Docker

Hello!

I am Sandeep Anuragi

I am here because I love to do docker Integration.

You can find me at blackmagiclinux@gmail.com

1.

What is Docker?

Let’s start with the first set of slides

Docker is a Lightweight virtual machine. But unlike a virtual machine, rather than create a whole virtual machine OS.

Docker Allows Applications to use same linux kernel as the system that they’re running on and only requires applications to shipped with things not already running on the host computer.

“

Docker is a tool designed to make it easier to create, deploy, and run applications by using containers.

Container allow to deploy an application with all of the parts it needs. Such as libraries and other dependencies, and deploy it as one package.

Who is Docker for?

Docker is a tool that is designed to benefit both developers and system administrators, making it a part of many DevOps (developers + operations) toolchains.

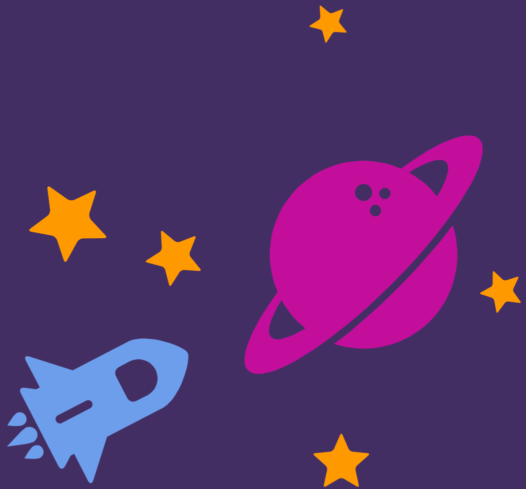
Docker is one of the container implementations available for deployment and supported by companies such as Red Hat in their Red Hat Enterprise Linux Atomic Host platform. Docker Hub provides a large set of containers developed by the community.

More info on how to use docker visit at http://trainingbasket.in

Understanding containers

- Builder : used to build a container
- Engine : used to run a container
- Orchestration: used to manage multi-container

Linux containers have facilitated a massive shift in **high-availability** computing, and there are many toolsets out there to help you run services in containers. Docker is one option among many, as defined by **Open Container Initiative (OCI)**, an industry standards organization meant to encourage innovation whilst avoiding the danger of vendor lock-in.



Big concept

DOCKER CORE ELEMENTS

DOCKER CORE ELEMENTS

Images

Images are read-only templates that contain a runtime environment that includes application libraries and applications. Images are used to create containers. Images can be created, updated, or downloaded for immediate consumption.

Registries

Registries store images for public or private use. The well-known public registry is Docker Hub, and it stores multiple images developed by the community, but private registries can be created to support internal image development under a company's discretion.

DOCKER CORE ELEMENTS CONT.

Containers

Containers are segregated user-space environments for running applications isolated from other applications sharing the same host OS.

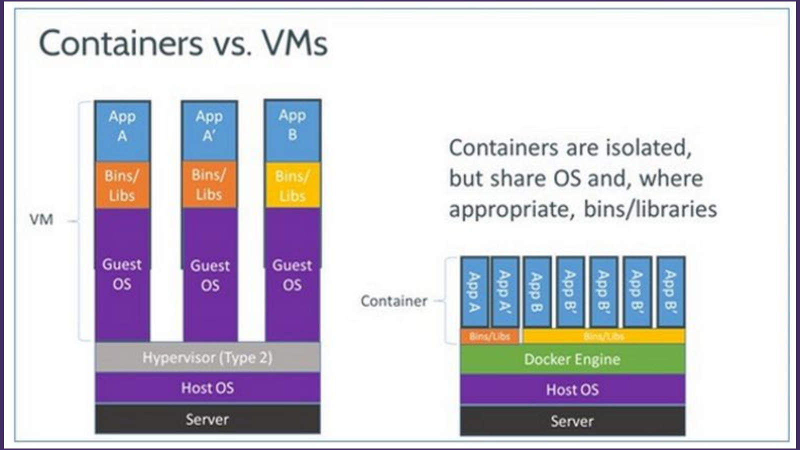
Namespaces

Namespaces The kernel can place specific system resources that are normally visible to all processes into a namespace

Control group

Control groups partition sets of processes and their children into groups in order to manage and limit the resources they consume.

