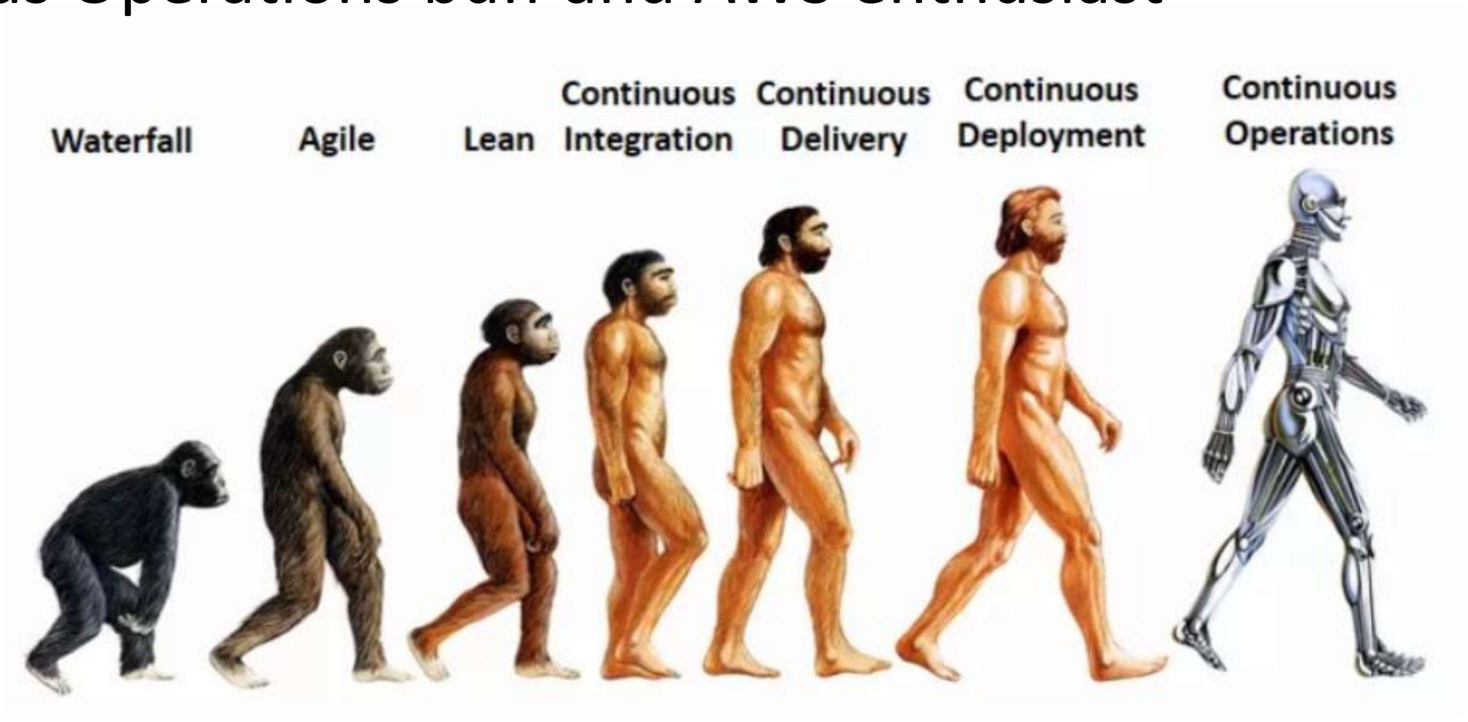
Introduction to Docker

The open platform to build, ship and run any application anywhere

About me

- 14+ Engineering experience (Cisco Systems, VMware)
- Continuous Operations buff and AWS enthusiast



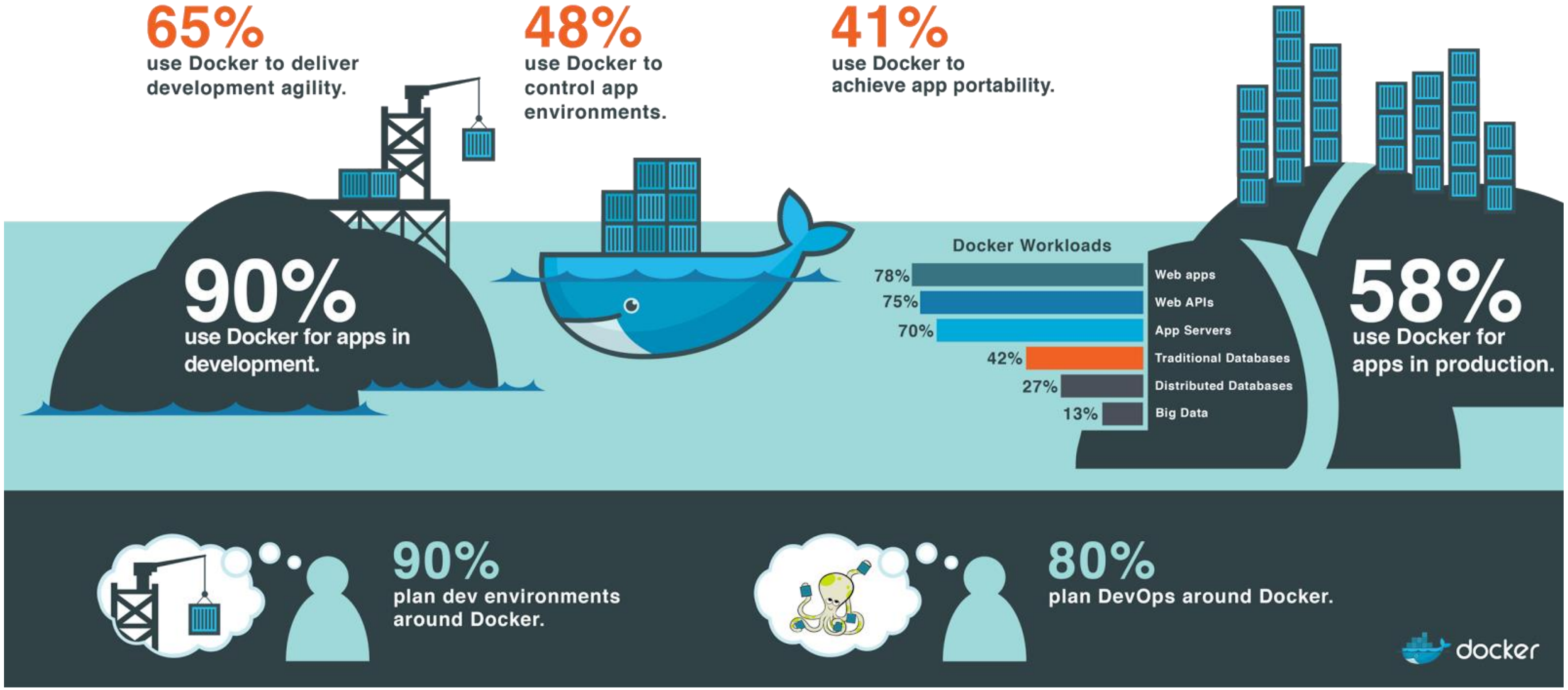
Session Logistics

- 3 hours (exercising time included)
- No docker experience required
- You will need:
  - Linux machine
    - Dedicated
    - VM
    - Cloud provided
  - Enthusiasm
- Familiar with basic Linux command line

