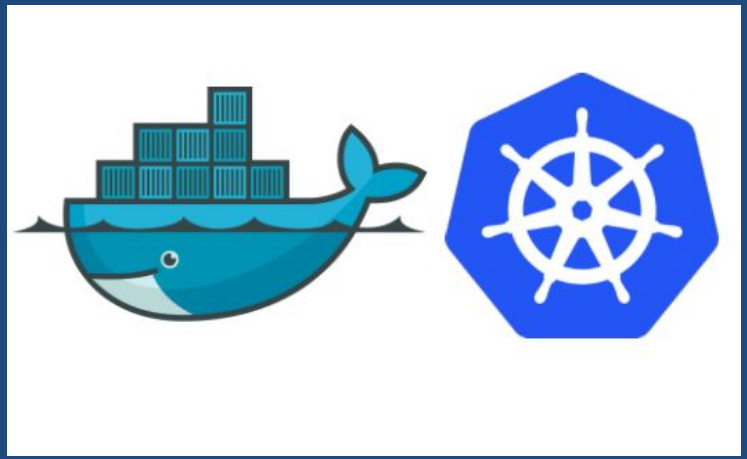
Docker Architecture



Session 4

23rd June 2018

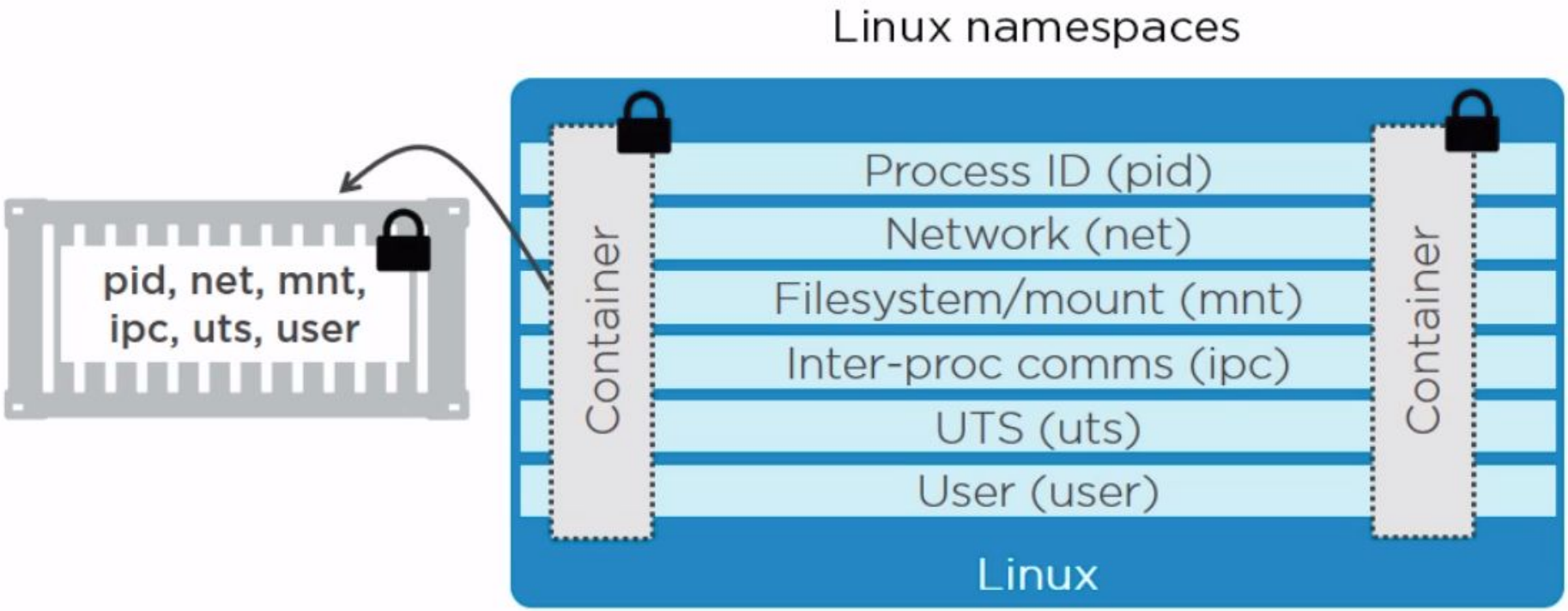
Rama Krishna Bhupathi

Docker Architecture

