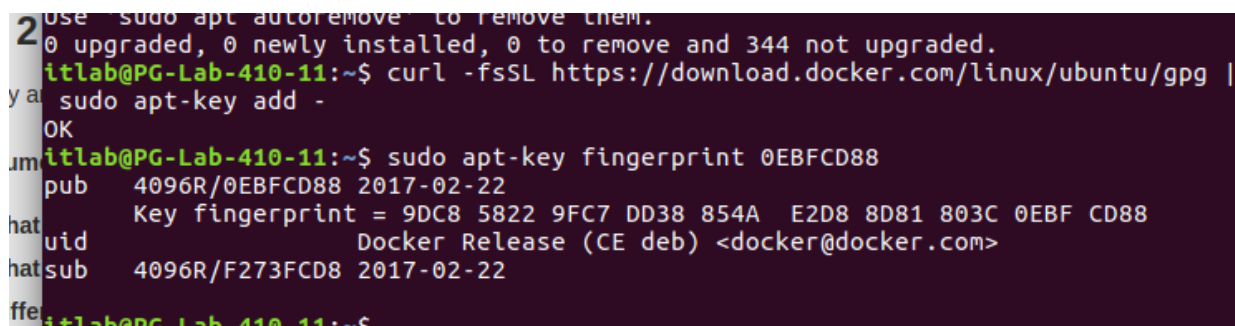
Step 1 :

Install Certificates and HTTPS access



```
itlab@PG-Lab-410-11: ~  
itlab@PG-Lab-410-11:~$ sudo apt-get install \  
> apt-transport-https \  
> ca-certificates \  
> curl \  
> software-properties-common  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
apt-transport-https is already the newest version (1.2.27).  
ca-certificates is already the newest version (20170717~16.04.1).  
curl is already the newest version (7.47.0-1ubuntu2.8).  
software-properties-common is already the newest version (0.96.20.7).  
The following packages were automatically installed and are no longer required:  
dkms libgsoap8 libqpdf17 libqt5x11extras5 libvncserver1  
linux-headers-4.4.0-104 linux-headers-4.4.0-104-generic  
linux-headers-4.4.0-109 linux-headers-4.4.0-109-generic  
linux-headers-4.4.0-112 linux-headers-4.4.0-112-generic  
linux-headers-4.4.0-116 linux-headers-4.4.0-116-generic  
linux-headers-4.4.0-31 linux-headers-4.4.0-31-generic  
linux-image-4.4.0-104-generic linux-image-4.4.0-109-generic  
linux-image-4.4.0-112-generic linux-image-4.4.0-116-generic  
linux-image-4.4.0-31-generic linux-image-extra-4.4.0-104-generic  
linux-image-extra-4.4.0-109-generic linux-image-extra-4.4.0-112-generic  
linux-image-extra-4.4.0-116-generic linux-image-extra-4.4.0-31-generic
```

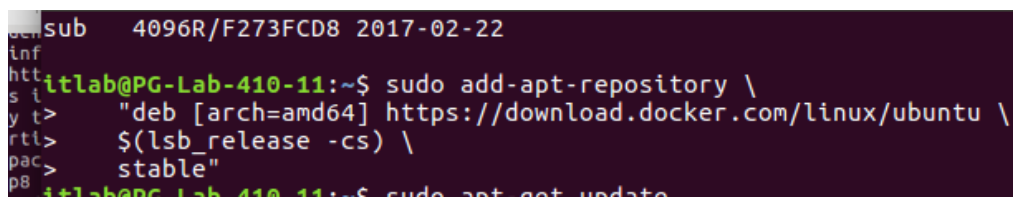
Step 2: Install Docker GPG Key and verify FingerPrint



```
2 use sudo apt autoremove to remove them.  
0 upgraded, 0 newly installed, 0 to remove and 344 not upgraded.  
itlab@PG-Lab-410-11:~$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg |  
sudo apt-key add -  
OK  
itlab@PG-Lab-410-11:~$ sudo apt-key fingerprint 0EBFCD88  
pub 4096R/0EBFCD88 2017-02-22  
Key fingerprint = 9DC8 5822 9FC7 DD38 854A E2D8 8D81 803C 0EBF CD88  
uid Docker Release (CE deb) <docker@docker.com>  
sub 4096R/F273FCD8 2017-02-22  
itlab@PG-Lab-410-11:~$
```

Step 3
:

Setup Docker Repo



```
sub 4096R/F273FCD8 2017-02-22  
itlab@PG-Lab-410-11:~$ sudo add-apt-repository \  
> "deb [arch=amd64] https://download.docker.com/linux/ubuntu \  
> $(lsb_release -cs) \  
> stable"  
itlab@PG-Lab-410-11:~$ sudo apt-get update
```

Step 4 : Install Docker

```
itlab@PG-Lab-410-11:~$ sudo apt-get install docker-ce
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
dkms libgsoap8 libqpdf17 libqt5x11extras5 libvncserver1
linux-headers-4.4.0-104 linux-headers-4.4.0-104-generic
linux-headers-4.4.0-109 linux-headers-4.4.0-109-generic
linux-headers-4.4.0-112 linux-headers-4.4.0-112-generic
linux-headers-4.4.0-116 linux-headers-4.4.0-116-generic
linux-headers-4.4.0-31 linux-headers-4.4.0-31-generic
linux-image-4.4.0-104-generic linux-image-4.4.0-109-generic
linux-image-4.4.0-112-generic linux-image-4.4.0-116-generic
linux-image-4.4.0-31-generic linux-image-extra-4.4.0-104-generic
linux-image-extra-4.4.0-109-generic linux-image-extra-4.4.0-112-generic
linux-image-extra-4.4.0-116-generic linux-image-extra-4.4.0-31-generic
virtualbox-dkms
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
aufs-tools cgroupfs-mount libseccomp2 pigz
The following NEW packages will be installed:
aufs-tools cgroupfs-mount docker-ce pigz
The following packages will be upgraded:
libseccomp2
```