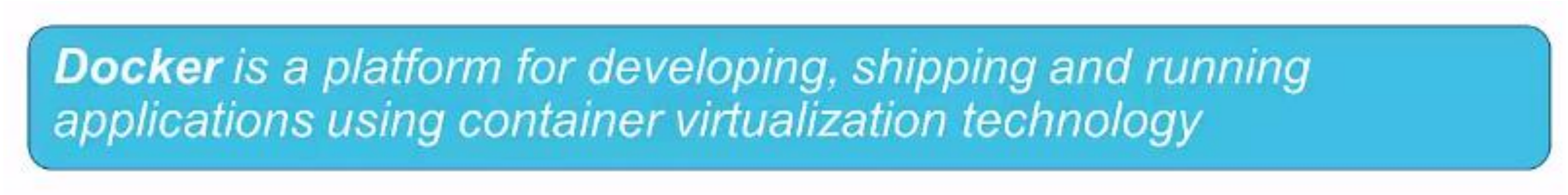
Agenda

- What is Docker
- Containers vs Virtual Machines
- Docker Platform Overview and Terminology
    - Docker Engine
    - Images
    - Containers
    - Registry
    - Repository
    - Docker Hub
    - Docker orchestration tools
- Intro to Images
- Getting Started with Containers

Docker community



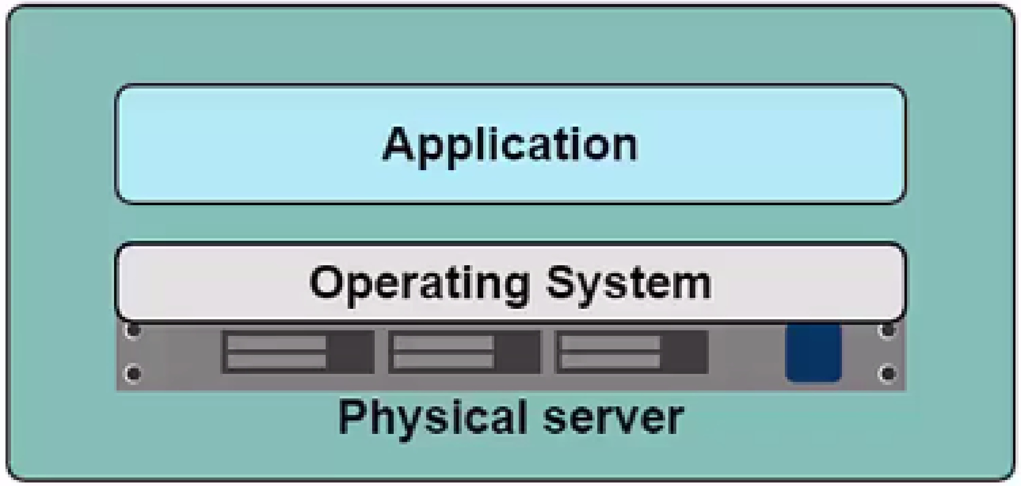
What is Docker


***Docker** is a platform for developing, shipping and running applications using container virtualization technology*

- The Docker Platform consist of multiple products/tools
  - Docker Engine
  - Docker Hub
  - Docker Machine
  - Docker Swarm
  - Docker Compose
  - Kitematic (docker ui)

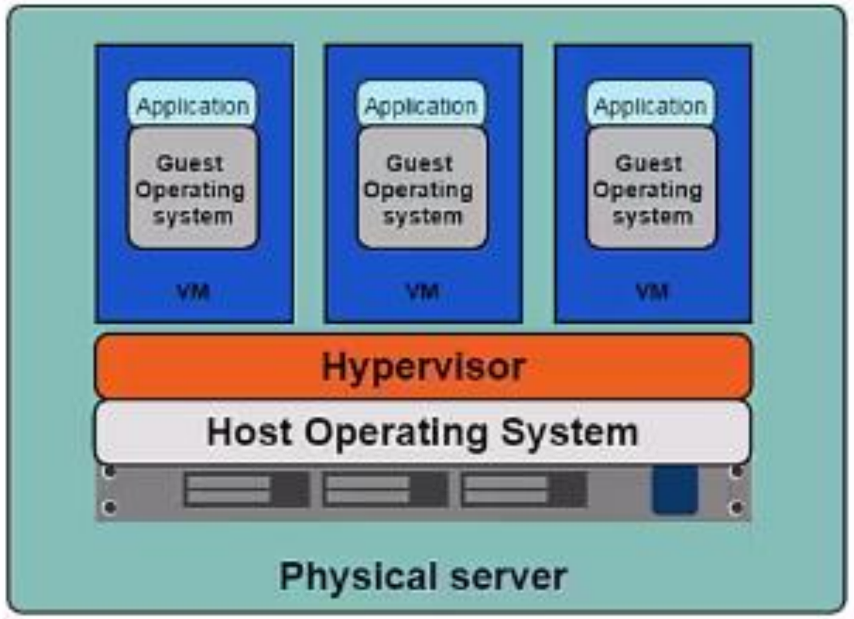
In the Dark Ages

- Slow deployment times
- Huge costs
- Wasted resources
- Difficult to scale
- Difficult to migrate
- Vendor lock in

