Cloud Security with

docker

James Knight
Dmitriy Burmistrov

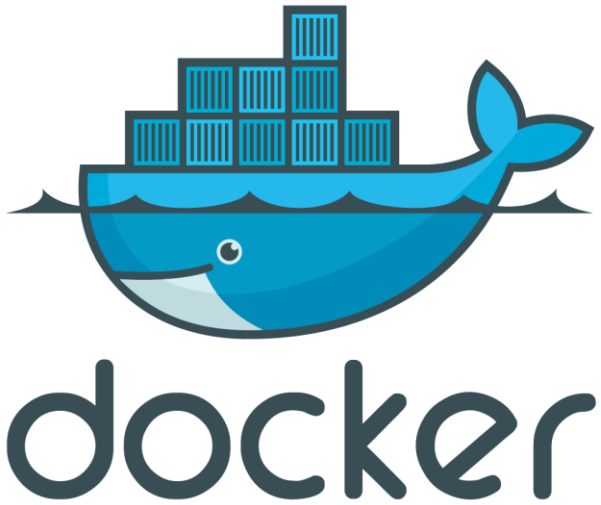
Introduction

“In a nutshell, here's what Docker can do for you:

1) It can get more applications running on the same hardware than other technologies

2) It makes it easy for developers to quickly create, ready-to-run containered applications

