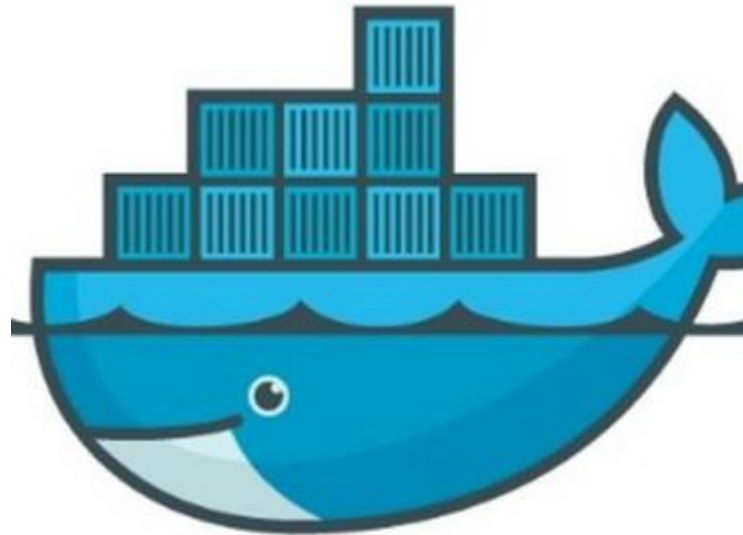
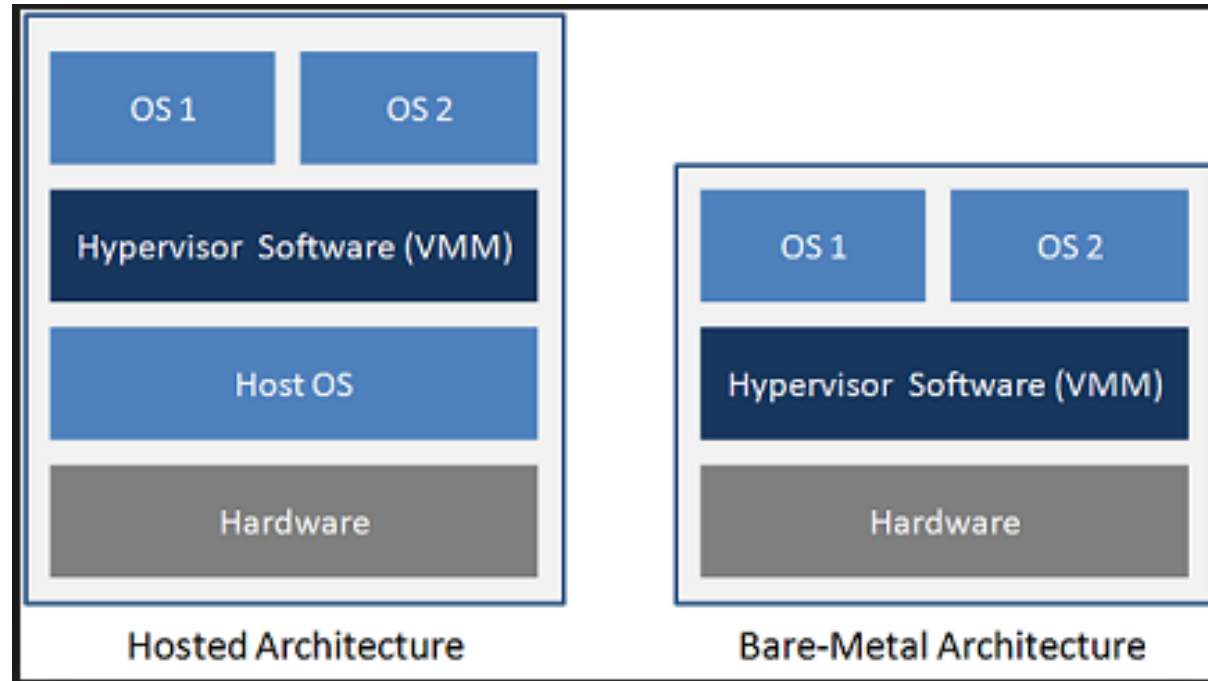


