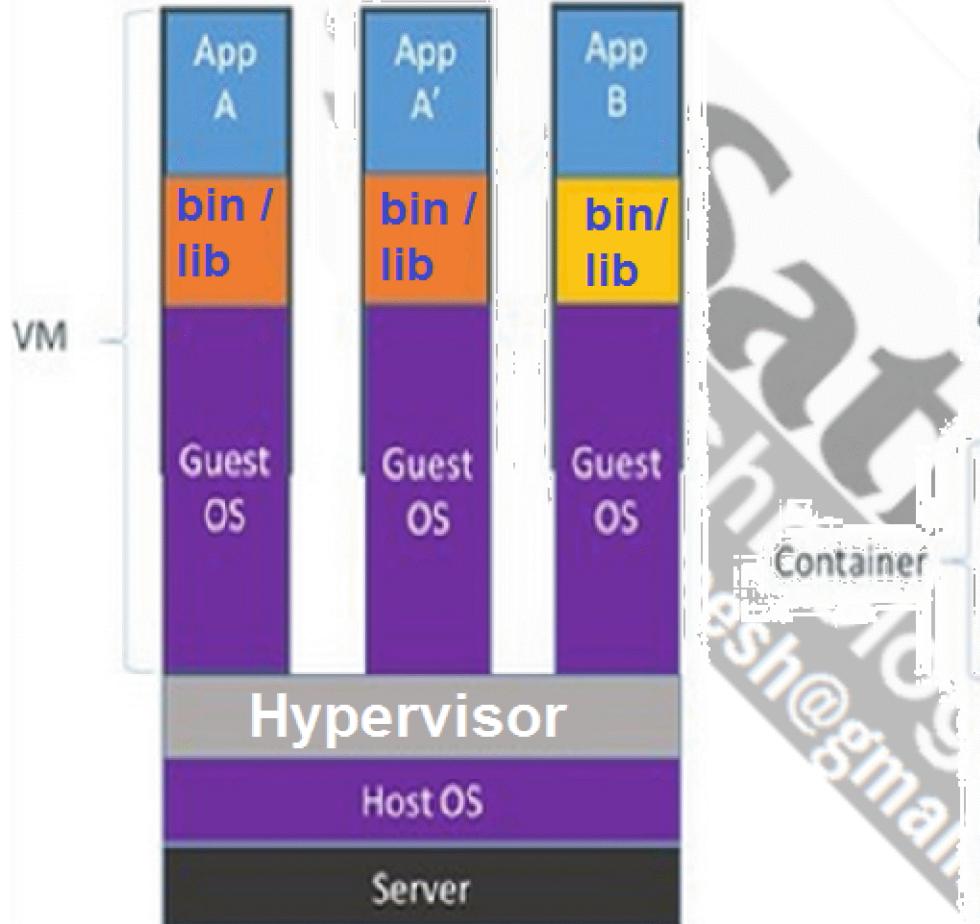
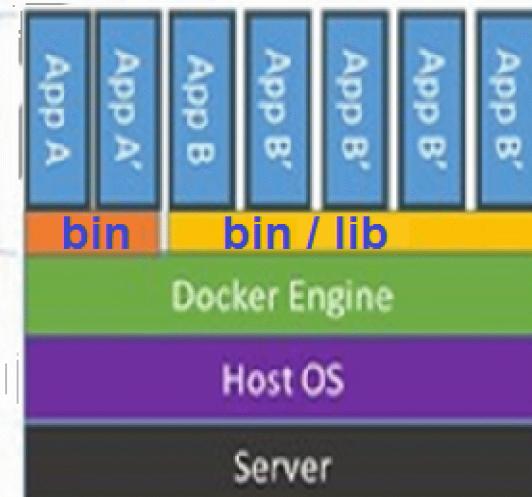