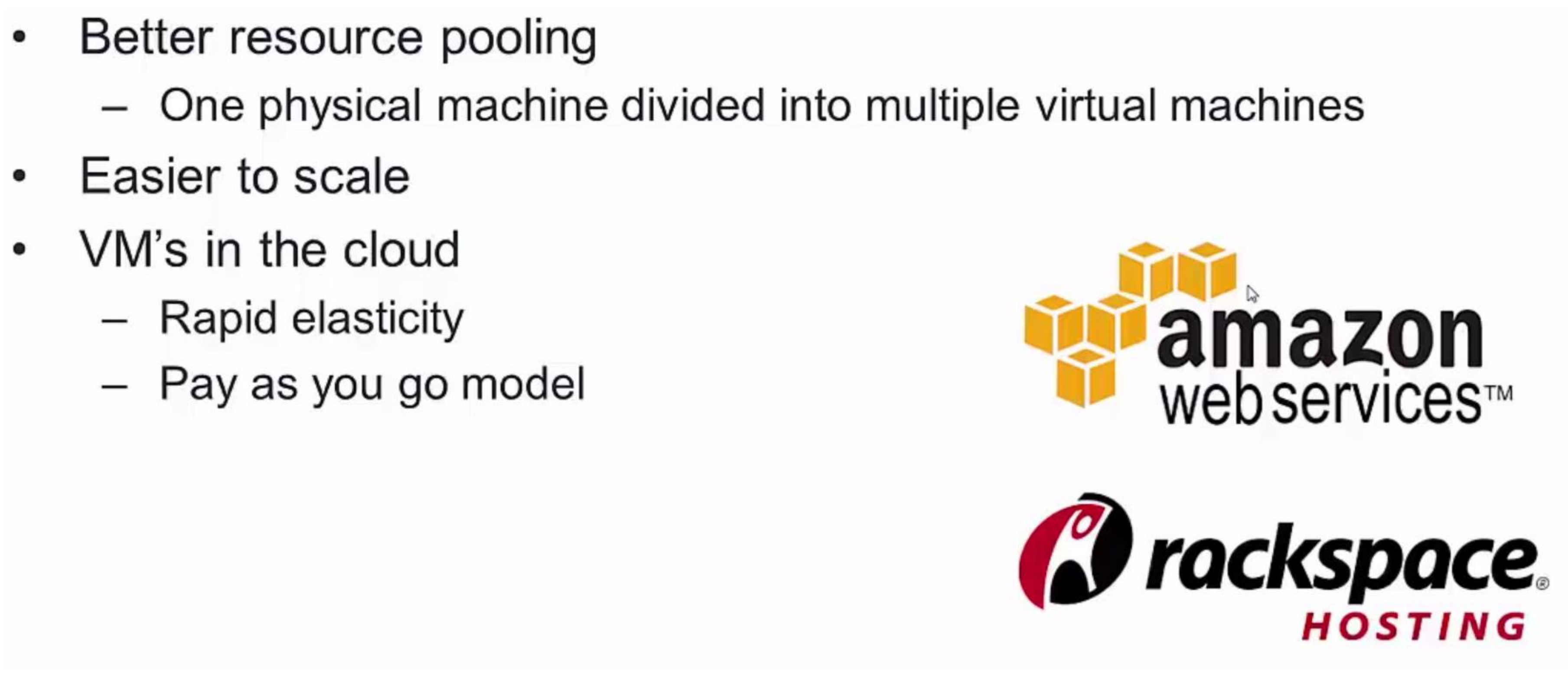


Hypervisor-based Virtualization

- One physical server can contain multiple applications
- Each application runs in a virtual machine (VM)



Benefits of VMs



- Better resource pooling
  - One physical machine divided into multiple virtual machines
- Easier to scale
- VM's in the cloud
  - Rapid elasticity
  - Pay as you go model

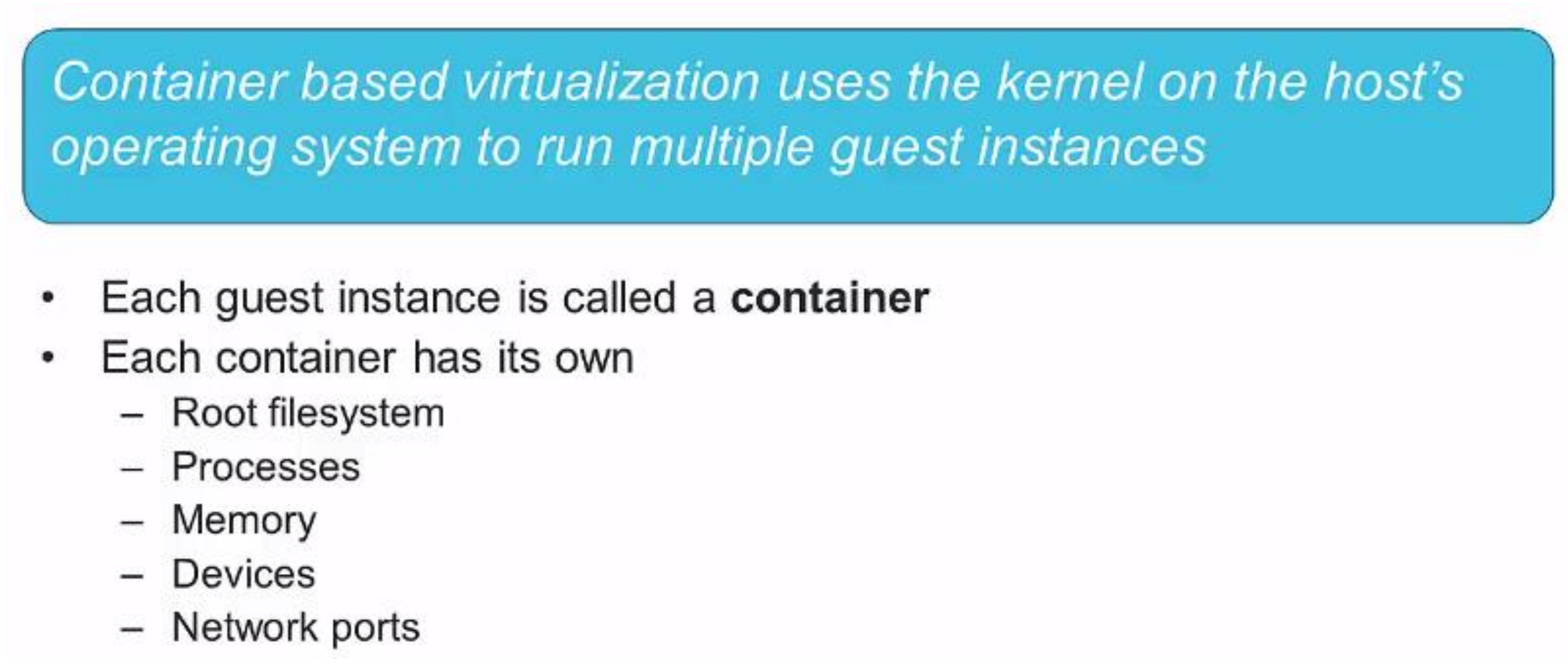
amazon web services™

rackspace® HOSTING

Limitations of VMs

- Each VM stills requires
  - CPU allocation
  - Storage
  - RAM
  - An entire guest operating system
- The more VM's you run, the more resources you need
- Guest OS means wasted resources
- Application portability not guaranteed