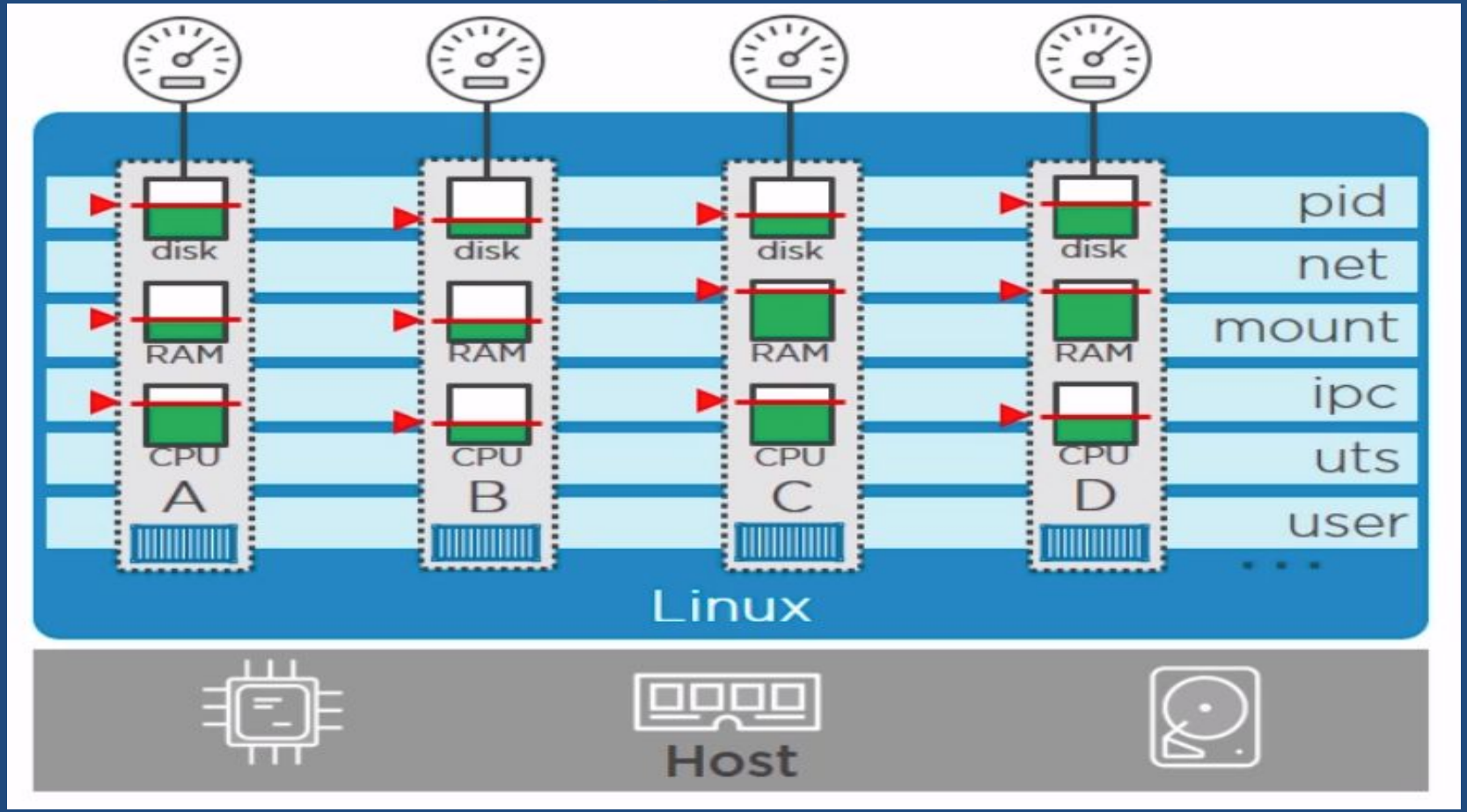
NameSpaces : Namespaces are about Isolation. Containers do not know the existence of one another

Control Groups: It is about grouping objects and setting limits.