DevOps by SATISH @ Sathya Technologies

## Containers vs. VMs



Containers are isolated,  
but share OS and, where  
appropriate, bins/libraries



## **VM 's**

---

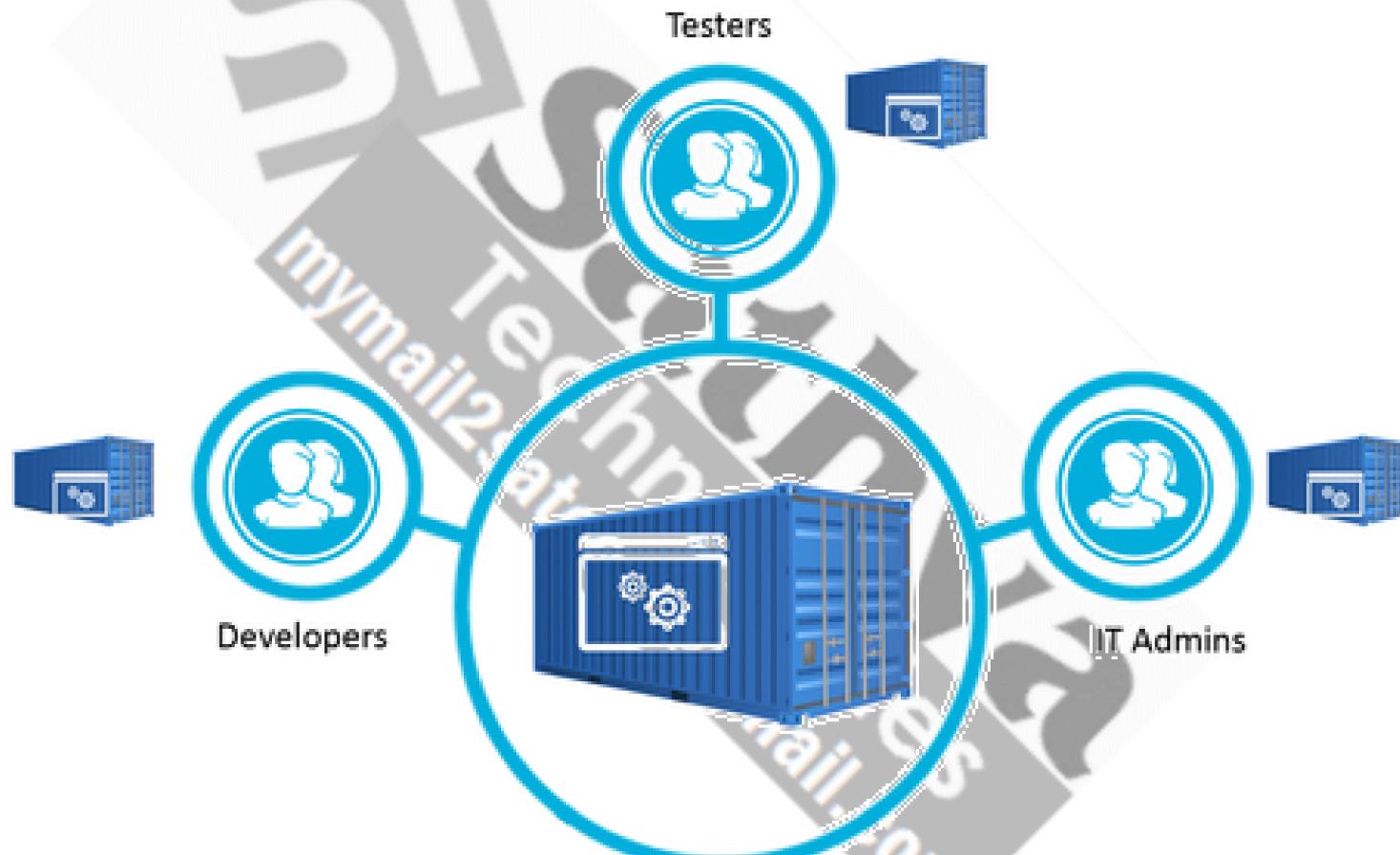
- 1) more time to create**
- 2) dedicated OS**
- 3) specific bin / lib**
- 4) dedicated resources**
- 5) memory wastage**
- 6) low performance**
- 7) complex configure.**
- 8) heavy weight (Gb's)**
- 9) memory can not share**

