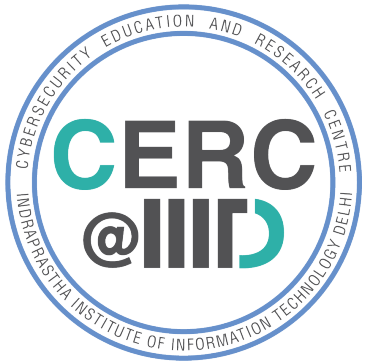
Docker

Why should I care ?

Muhammad Falak R Wani
falakreyaz@gmail.com

Cybersecurity Education and Research Centre – *(CERC)*
Department of Computer Science
IIIT-D



GDG DevFest ”17

INDRAPRASTHA INSTITUTE of INFORMATION TECHNOLOGY DELHI

Agenda

1. Foundations
   - History
   - Containers
   - Takeaways
2. Docker 101
   - Current Trends
   - Survival Skills
   - Basic Skills

Agenda

1. Foundations
   - History
   - Containers
   - Takeaways
2. Docker 101
   - Current Trends
   - Survival Skills
   - Basic Skills

INDRAPRASTHA INSTITUTE of INFORMATION TECHNOLOGY DELHI

Lets turn the dial back ...

It started on 1979

Finally we did realize it :

- Bill Joy creates **chroot** in 1979.
- **VMware** joins in on 1998. (HW Virt)
- Solaris **Jails** in 2000. (OS Virt)
- Solaris **Zones** in 2004. (Refinement of Jails)
- Google in 2007 **Process Containers** (**cgroups**)
- **LXC** was introduced in 2008
- dotCloud Open Sourced **Docker** in 2013.

Agenda

1. Foundations
   - History
   - Containers
   - Takeaways
2. Docker 101
   - Current Trends
   - Survival Skills
   - Basic Skills

What is a Container ?

A boundary box

Solution to the problem of how to get software to run reliably when moved from one computing environment to another.

What is a Container ?

