Containerized GPU

Maruthi S. Inukonda

26th Jun 2018

Speaker

I am

- Maruthi Seshidhar Inukonda
- Currently Ph.D(CSE) student at CANDLE lab@CSE, IIT Hyderabad.
- M.Tech (CSE) from IIT Roorkee
- Worked in Operating Systems' Storage stack development for 13.5 years in Industry (Veritas, NetApp, EMC^2)
- Using/Working on Containers from 2014.
- Member of CCICI, SNIA, IEEE, Linux, Openstack, Ceph communities

Agenda

- Virtualization
- Virtual machines
- Containers
- Physical machine vs Virtual machine vs Container
- Dockers
- GPU Dockers & Benchmarking
- Multi-tenancy & CUDA

Virtualization

What is a Virtualization? (1/4)

- A system software that provides multi-tenancy for sharing of hardware resources. It also provides isolation.

Civil engineering example

- Dormitory complex with shared resources.
  Viz., lifts, generator, security staff, water pumps, television, restrooms, hall, bedroom.

- Hostel complex having multiple rooms with shared resources.
  Viz., all the above but not study/bedroom.

- Apartment complex having multiple flats with shared resources.
  Viz., all the above but not hall, restroom, study/bedroom, television.

What is a Virtualization? (2/4)

Computing hardware resources

- CPU, RAM, Disk, NIC.
- Accelerators (GPU)

Terminology

- Hypervisor: A software that emulates hardware/run-time environment.
- Guest: OS running in VM.
- Host: OS running on physical hardware.

What is a Virtualization? (3/4)

Types

- Full Virtualization:
  - Emulates entire computer system.
  - Runs unmodified guest OS (kernel and userspace).

  Eg. VirtualBox, XenServer, VMWare, Hyper-V, etc

- Para Virtualization:
  - Emulates entire computer system.
  - Runs modified guest OS kernel, and unmodified guest OS userspace.

  Eg. KVM

Note: We will not talk about Para virtualization

What is a Virtualization? (4/4)

Types

- Process Virtualization :
  - Emulates one programming language run-time environment.
  - Runs an application written in single programming language.

  Eg. Java/Python VM

- Operating System Virtualization :
  - Does not emulate any hardware. Directly uses host hardware.
  - Runs an user-space (application and libraries) of OS.

  Eg. Containers/Dockers

Note: We will not talk about Process virtualization

Virtual Machines

What is a Virtual Machine?

- Virtual Machine (VM) is an emulation of entire computer system, as result of full virtualization.
- Each VM runs its own operating system instance.
- Civil Engineering example :
  - Apartment complex having multiple flats.
  - Each flat has virtualized resources (generator , lift, security guard, water pump, etc.)