3) It makes managing and deploying applications much easier. ” - ZDNET



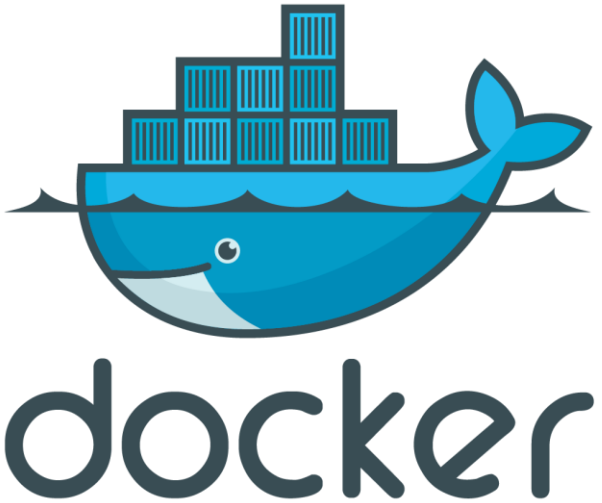
Presentation Outline

Background Information

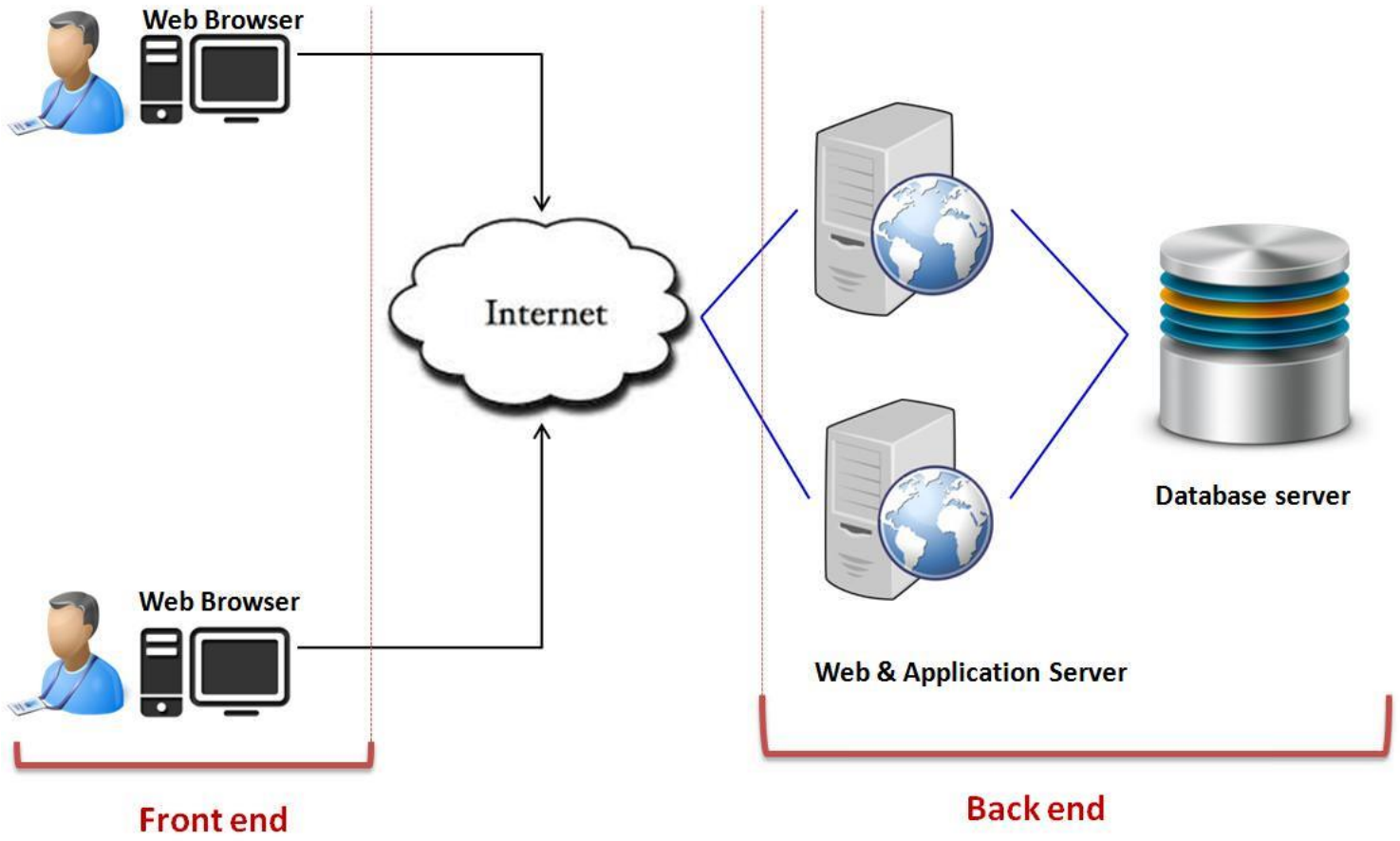
Overview of Technology

Interactive Walkthrough

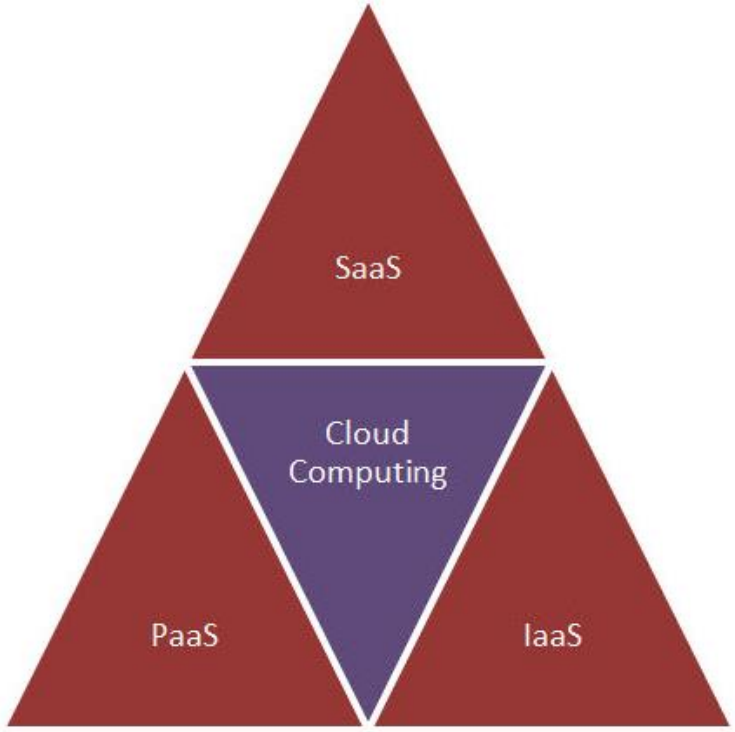
Background Information



Overview of Cloud Architecture



Spectrum of Clouds and Cloud Services



Private

Hybrid

Public

Software as a Service (SaaS)

User’s only have access to applications

Platform as a Service (PaaS)

Developers have access to a preset platform where they can update code but have restricted access