- Hi-Tech Institution | ph 7092 90 91 92, 82202 17640

Docker Basics



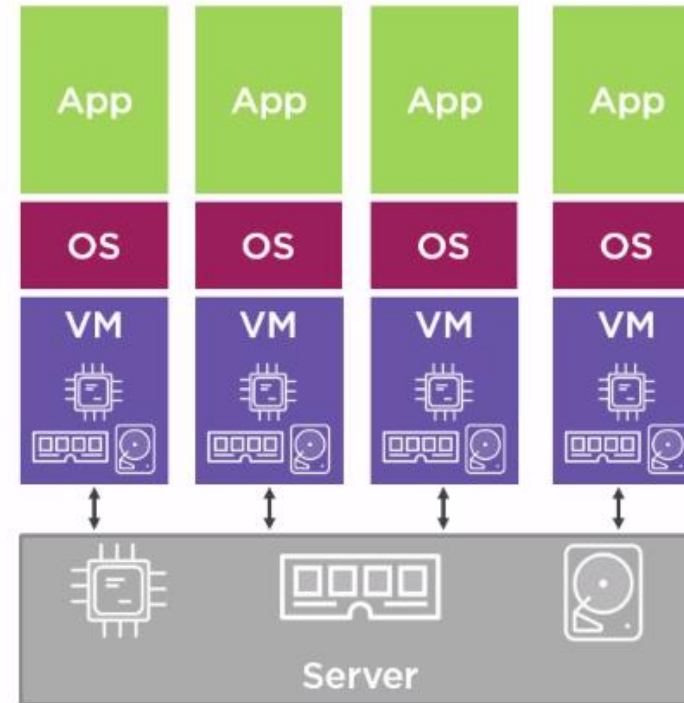
What is Hypervisor



- Hi-Tech Institution | ph 7092 90 91 92, 82202 17640

Issues with Hypervisors

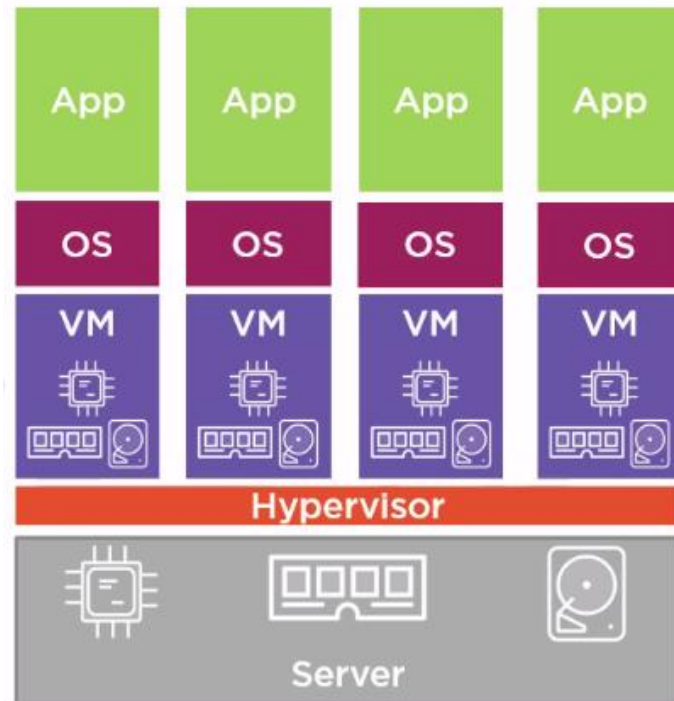
- Involves cost for
 - RAM
 - CPU
 - HDD
 - OS Licenses



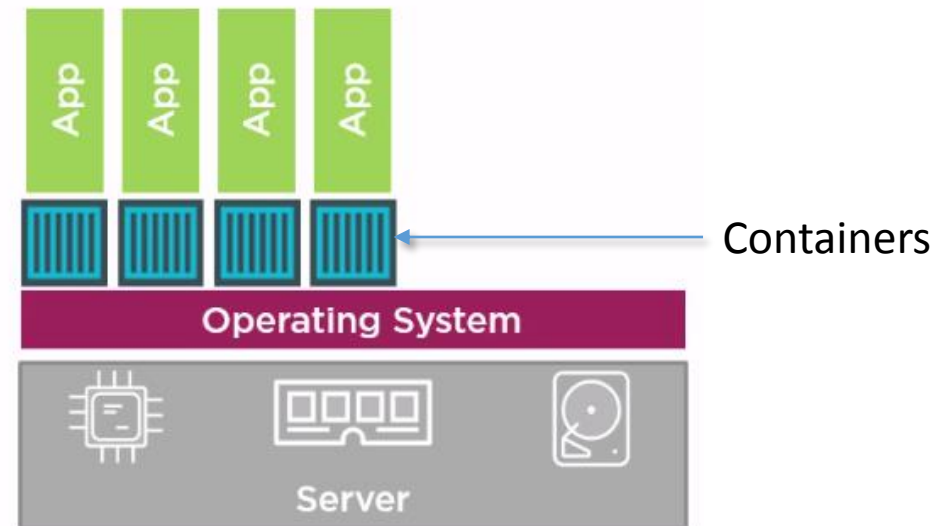
- Hi-Tech Institution | ph 7092 90 91 92, 82202 17640