## **Docker Containers**

---

- 1) less time**
- 2) common OS**
- 3) common bin / lib**
- 4) common resources**
- 5) no Memory wastage**
- 6) high performance**
- 7) simple configurations**
- 8) light memory (Mb's)**
- 9) can share momory**

# Application shipment



**Application shipment  
across departments**  
DevOps by SATISH @ Sathya Technologies

# Run Platforms

- Various Linux distributions (Ubuntu, Fedora, RHEL, Centos, openSUSE, ...)
- Cloud (Amazon EC2, Google Compute Engine, Rackspace)
- 2014-10: Microsoft announces plans to integrate Docker with next release of Windows Server

# Install Docker Engine

- Recommended Installation:
- `# curl -fsSL https://test.docker.com | sh`
- <https://docs.docker.com/engine/installation/>
- Installing Docker on Windows 10:
  - <https://docs.docker.com/docker-for-windows/>

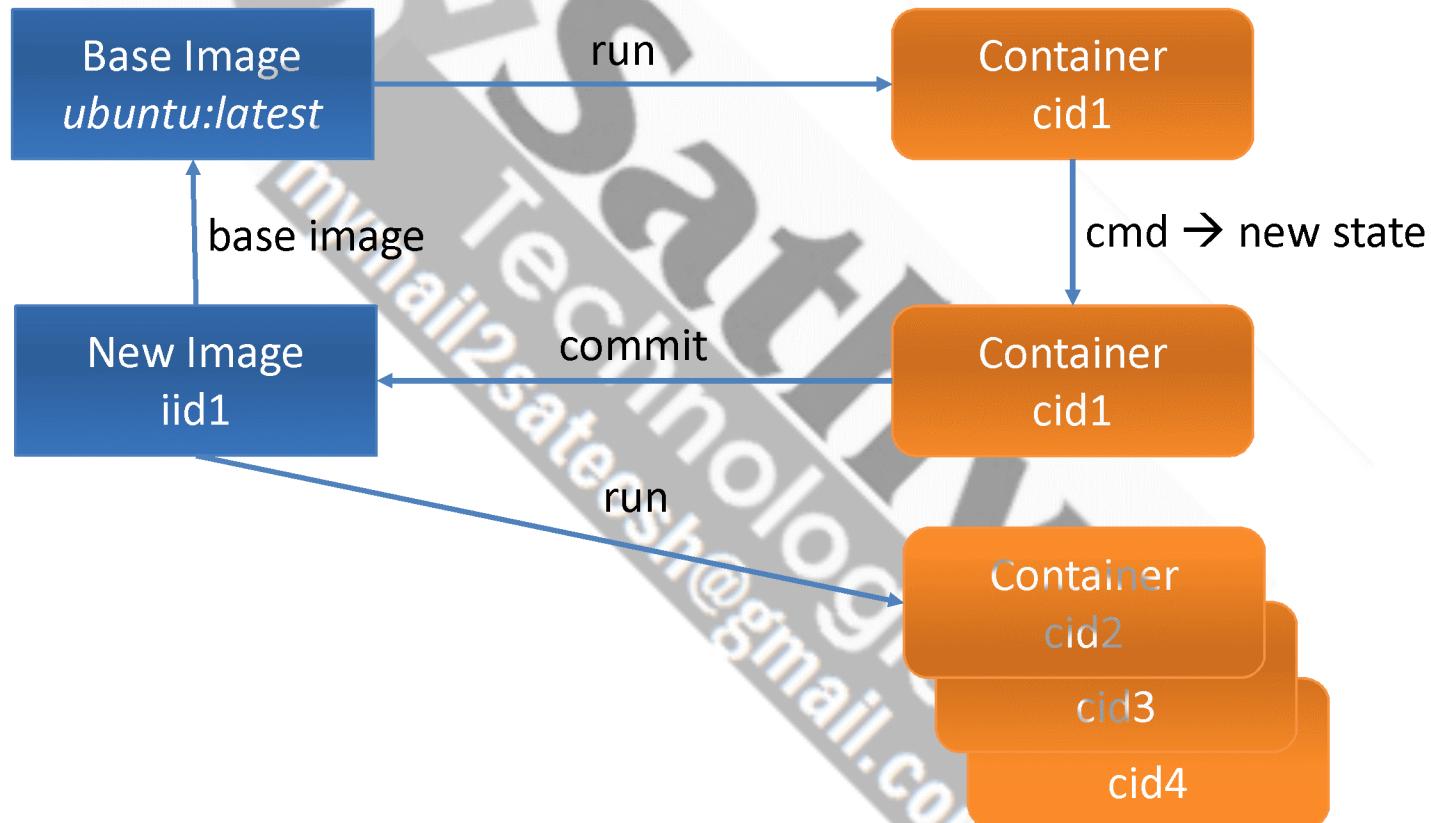
# Terminology - Image

- Persisted snapshot that can be run
  - ***images***: List all local images
  - ***run***: Create a container from an image and execute a command in it
  - ***tag***: Tag an image
  - ***pull***: Download image from repository
  - ***push***: to Push an image to DockerHub
  - ***rmi***: Delete a local image
    - This will also remove intermediate images if no longer used

# Terminology - Container

- Runnable instance of an image
  - ***ps***: List all running containers
  - ***ps -a***: List all containers (incl. stopped)
  - ***top***: Display processes of a container
  - ***start***: Start a stopped container
  - ***stop***: Stop a running container
  - ***pause***: Pause all processes within a container
  - ***rm***: Delete a container
  - ***commit***: Create an image from a container
