



IIUG  2016

May 4<sup>th</sup> - 8<sup>th</sup>

Ponte Vedra Beach  
Florida, USA

# Informix HA using Docker containers

Nagaraju Inturi

Informix Replication team

[nagaraju@us.ibm.com](mailto:nagaraju@us.ibm.com)

# What is Docker

- Open-source project
- Automates the deployment of applications inside software containers
- Docker uses the resource isolation features of the Linux kernel such as cgroups and kernel namespaces

# What is Docker

Docker allows you to package an application with all of its dependencies into a standardized unit for software development.

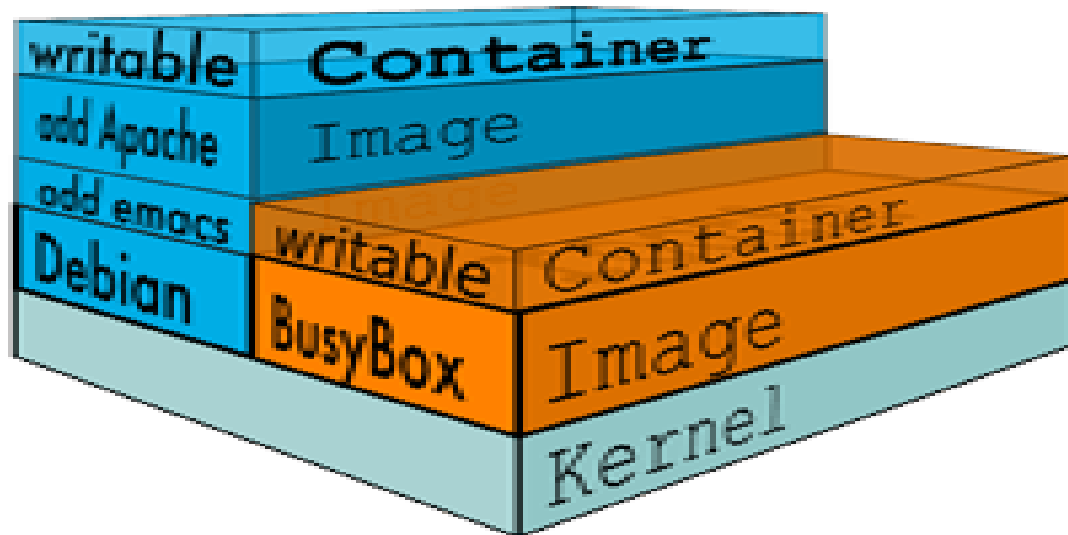
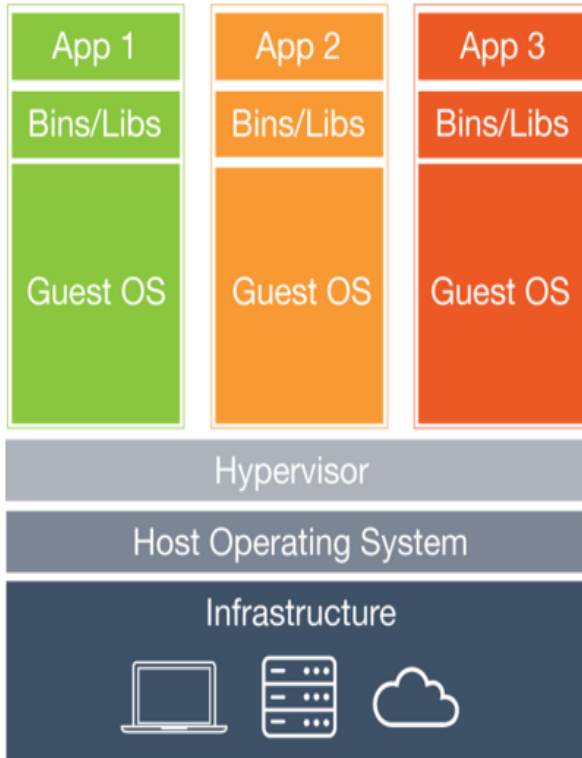


Image source: [docker.com](https://docker.com)

# What is Docker

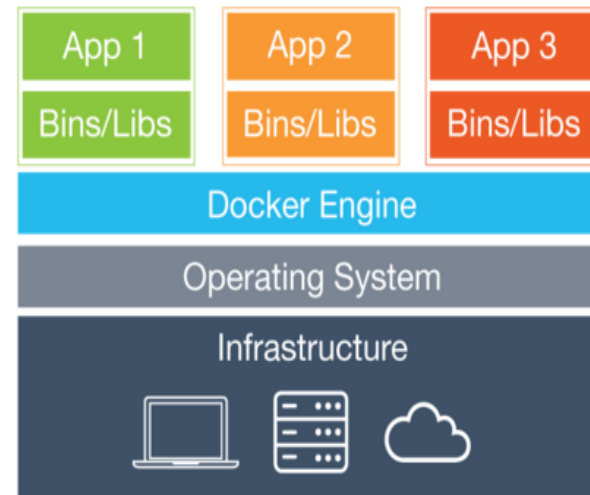
- In docker environment what developer builds is a tar file – called docker image.
- Docker Engine overlays this tar file on the underlying Linux OS using overlay filesystem called AUFS.
- Docker runtime environment – called container—uses this overlay and runs a command with a certain environment inside the docker image.