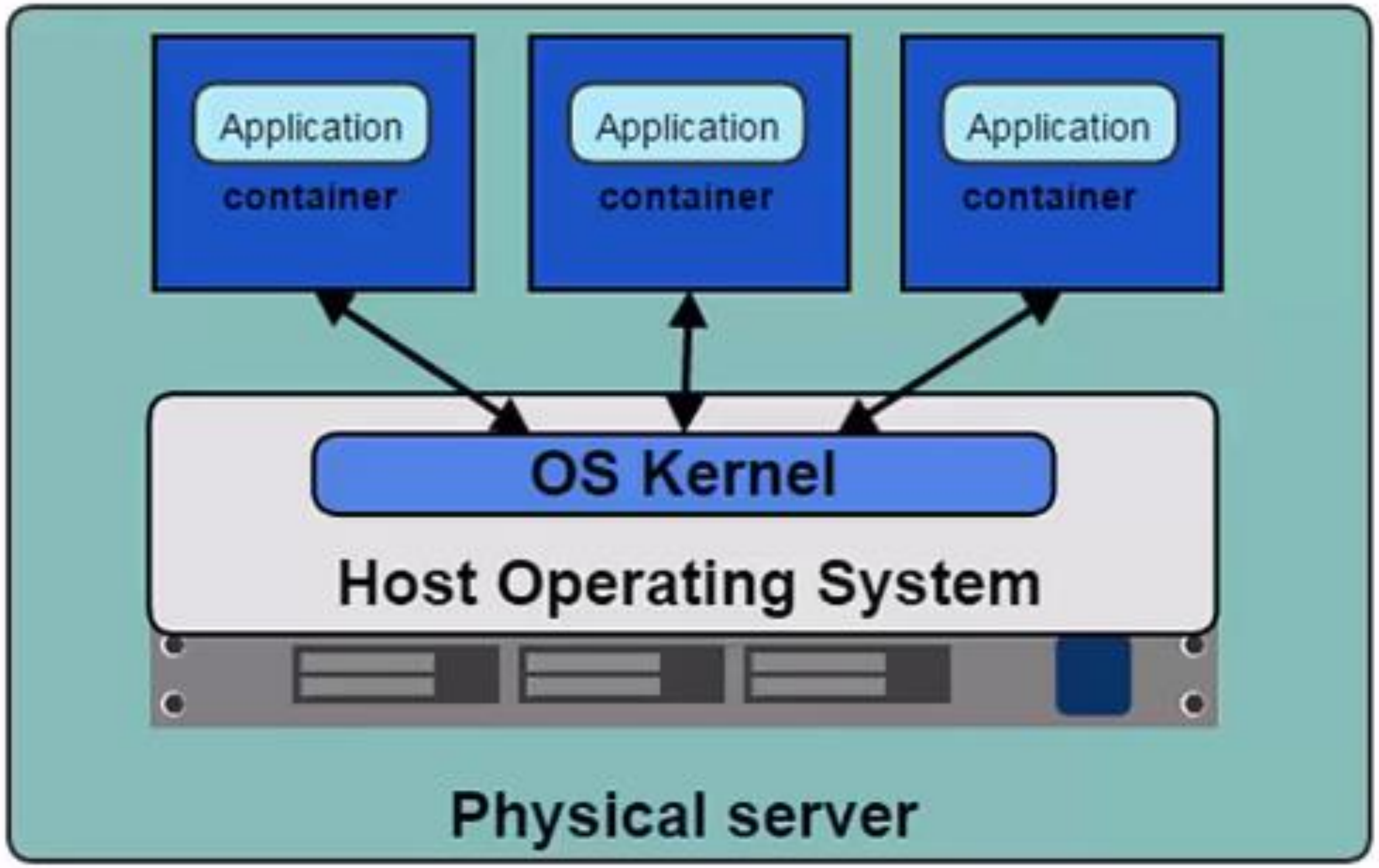
Introducing Containers



Container based virtualization uses the kernel on the host's operating system to run multiple guest instances

- Each guest instance is called a **container**
- Each container has its own
  - Root filesystem
  - Processes
  - Memory
  - Devices
  - Network ports

Containers

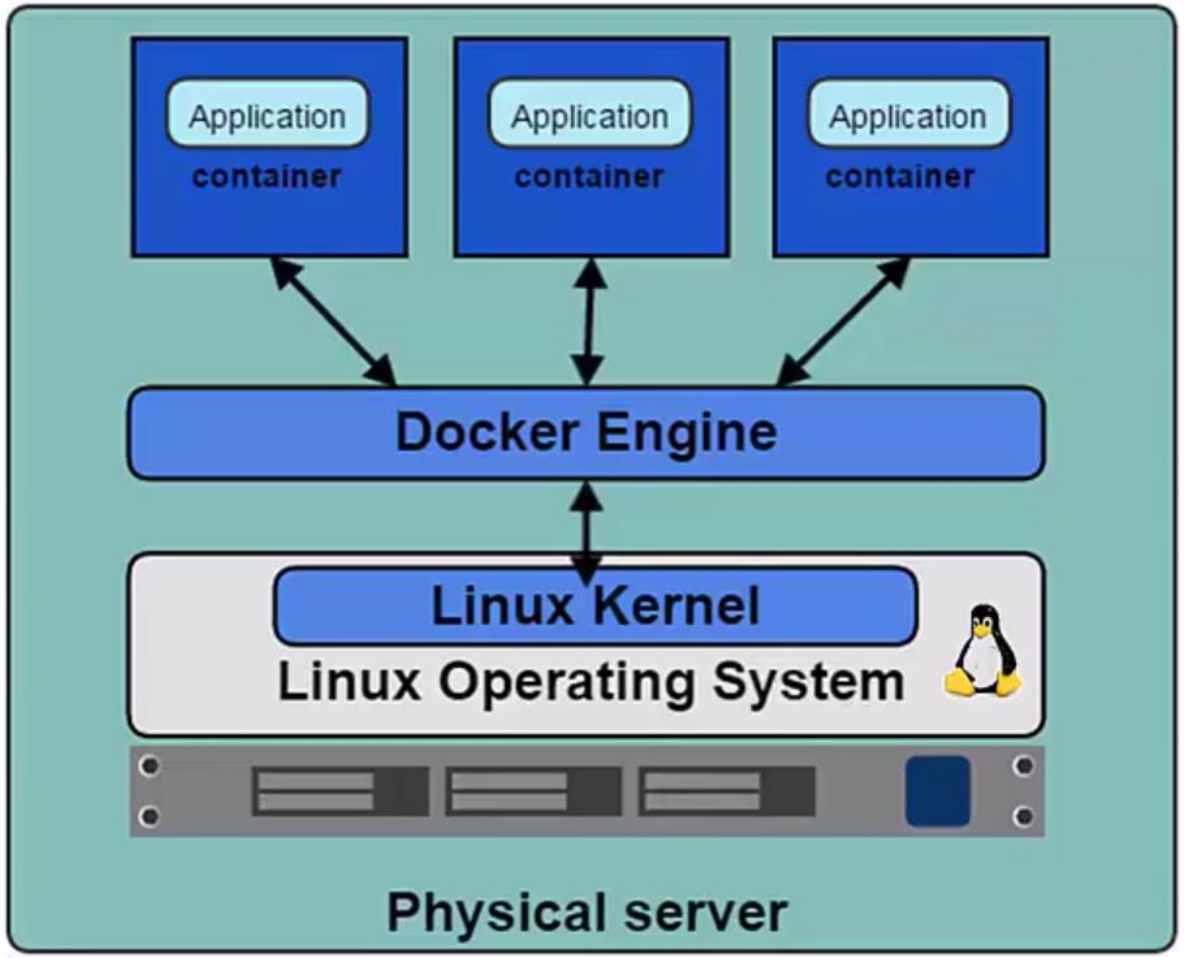

Application
container
Application
container
Application
container
OS Kernel
Host Operating System
Physical server