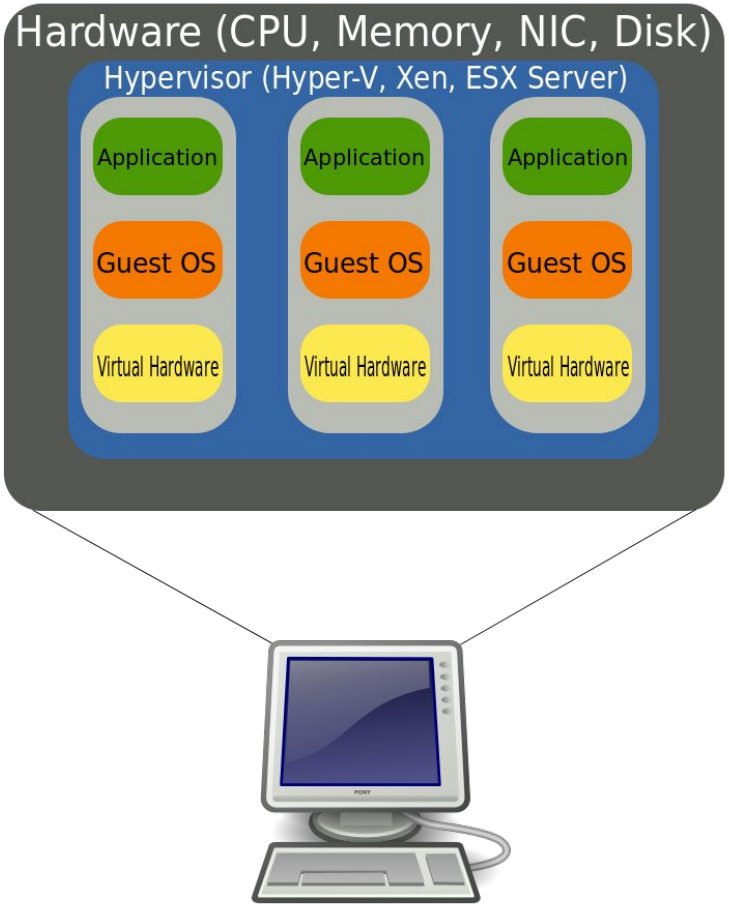

Image Courtesy: Wikipedia

Containers

What is a Container? (1/2)

- Linux Containers (LXC) is an operating-system-level virtualization method.
- For running multiple isolated Linux systems (containers) on a control host using a single Linux kernel.
- Directly runs on hardware. (No per-instruction level trapping)
- An unprivileged user on host can be privileged user on guest.
- Civil Engineering example :
  - Hostel complex having multiple rooms with shared resources.
    Viz., all the above but not study/bedroom.

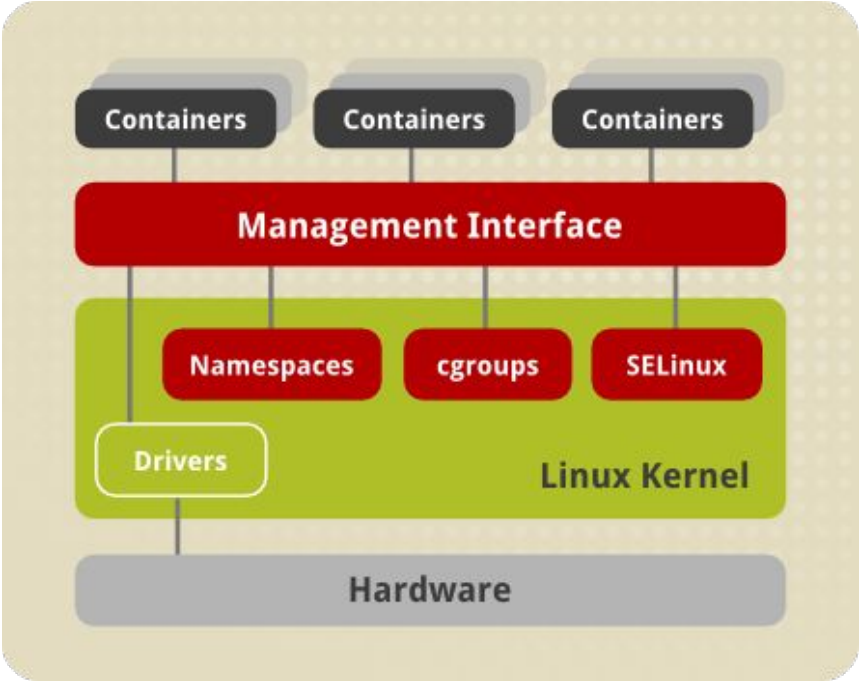

Image Courtesy: Redhat Customer Portal

What is a Container? (2/2)

It is implemented using following features in Linux

- Advanced Multi-layer Union FS (AUFS) or Overlay FS
- Kernel namespaces
- Cgroups
- Capabilities
- Netfilter, Netlink
- Bind mount
- Role-Based Access Control (RBAC)
  - Eg. SELinux, AppArmor