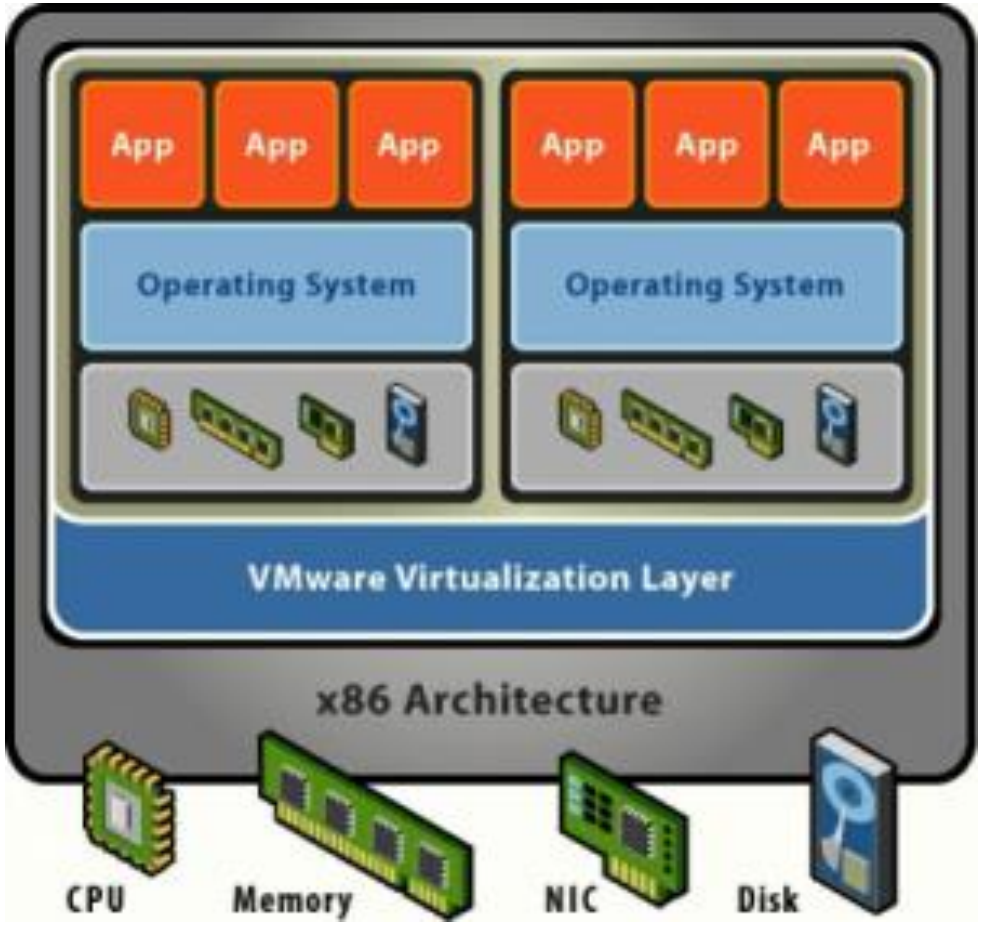
Infrastructure as a Service (IaaS)

Administrators have full control over their system and can customize operating system as well as have as many different platforms as they choose

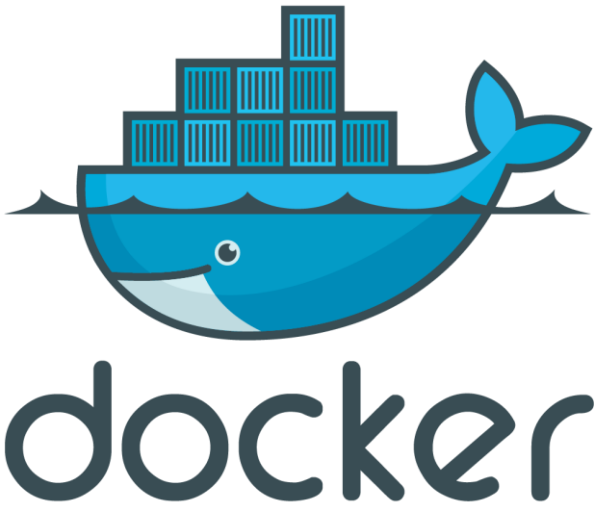
Virtualization

Process Virtual Machines – Java's JVM

System Virtual Machines – VMware



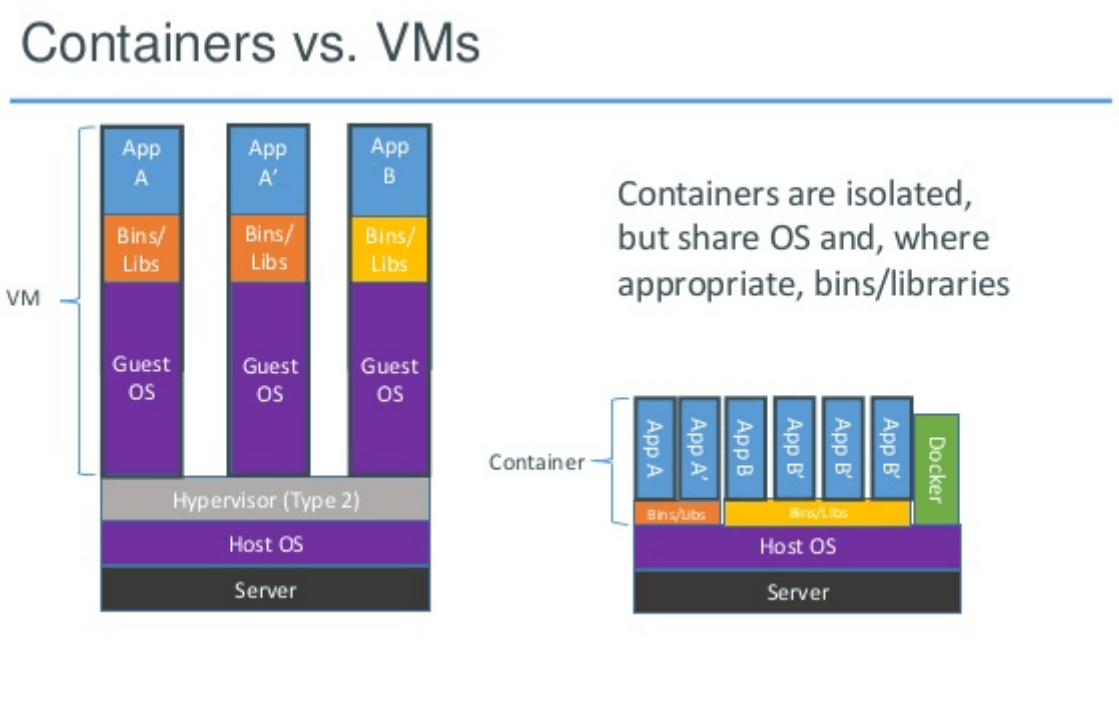
Overview of Technology



Overview

Containers use shared operating systems and rest on top of a single Linux instance meaning they are much more efficient than hypervisors in system resource terms