- Hi-Tech Institution | ph 7092 90 91 92, 82202 17640

Hypervisor vs Container

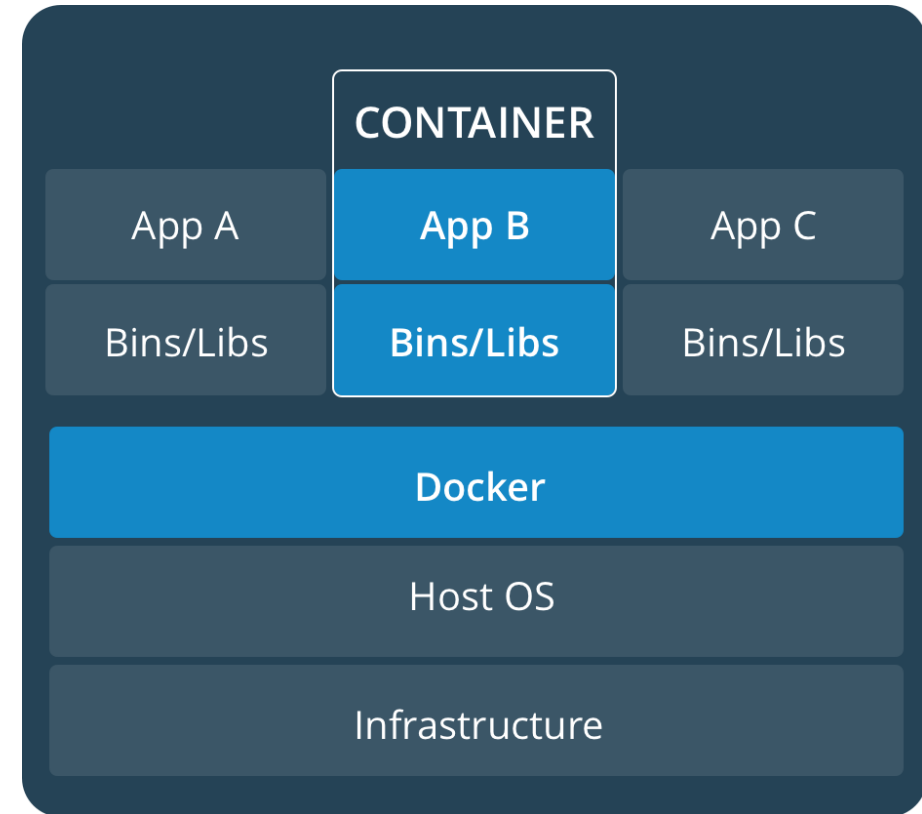
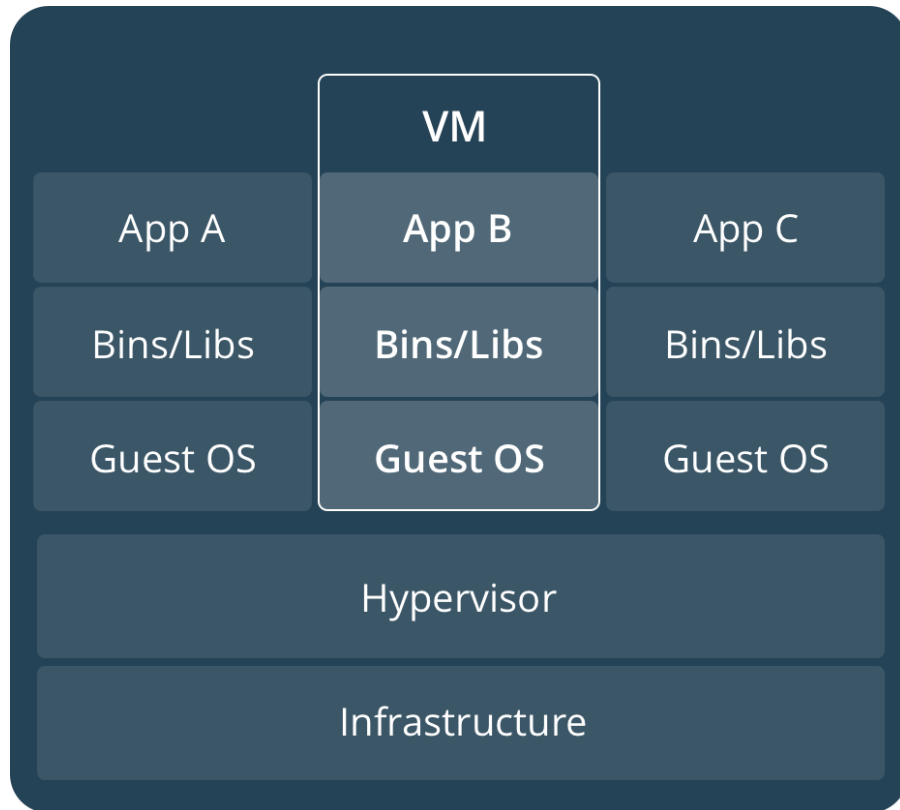


Hypervisor
Architecture



Container
Architecture

Hypervisor vs Container



- Hi-Tech Institution | ph 7092 90 91 92, 82202 17640

What are containers

- An **image** is a lightweight, stand-alone, executable package that includes everything needed to run a piece of software, including the code, a runtime, libraries, environment variables, and config files.
- A **container** is a runtime instance of an image—what the image becomes in memory when actually executed. It runs completely isolated from the host environment by default, only accessing host files and ports if configured to do so.
- **Containers run apps natively on the host machine's kernel.** They have better performance characteristics than virtual machines that only get virtual access to host resources through a hypervisor. Containers can get native access, each one running in a discrete process, taking no more memory than any other executable.

- Hi-Tech Institution | ph 7092 90 91 92, 82202 17640

Why containers?