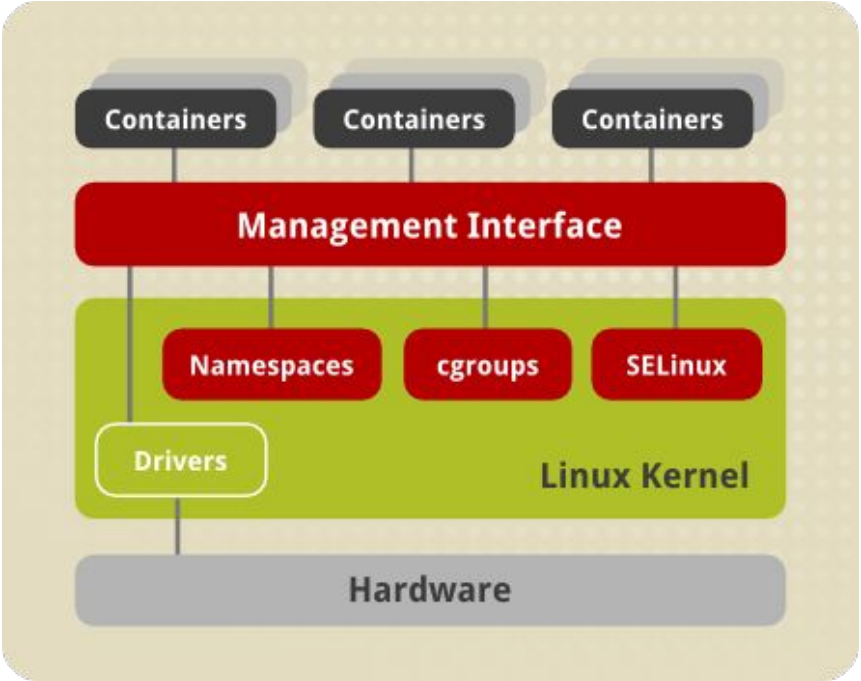

Image Courtesy: Redhat Customer Portal

Physical Machine vs Virtual Machine vs Container

Physical Machine vs Virtual Machine vs Container

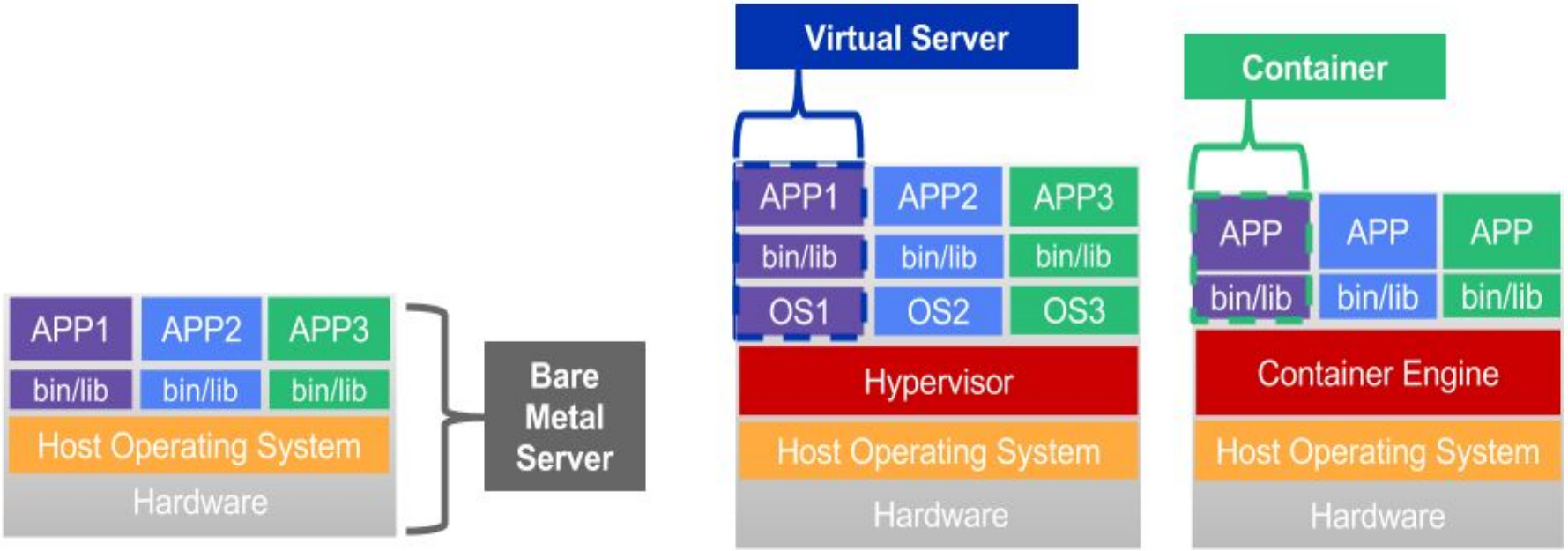

Image Courtesy: Kumulus Technologies

Dockers

What is a Docker?

