Docker Getting Started

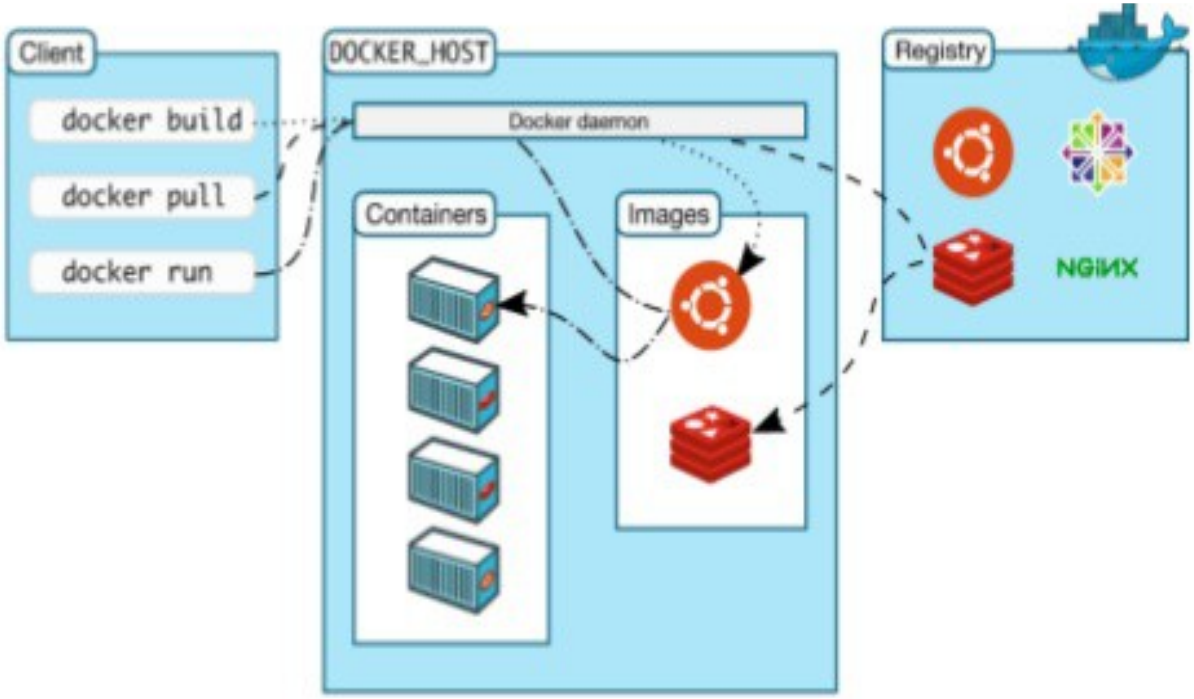
Sang Shin
JPassion.com
"Code with Passion!"

Topics

- Docker architecture
- Docker installation
- Docker machine
- Docker client
- Docker images pulled and installed
- Docker images in the Docker registry
- Types of Docker images
- Kitematic

Docker Architecture

Docker Architecture



What makes up Docker?

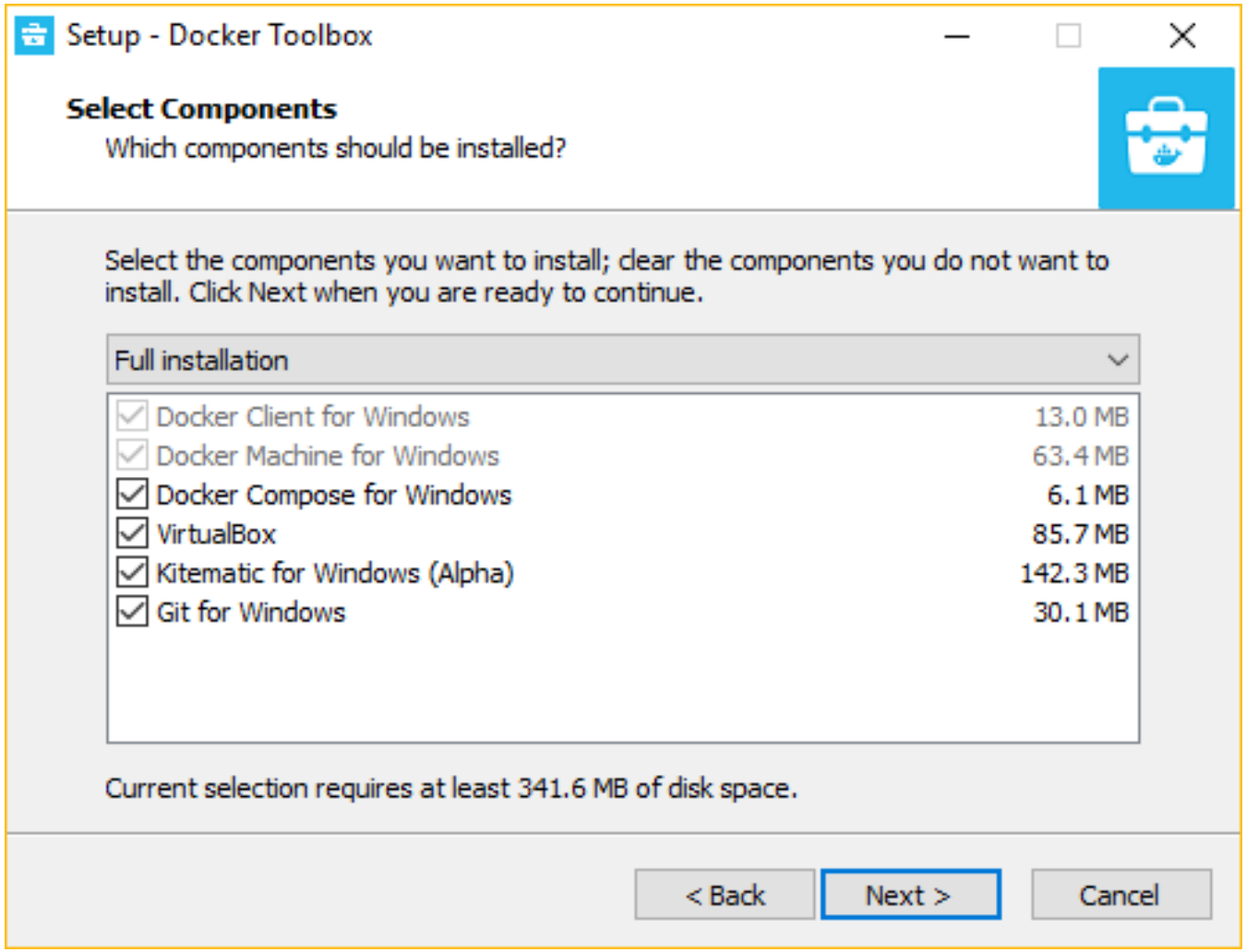
- Images
  - The file system and configuration of our application which are used to create containers
- Containers
  - Running instances of Docker images - containers run the actual applications
- Docker daemon
  - The background service running on the host that manages building, running and distributing Docker containers
- Docker client
  - Command line tool that allows the user to interact with the Docker daemon
- Docker Hub (Docker Registry)
  - A registry of Docker images

Docker Installation

Installation of Docker

- If your platform meets the hardware/OS requirement, you can install Docker natively on your platform
  - https://docs.docker.com/engine/getstarted/step_one/
  - Windows 10 Pro (Install “Docker for Windows”)
  - Mac OS 10.10.3 Yosemite or newer (Install “Docker for Mac”)
  - Linux (Install “Docker for Linux”)
- If you have old models of Windows or Mac, install Docker Toolbox
  - https://www.docker.com/products/docker-toolbox
  - Windows 7, 8, Windows 10 Home
  - Old version of Mac OS