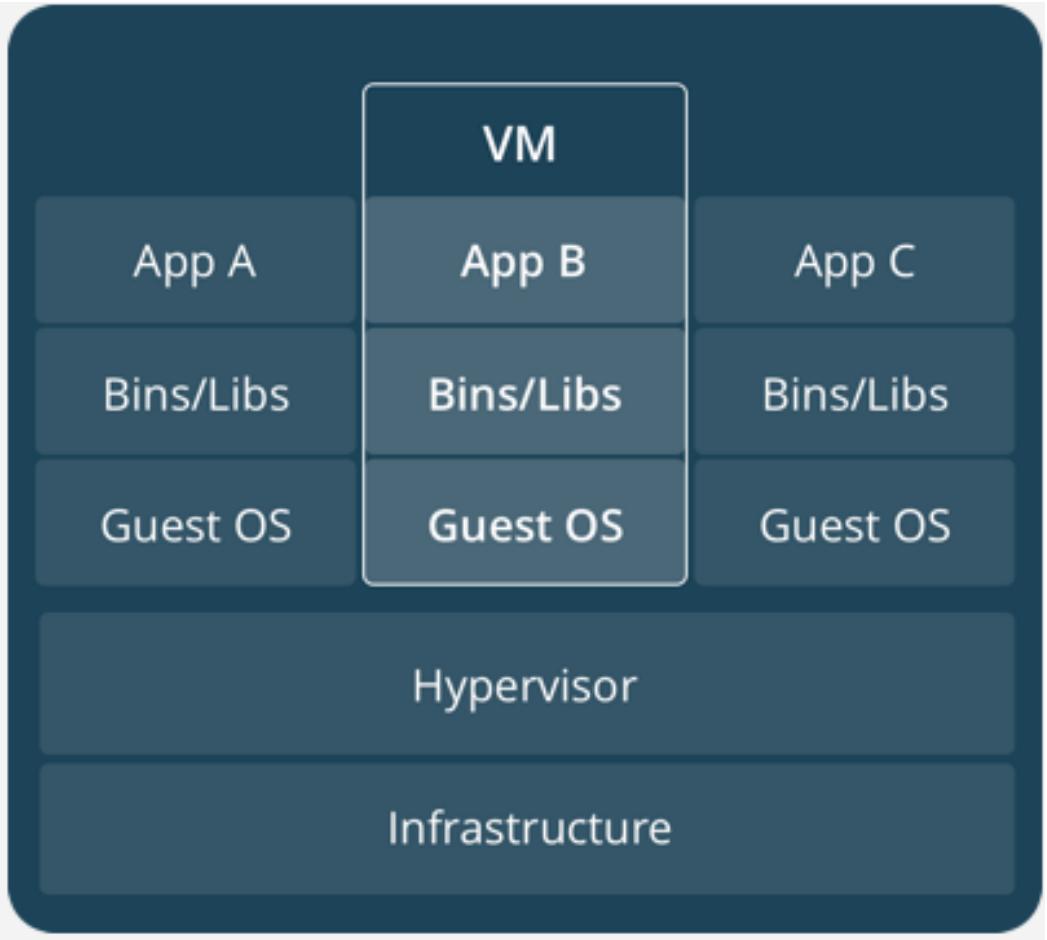
A boundary box

Solution to the problem of how to get software to run reliably when moved from one computing environment to another.

But wait!
VM's also do the same...