Docker NameSpaces



Docker Control-Groups



Docker Persistence

Containers are non-persistent (containers get their own non-persistent storage).It is usually on the block storage of the local host managed by storage drivers/graph driver .Under /var/lib/docker

- Immutable
- Non-persistent
- Stateless
- Short Lived

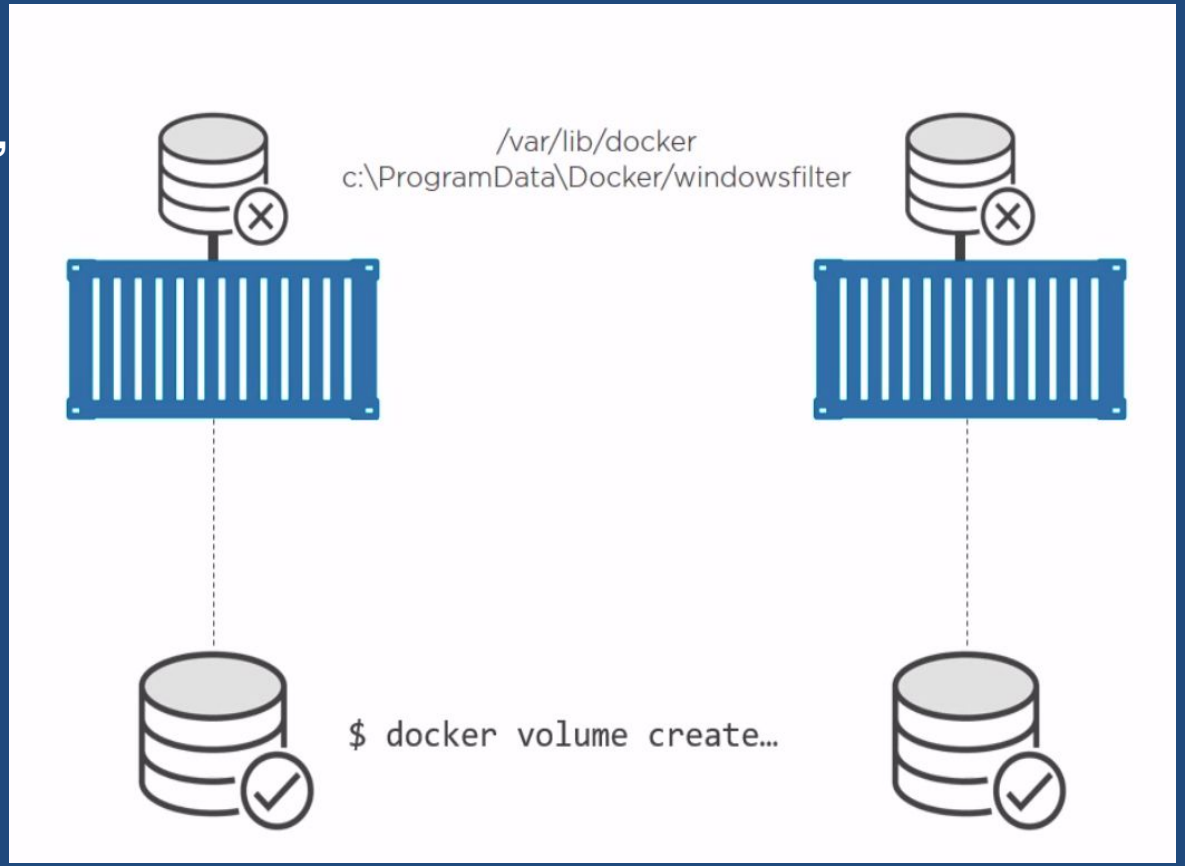
So how do you maintain the state of a container over its lifecycle?

Docker Volumes

Use “docker volume create..’
To create and attach a
Volume to a container

Volumes can persist
Even after the container
Ceases to exist.

Volume can be any storage
Device.



Volumes can exist by
Themselves without being attached to any container.

Docker Volume

Go to
https://labs.play-with-docker.com

Let us create a volume.

```
docker volume create myvolume
docker volume inspect myvolume
Docker volume rm myvolume
```

Let us Create a volume and attach it to the container

```
docker container run -dit --name volume_test --mount
source=ubervol,target=/volume_dir alpine:latest
```

Log into the container and do a ls -l and check for volume_dir

```
docker container exec -it volume_test sh
```

Docker Networking

User-defined bridge networks : When you need multiple containers to communicate on the same Docker host.

Host networks :When the network stack should not be isolated from the Docker host, but you want other aspects of the container to be isolated.

Overlay networks: When you need containers running on different Docker hosts to communicate, or when multiple applications work together using swarm services.

Macvlan networks : When you are migrating from a VM setup or need your containers to look like physical hosts on your network, each with a unique MAC address.