Extremely more efficient compared to technologies like Hyper-V, KVM, Xen



Docker and Cloud Security

Confidentiality

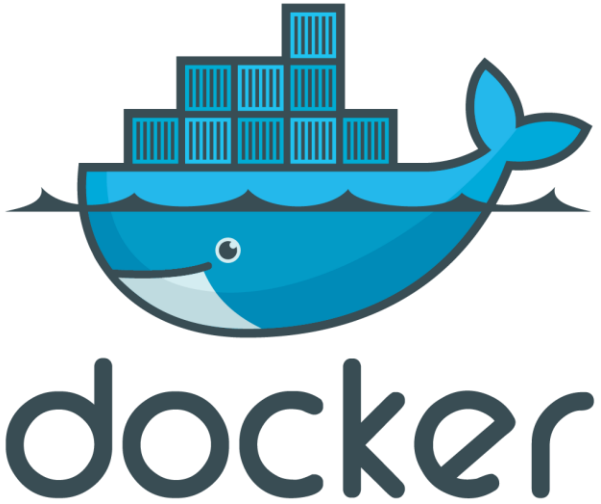
Able to segregate containers in multitenant environments protecting information from unauthorized individuals

Integrity

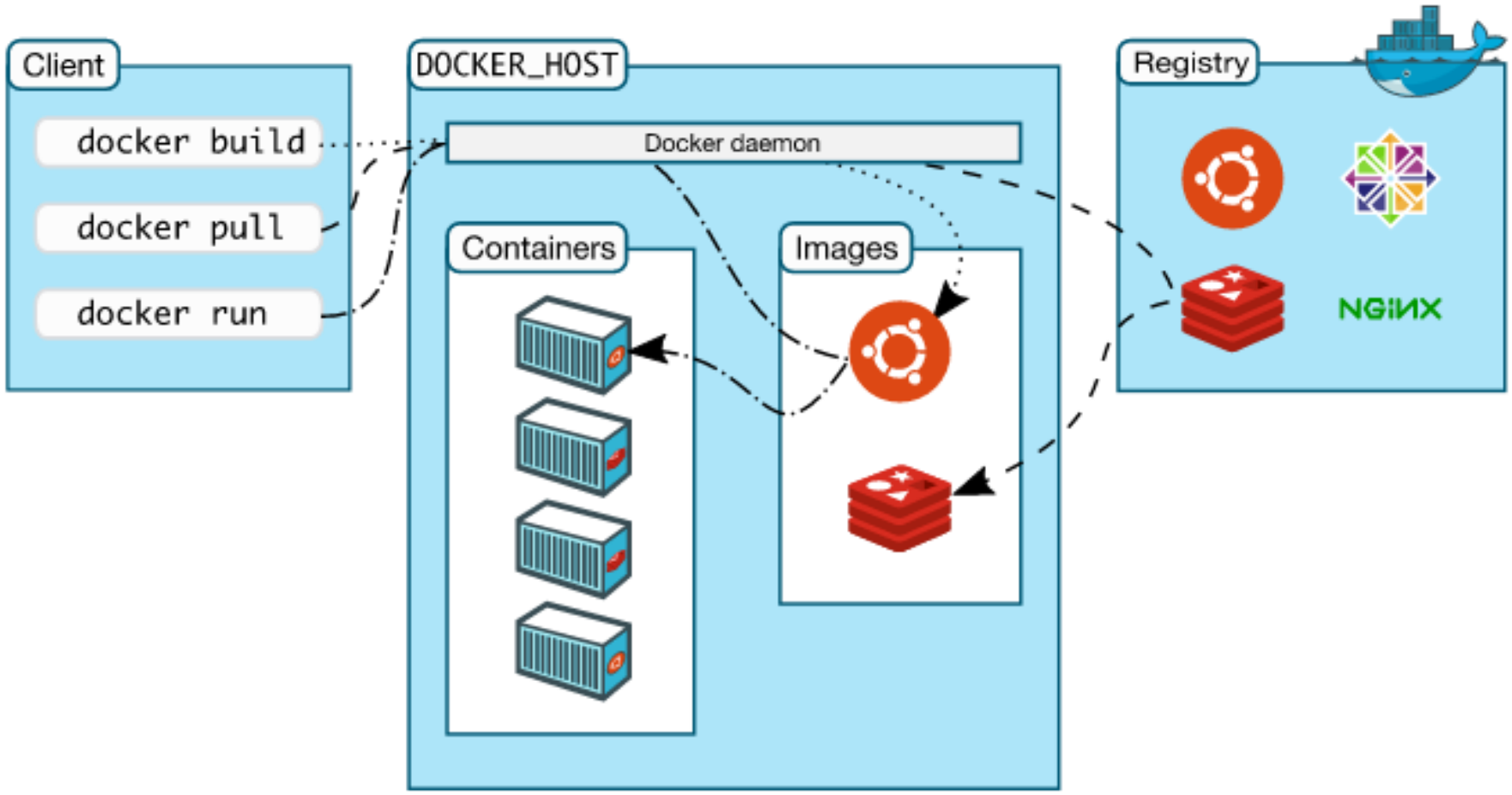
Able to isolate software with containers so that in a worst case scenario, if software crashes, other software is unaffected in other containers and also the container can be easily restarted