Apps installed in an OS

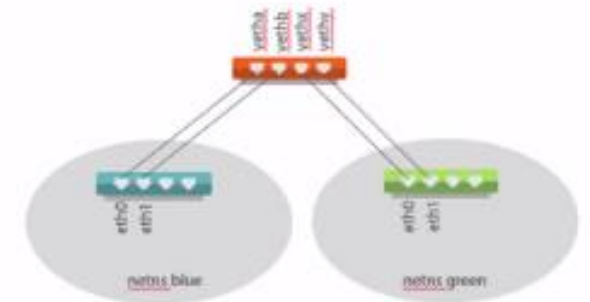
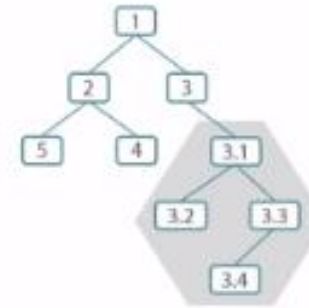
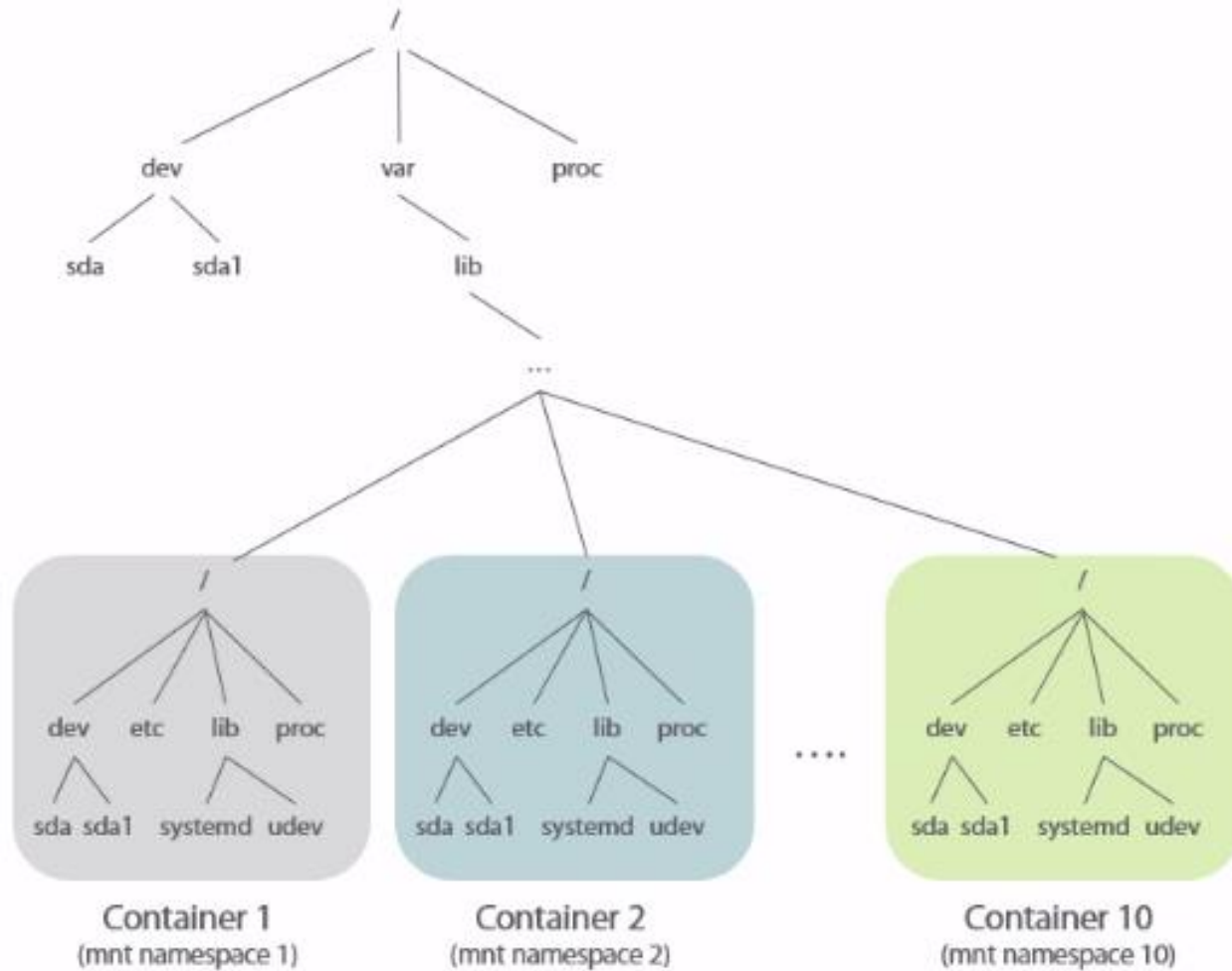


Apps installed through Containers



- Hi-Tech Institution | ph 7092 90 91 92, 82202 17640

How Containers Work?