DOCKER CONTAINER IMAGES



1. Each image in Docker consists of a series of layers that are combined into what is seen by the containerized applications a single virtual file system.
2. Docker images are immutable; any extra layer added over the preexisting layers overrides their contents without changing them directly.

There are two approaches to create a new image:

Using a running container

An immutable image is used to start a new container instance and any changes or updates needed by this container are made to a read/write extra layer.

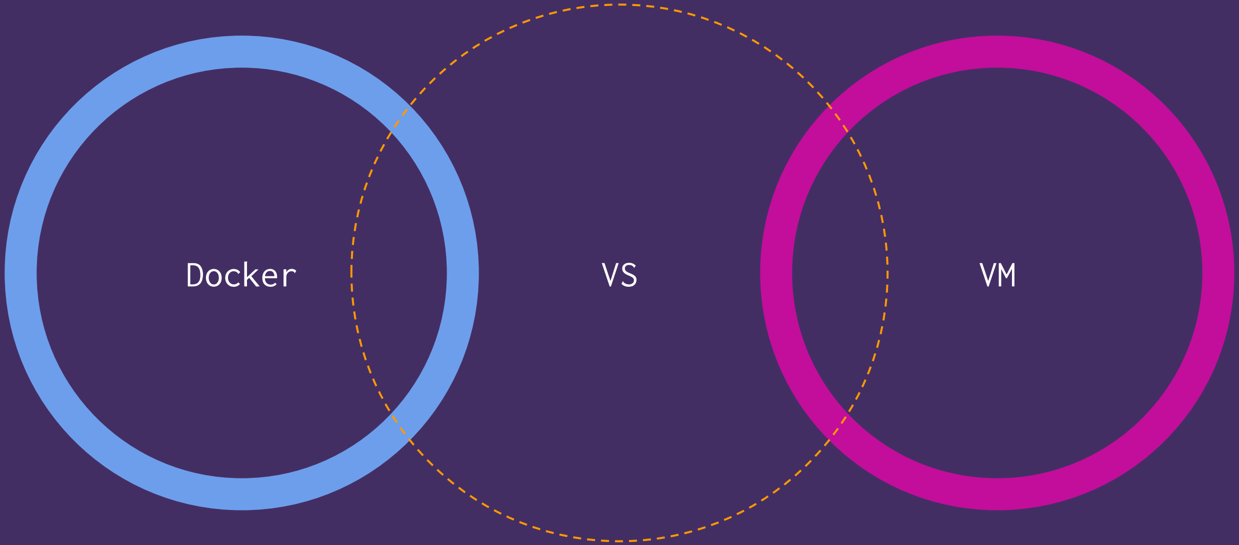
this approach is the easiest way to create images, but it is not a recommended approach because the image size might become large due to unnecessary files, such as temporary files and logs

Using a Dockerfile

Alternatively, container images can be built from a base image using a set of steps called instructions. Each instruction creates a new layer on the image that is used to build the final container image.

Want big impact?
Use big image.

Docker vs. VM – who wins?



Virtual Machines

What are VMs?

A virtual machine (VM) is an emulation of a computer system. Put simply, it makes it possible to run what appear to be many separate computers on hardware that is actually one computer.

Benefits of VMs?

- All OS resources available to apps
- Established management tools
- Established security tools
- Better known security control

Containers

What are Containers?

With Containers, instead of virtualizing the underlying computer like a virtual machine(VM), just is the OS is virtualized.

Benefits of Containers?

- Reduce IT management resources.
- Reduce size of snapshots
- Quicker spinning up apps
- Less code to transfer, migrate, upload workloads
- Reduce & simplified security updates.