Step 5 : Verify if it was installed by running Hello

```
itlab@PG-Lab-410-11:~$ sudo 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/engine/userguide/

itlab@PG-Lab-410-11:~$
```

Step 6 : Create Custom Image with Ubuntu as base layer

```
root@10f4f4859d6c: /
itlab@PG-Lab-410-11:~$ sudo docker run --name my-redis -it ubuntu:latest bash
Unable to find image 'ubuntu:latest' locally
latest: Pulling from library/ubuntu
c64513b74145: Pull complete
01b8b12bad90: Pull complete
c5d85cf7a05f: Pull complete
b6b268720157: Pull complete
e12192999ff1: Pull complete
Digest: sha256:3f119dc0737f57f704ebecac8a6d8477b0f6ca1ca0332c7ee1395ed2c6a82be7
Status: Downloaded newer image for ubuntu:latest
root@10f4f4859d6c:/# apt-get update
Get:1 http://archive.ubuntu.com/ubuntu bionic InRelease [242 kB]
Get:2 http://security.ubuntu.com/ubuntu bionic-security InRelease [83.2 kB]
Get:3 http://archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
Get:4 http://security.ubuntu.com/ubuntu bionic-security/universe Sources [10.3 kB]
Get:5 http://archive.ubuntu.com/ubuntu bionic-backports InRelease [74.6 kB]
Get:6 http://security.ubuntu.com/ubuntu bionic-security/multiverse amd64 Packages [1364 B]
Get:7 http://security.ubuntu.com/ubuntu bionic-security/main amd64 Packages [157 kB]
Get:8 http://archive.ubuntu.com/ubuntu bionic/universe Sources [11.5 MB]
Get:9 http://security.ubuntu.com/ubuntu bionic-security/universe amd64 Packages
```


Step 7 : Run Docker File

```
root@e87c6bd3f964: /chanaserver
root@5241058a1724:/# exit
exit
itlab@PG-Lab-410-11:~$ sudo docker run --name py-hardik -p 8000:8000 -it ubuntu:latest bash
root@e87c6bd3f964:/# apt-get install python3
Reading package lists... Done
Building dependency tree
Reading state information... Done
E: Unable to locate package python3
root@e87c6bd3f964:/# apt-get update
Get:1 http://security.ubuntu.com/ubuntu bionic-security InRelease [83.2 kB]
Get:2 http://archive.ubuntu.com/ubuntu bionic InRelease [242 kB]
Get:3 http://archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
Get:4 http://security.ubuntu.com/ubuntu bionic-security/universe Sources [10.3 kB]
Get:5 http://archive.ubuntu.com/ubuntu bionic-backports InRelease [74.6 kB]
Get:6 http://security.ubuntu.com/ubuntu bionic-security/main amd64 Packages [157 kB]
Get:7 http://archive.ubuntu.com/ubuntu bionic/universe Sources [11.5 MB]
Get:8 http://security.ubuntu.com/ubuntu bionic-security/universe amd64 Packages [46.3 kB]
Get:9 http://security.ubuntu.com/ubuntu bionic-security/multiverse amd64 Packages [1364 B]
Get:10 http://archive.ubuntu.com/ubuntu bionic/universe amd64 Packages [11.3 MB]
```

Step 8:

Access Python Server From Outside

```
Step 7 :  
Run Docker File  
root@e87c6bd3f964: /chanaserver  
Processing triggers for libc-bin (2.27-3ubuntu1) ...  
root@e87c6bd3f964: /# python3 -m http.server  
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...  
172.17.0.1 - - [01/Aug/2018 10:00:19] "GET / HTTP/1.1" 200 -  
172.17.0.1 - - [01/Aug/2018 10:00:19] code 404, message File not found  
172.17.0.1 - - [01/Aug/2018 10:00:19] "GET /favicon.ico HTTP/1.1" 404 -  
^C  
Keyboard interrupt received, exiting.  
root@e87c6bd3f964: /# cat > index.py  
Hello World  
^C  
root@e87c6bd3f964: /# python3 -m http.server  
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...  
172.17.0.1 - - [01/Aug/2018 10:01:21] "GET / HTTP/1.1" 200 -  
172.17.0.1 - - [01/Aug/2018 10:01:23] "GET /index.py HTTP/1.1" 200 -  
^C  
Keyboard interrupt received, exiting.  
root@e87c6bd3f964: /# cd bin  
root@e87c6bd3f964: /bin# python3 -m http.server  
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...  
172.17.0.1 - - [01/Aug/2018 10:03:11] "GET / HTTP/1.1" 200 -  
^C  
Keyboard interrupt received, exiting.  
root@e87c6bd3f964: /bin# ^C
```



The screenshot shows a web browser window with the address bar set to 'localhost:8000'. The browser displays a directory listing for the root directory ('/').

Directory listing for /

- [.dockerenv](#)
- [bin/](#)
- [boot/](#)
- [chanaserver/](#)
- [dev/](#)
- [etc/](#)
- [home/](#)
- [index.py](#)
- [lib/](#)
- [lib64/](#)
- [media/](#)
- [mnt/](#)
- [opt/](#)
- [proc/](#)
- [root/](#)
- [run/](#)
- [sbin/](#)
- [srv/](#)
- [sys/](#)
- [tmp/](#)
- [usr/](#)
- [var/](#)

Conclusion:

Hence I learnt how to install Docker and create my own container. Also I learnt difference between container and hypervisors.