Docker Toolbox Installation

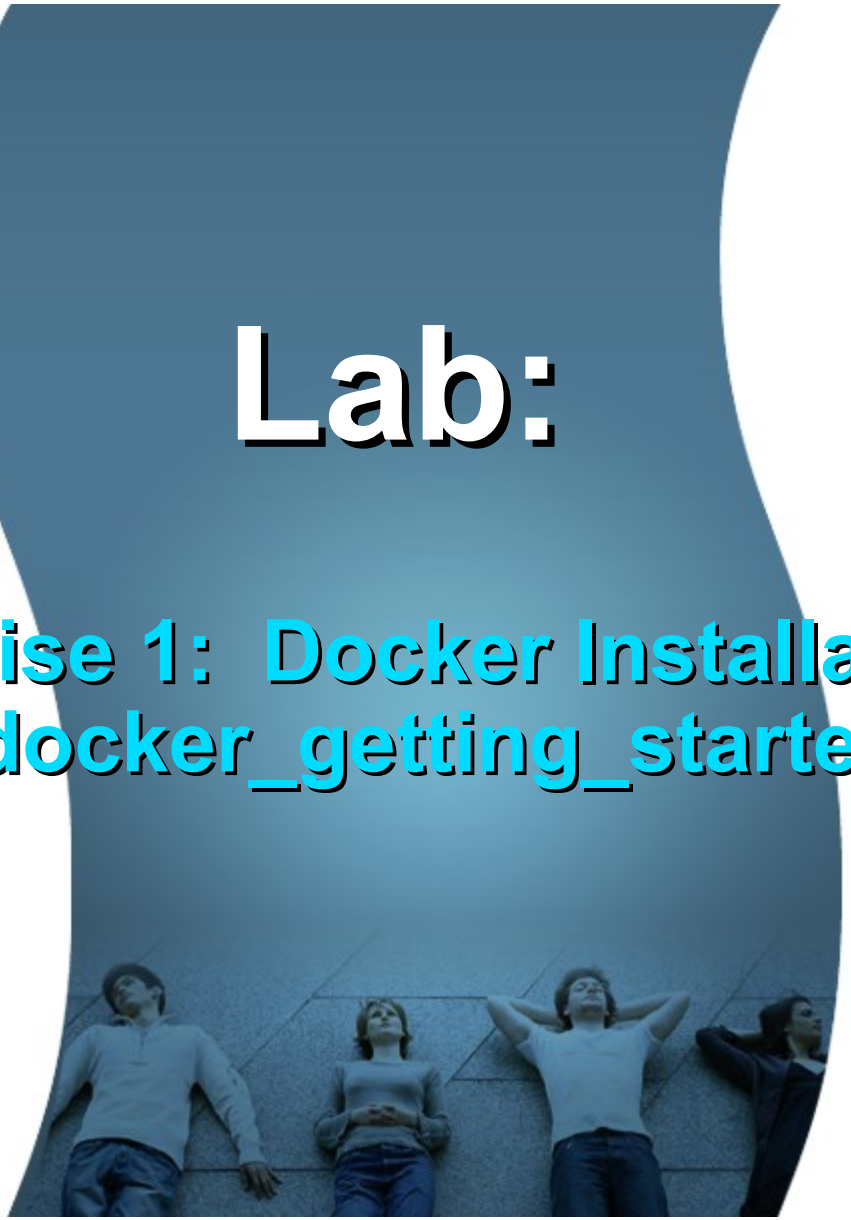


Lab: Check Docker/Docker-Machine versions - remove

```
C:\>docker-machine -v
docker-machine version 0.10.0, build 76ed2a6

C:\>docker -v
Docker version 17.03.0-ce, build 60ccb22
```

Lab:

Exercise 1: Docker Installation
1651_docker_getting_started.zip



Docker Machine

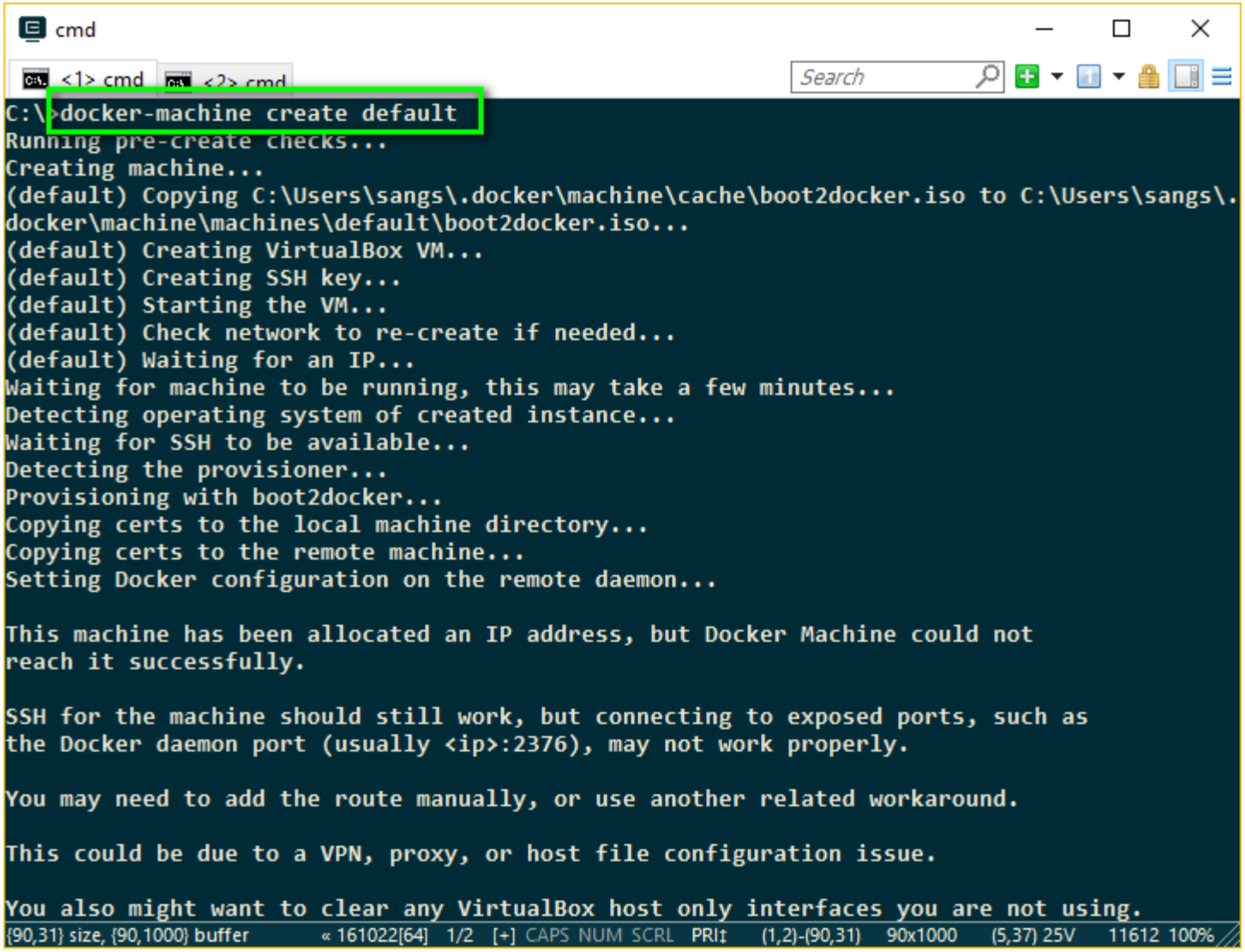
What is Docker Machine?

- Docker Machine is a tool that lets you install Docker Engine on virtual hosts, and manage the hosts with docker-machine commands
- You can use Docker Machine to create Docker hosts
  - on your local Mac or Windows box
  - on your company network in your data center
  - on cloud providers like AWS
- Using docker-machine commands, you can start, inspect, stop, and restart a managed host, upgrade the Docker client and daemon, and configure a Docker client to talk to your host
  - docker-machine --help

Lab: Docker machine commands help

```
C:\>docker-machine --help
Usage: docker-machine [OPTIONS] COMMAND [arg...]

Create and manage machines running Docker.

Version: 0.9.0, build 15fd4c7

Author:
  Docker Machine Contributors - <https://github.com/docker/machine>

Options:
  --debug, -D                                         Enable debug mode
  --storage-path, -s "C:\Users\sangs\.docker\machine"   Configures storage path [$MACHINE_STORAGE_PATH]
  --tls-ca-cert                                       CA to verify remotes against [$MACHINE_TLS_CA_CERT]
  --tls-ca-key                                        Private key to generate certificates [$MACHINE_TLS_CA_
  --tls-client-cert                                   Client cert to use for TLS [$MACHINE_TLS_CLIENT_CERT]
  --tls-client-key                                    Private key used in client TLS auth [$MACHINE_TLS_CLIE
  --github-api-token                                  Token to use for requests to the Github API [$MACHINE_
N]
  --native-ssh                                        Use the native (Go-based) SSH implementation. [$MACHIN
  --bugsnag-api-token                                 BugSnag API token for crash reporting [$MACHINE_BUGSNA
  --help, -h                                          show help
  --version, -v                                       print the version

Commands:
  active               Print which machine is active
  config               Print the connection config for machine
  create               Create a machine
  env                  Display the commands to set up the environment for the Docker client
  inspect              Inspect information about a machine
  ip                   Get the IP address of a machine
  kill                 Kill a machine
  ls                   List machines
  provision            Re-provision existing machines
  regenerate-certs     Regenerate TLS Certificates for a machine
  restart              Restart a machine
  rm                   Remove a machine
  ssh                  Log into or run a command on a machine with SSH.
  scp                  Copy files between machines
  start                Start a machine
  status               Get the status of a machine
  stop                 Stop a machine
```