Availability

The efficiency of system resources allows for increased scalability as well as quick start up of containers



Docker Architecture

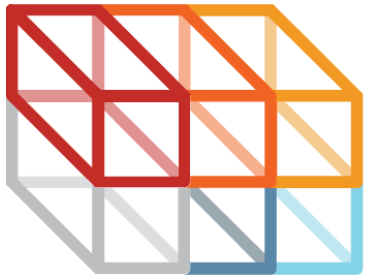


Hardening Docker Security with Triton

Triton implements Docker on SmartOS

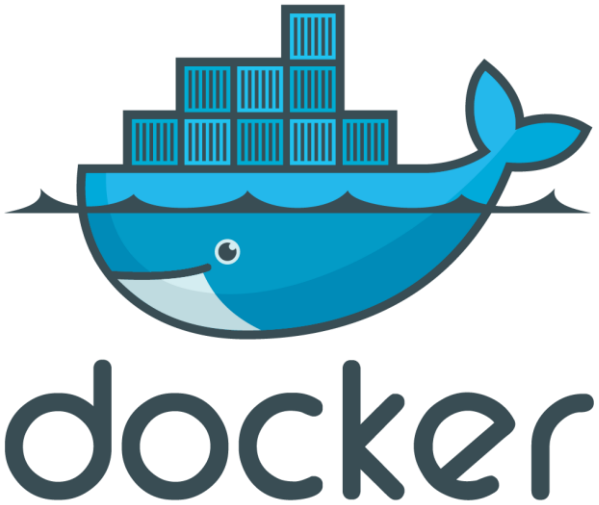
SmartOS is Unix based operating system whereas Docker is a Linux based technology

Increases performance and security between containers which is important with clouds and multitenant situations



Joyent
TRITON™

Interactive Walkthrough



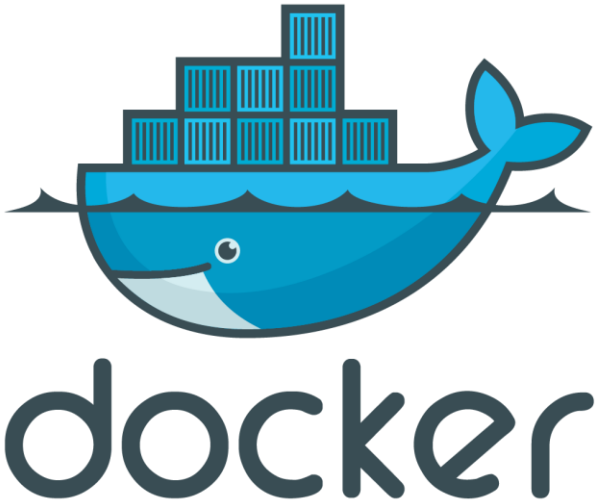
Installing Docker on Kali Linux

Very straightforward tutorial available: (Kali 2.0 is compatible with Debian Jessie)
https://docs.docker.com/engine/installation/linux/debian/

docker

In a terminal, try "docker"

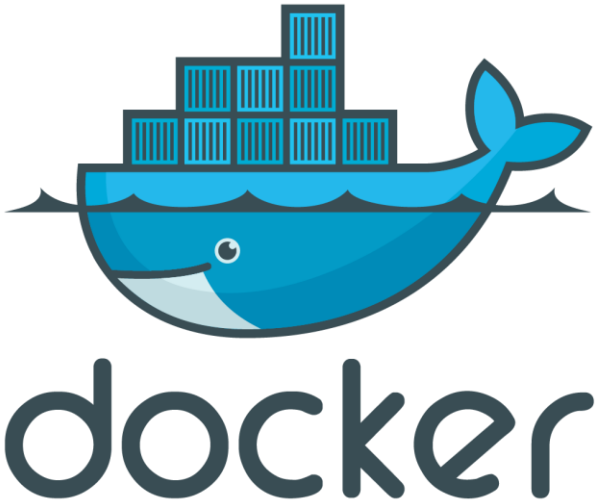
A listing of docker commands should appear



docker images

In a terminal, try "docker images"