- Docker is a company that provides software (also called Docker) that allows you to build, run and manage software containers.
- It makes container deployment and administration quite easy.
- It allows re-use of containers created by others.

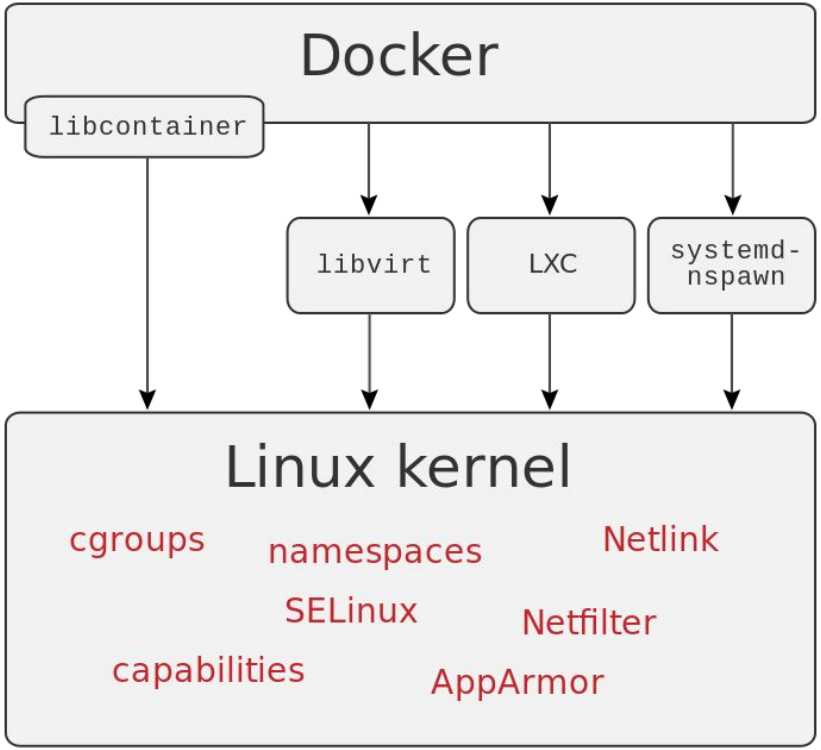

Image Courtesy: Maklaan's Docker blog post

Docker components

Docker has three major components

- Docker repository/hub
- Docker image
- Docker container

Image Courtesy: Wikipedia

Docker hub

https://hub.docker.com/search/?isAutomated=0

Search

Docker Store is the new place to discover public Docker content. Check it out →

nvidia

Dashboard Explore Organizations Create maruthisi

Repositories (774)

All

nvidia/cuda public	299 STARS	1M+ PULLS	DETAILS

Pulling images

- To pull/download docker images, use
  `docker pull <image>:<tag>`

```
$ docker pull nvidia/cuda:latest
latest: Pulling from nvidia/cuda
b234f539f7a1: Pull complete
...
d056eaf3dff4: Pull complete
7dc790b5527b: Pull complete
89bce857a556: Downloading [=========================================>        ] 463.4
MB/580.1 MB89bce857a556: Pull complete
Digest: sha256:3cdf1b5becfde8772e15dab594bc76de1cbbefd6c0f8533748854ab47e109ad1
Status: Downloaded newer image for nvidia/cuda:latest
```