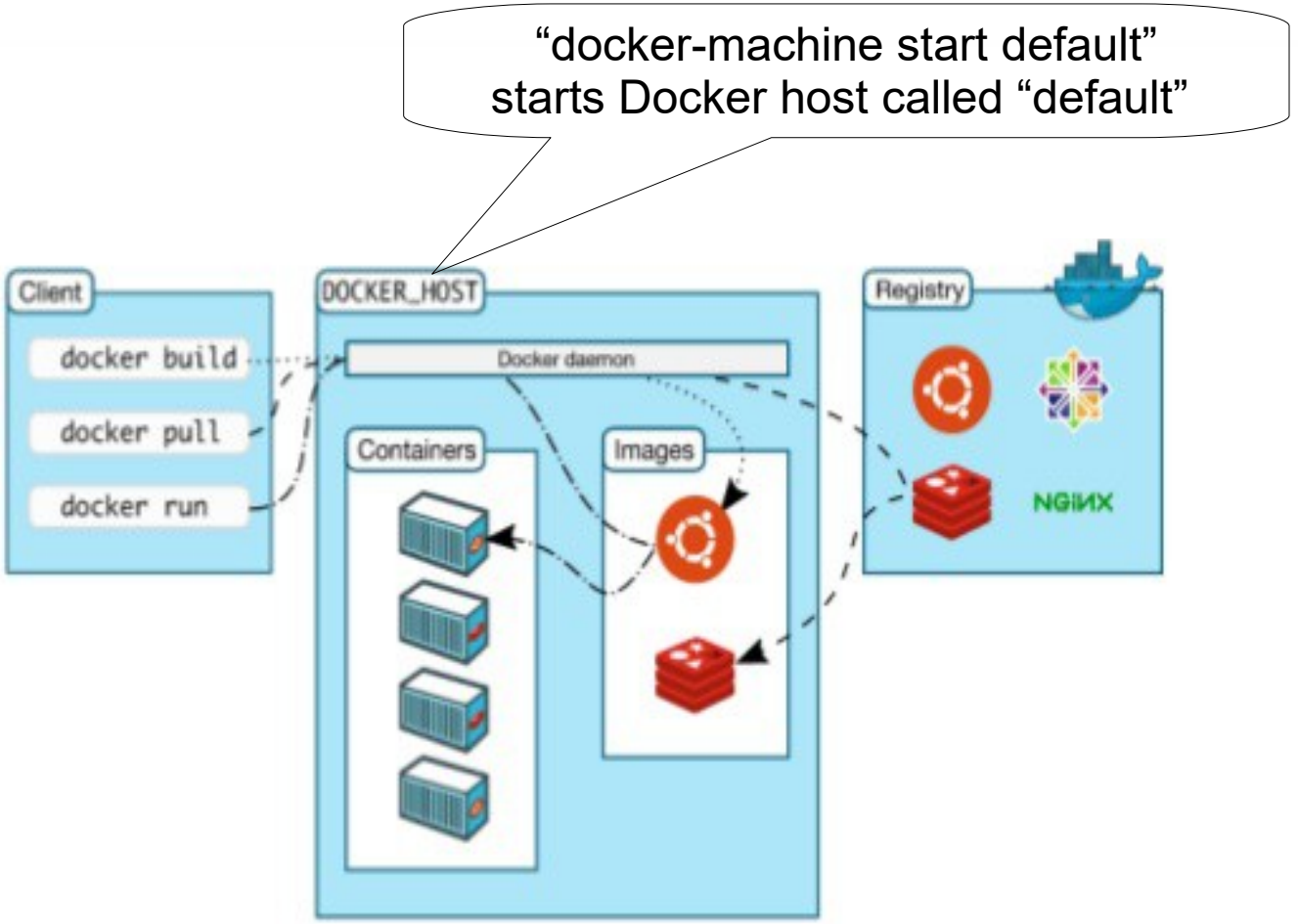
Lab: Create/Start “default” Docker machine



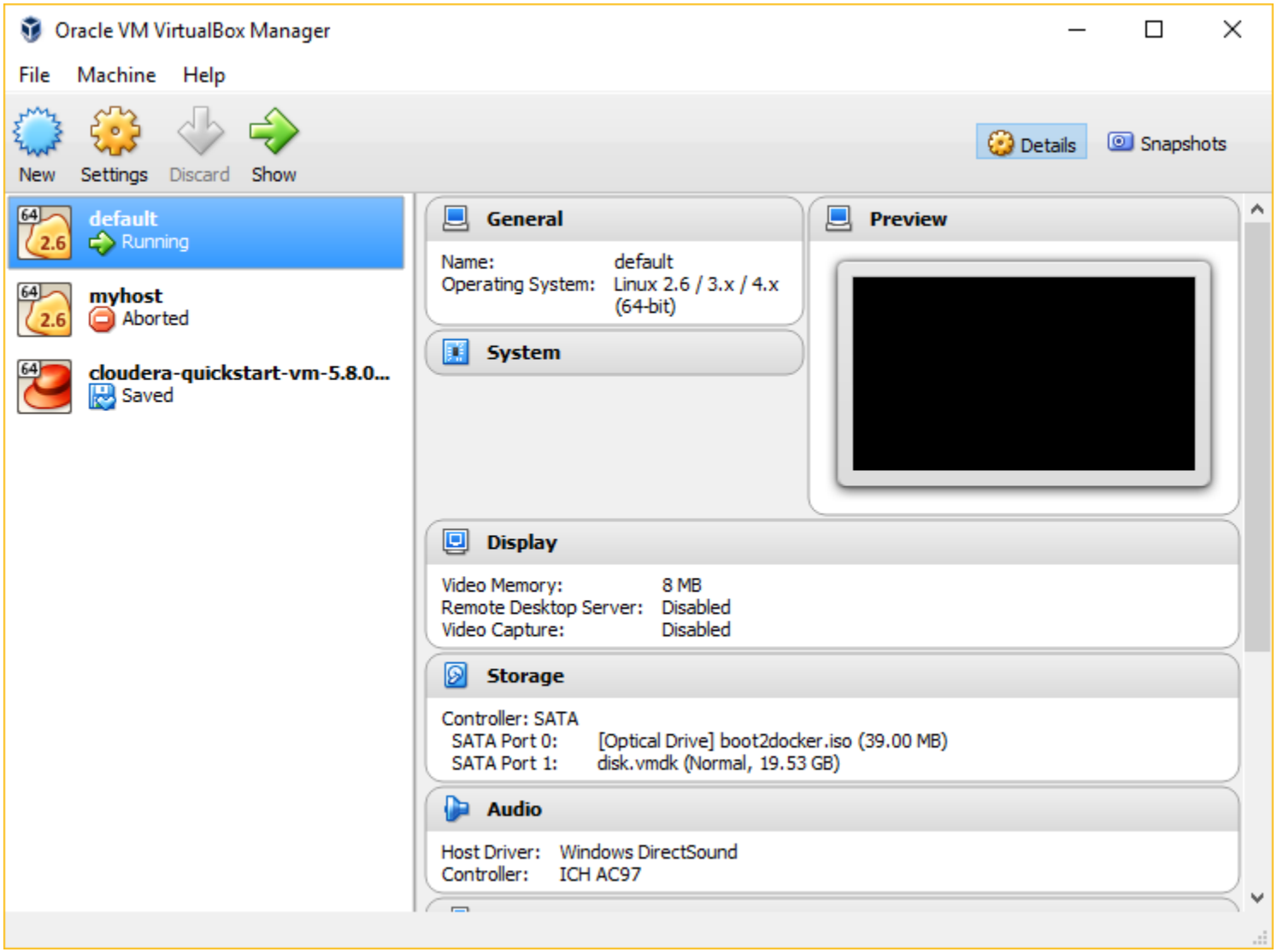
Lab: Start and Stop Docker machine

- *docker-machine start (or docker-machine start default)*
  - Starts “default” Docker machine (docker daemon)
- *docker-machine status (or docker-machine status default)*
  - Lists the status of default docker machine
- *docker-machine stop (or docker-machine stop default)*
  - Stops “default” Docker machine (docker daemon)