Containers vs VMs

- Containers are more lightweight
- No need to install guest OS
- Less CPU, RAM, storage space required
- More containers per machine than VMs
- Greater portability

Docker Engine

- **Docker Engine** is the program that enables containers to be built, shipped and run.
- Docker Engine uses Linux Kernel namespaces and control groups
- Namespaces give us the isolated workspace



EXERCISE

Install Docker Engine

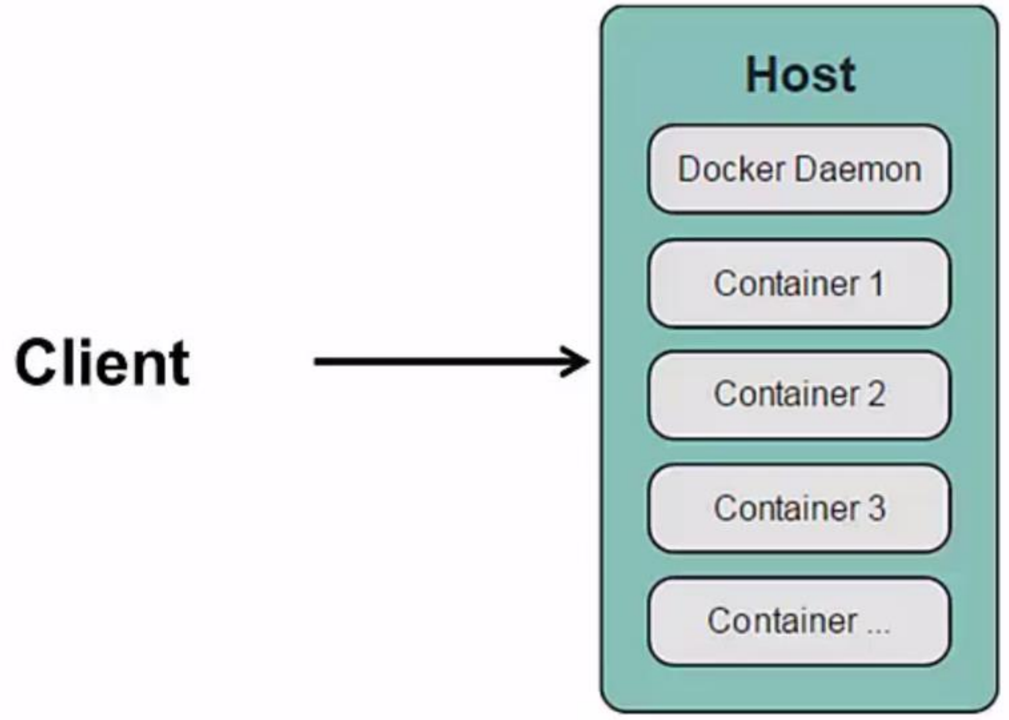
- Install Docker Engine
  - Ubuntu
  - CentOS
  - Windows
- Post install steps
- Docker run hello-world

Hint:

- Download putty
- Enable sshd
  - sudo apt-get install ssh
  - sudo ufw allow 22
- Enable NAT path to the VM
  - https://www.youtube.com/watch?v=oNK1fXhxQHs

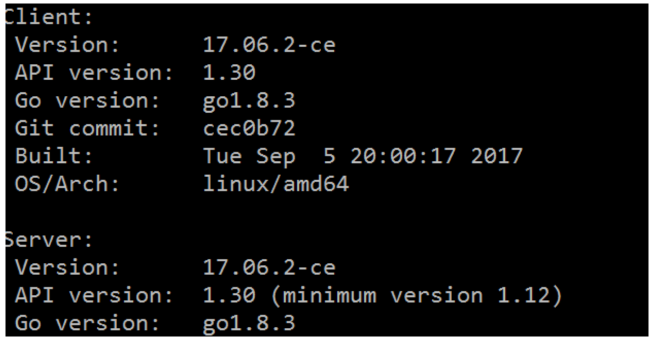
Docker Client and Daemon

- Client / Server architechture
- Client takes user inputs and send them to the daemon
- Daemon builds, runs and distributes containers
- Client and daemon can run on the same host or on different hosts
- CLI client and GUI (Kitematic)



Checking Docker Client and Engine Version

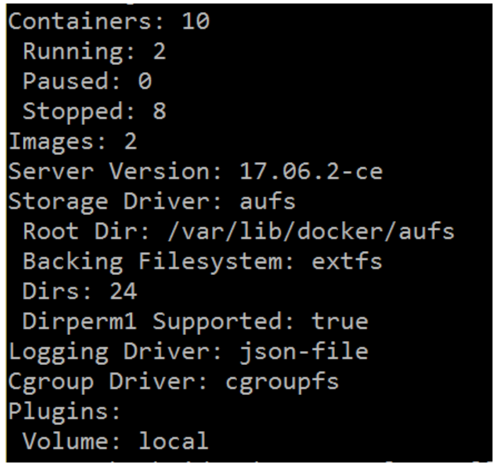
- Docker version



```
Client:
 Version:      17.06.2-ce
 API version:  1.30
 Go version:   go1.8.3
 Git commit:   cec0b72
 Built:        Tue Sep  5 20:00:17 2017
 OS/Arch:      linux/amd64

Server:
 Version:      17.06.2-ce
 API version:  1.30 (minimum version 1.12)
 Go version:   go1.8.3
```