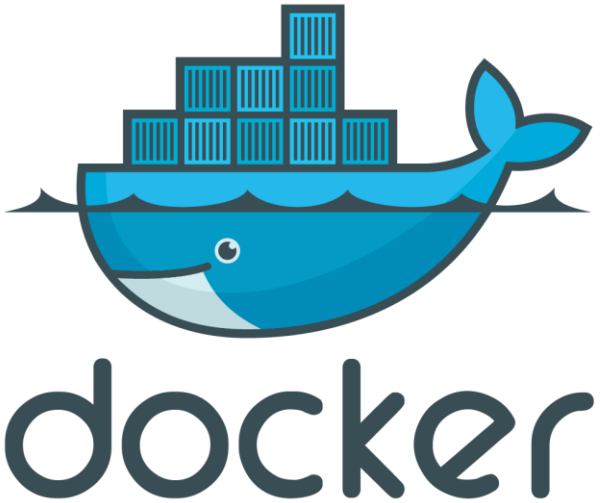
A listing should appear of docker images available to launch as containers



docker run

Try "docker run hello-world"

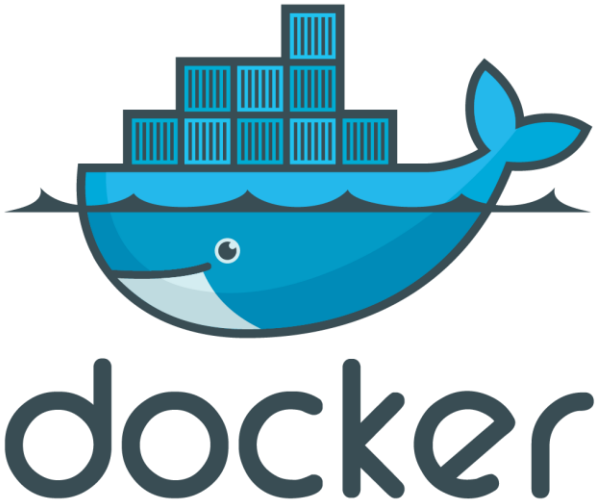
This will show a welcome message in the terminal



docker search

Try "docker search apache-php"

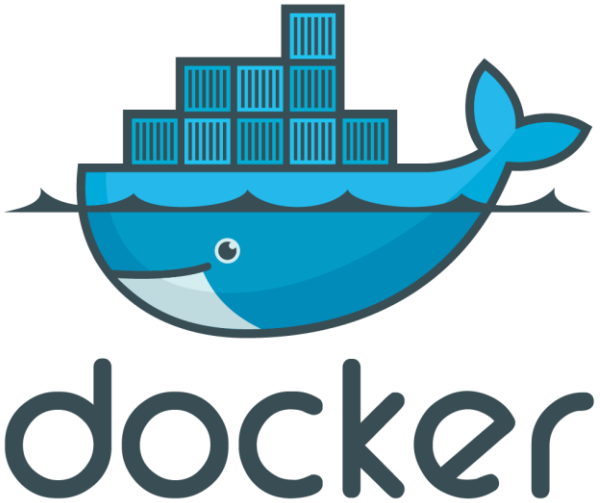
A listing of available images that match the query appear



docker pull

Try "docker pull tutum/apache-php"

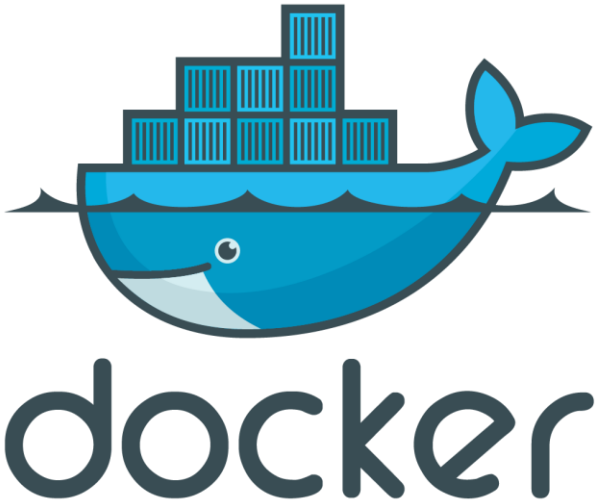
The Ubuntu image is downloaded



docker run

Try "docker run –d --name app –p 80:80 -e MY_VAR=value tutum/apache-php"

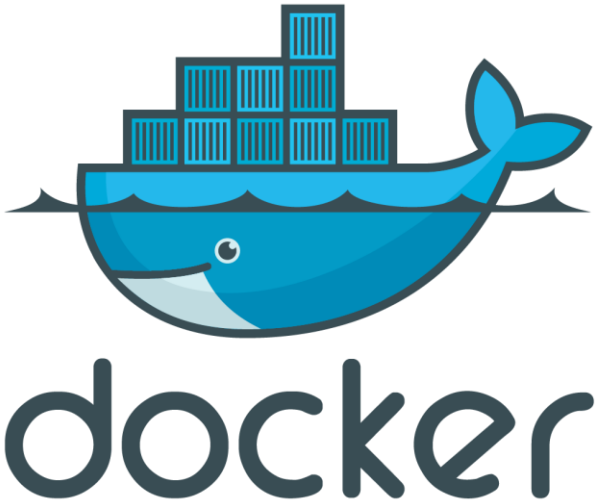
The container is started from the image in the background



docker exec

Try "docker exec -it app bash"