# Container vs. Virtual Machine



## Virtual Machines

Each virtual machine includes the application, the necessary binaries and libraries and an entire guest operating system - all of which may be tens of GBs in size.



## Containers

Containers include the application and all of its dependencies, but share the kernel with other containers. They run as an isolated process in userspace on the host operating system. They're also not tied to any specific infrastructure – Docker containers run on any computer, on any infrastructure and in any cloud.

# Docker Value addition

- To the developer:
  - The developer can build software on their choice of Linux distribution along “with all the dependencies”. Then at deployment time everything is exactly the same. The developer is responsible and able to ensure all of the dependencies are met.
- To the operations Engineer
  - Operations can keep deploying their chosen OS, but the developers can drop an image based on a different distro with the right tools on top of that. Operations can manage/patch the host OS as necessary. With docker the two don't start conflicting with each other.

# Docker terminology

- Container :
  - A container is a runtime instance of a docker image.
- Image
  - Docker images are the basis of containers. An Image is an ordered collection of root filesystem changes and the corresponding execution parameters for use within a container runtime. An image typically contains a union of layered filesystems stacked on top of each other. An image does not have state and it never changes.
- Docker Engine:
  - A lightweight runtime and robust tooling that builds and runs your Docker containers.

# Docker terminology

- Dockerfile
  - A Dockerfile is a text document that contains all the commands you would normally execute manually in order to build a Docker image. Docker can build images automatically by reading the instructions from a Dockerfile.
- Compose
  - Compose is a tool for defining and running complex applications with Docker. With compose, you define a multi-container application in a single file, then spin your application up in a single command which does everything that needs to be done to get it running.
- Docker Machine:
  - Docker Machine provisions the hosts, installs Docker Engine on them, and then configures the Docker client to talk to the Docker Engines. Docker Machine can create hosts on most major virtualization hypervisors and in cloud service providers.
- Docker Registry:
  - Docker Registry is an open source application dedicated to the storage and distribution of your Docker images.



# Docker terminology

- Docker Swarm:
  - Docker Swarm provides native clustering capabilities to turn a group of Docker engines into a single, virtual Docker Engine.
- Data volume
  - A data volume is a specially-designated directory within one or more containers that bypasses the Union File System. Data volumes are designed to persist data, independent of the container's life cycle. Docker therefore never automatically delete volumes when you remove a container, nor will it “garbage collect” volumes that are no longer referenced by a container.
- Aufs
  - aufs (advanced multi layered unification filesystem) is a Linux filesystem that Docker supports as a storage backend. It implements the union mount for Linux file systems.

# Docker commands used in this LAB

- `docker build [OPTIONS] PATH | URL | -`
  - Builds Docker images from a Dockerfile and a “context”.
  - Example: `$ docker build -t iiug/apache:2.0 .`
- `docker run [OPTIONS] IMAGE [COMMAND] [ARG...]`
  - To create and start docker container from docker image
  - `$ docker run --name ubuntu_bash --rm -i -t ubuntu bash`
    - This will create a container named `ubuntu_bash` and start a Bash session.
- `docker exec [OPTIONS] CONTAINER COMMAND [ARG...]`
  - Runs a new command in a running container.
  - `$ docker exec -it ubuntu_bash bash`
    - This will create a new Bash session in the container `ubuntu_bash`.

# Dockerfile instructions

- **FROM <image>:<tag>**
  - This instruction sets the Base Image for subsequent instructions.
- **MAINTAINER <name>**
  - This instruction allows you to set the Author field of the generated images.
- **RUN**
  - The RUN instruction will execute any commands in a new layer on top of the current image and commit the results. The resulting committed image will be used for the next step in the Dockerfile.
- **ENTRYPOINT**
  - An ENTRYPOINT allows you to configure a container that will run as an executable.
- **CMD**
  - The main purpose of a CMD is to provide defaults for an executing container. CMD can be used to provide default arguments for the ENTRYPOINT
- **VOLUME ["/data"]**
  - The VOLUME instruction creates a mount point with the specified name and marks it as holding externally mounted volumes from native host or other containers.

# Dockerfile instructions

- EXPOSE <port> [<port>...]
  - The EXPOSE instruction informs Docker that the container listens on the specified network ports at runtime.
- ENV
  - The ENV instruction sets the environment variable <key> to the value <value>.
- ADD
  - The ADD instruction copies new files, directories or remote file URLs from <src> and adds them to the filesystem of the container at the path <dest>.
- USER username
  - The USER instruction sets the user name or UID to use when running the image and for any RUN, CMD and ENTRYPOINT instructions that follow it in the Dockerfile.
- WORKDIR /path/to/workdir
  - The WORKDIR instruction sets the working directory for instructions that follow it in the Dockerfile.

# What are we building in this LAB ?

- Docker image for Informix server.
- Docker image for Informix connection manager
- Informix HA cluster with 6 containers.
  - Primary
  - SDS
  - HDR
  - RSS
  - CM1
  - CM2

# Block Diagram

Docker containers for HA cluster