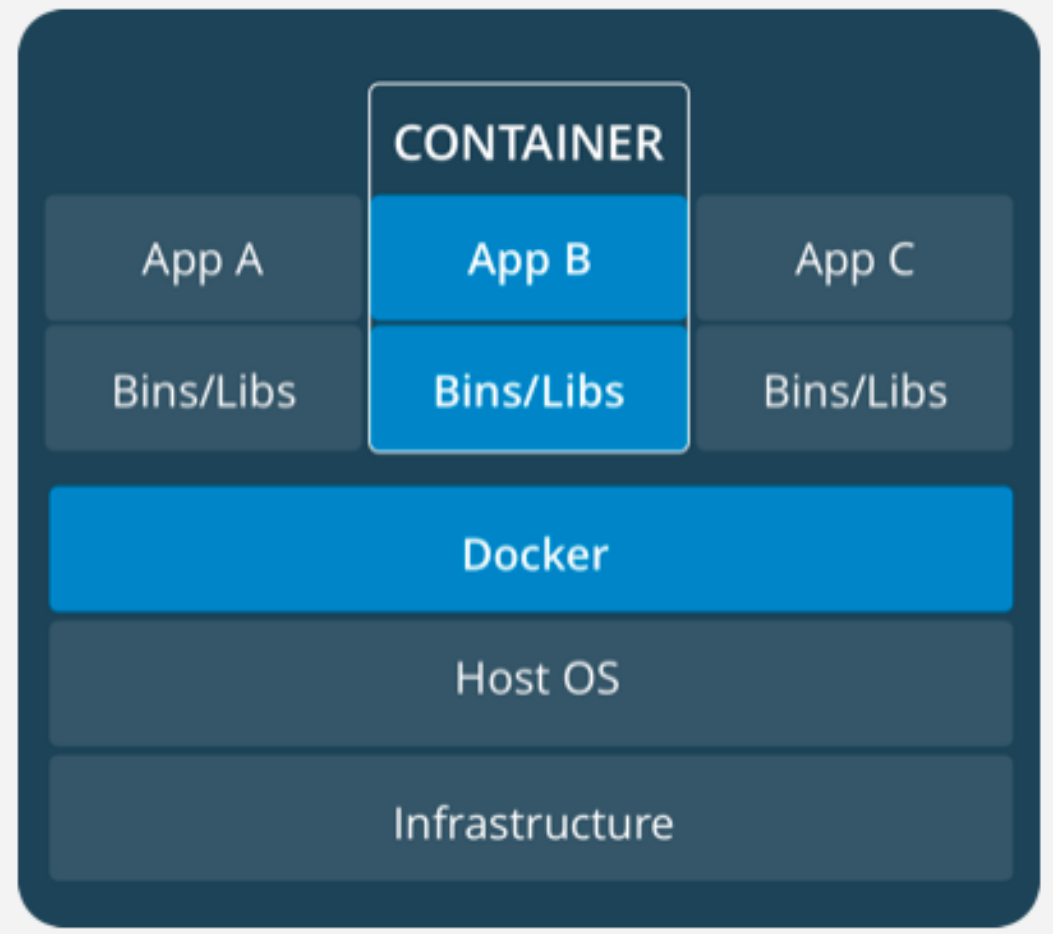
A typical VM

Very High overhead



A typical Container I

Very Low overhead



INDRAPRASTHA INSTITUTE of INFORMATION TECHNOLOGY DELHI

A typical Container II

How's it done ?

- namespaces: Wraps a global system resource in an abstraction.
- cgroups: Limits, accounts for, and isolates the resource usage.

These are just pointers, so that you can look them up. Can't say much due to the time constraint.

Agenda

1. Foundations
   - History
   - Containers
   - Takeaways
2. Docker 101
   - Current Trends
   - Survival Skills
   - Basic Skills