Docker Bridge-Networks

By default all containers use bridge networks .Bridge networks apply to containers running on the same Docker daemon host.

When you start Docker, a default bridge network (also called bridge) is created automatically, and newly-started containers connect to it unless otherwise specified.

docker network ps docker network inspect <id>

Create a container and inspect the container

Docker Bridge-Networks (User Defined)

- User Defined Bridge Networks are superior to the Default ones.
- User-defined bridges provide better isolation and interoperability between containerized applications.
- User-defined bridges provide automatic DNS resolution between containers.
- Containers can be attached and detached from user-defined networks on the fly.

More Info at
https://docs.docker.com/network/bridge/#differences-between-user-defined-bridges-and-the-default-bridge

Docker Bridge-Networks (Exercise)

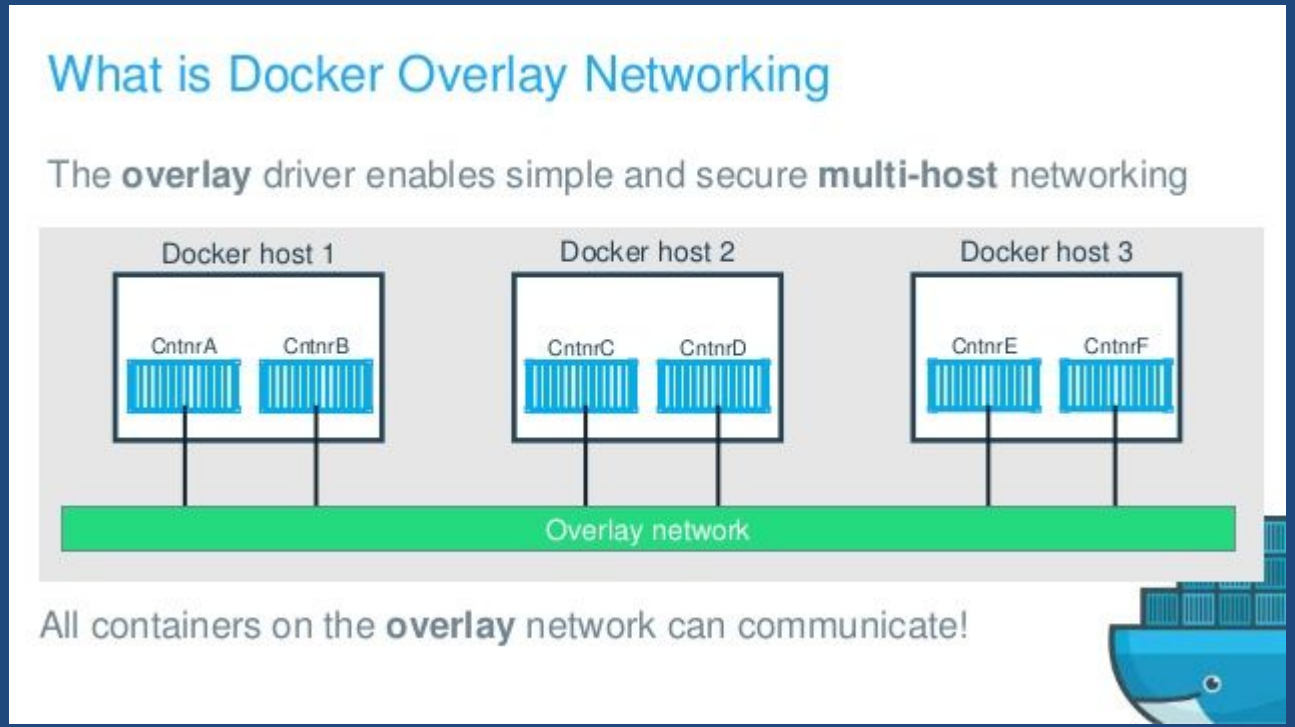
Go to
https://labs.play-with-docker.com

And Let us do the Exercise. Download the instructions document
User-Defined-Bridge-Networks-Exercise.pdf

Follow the instructions therein

OverLay Networks

The overlay network driver creates a distributed network among multiple Docker daemon hosts. This network sits on top of (overlays) the host-specific networks, allowing containers connected to it (including swarm service containers) to communicate securely and from the correct Docker daemon host and the correct destination container.



Course Materials

All the course materials are available on GitHub. Access the following URL.

https://github.com/ramakris/gentfg-kubernetes

Questions ?