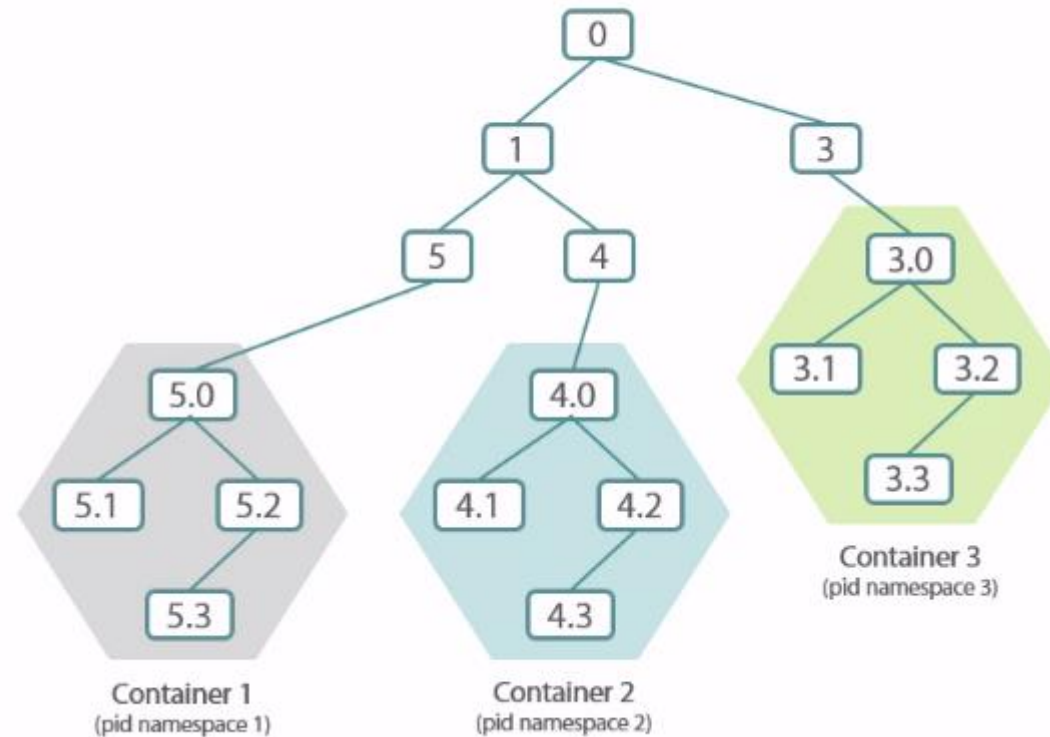
Kernel Namespaces

The **pid** Namespace

The **net** Namespace

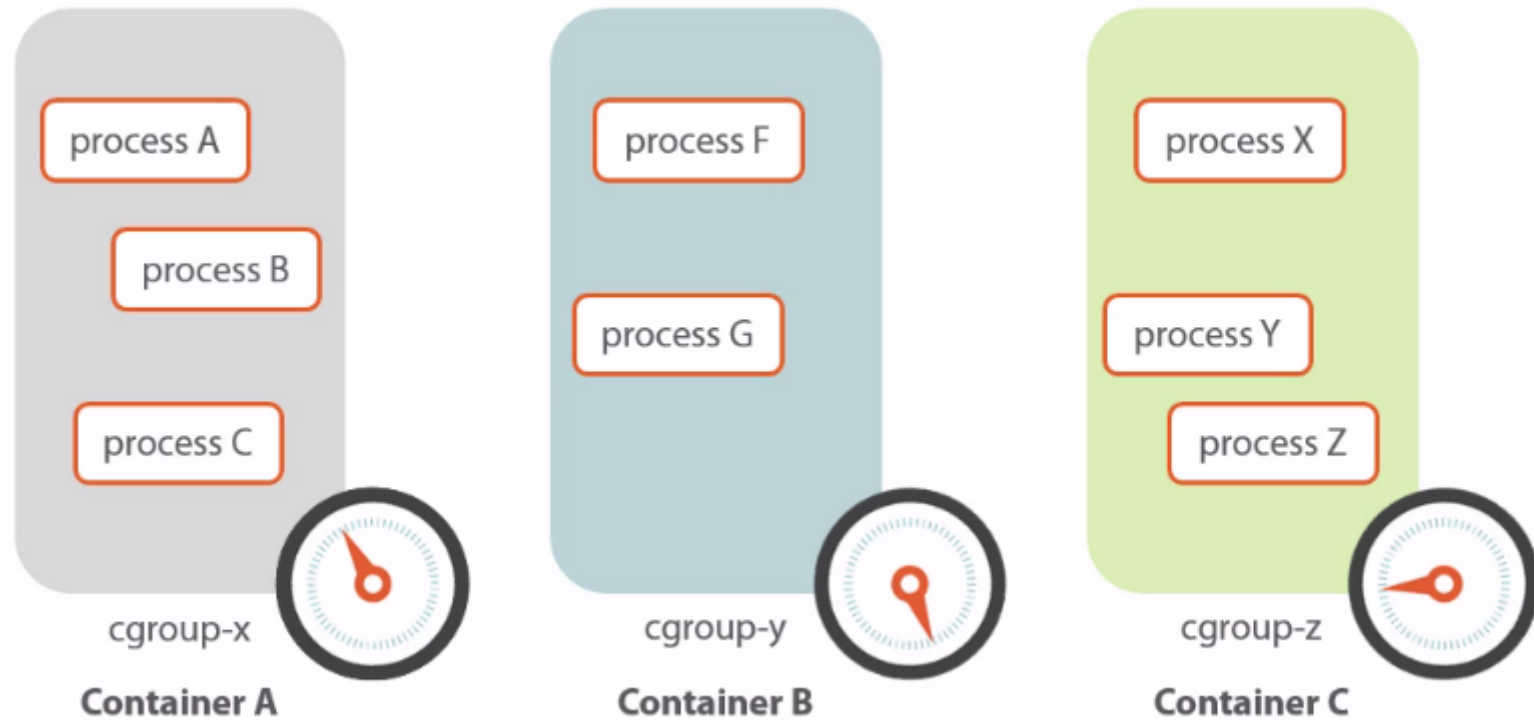
The **mnt** Namespace

The **user** Namespace



- Hi-Tech Institution | ph 7092 90 91 92, 82202 17640

Control Groups(cgroups)



Other underlying technologies

- Union file systems
 - Union file systems, or UnionFS, are file systems that operate by creating layers, making them very lightweight and fast. Docker Engine uses UnionFS to provide the building blocks for containers. Docker Engine can use multiple UnionFS variants, including AUFS, btrfs, vfs, and DeviceMapper.
- Container format
 - Docker Engine combines the namespaces, control groups, and UnionFS into a wrapper called a container format. The default container format is **libcontainer**.
 - Hi-Tech Institution | ph 7092 90 91 92, 82202 17640