Containers provider Facilities

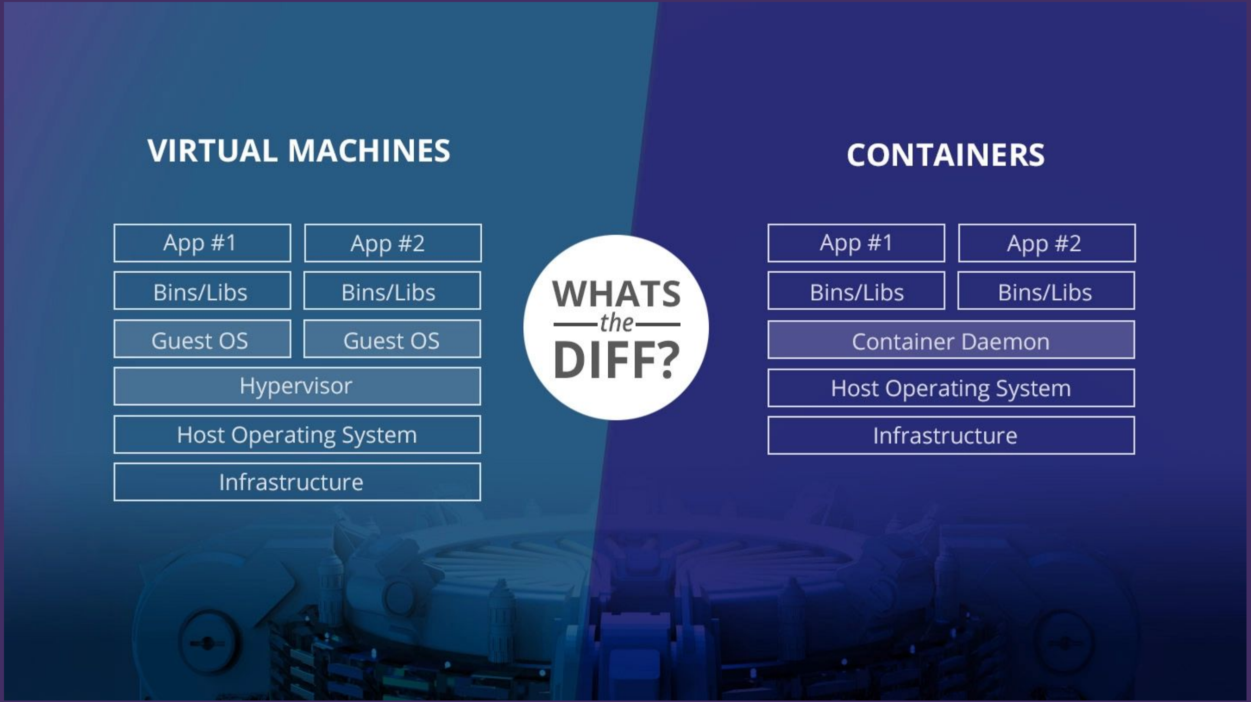
Popular VM Providers

- Vmware vSphere
- VirtualBox
- Xen
- Hyper-V
- KVM

Popular Containers Providers

- Linux containers
  - LXC
  - LXD
  - CGManager
- Docker
- Windows Server Containers

What’s the Diff: VMs vs Containers



There are two approaches to create a new image:

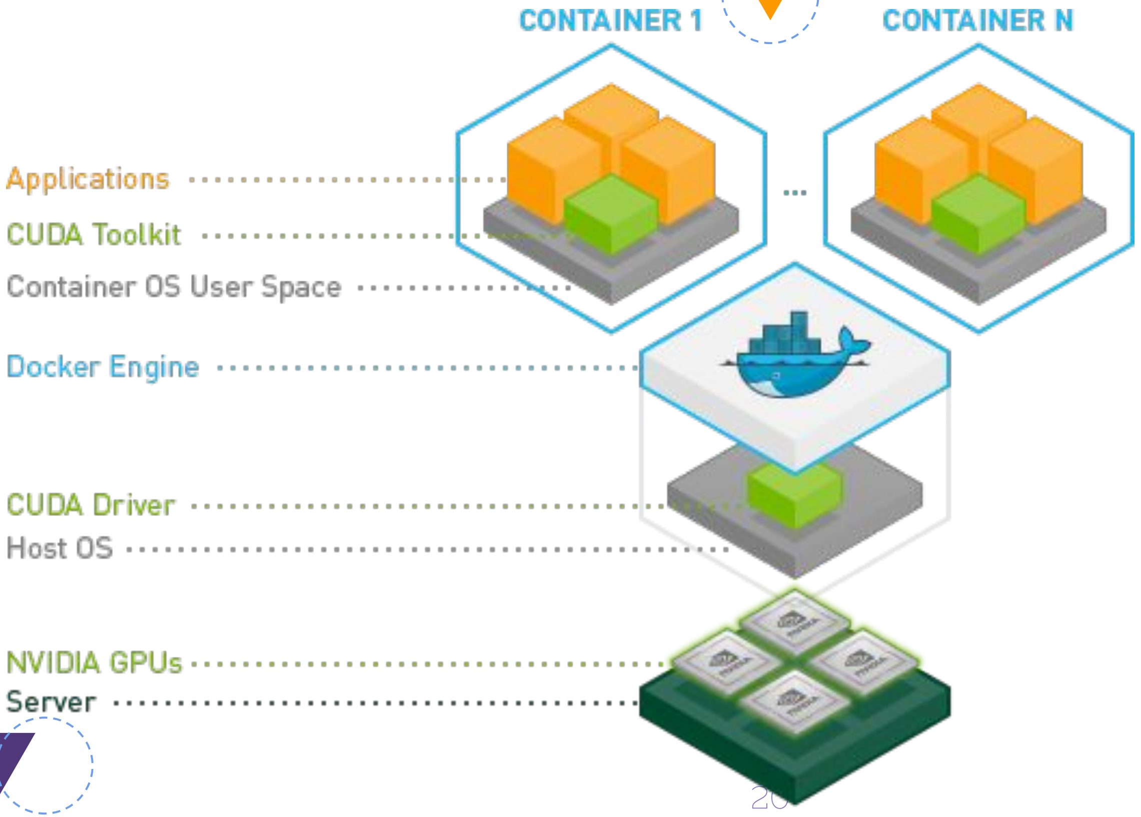
VMS

- Heavyweight
- Limited performance
- Each VM run in its own OS
- Hardware-level virtualization
- Startup time in minutes
- Allocates required memory
- Fully isolated