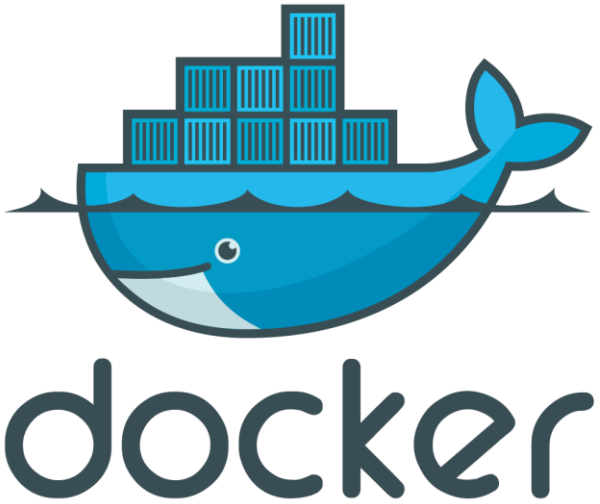
A shell is spawned into the container!



docker ps

Try "docker ps"

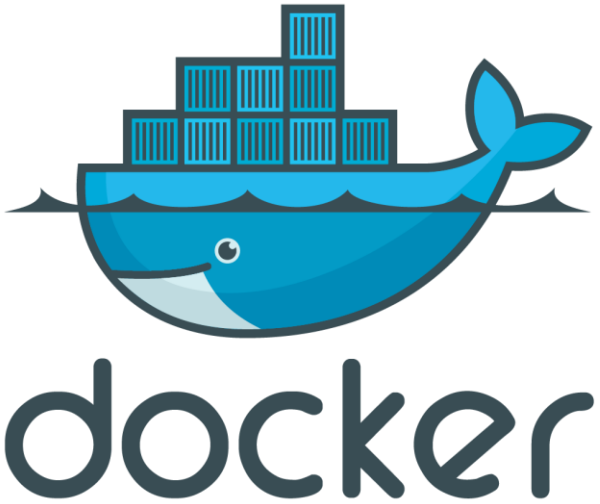
A listing of running containers should appear



docker stop

Try "docker stop app"

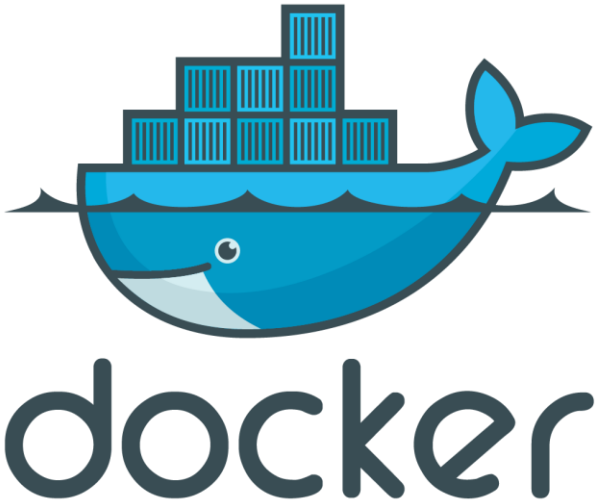
A listing of running containers should appear



docker ps

Try "docker ps"

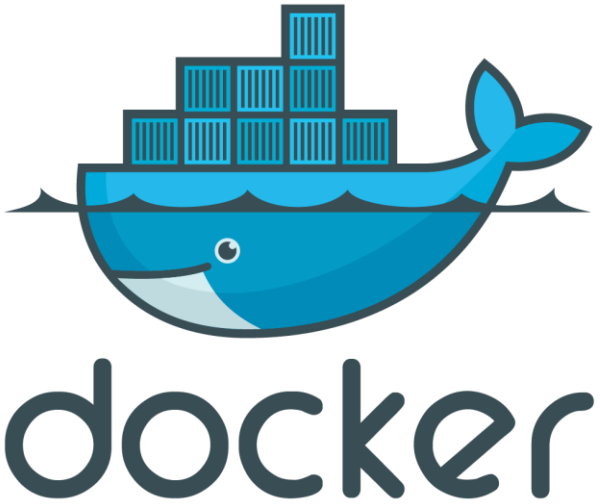
No containers... really?



docker ps -a

Try "docker ps -a"

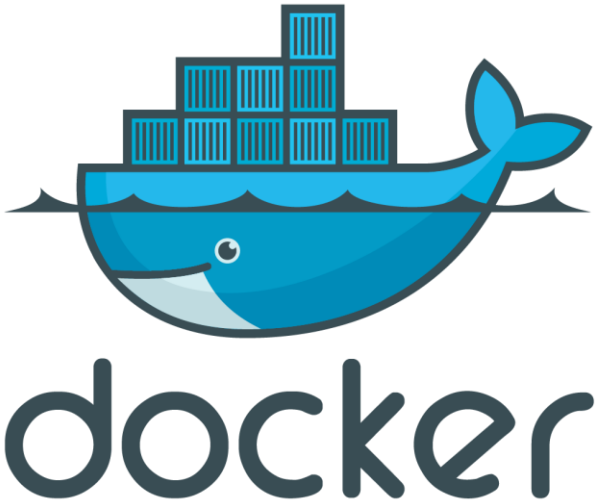
Ahah! There you are!!!



docker start

Try "docker start app"

Ahah! There you are!!!