Introduction to Docker

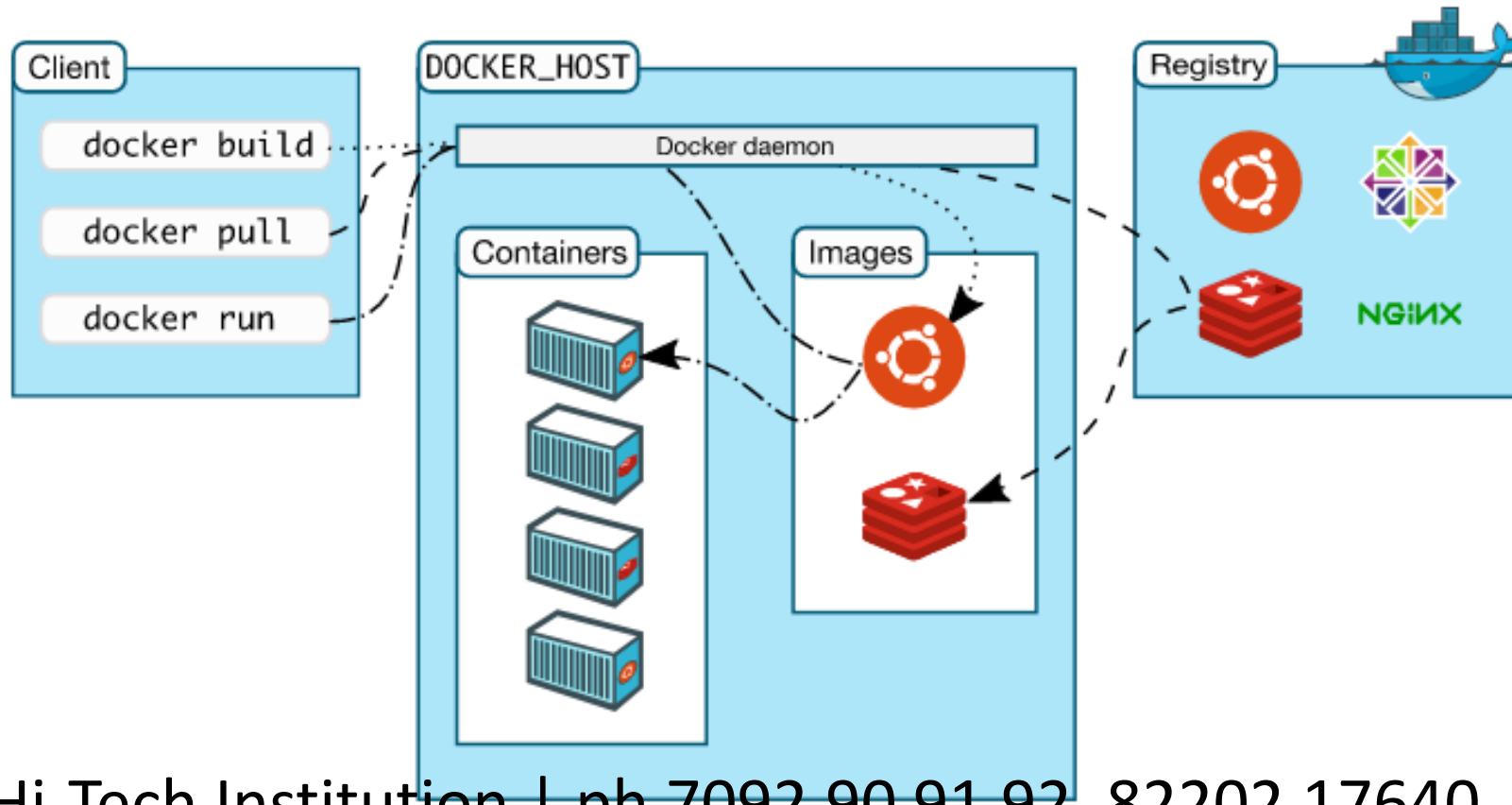
- Docker was released as an open source project by dotCloud, a platform as a service company, in 2013, founded by Solomon Hykes
- Docker relies on Linux kernel features, such as namespaces and cgroups, to ensure resource isolation and to package an application along with its dependencies.

• Hi-Tech Institution | ph 7092 90 91 92, 82202 17640

Docker Overview

- Docker provides the ability to package and run an application in a loosely isolated environment called a container.
- The isolation and security allow you to run many containers simultaneously on a given host.
- Containers are lightweight because they don't need the extra load of a hypervisor, but run directly within the host machine's kernel.
- This means you can run more containers on a given hardware combination than if you were using virtual machines. You can even run Docker containers within host machines that are actually virtual machines!
 - Hi-Tech Institution | ph 7092 90 91 92, 82202 17640