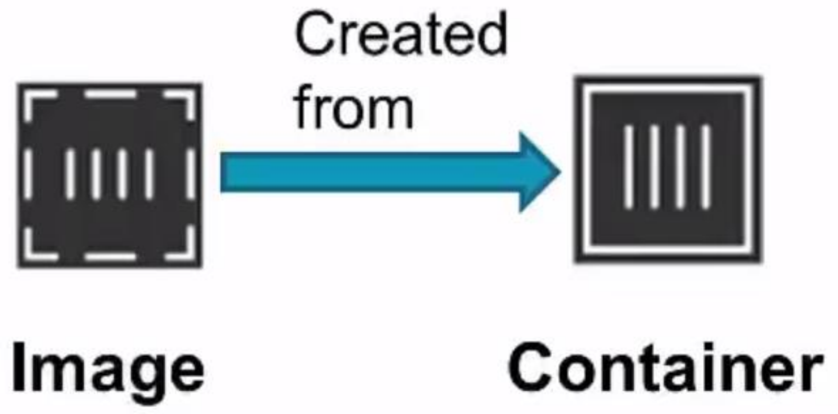
- Docker info



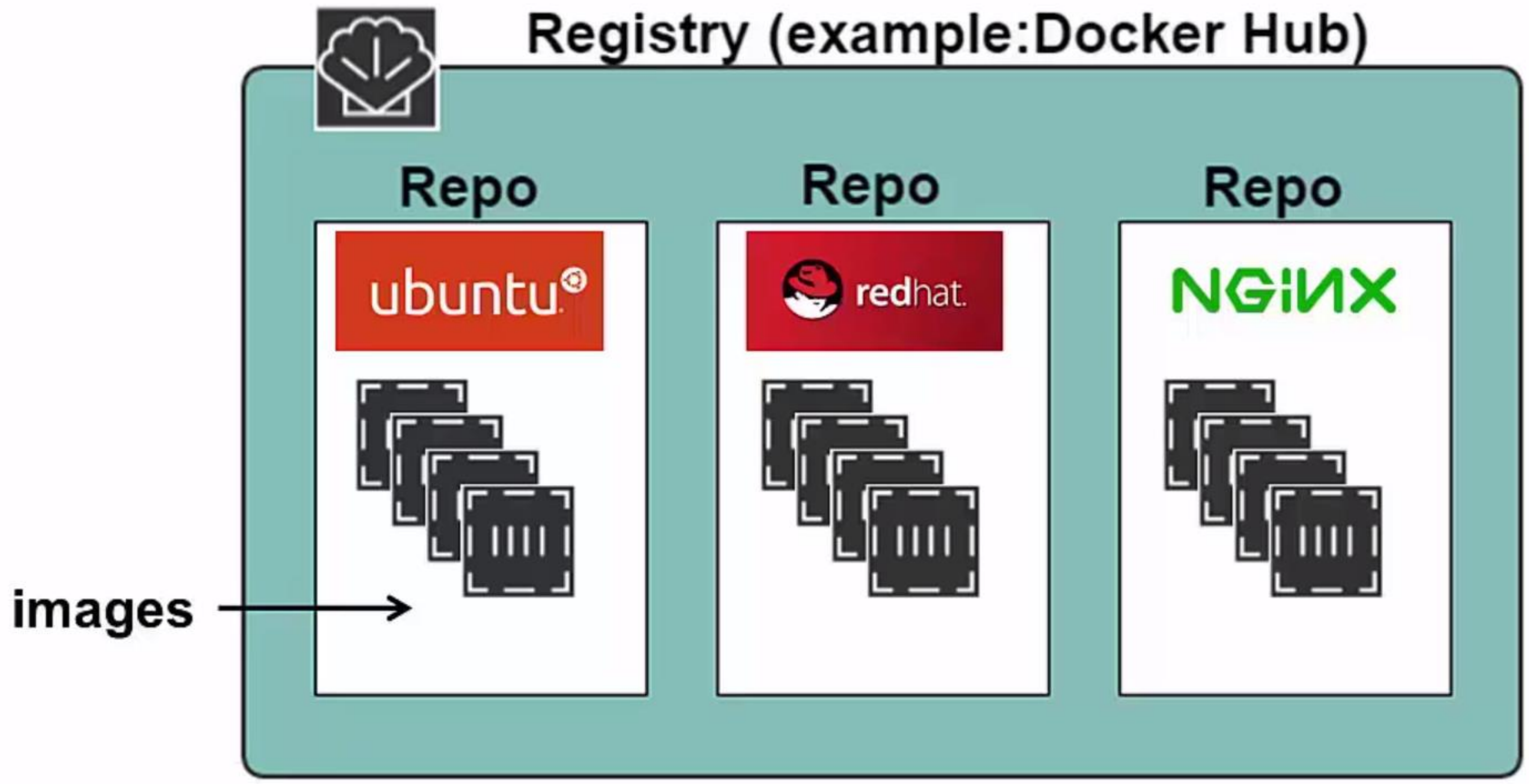
```
Containers: 10
 Running: 2
 Paused: 0
 Stopped: 8
Images: 2
Server Version: 17.06.2-ce
Storage Driver: aufs
 Root Dir: /var/lib/docker/aufs
 Backing Filesystem: extfs
 Dirs: 24
 Dirperm1 Supported: true
Logging Driver: json-file
Cgroup Driver: cgroupfs
Plugins:
 Volume: local
```

Docker Images and Containers

- Images
  - Read only template used to create containers
  - Built by you or other Docker users
  - Stored in the Docker Hub or your local Registry
- Containers
  - Isolated application platform
  - Contains everything needed to run your application
  - Based on one or more images