Docker Machine



Default Docker Machine called “default”



Lab: More Docker Machine commands

- *docker-machine ls*
  - List all docker machines
- *docker-machine ip*
  - Gets the IP address of the “default” docker machine
- *docker-machine url*
  - Gets the URL of the “default” docker machine

Show Docker Machines

```
C:\Users\sangs>docker-machine create another-machine
...
C:\Users\sangs>docker-machine ls
NAME              ACTIVE   DRIVER       STATE     URL                         SWARM   DOCKER        ERRORS
another-machine   -        virtualbox   Running   tcp://192.168.99.101:2376           v17.03.0-ce
default           *        virtualbox   Running   tcp://192.168.99.100:2376           v17.03.0-ce

C:\Users\sangs>docker-machine ip
192.168.99.100

C:\Users\sangs>docker-machine url
tcp://192.168.99.100:2376

C:\Users\sangs>docker-machine ip another-machine
192.168.99.101

C:\Users\sangs>docker-machine url another-machine
tcp://192.168.99.101:2376

C:\Users\sangs>docker-machine active
default
```

a machine is considered active if the DOCKER_HOST environment variable points to it

Connecting another running docker-machine

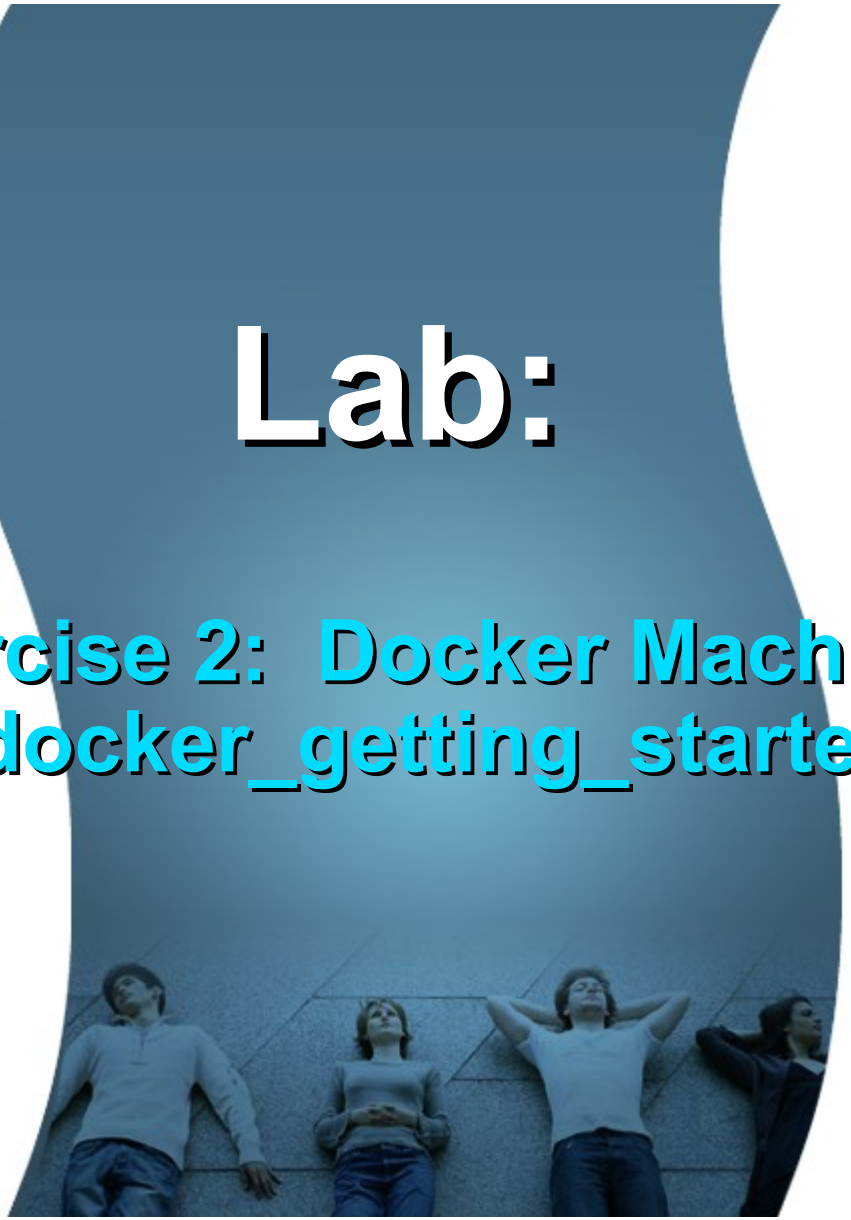
Open another terminal

```
C:\Users\sangs>docker-machine ls
NAME              ACTIVE   DRIVER       STATE     URL                         SWARM   DOCKER      ERRORS
another-machine   -        virtualbox   Running   tcp://192.168.99.101:2376           v17.03.0-ce
default           -        virtualbox   Running   tcp://192.168.99.100:2376           v17.03.0-ce

C:\Users\sangs>docker-machine active
No active host found

C:\Users\sangs>docker-machine env another-machine
SET DOCKER_TLS_VERIFY=1
SET DOCKER_HOST=tcp://192.168.99.101:2376
SET DOCKER_CERT_PATH=C:\Users\sangs\.docker\machine\machines\another-machine
SET DOCKER_MACHINE_NAME=another-machine
SET COMPOSE_CONVERT_WINDOWS_PATHS=true
REM Run this command to configure your shell:
REM     @FOR /f "tokens=*" %i IN ('docker-machine env another-machine') DO @%i

C:\Users\sangs>@FOR /f "tokens=*" %i IN ('docker-machine env another-machine') DO @%i

C:\Users\sangs>docker-machine ls
NAME              ACTIVE   DRIVER       STATE     URL                         SWARM   DOCKER      ERRORS
another-machine   *        virtualbox   Running   tcp://192.168.99.101:2376           v17.03.0-ce
default           -        virtualbox   Running   tcp://192.168.99.100:2376           v17.03.0-ce

C:\Users\sangs>docker-machine active
another-machine
```

a machine is considered active if the DOCKER_HOST environment variable points to it

Lab:

Exercise 2: Docker Machine
1651_docker_getting_started.zip



Docker Client

Lab: Set up docker client connection to Docker machine

- *docker-machine env*
  - Display the commands to set up the environment for the Docker client
- *@FOR /f "tokens=*" %i IN ('docker-machine env') DO @%i*
  - Sets up environment variables

- *docker version*
  - Shows both client and server version
- *docker info*
  - Display system-wide information

Show Docker Client and Server Versions

```
C:\Users\sangs>docker version
Client:
 Version:      17.03.0-ce
 API version:  1.26
 Go version:   go1.7.5
 Git commit:   60ccb22
 Built:        Thu Mar  2 01:11:00 2017
 OS/Arch:      windows/amd64

Server:
 Version:      17.03.0-ce
 API version:  1.26 (minimum version 1.12)
 Go version:   go1.7.5
 Git commit:   3a232c8
 Built:        Tue Feb 28 07:52:04 2017
 OS/Arch:      linux/amd64
 Experimental: false
```

Show System-wide information

```
C:\Users\sangs>docker info
Containers: 4
 Running: 0
 Paused: 0
 Stopped: 4
Images: 6
Server Version: 17.03.0-ce
Storage Driver: aufs
 Root Dir: /mnt/sda1/var/lib/docker/aufs
 Backing Filesystem: extfs
 Dirs: 69
 Dirperm1 Supported: true
Logging Driver: json-file
Cgroup Driver: cgroupfs
Plugins:
 Volume: local
 Network: bridge host macvlan null overlay
Swarm: inactive
Runtimes: runc
Default Runtime: runc
...
```

Docker Command Help

```
C:\>docker --help
Usage:  docker COMMAND

A self-sufficient runtime for containers

Options:
      --config string      Location of client config files (default "C:\Users\sangs\.docker")
  -D, --debug              Enable debug mode
      --help               Print usage
  -H, --host list          Daemon socket(s) to connect to (default [])
...

Management Commands:
  container   Manage containers
  image       Manage images
  network     Manage networks
  node        Manage Swarm nodes
  plugin      Manage plugins
  secret      Manage Docker secrets
  service     Manage services
  stack       Manage Docker stacks
  swarm       Manage Swarm
  system      Manage Docker
  volume      Manage volumes

Commands:
```