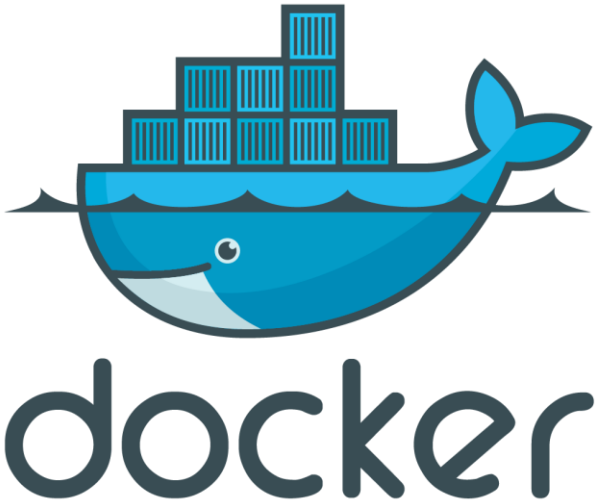
docker build

In the provided VM, navigate to "~/apache-php"

Open Dockerfile in your favourite text editor

Try "docker build UTM/student-apache-php ."

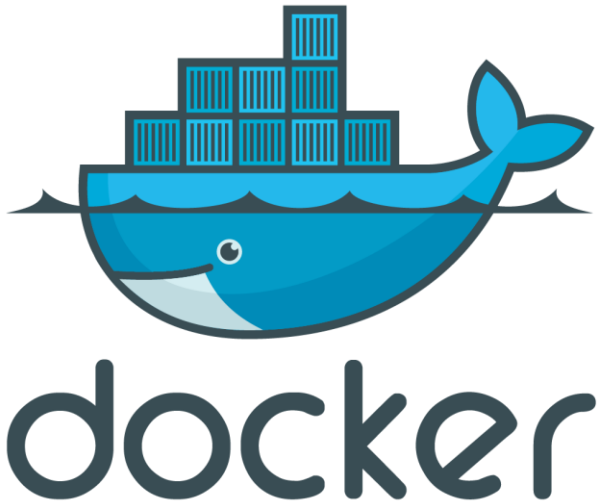
Reference: https://github.com/tutumcloud/apache-php



Next Steps

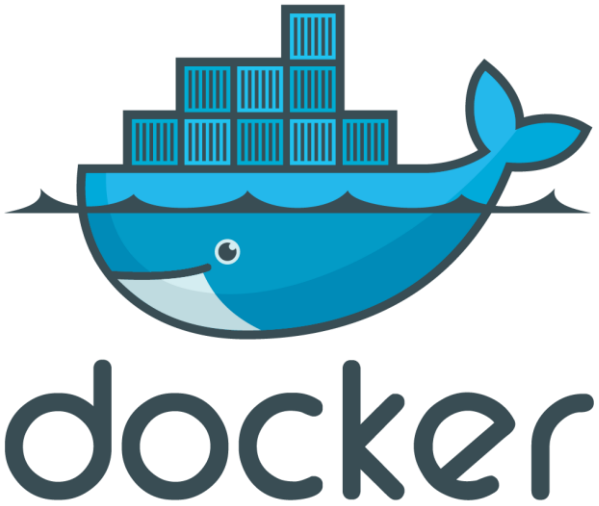
Now that you have the basics of docker, see if you can research how to :

- 1) Use a volume to connect the local filesystem to a container's filesystem
- 2) Use docker "links" to get two containers to talk to each other
- 3) Check out available images on DockerHub
- 4) Use docker-compose to define multicontainer configuration



The End!

Thanks for your time!



References

http://www.infoworld.com/article/2936623/

http://www.guru99.com/cloud-computing-for-beginners.html

http://stackoverflow.com/questions/16820336/what-is-saas-paas-and-iaas-with-examples

https://www.heroku.com/platform#platform-diagram-detail

http://stackoverflow.com/questions/16047306/how-is-docker-different-from-a-normal-virtual-machine

http://www.zdnet.com/article/what-is-docker-and-why-is-it-so-darn-popular/

https://docs.docker.com/v1.8/introduction/understanding-docker/

https://www.sans.org/reading-room/whitepapers/linux/securing-linux-containers-36142

https://www.joyent.com/blog/triton-docker-and-the-best-of-all-worlds