Advantages & Disadvantages

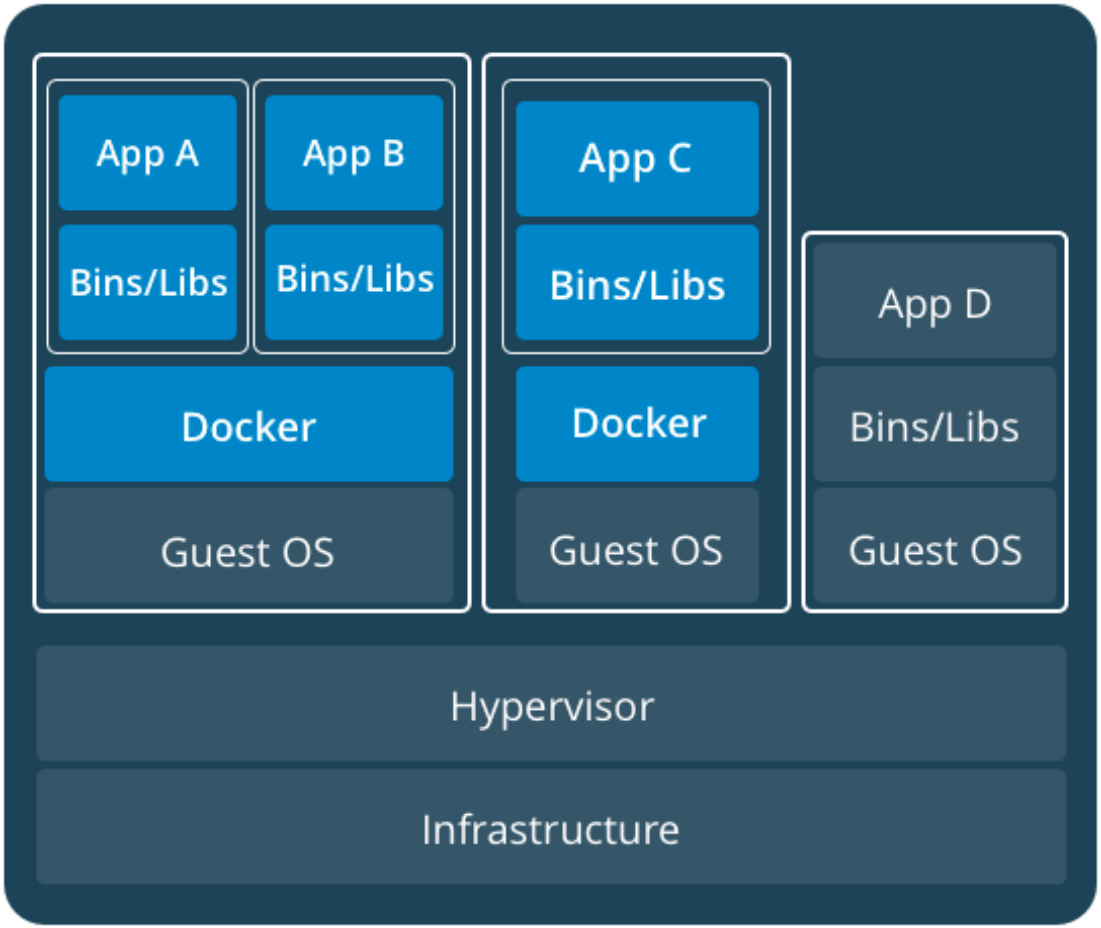
Lets draw a line

- Size: VM's come with a lot of baggage (OS)
- Start-up Time: A container uses your host OS, so starts almost instantly.
- Micro-Services: Split the app in to modules.

- Can't run a mix of OS's.
- Can't work on the Kernel Level*.