Docker Architecture



- Hi-Tech Institution | ph 7092 90 91 92, 82202 17640

- Hi-Tech Institution | ph 7092 90 91 92, 82202 17640

Docker Architecture

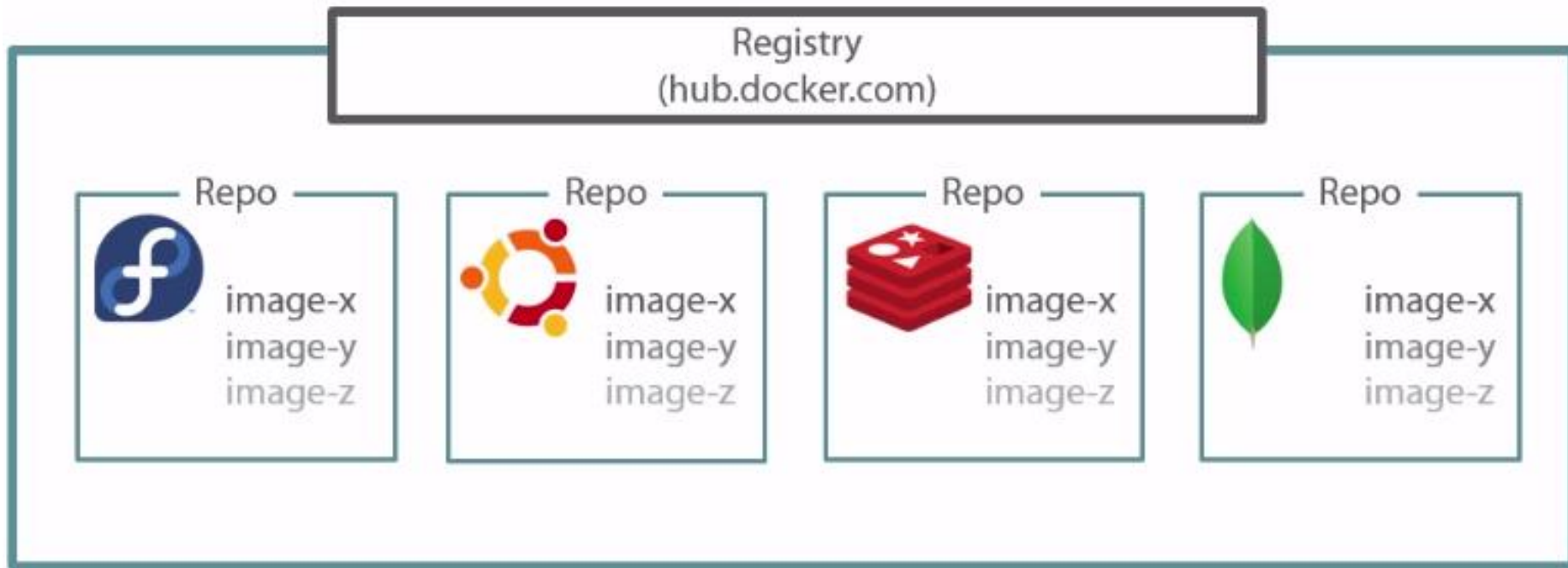
- The Docker daemon
 - The Docker daemon (dockerd) listens for Docker API requests and manages Docker objects such as images, containers, networks, and volumes.
- The Docker client
 - The Docker client (docker) is the primary way that many Docker users interact with Docker. When you use commands such as **docker run**, the client sends these commands to dockerd, which carries them out.
- Docker registries
 - A Docker registry stores Docker images. Docker Hub and Docker Cloud are public registries that anyone can use, and Docker is configured to look for images on Docker Hub by default.

- Hi-Tech Institution | ph 7092 90 91 92, 82202 17640

Docker Architecture

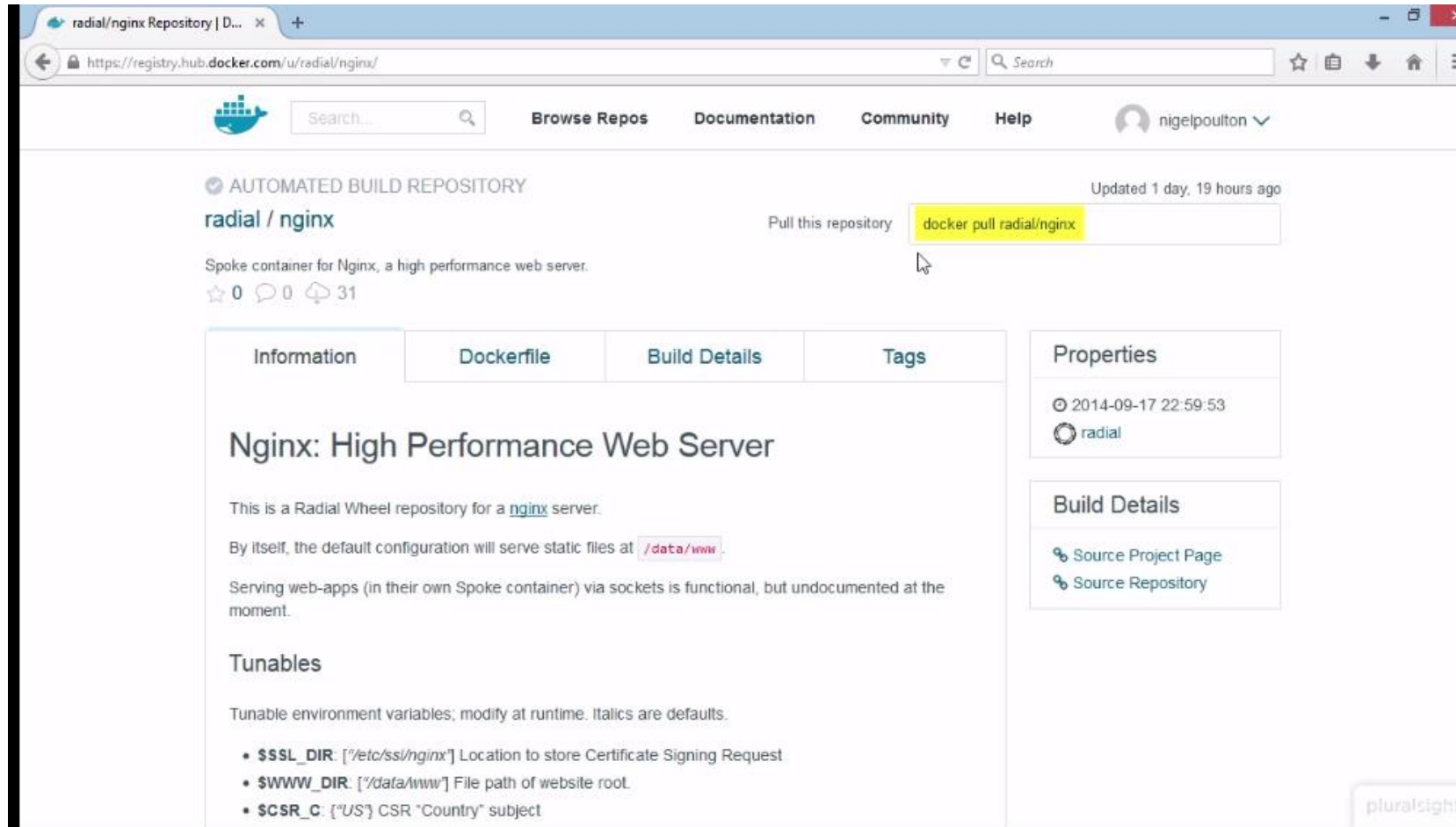
- Docker objects
 - IMAGES
 - An image is a read-only template with instructions for creating a Docker container. Often, an image is based on another image, with some additional customization. For example, you may build an image which is based on the ubuntu image, but installs the Apache web server and your application, as well as the configuration details needed to make your application run.
 - CONTAINERS
 - A container is a runnable instance of an image. You can create, run, stop, move, or delete a container using the Docker API or CLI. You can connect a container to one or more networks, attach storage to it, or even create a new image based on its current state.
 - SERVICES
 - Services allow you to scale containers across multiple Docker daemons, which all work together as a swarm with multiple managers and **workers**.