Docker Command Help

```
C:\>docker system --help

Usage:  docker system COMMAND

Manage Docker

Options:
      --help   Print usage

Commands:
  df          Show docker disk usage
  events      Get real time events from the server
  info        Display system-wide information
  prune       Remove unused data

Run 'docker system COMMAND --help' for more information on a command.

C:\>docker system df
TYPE                TOTAL               ACTIVE              SIZE                RECLAIMABLE
Images              6                   4                   1.438 GB            706.8 MB (49%)
Containers          4                   2                   8.79 kB             0 B (0%)
Local Volumes       4                   1                   293.3 MB            86.45 MB (29%)
```

http://files.zeroturnaround.com/pdf/zt_docker_cheat_sheet.pdf

docker Cheat Sheet

REBELLABS by ZEROTURNAROUND

Glossary

Layer - a set of read-only files to provision the system

Image - a read-only layer that is the base of your container. Might have a parent image

Container - a runnable instance of the image

Registry / Hub - central place where images live

Docker machine - a VM to run Docker containers (Linux does this natively)

Docker compose - a utility to run multiple containers as a system

Useful one-liners

Download an image
```
docker pull image_name
```

Start and stop the container
```
docker [start|stop] container_name
```

Create and start container, run command
```
docker run -ti --name container_name
    image_name command
```

Create and start container, run command, destroy container
```
docker run --rm -ti image_name command
```

Example filesystem and port mappings
```
docker run -it --rm -p 8080:8080 -v
/path/to/agent.jar:/agent.jar -e
JAVA_OPTS="-javaagent:/agent.jar"
tomcat:8.0.29-jre8
```

Docker cleanup commands

Kill all running containers
```
docker kill $(docker ps -q)
```

Delete dangling images
```
docker rmi $(docker images -q -f
    dangling=true)
```

Remove all stopped containers
```
docker rm $(docker ps -a -q)
```

Docker machine commands

Use docker-machine to run the containers

Start a machine
```
docker-machine start machine_name
```

Configure docker to use a specific machine
```
eval "$(docker-machine env machine_name)"
```

Docker compose syntax

docker-compose.yml file example
```
version: "2"
services:
web:
  container_name: "web"
  image: java:8 # image name
  # command to run
  command: java -jar /app/app.jar
  ports: # map ports to the host
    - "4567:4567"
  volumes: # map filesystem to the host
    - ./myapp.jar:/app/app.jar
mongo: # container name
  image: mongo # image name
```

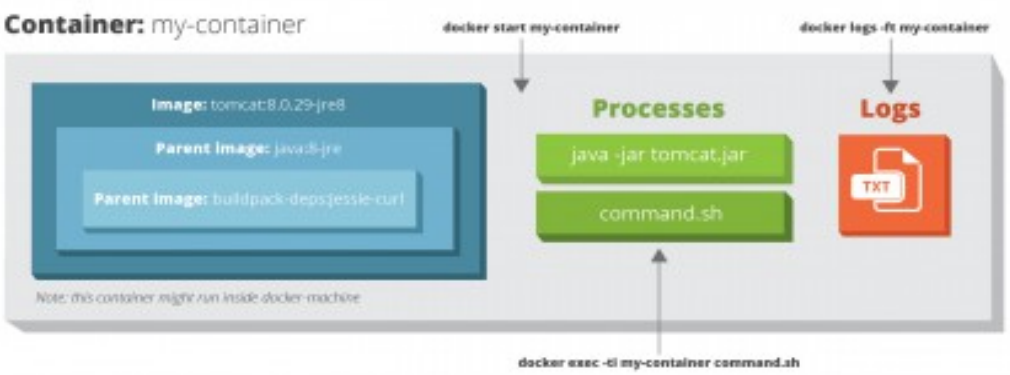
Create and start containers
```
docker-compose up
```

Interacting with a container

Run a command in the container
```
docker exec -ti container_name command.sh
```

Follow the container logs
```
docker logs -ft container_name
```

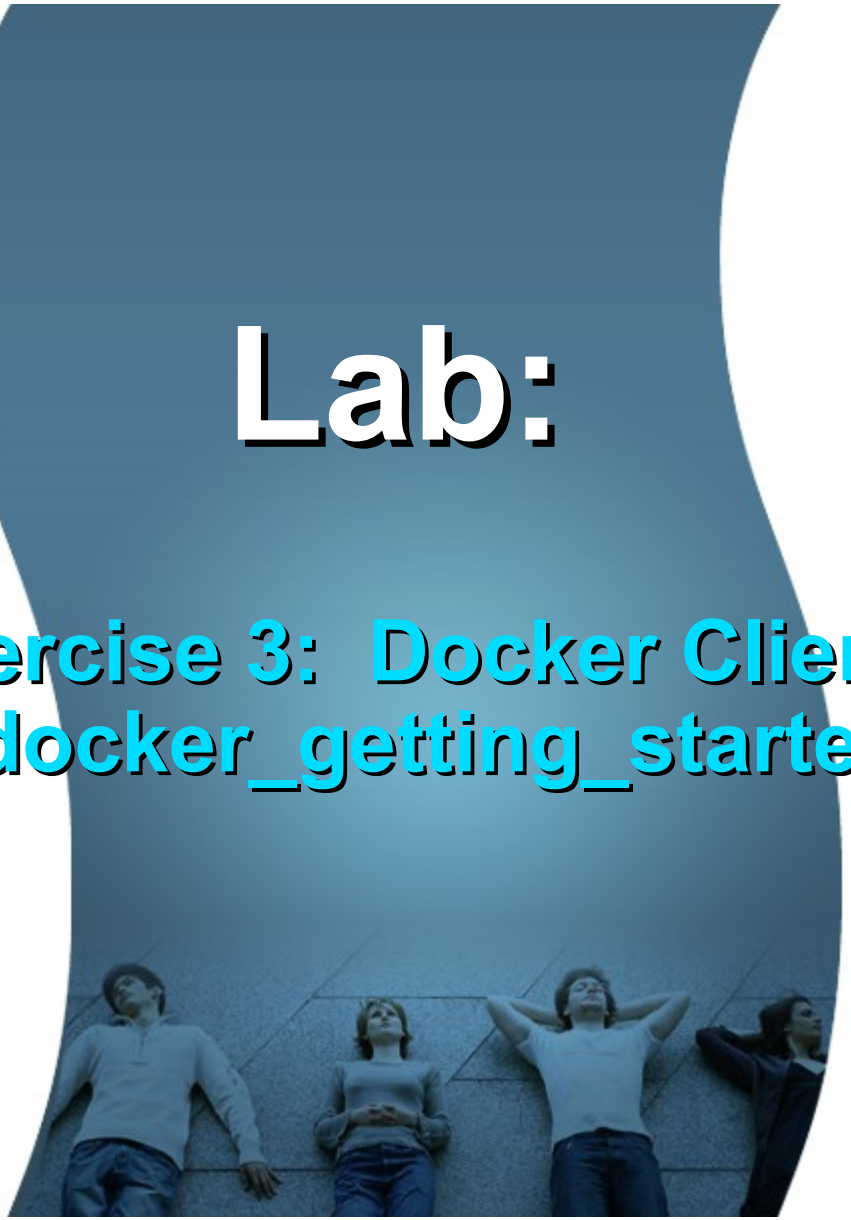
Save a running container as an image
```
docker commit -a "commit message" -a "author"
    container_name username/image_name:tag
```