Hybrid Approach

Best of both the worlds



INDRAPRASTHA INSTITUTE of INFORMATION TECHNOLOGY DELHI

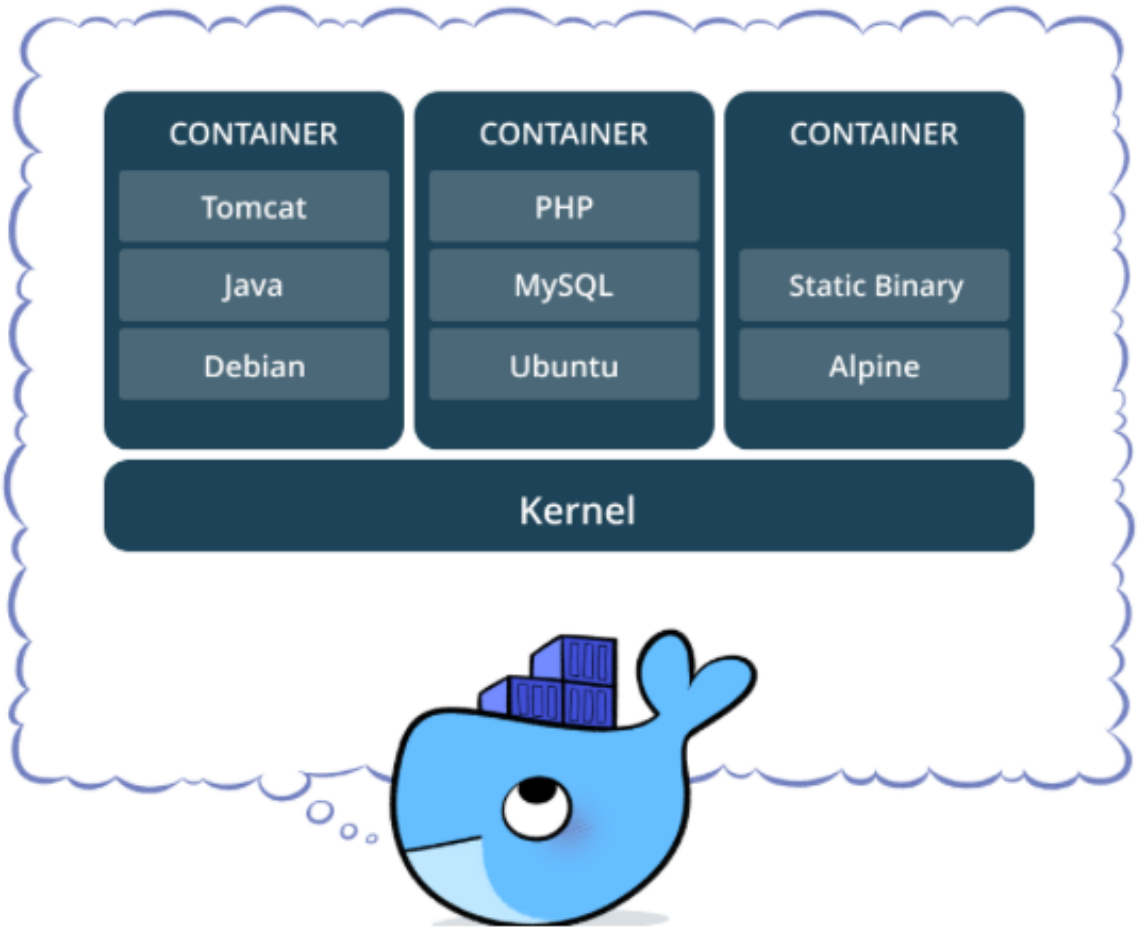
Agenda

1. Foundations
   - History
   - Containers
   - Takeaways
2. Docker 101
   - Current Trends
   - Survival Skills
   - Basic Skills

INDRAPRASTHA INSTITUTE of INFORMATION TECHNOLOGY DELHI

Should I care ?