Docker container life-cycle

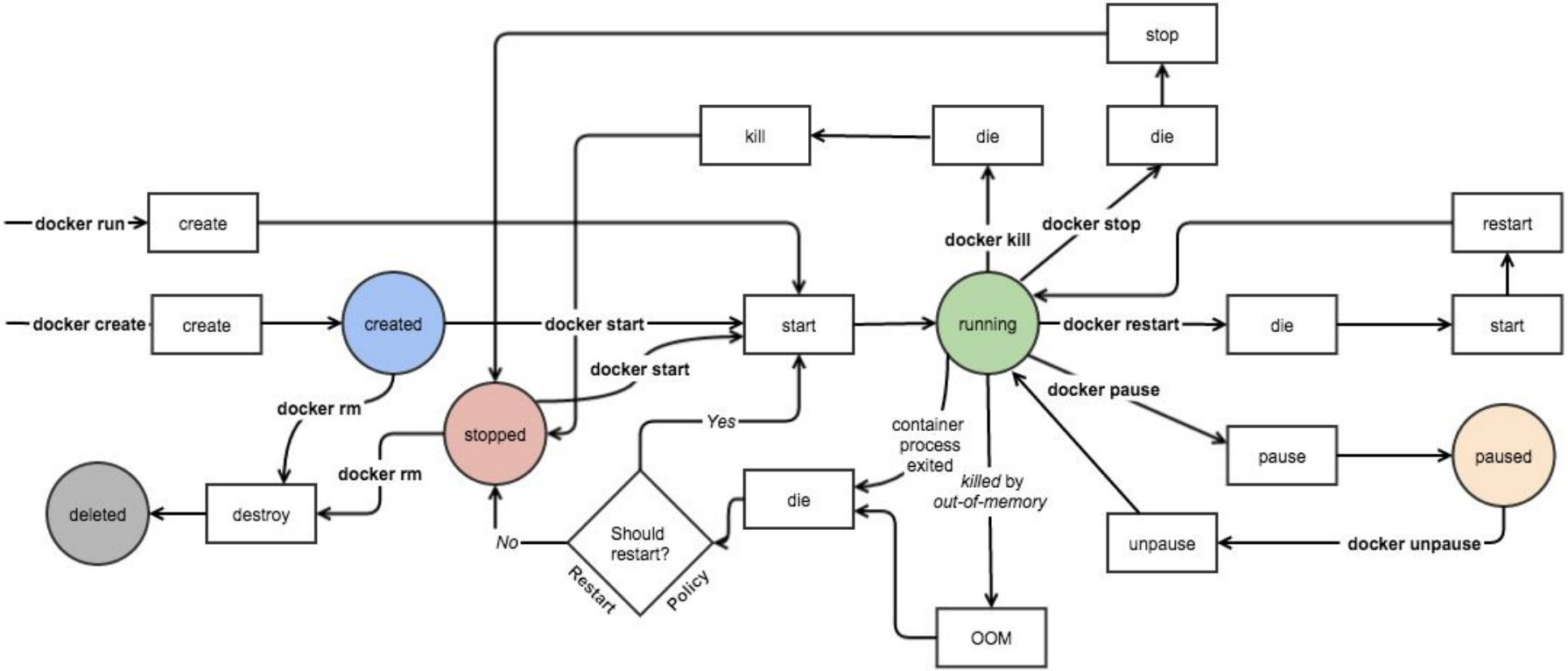

Image Courtesy: Nitin Agarwal @ medium.com

Listing images

- To list downloaded images, use `docker images`

```
$ docker images
REPOSITORY              TAG                 IMAGE ID            CREATED         SIZE
nvidia/opencl           latest              b546828c2b30        12 days ago     116MB
centos                  latest              49f7960eb7e4        2 weeks ago     200MB
nvidia/cuda             latest              9337ecb4311e        7 weeks ago     2.24GB
ros                     kinetic-ros-base    2e1693285910        8 weeks ago     1.18GB
ubuntu                  latest              452a96d81c30        8 weeks ago     79.6MB
autoware/autoware        1.6.0-kinetic      8fc60a26cc84        6 months ago    7.65GB
```

Running a container

- To run a container, use
  `docker run -it --name <name> <image>:<tag> <program>`

```
$  docker run -it --name centos1 centos:latest bash
[root@e7f7395af134 /]#

[root@e7f7395af134 /]# cat /etc/redhat-release
CentOS Linux release 7.5.1804 (Core)

[root@e7f7395af134 /]#
```