BROUGHT TO YOU BY JRebel

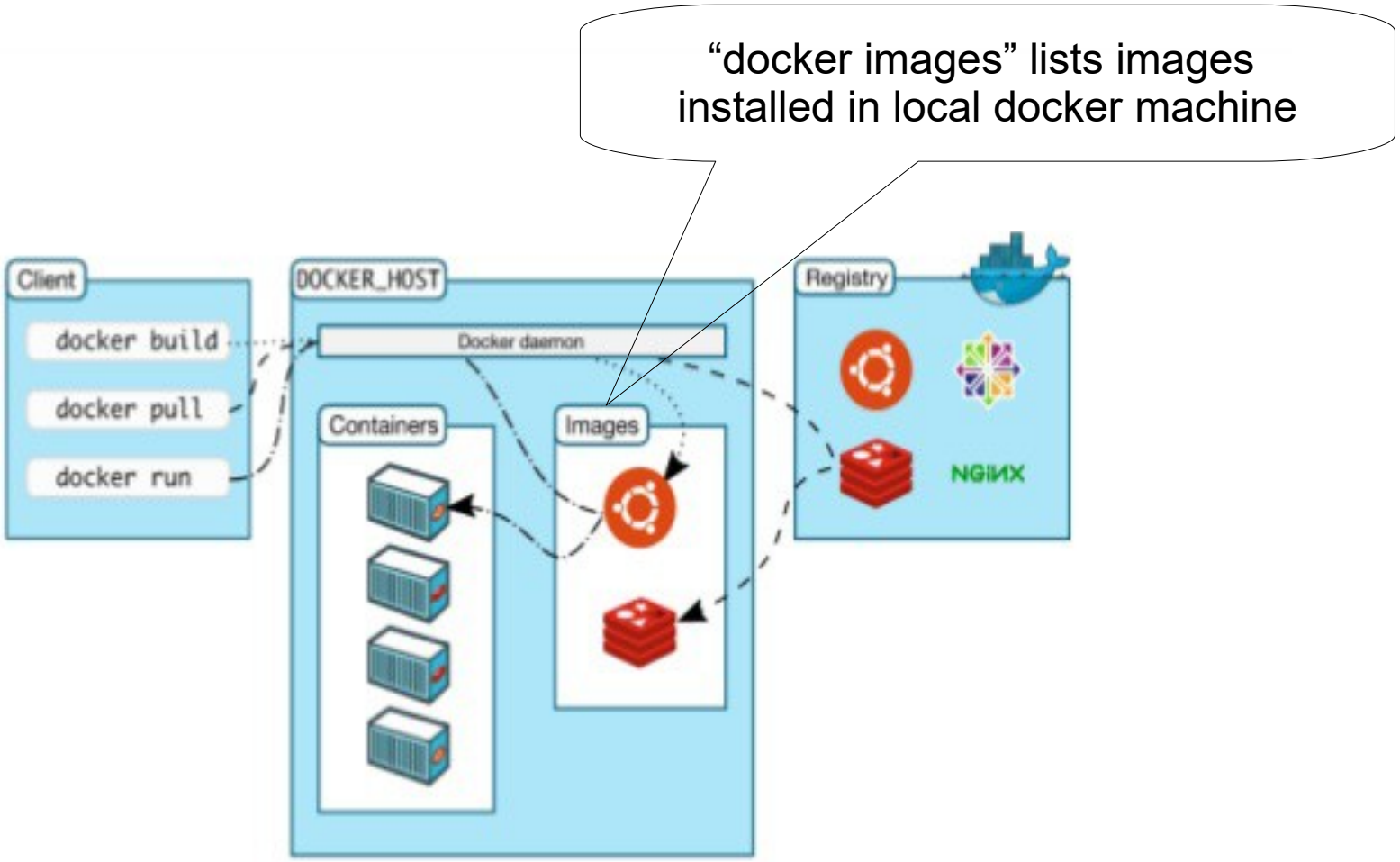
Lab:

Exercise 3: Docker Client
1651_docker_getting_started.zip



Docker Images Installed

Docker Images



Lab: Docker Images

- docker images (detailed info) or docker images -q (displays just ids)
  - Get the list of installed images
- docker pull ubuntu:12.04, docker pull ubuntu, docker pull ubuntu:latest
  - Pulls and installs the image (only if it has not been pulled before)
- docker inspect <image-name>
  - Get details of the image
- docker history <image-name>
  - Shows history of the image – each line represents a layer with exact size
- docker rmi <image-name> <image-name> …
  - Removes one or more images
- docker save <image-name> -o <name>.tar and docker load -i ./<name>

Show All Images Locally Installed

```
PS C:\Users\sangs> docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
fedora	latest	1f8ec1108a3f	4 weeks ago	230 MB
nginx	latest	db079554b4d2	4 weeks ago	182 MB
mongo	latest	ad974e767ec4	5 weeks ago	402 MB
openjdk	latest	8dde5631d4aa	6 weeks ago	641 MB
dockercloud/cli	latest	c4eadf2dd53a	6 weeks ago	63.8 MB
ubuntu	latest	f49eec89601e	2 months ago	129 MB
registry	latest	d1e32b95d8e8	2 months ago	33.2 MB
hello-world	latest	48b5124b2768	2 months ago	1.84 kB

Show Image ID's only

```
PS C:\Users\sangs> docker images -q
1f8ec1108a3f
db079554b4d2
ad974e767ec4
8dde5631d4aa
c4eadf2dd53a
f49eec89601e
d1e32b95d8e8
48b5124b2768
```

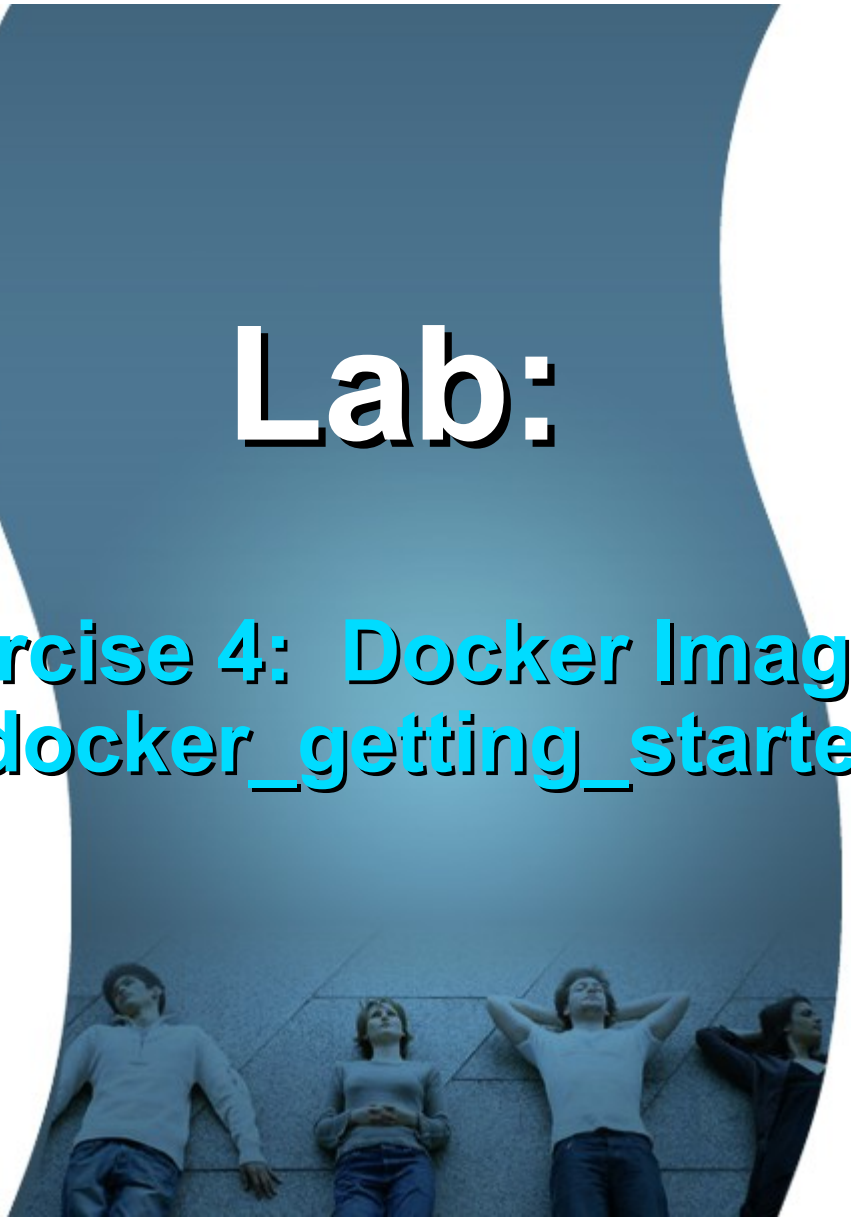
Lab: Docker Images – removal of images

- docker rmi $(docker images -q)
  - Remove all images
  - On Windows, run it over powershell
- docker images --filter “dangling=true”
  - Find dangling images
- docker rmi $(docker images --filter “dangling=true” -q)
  - Search all dangling images and remove them

- docker images prune
  - Prunes unused images
  - Same as above

Lab:

Exercise 4: Docker Images
1651_docker_getting_started.zip



Docker Images In the Registry

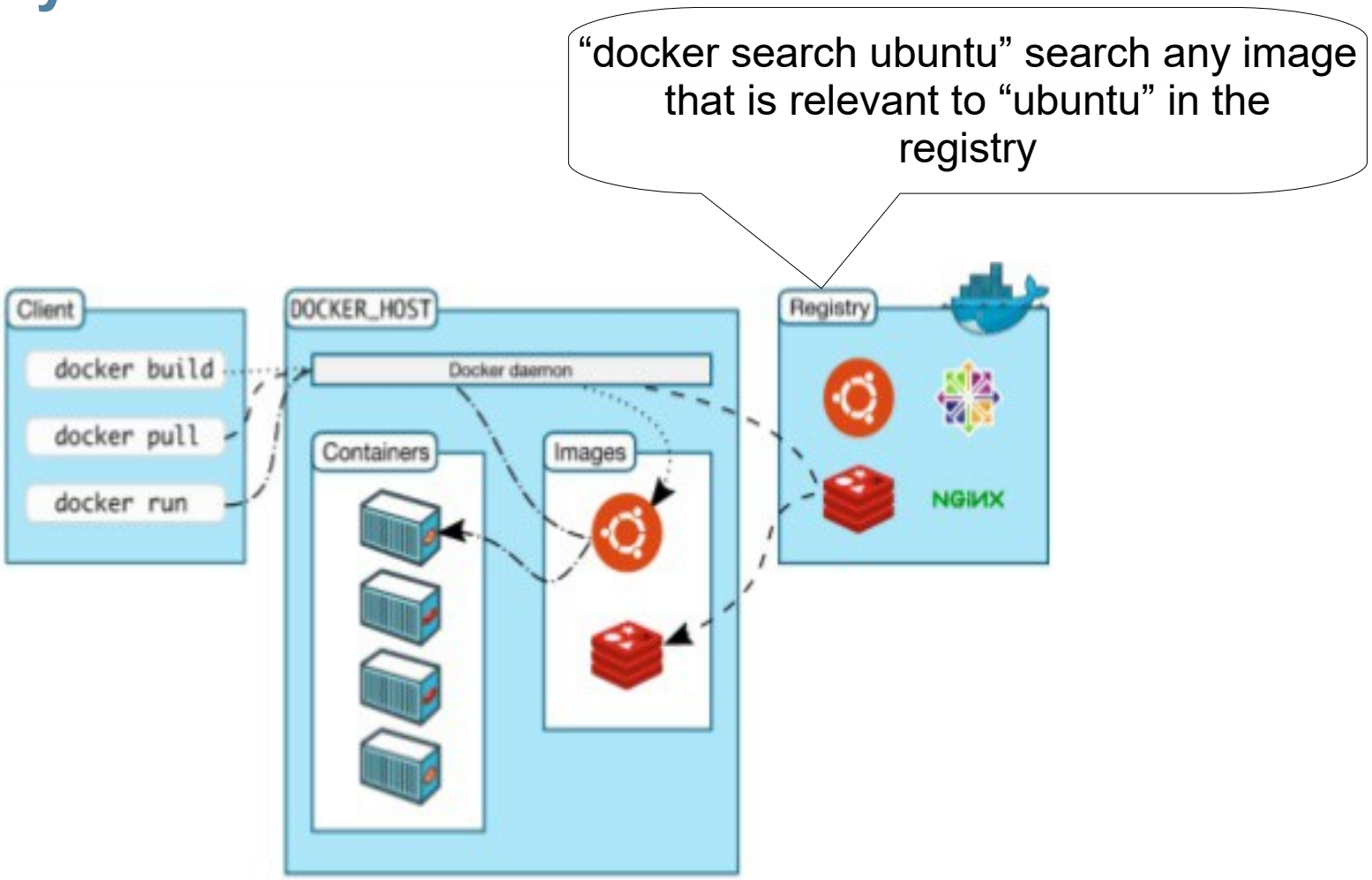
Lab: Docker Images

- docker search <image-name>
  - Searches <image-name> from docker repository
- <image-name> syntax
  - username/imagename:version for user image
  - imagename for official image

Search for an Image from Image Registry