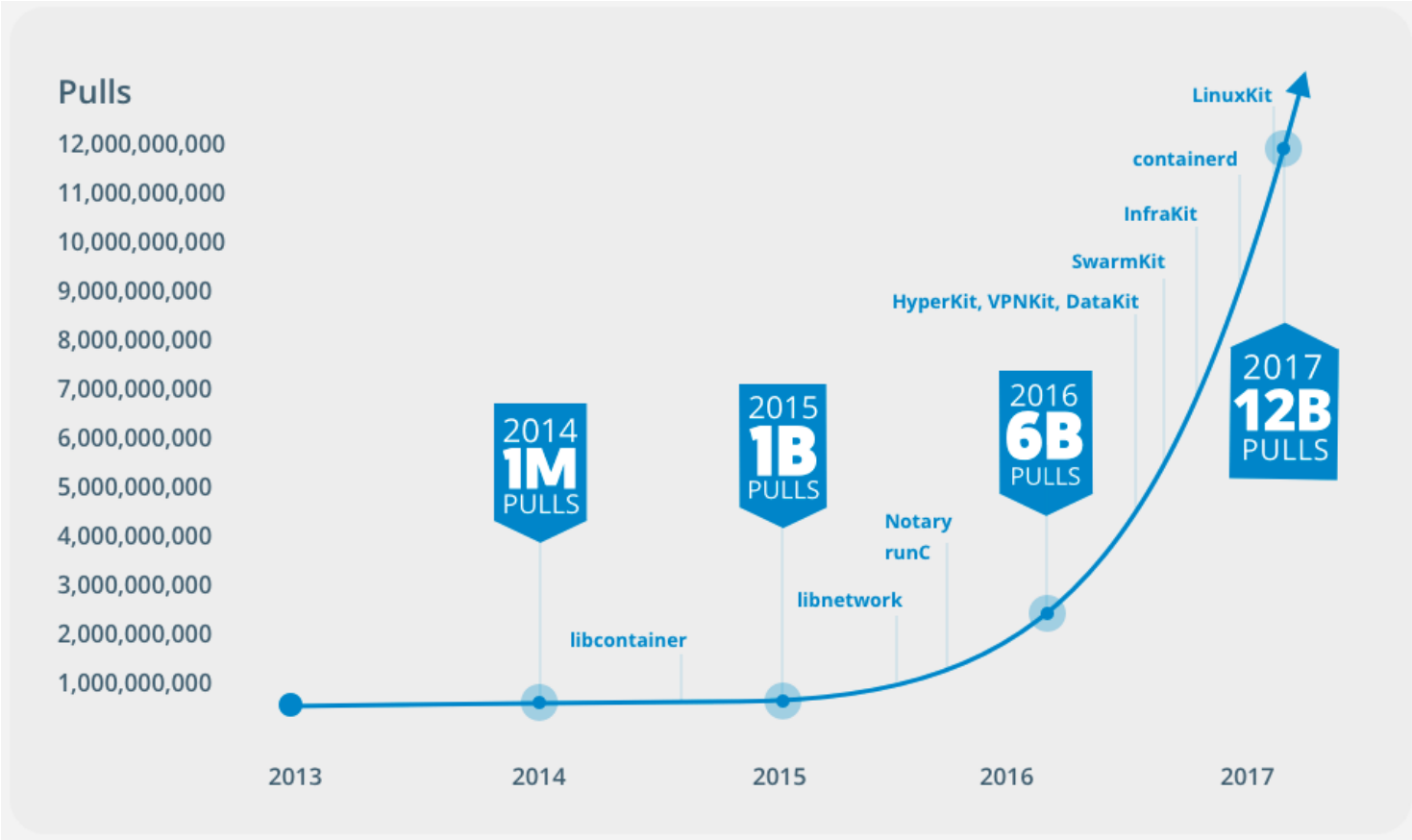
You need to decide



Should I care ?

Obviously...

Docker will do the same to apt, what apt did to *.tar.gz



Agenda

1. Foundations
   - History
   - Containers
   - Takeaways
2. Docker 101
   - Current Trends
   - Survival Skills
   - Basic Skills

Instantiate a Container

docker container run

docker *container* **run** [options] image-name `command`

– or –

docker **run** [options] image-name `command`

- docker **run** hello-world
- docker **run** alpine `ping -c 5 www.google.com`
- docker **run** -d ubuntu `sleep 600`
- docker **run** -it ubuntu `bash`

INDRAPRASTHA INSTITUTE of INFORMATION TECHNOLOGY DELHI

View Containers

docker container ls

docker *container* **ls** [options]

– or –

docker **ps** [options]

- docker ps
- docker ps -s
- docker ps -a
- docker ps -as

Agenda

1. Foundations
   - History
   - Containers
   - Takeaways
2. Docker 101
   - Current Trends
   - Survival Skills
   - Basic Skills

Miscellaneous I

docker {stop, start, remove, pull}

- **Stop** a running container:
  `docker stop #ash`
- **Start** a stopped container:
  `docker start #ash`
- **Remove** a stopped container:
  `docker rm #ash`

- **Pull** an Image from registry:
  `docker pull image-name`
- **Remove** a docker image:
  `docker rmi img-#ash`

Miscellaneous II

docker {stats, exec, attach, logs, inspect}

- Docker **stats** – I/O, MEM, CPU:

  `docker stats`

- **Exec** in a running container:

  `docker exec -it #ash`
- **Attach** to a detached container:

  `docker start #ash`
- **Logs** for a container:

  `docker rm #ash`
- **Inspect** a container:

  `docker inspect #ash`

INDRAPRASTHA INSTITUTE of INFORMATION TECHNOLOGY DELHI

- **Publishing Ports**:

```
docker run -it -p hPort:cPort img cmd
```

- **Sharing Files**:

```
docker run -it -v hPath:cPath img cmd
```

Creating Docker Images

docker commit & Dockerfile

- **Saving** container state:

```
docker commit #ash tag
```

- **Dockerfile**:

```
docker build -t tag /path/
```

Dockerfile

Creating docker Images

```
FROM alpine:latest

LABEL maintainer "mfrw <falakreyaz@gmail.com>"

RUN apk --update add tor && adduser -D anon
COPY torrc /etc/tor/torrc

EXPOSE 9050 9051
USER anon

CMD [ "tor" ]
```

Summary

Let's wrap it up

- OS Level Virtualization.
- Docker Images.
- Docker pull/push
- Docker Containers.
- Dockerfile