Listing running containers

- To list running container, use
  `docker ps`

```
$ docker ps
CONTAINER ID   IMAGE           COMMAND   CREATED         STATUS        PORTS   NAMES
e7f7395af134   centos:latest   "bash"    8 seconds ago   Up 7 seconds          centos1
```

Exiting from a container after stopping

- To stop and exit from a container, use `exit`

```
[root@e7f7395af134 /]# exit

$
```

Listing all containers

- To list all (running/exited) container, use
  `docker ps -a`

```
$ docker ps -a
CONTAINER ID   IMAGE          COMMAND   CREATED        STATUS                     PORTS    NAMES
e7f7395af134   centos:latest  "bash"    8 seconds ago  Exited (0) 5 seconds ago
centos1
```

Exiting from a container without stopping

- To exit from a container without stopping, press `Ctrl+p Ctrl+q`

```
[root@e7f7395af134 /]#  Ctrl+p Ctrl+q    read escape sequence

$
```

Creating a container

- To create a container, use
  `docker create --name <name> -it <image>:<tag> <program>`

```
$ docker create --name centos2 -it centos:latest bash
916dc303760db834c1aa4a9b591605ab56c91aba97bceb6fda2ca0db564f489a

$ docker ps -a
CONTAINER ID   IMAGE            COMMAND     CREATED              STATUS    PORTS   NAMES
e7f7395af134   centos:latest   "bash"      31 seconds ago       Created           centos1
916dc303760d   centos:latest   "bash"      About a minute ago   Created           centos2
```

Starting a container

- To start a container, use
  `docker start <name_or_id>`

```
$ docker start centos2
centos2

$ docker ps -a
CONTAINER ID   IMAGE              COMMAND      CREATED          STATUS        PORTS       NAMES
e7f7395af134   centos:latest   "bash"          4 minutes ago    Up 3 minutes
centos1
916dc303760d   centos:latest   "bash"          3 minutes ago    Up 2 minutes
centos2
```

Attaching to a running container

- To start a container, use
  `docker attach <name_or_id>`

```
$ docker attach centos2
[root@916dc303760d /]#
```

GPU Dockers

nvidia-docker

- To launch a nvidia GPU docker, use
  `nvidia-docker run -it --name <name> <image>:<tag> <program>`

```
$ docker images
REPOSITORY              TAG                 IMAGE ID            CREATED         SIZE
nvidia/opencl           latest              b546828c2b30        12 days ago     116MB
nvidia/cuda             latest              9337ecb4311e        7 weeks ago     2.24GB
ros                     kinetic-ros-base    2e1693285910        8 weeks ago     1.18GB
autoware/autoware        1.6.0-kinetic      8fc60a26cc84        6 months ago    7.65GB
```

```
$ nvidia-docker run -it --name nvcuda1 -v /mnt:/mnt nvidia/cuda:latest  bash
root@ba47b6c2ad70:/#
```

Performance difference between bare metal and container

A mean pooling program is run on a matrix of 4096x4096 256 times.
An element is updated with mean of its left, right, top and bottom elements.

```
On bare metal:

$ time ./cs18resch01001_Prog
real    0m14.549s
user    0m9.768s
sys    0m4.743s

In container

# time ./cs18resch01001_Prog
real    0m14.656s
user    0m2.057s
sys    0m12.603s
```

Multi-tenancy & CUDA

Launching multiple nvidia/cuda containers

```
$ nvidia-docker run -it --name nvcuda1 -v /mnt:/mnt nvidia/cuda:latest  bash
root@ba47b6c2ad70:/#

root@ba47b6c2ad70:/#  time ./cs18resch01001_Prog
real    0m28.756s
user    0m4.020s
sys    0m24.702s


$ nvidia-docker run -it --name nvcuda2 -v /mnt:/mnt nvidia/cuda:latest  bash
root@22dd291ccb73:/#

root@22dd291ccb73:~#  time ./cs18resch01001_Prog

real    0m28.778s
user    0m4.281s
sys    0m24.507s
```

Q & A