```
C:\Users\sangs>docker search ubuntu
NAME                                DESCRIPTION                                     STARS     OFFICIAL   AUTOMATED
ubuntu                              Ubuntu is a Debian-based Linux operating s...   5725      [OK]
rastasheep/ubuntu-sshd              Dockerized SSH service, built on top of of...   77                   [OK]
ubuntu-upstart                      Upstart is an event-based replacement for ...   71        [OK]
consol/ubuntu-xfce-vnc              Ubuntu container with "headless" VNC sessi...   45                   [OK]
ubuntu-debootstrap                  debootstrap --variant=minbase --components...   30        [OK]
torusware/speedus-ubuntu            Always updated official Ubuntu docker imag...   27                   [OK]
nuagebec/ubuntu                     Simple always updated Ubuntu docker images...   17                   [OK]
nickistre/ubuntu-lamp               LAMP server on Ubuntu                           16                   [OK]
...
```

Registry


“docker search ubuntu” search any image that is relevant to “ubuntu” in the registry
Client
docker build
docker pull
docker run
DOCKER_HOST
Docker daemon
Containers
Images
Registry
NGINX

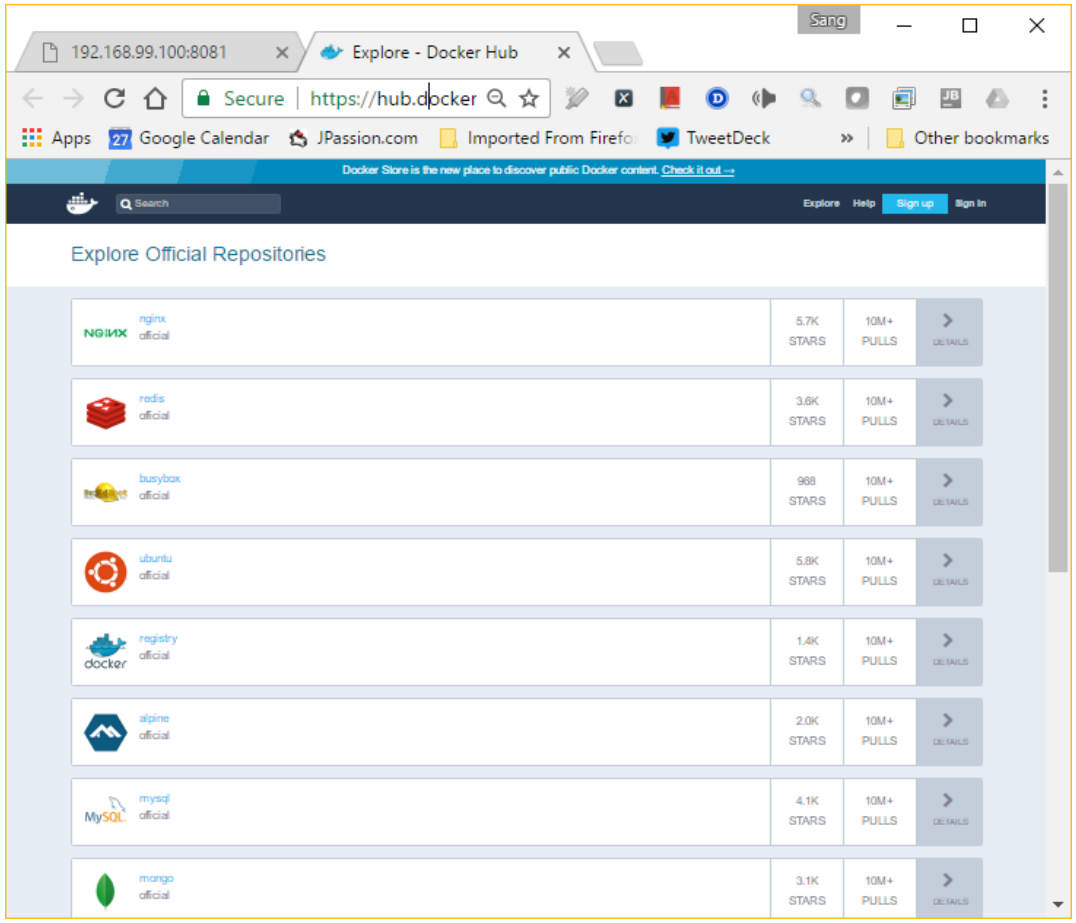
Types of Docker Images

Types of Docker Images

- Official images vs User images
  - Official images are Docker sanctioned images. These are not prefixed by an organization or user name. In the list of images above, the python, node, alpine and nginx images are official (base) images.
  - User images are images created and shared by users like you. They build on base images and add additional functionality. Typically these are formatted as user/image-name. The user value in the image name is your Docker Hub user or organization name
- Base images vs Child images
  - Base images are images that have no parent images, usually images with an OS like ubuntu, alpine or debian
  - Child images are images that build on base images and add additional functionality.

Official Docker Images

- https://hub.docker.com/explore/

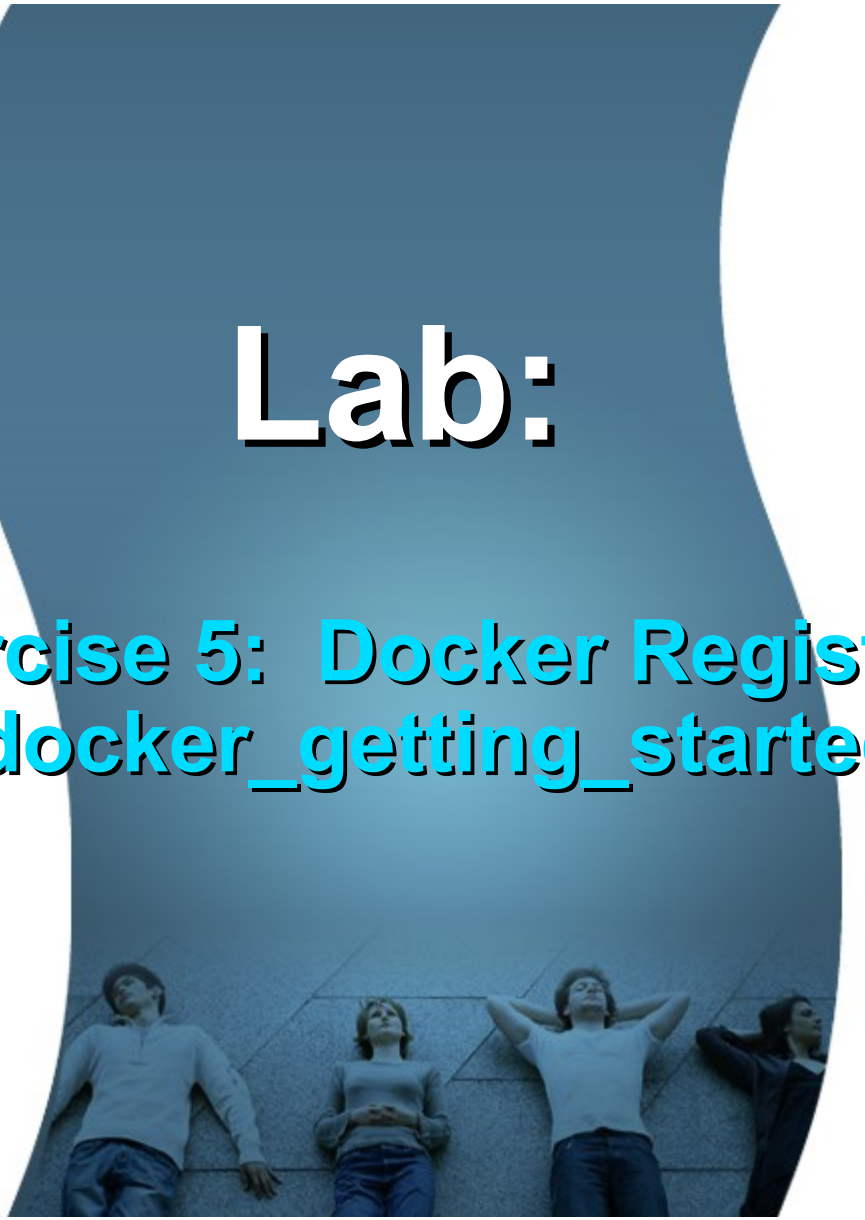


Lab: See history of image