  - ***attach*** : to Enter to running container

# Image vs. Container



# Hello World

## Simple Command - Ad-Hoc Container

- `docker run ubuntu echo Hello  
World`
  - `docker images [-a]`
  - `docker ps -a`

to pull the tomcat image from Docker Hub

```
#docker pull tomcat:8.0
```

```
#docker pull tomcat
```

to create a Container from Image

```
# docker run -it ubuntu
```

```
# docker run -d -p 9090:8080 tomcat:8.0
```

(-d : detached mode)

(-p : port Mapping)

( -it : interactive terminal)

```
#find / -name webapps
```

```
# docker run -it tomcat bash
```

**to create an Image from Container :**

---

```
#docker ps -a  
#docker commit <container id> ImageName  
#docker images
```

**to List all the Container IDs**

---

```
#docker ps -aq
```

**to Remove all the containers**

---

```
# docker rm $(docker ps -aq)
```

**to Remove a specific container:**

---

```
# docker rm 47e1bce974ef
```

**to Remove an Image:**

---

```
# docker rmi <image id>
```

**to Remove all the Images:**

---

```
# docker rmi $(docker images -aq)
```

# Docker Push

```
# docker run -it ubuntu bash  
# apt-get update  
# apt-get install apache2  
(press ctrl+pq to quit from Docker Container  
without stopping the Container)  
#docker ps  
#docker commit <container id> <Image Name>  
#docker tag <img Name> sathyadevops/apache  
#docker login  
#docker push sathyadevops/apache
```

```
# docker inspect <con id> / <image id>
# docker network ls
# docker network inspect bridge
# docker start <container id>
# docker attach <container id>
# docker stop <container id>
# docker run -itd centos /bin/bash
    yum install httpd -d
# docker commit <cont id> centos_httpd
# docker run -itd -p 80:80 centos_httpd
    /usr/sbin/bash
```