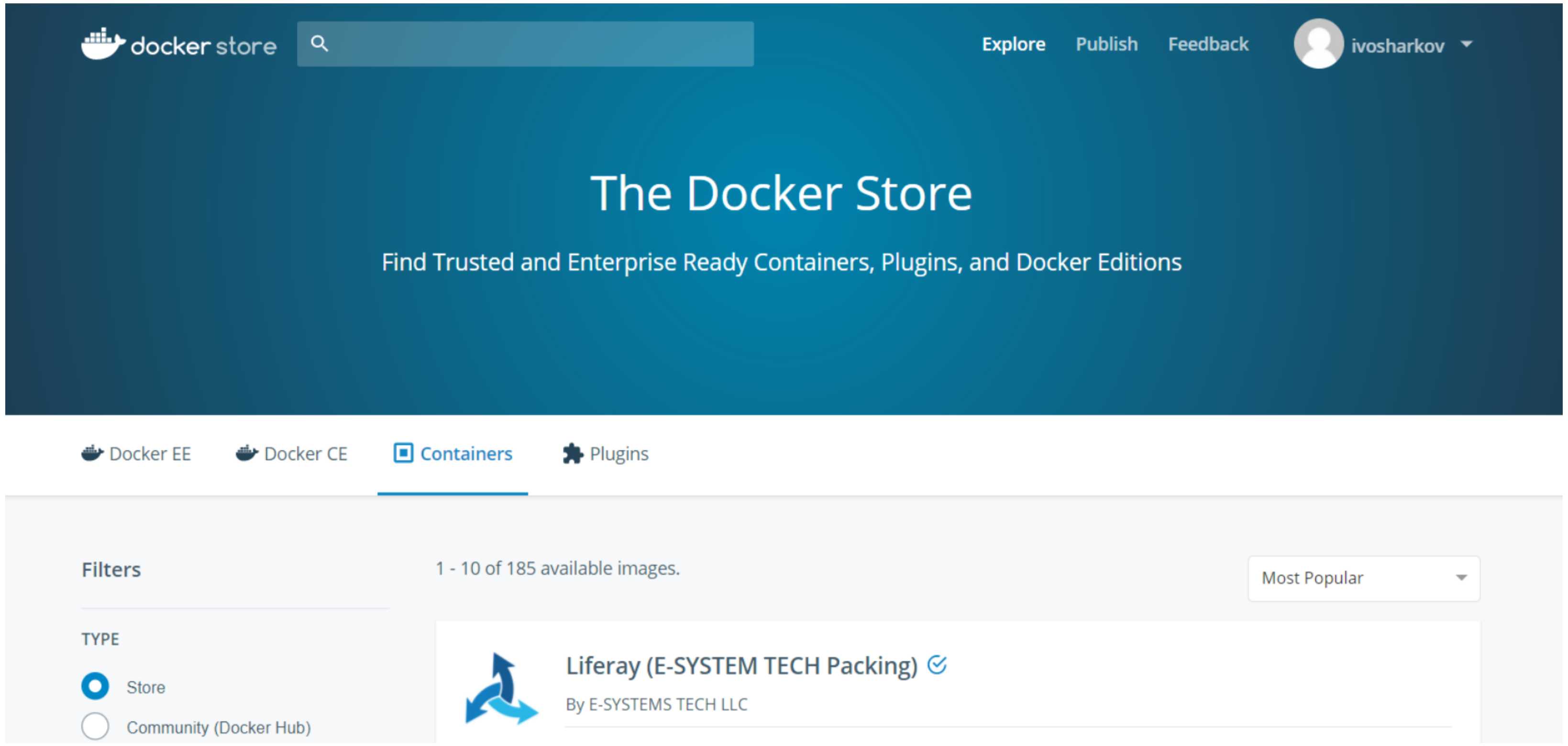


Registry and Repositories



Docker Hub & Docker Store

***Docker Hub** is the public registry that contains a large number of images available for your use*



The Docker Store

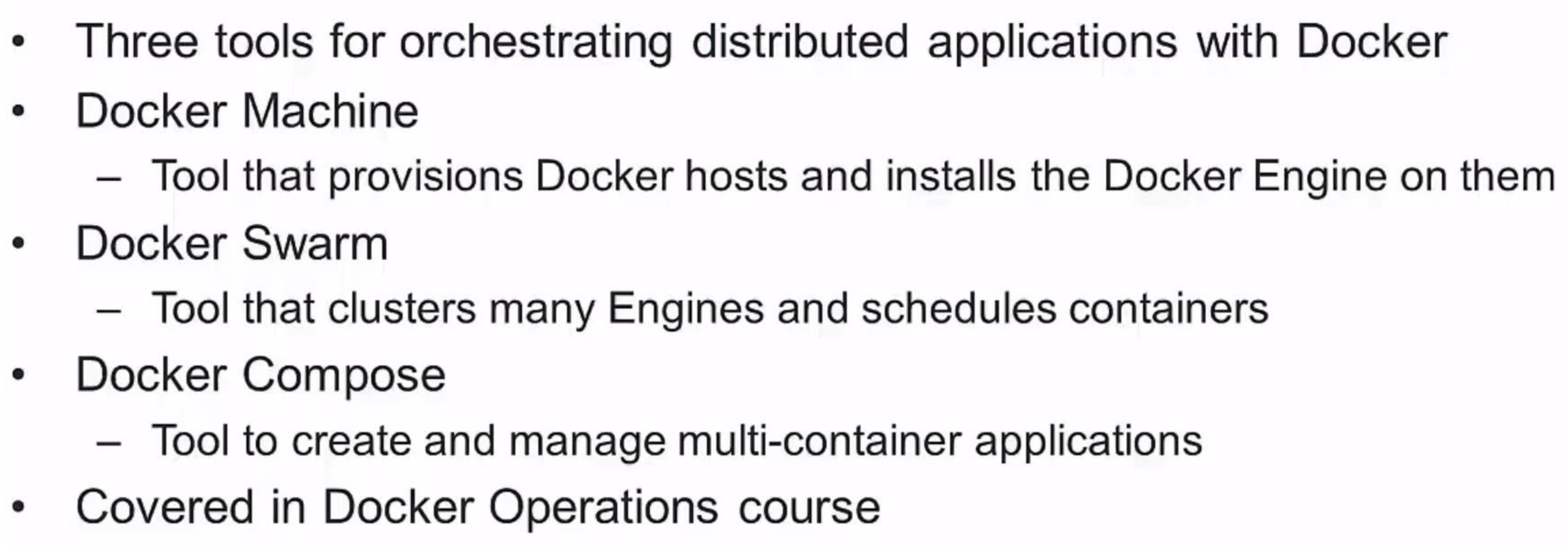
Find Trusted and Enterprise Ready Containers, Plugins, and Docker Editions

1 - 10 of 185 available images.

Liferay (E-SYSTEM TECH Packing)

By E-SYSTEMS TECH LLC

Docker Orchestration Tools



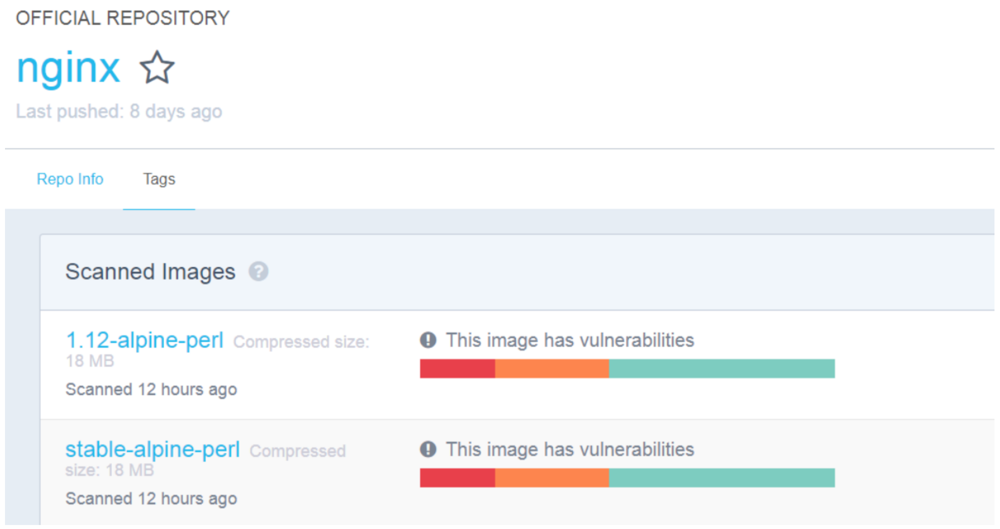
- Three tools for orchestrating distributed applications with Docker
- Docker Machine
  - Tool that provisions Docker hosts and installs the Docker Engine on them
- Docker Swarm
  - Tool that clusters many Engines and schedules containers
- Docker Compose
  - Tool to create and manage multi-container applications
- Covered in Docker Operations course

Docker Hub Walkthrough

- Demo
  - Repositories
  - Dockerfiles
  - Tags
  - The Social part

Image Tags

- Images are specified by repository: tag
- The same image may have multiple tags
- The default tag is latest
- Available tags could be looked up at the repository (alpine,java,nginx)