```
C:\>docker history seqvence/static-site
IMAGE          CREATED          CREATED BY                                      SIZE          COMMENT
f589ccde7957       12 months ago       /bin/sh -c #(nop) CMD ["/bin/sh" "-c" "cd ...   0 B
<missing>          12 months ago       /bin/sh -c #(nop) WORKDIR /usr/share/nginx...   0 B
<missing>          12 months ago       /bin/sh -c #(nop) COPY file:c8203f6bfe2ff6...   8.75 kB
<missing>          12 months ago       /bin/sh -c mkdir -p /usr/share/nginx/html       0 B
<missing>          12 months ago       /bin/sh -c #(nop) ENV AUTHOR=Docker             0 B
<missing>          12 months ago       /bin/sh -c #(nop) CMD ["nginx" "-g" "daemo...   0 B
<missing>          12 months ago       /bin/sh -c #(nop) EXPOSE 443/tcp 80/tcp         0 B
<missing>          12 months ago       /bin/sh -c ln -sf /dev/stdout /var/log/ngi...   22 B
<missing>          12 months ago       /bin/sh -c apt-key adv --keyserver hkp://p...   65.4 MB
<missing>          12 months ago       /bin/sh -c #(nop) ENV NGINX_VERSION=1.9.12...   0 B
<missing>          13 months ago       /bin/sh -c #(nop) MAINTAINER NGINX Docker ...   0 B
<missing>          13 months ago       /bin/sh -c #(nop) CMD ["/bin/bash"]             0 B
<missing>          13 months ago       /bin/sh -c #(nop) ADD file:b5391cb13172fb5...   125 MB
```

Lab:

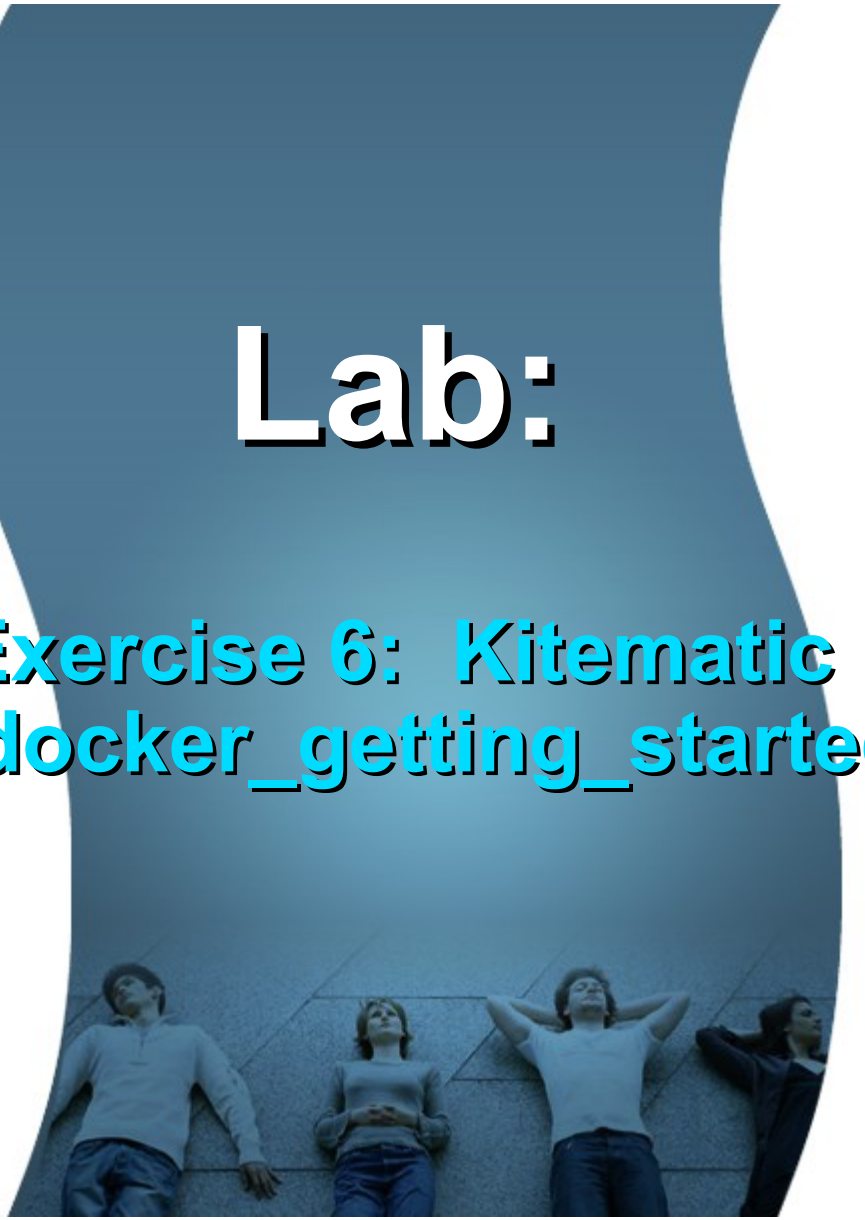
Exercise 5: Docker Registry
1651_docker_getting_started.zip



Kitematic

Lab:

Exercise 6: Kitematic
1651_docker_getting_started.zip



Code with Passion!
JPassion.com