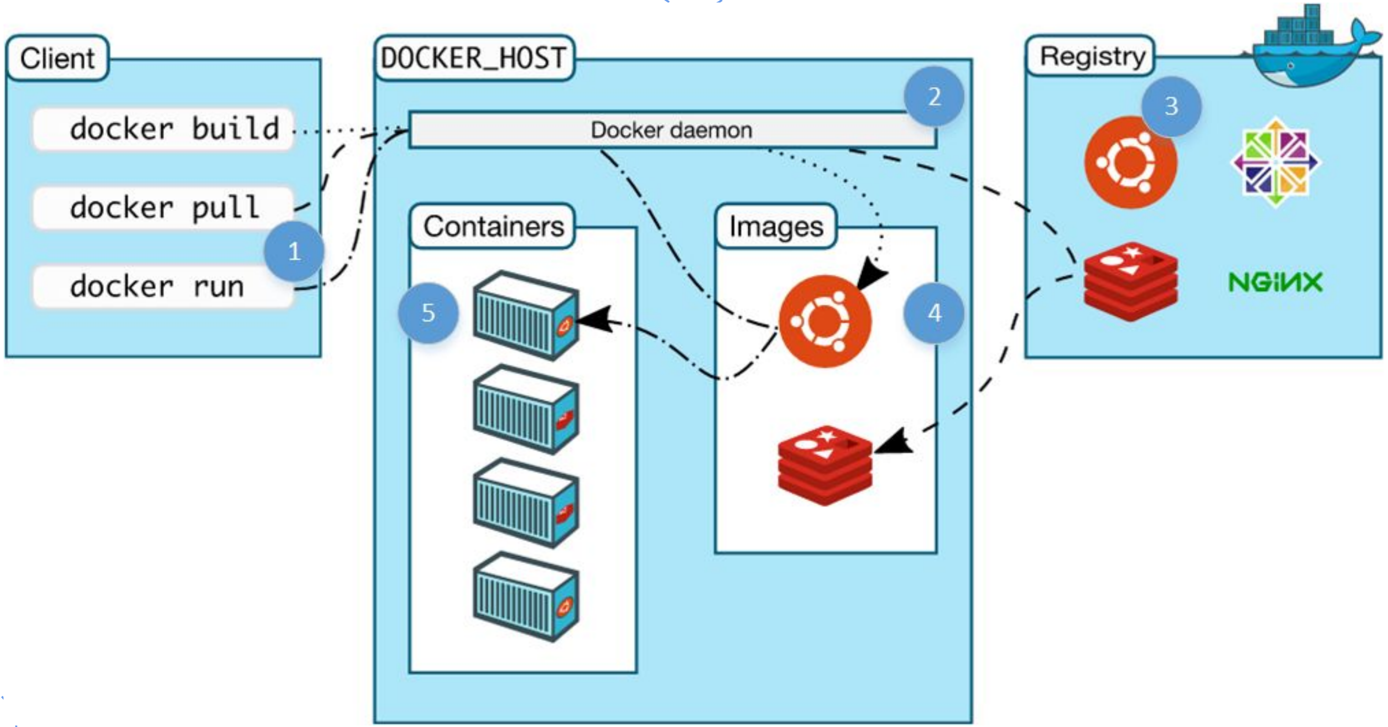
Containers

- Lightweight
- Native performance
- All containers share the host OS
- OS-level Virtualization
- Startup time in milliseconds
- Require less memory space
- Process level isolation

Internally Features



Working Role



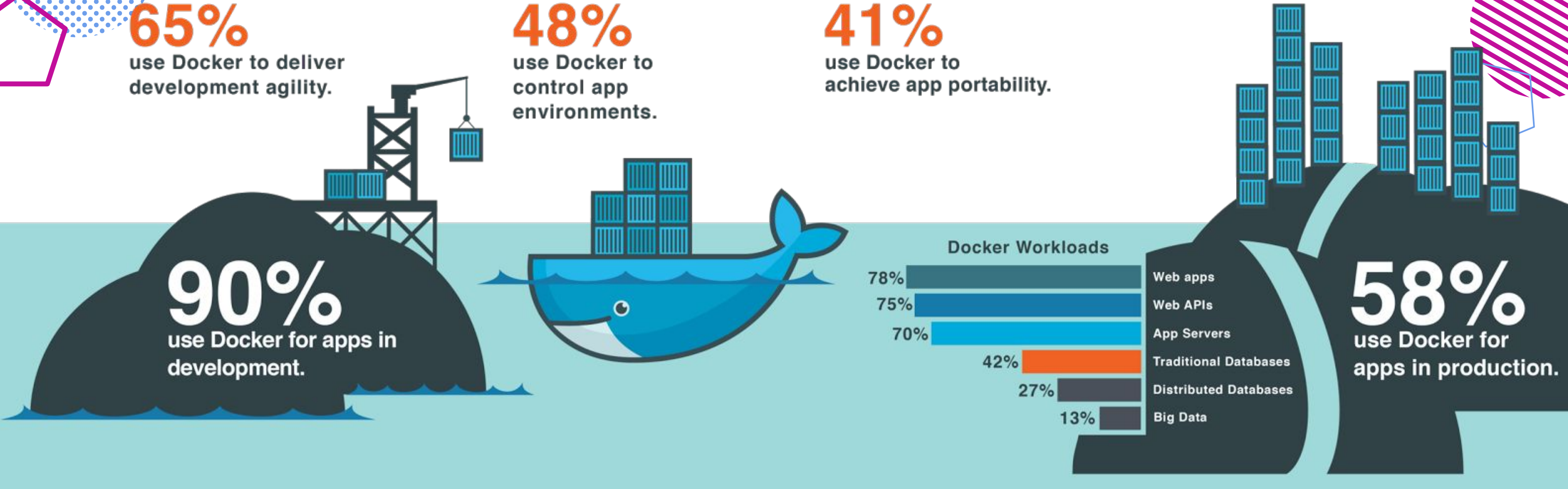
65% use Docker to deliver development agility.

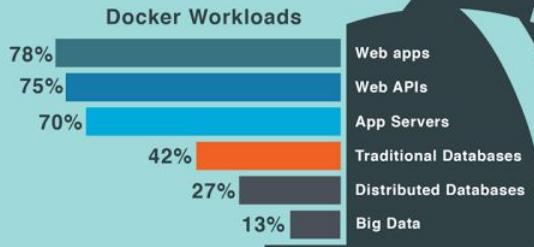
48% use Docker to control app environments.

41% use Docker to achieve app portability.

90% use Docker for apps in development.

58% use Docker for apps in production.





90% plan dev environments around Docker.

80% plan DevOps around Docker.

docker

Our process is easy

BUILD

SHIP

RUN

Thanks For Join Us

blackmagiclinux@gmail.com

Are you Ready to face of challenges

Comment and Survey fo this course