OFFICIAL REPOSITORY

nginx

Last pushed: 8 days ago

Repo Info | Tags

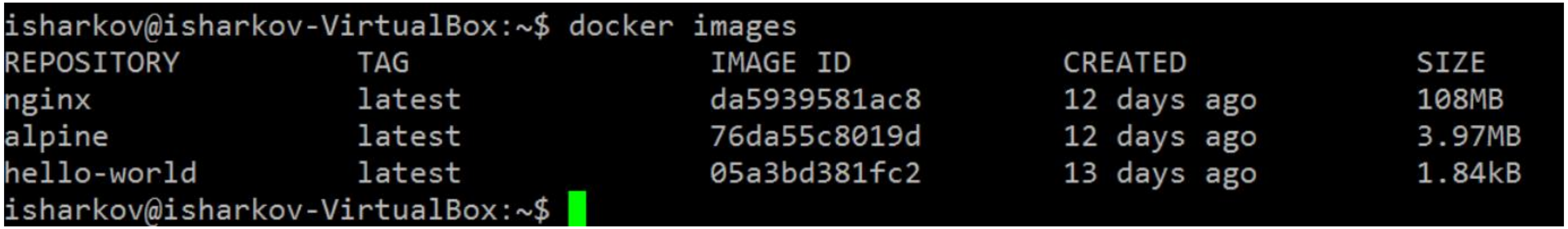
Scanned Images

1.12-alpine-perl Compressed size: 18 MB
Scanned 12 hours ago
This image has vulnerabilities

stable-alpine-perl Compressed size: 18 MB
Scanned 12 hours ago
This image has vulnerabilities

Display Local Images

- Run docker images
- Local and remote repos
  - docker run alpine
  - Do it again



```
isharkov@isharkov-VirtualBox:~$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
nginx               latest              da5939581ac8        12 days ago         108MB
alpine              latest              76da55c8019d        12 days ago         3.97MB
hello-world         latest              05a3bd381fc2        13 days ago         1.84kB
isharkov@isharkov-VirtualBox:~$
```

EXERCISE

Create a Docker Hub Account

1. Go to https://hub.docker.com/account/signup/ and signup for an account if you do not already have one.
   No credit card details are needed
2. Find your confirmation email and activate your account
3. Browse some of the repositories
4. Search for some images of your favourite dev tools, languages, servers etc…
   a) (examples: Java, Perl, Maven, Tomcat, NGINX, Apache)

Creating a Container

- Use docker run command
- Syntax

```
docker run [options] [image_name] [commands] [args]
```

- Image is specified by repository:tag
- Type

```
docker run alpine echo “Hello World”
```

Run a Simple Container

EXERCISE

1. On your terminal type
   ```
   docker run ubuntu:14.04 echo "hello world"
   ```
2. Observe the output
3. Then type
   ```
   docker run ubuntu:14.04 ps ax
   ```
4. Observe the output
5. Notice the much faster execution time compared to the first container that was run. This is due to the fact that Docker now has the Ubuntu 14.04 image locally and thus does not need to download the image

Container with Terminal

- Container started with –i & -t
    - -I – interactive
    - -t – terminal

- Demo

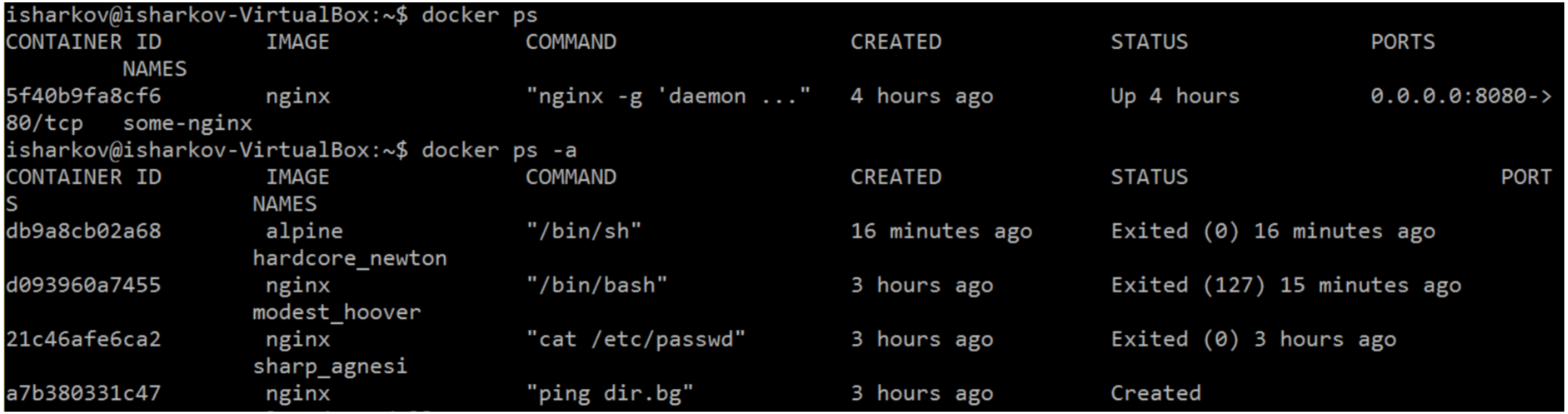
```
docker run –I –t nginx
docker run –d nginx
docker exec –I –t nginxId
```

Container ID

- Container can be specified using their ID or name
- Long and short ID
- Short ID and name can be obtained with : docker ps
- Long ID by: docker inspect [some_docker_container_name]

Find Your Containers

- Use docker ps to list running containers
- Use docker ps –a to list all containers (recently stopped are included)



```
isharkov@isharkov-VirtualBox:~$ docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS
         NAMES
5f40b9fa8cf6        nginx               "nginx -g 'daemon ..."   4 hours ago         Up 4 hours          0.0.0.0:8080->
80/tcp   some-nginx
isharkov@isharkov-VirtualBox:~$ docker ps -a
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS                       PORT
S                   NAMES
db9a8cb02a68        alpine              "/bin/sh"                16 minutes ago      Exited (0) 16 minutes ago
                   hardcore_newton
d093960a7455        nginx               "/bin/bash"              3 hours ago         Exited (127) 15 minutes ago
                   modest_hoover
21c46afe6ca2        nginx               "cat /etc/passwd"        3 hours ago         Exited (0) 3 hours ago
                   sharp_agnesi
a7b380331c47        nginx               "ping dir.bg"            3 hours ago         Created
```

Running in Detached Mode

- Known as running in the background or as a daemon
- Use –d flag
- Where is the output ? (hint use docker logs (-f) container_id)

- Example

```
docker run –d alpine ping dir.bg
docker logs –f alpine_container_id
```

Port Mapping

- How to interact with myApp
  - -P for port mapping
  - docker run -d -p 8080:80 nginx

Docker most used commands

- Docker images shows all images
- docker ps shows running containers.
- docker logs gets logs from
- docker inspect looks at all the info on a container (including IP address).
- docker events gets events from container.
- docker port shows public facing port of container.
- docker top shows running processes in container.
- docker stats shows containers' resource usage statistics.
- docker diff shows changed files in the container's FS.

Benefits of Docker

- Separation of concerns
  - Developers focus on building apps
  - QA focus on testing
  - System admins focus on deployment
- Fast development cycle
- Application portability and
  - Build in one environment, test and ship to another
- Scalability spin-up new container if needed
  - Easily spin-up new container on demand
- Cost efficiency

Questions

References

- Docker cheat sheet

  https://github.com/eon01/DockerCheatSheet
- Create a best Image

  https://docs.docker.com/engine/userguide/eng-image/baseimages/
- Docker FAQs

  https://docker-curriculum.com/
- Google ☺