Registries and Repositories



- Hi-Tech Institution | ph 7092 90 91 92, 82202 17640

Registries and Repositories



The screenshot shows a web browser window displaying the Docker Hub page for the `radial/nginx` repository. The browser's address bar shows the URL `https://registry.hub.docker.com/u/radial/nginx/`. The page header includes the Docker logo, a search bar, and navigation links for `Browse Repos`, `Documentation`, `Community`, and `Help`. The user `nigelpoulton` is logged in.

The repository page for `radial / nginx` is shown, marked as an `AUTOMATED BUILD REPOSITORY`. It includes a description: "Spoke container for Nginx, a high performance web server." and statistics: 0 stars, 0 comments, and 31 pulls. A button labeled "Pull this repository" is next to a yellow box containing the command `docker pull radial/nginx`.

The main content area has tabs for `Information`, `Dockerfile`, `Build Details`, and `Tags`. The `Information` tab is active, displaying the title `Nginx: High Performance Web Server`. The description states: "This is a Radial Wheel repository for a `nginx` server. By itself, the default configuration will serve static files at `/data/www`. Serving web-apps (in their own Spoke container) via sockets is functional, but undocumented at the moment."

Under the `Tunables` section, it says: "Tunable environment variables; modify at runtime. Italics are defaults." and lists the following variables:

- `$SSL_DIR`: `[/etc/ssl/nginx]` Location to store Certificate Signing Request
- `$WWW_DIR`: `[/data/www]` File path of website root.
- `$CSR_C`: `{US}` CSR "Country" subject

On the right side, the `Properties` section shows the build time `2014-09-17 22:59:53` and the user `radial`. The `Build Details` section provides links to the `Source Project Page` and `Source Repository`.

Install Docker in Linux

- The Docker installation package available in the official Ubuntu 16.04 repository may not be the latest version. To get the latest and greatest version, install Docker from the official Docker repository. This section shows you how to do just that.

- First, add the GPG key for the official Docker repository to the system:

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -

- Add the Docker repository to APT sources:

sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu \$(lsb_release -cs) stable"

- Hi-Tech Institution | ph 7092 90 91 92, 82202 17640

- Hi-Tech Institution | ph 7092 90 91 92, 82202 17640

Install Docker in ubuntu

- Next, update the package database with the Docker packages from the newly added repo:

sudo apt-get update

Make sure you are about to install from the Docker repo instead of the default Ubuntu 16.04 repo:

apt-cache policy docker-ce

You should see output similar to the follow:

docker-ce:

Installed: (none)

Candidate: 17.03.1~ce-0~ubuntu-xenial

Version table:

17.03.1~ce-0~ubuntu-xenial 500

500 [https://download.docker.com/linux/ubuntu xenial/stable amd64](https://download.docker.com/linux/ubuntu/xenial/stable/amd64) Packages

Install Docker in RHEL7/Centos

- `yum-config-manager --add-repo`
`https://download.docker.com/linux/centos/docker-ce.repo`
- `yum install`
`http://mirror.centos.org/centos/7/extras/x86_64/Packages/container-selinux-2.68-1.el7.noarch.rpm`
- `yum install docker-ce`

Install Docker in Linux

- Finally, install Docker:

sudo apt-get install -y docker-ce

- Docker should now be installed, the daemon started, and the process enabled to start on boot. Check that it's running:

sudo systemctl status docker

To install Ubuntu in docker

\$docker search Ubuntu

\$docker pull ubuntu

Docker commands

- Docker -v
 - Shows the current version of docker installed in the docker host(server in which docker is installed)
- Docker images
 - Shows list of pulled images
- **Docker run -it <name of image>**
 - Runs the docker container from it's name
- **docker run --name nginx -it nginx**
 - This would give a name for the container nginx which will be displayed with the following command
- Docker ps
 - Shows list of running containers
- Docker ps -a
 - Shows list of recently closed containers with their ID

Docker commands

- **Docker start ContainerID**
 - Start a container using the Container ID
- **Docker attach ContainerID**
 - Opens the prompt inside the container
- Exit from the container prompt without stopping the container
 - **Ctrl+P+Q**
- Docker info
 - Shows information related to docker engine
- **Docker rm ContainerID**
 - This deletes a container
- Check normal Linux commands inside the container

Docker commands

- **docker cp SRC_PATH CONTAINER:DEST_PATH**
 - E.g. `#docker cp new.tar test:/root`
- **docker exec -d nameofcontainer** commands
 - E.g. `#docker exec -d test touch /opt/file`
 - This executes commands inside a running container

Ph or Whats app : 7092 90 91 92, 82202 17640

