Intro to Docker

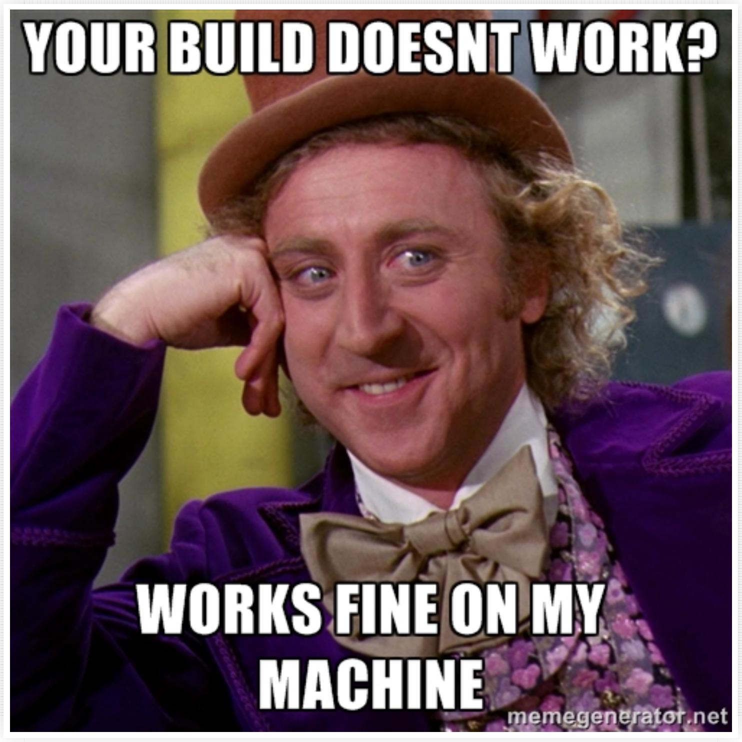
Container technology so easy it should be illegal not to use it.

Linux User Group of Davis (LUGOD)
05/16/2016

Outline

- Slides available at http://goo.gl/cj70Fw
- Why Docker?
- What is Docker?
- Basic docker terminologies and commands
- Hands-on exercise:
  - build a Docker image
  - run Docker container to update theHackerWithin Davis chapter website

Why Docker?


YOUR BUILD DOESNT WORK?
WORKS FINE ON MY MACHINE
memegenerator.net

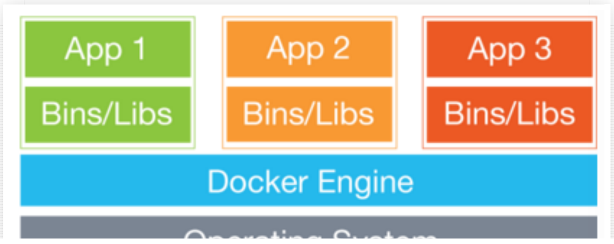
Why Docker?

- Helps reproduce an exact software environment
- Good for:
  - software development / deployment
  - reproducible science!
  - teaching (and grading)

What is Docker?

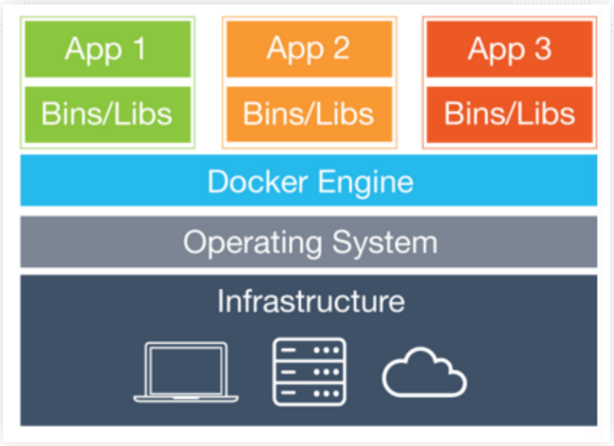
An open-source technology that allows you to package an app with all its dependencies into a standardized unit for software development.

-Docker website



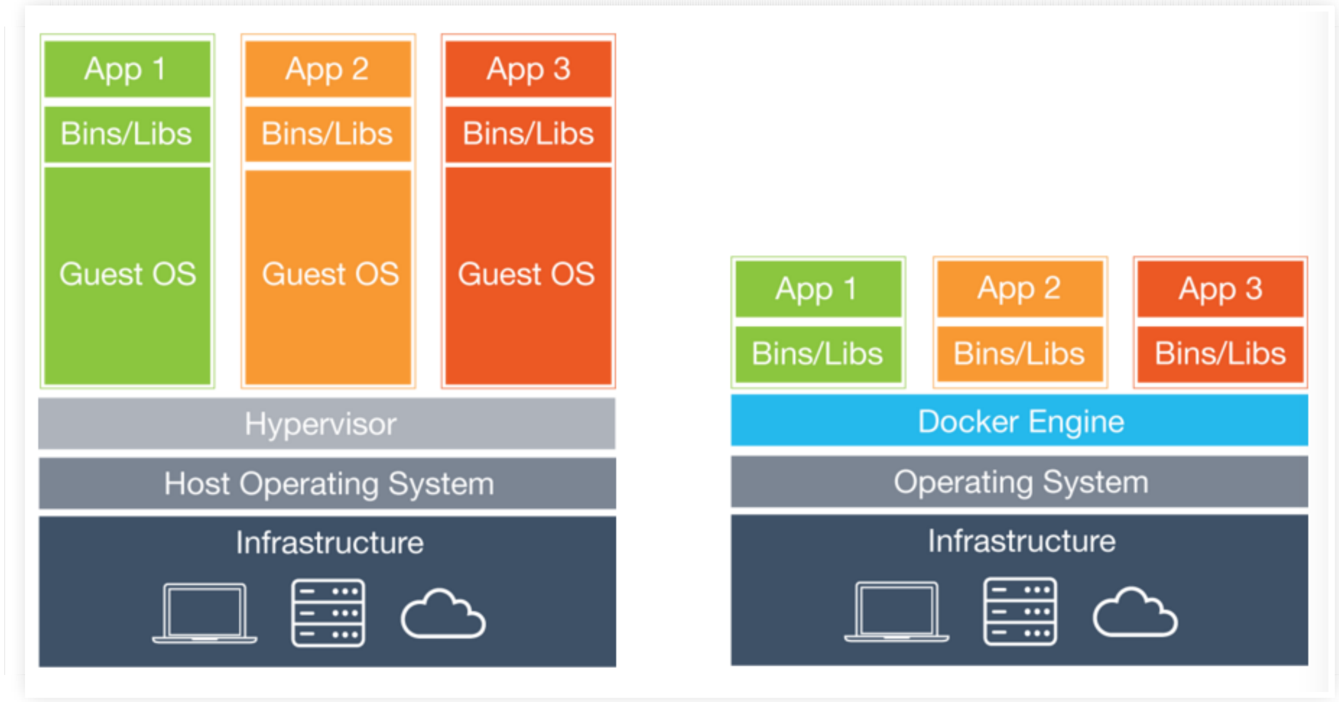
What is Docker?

- One container for running approximately one app



0

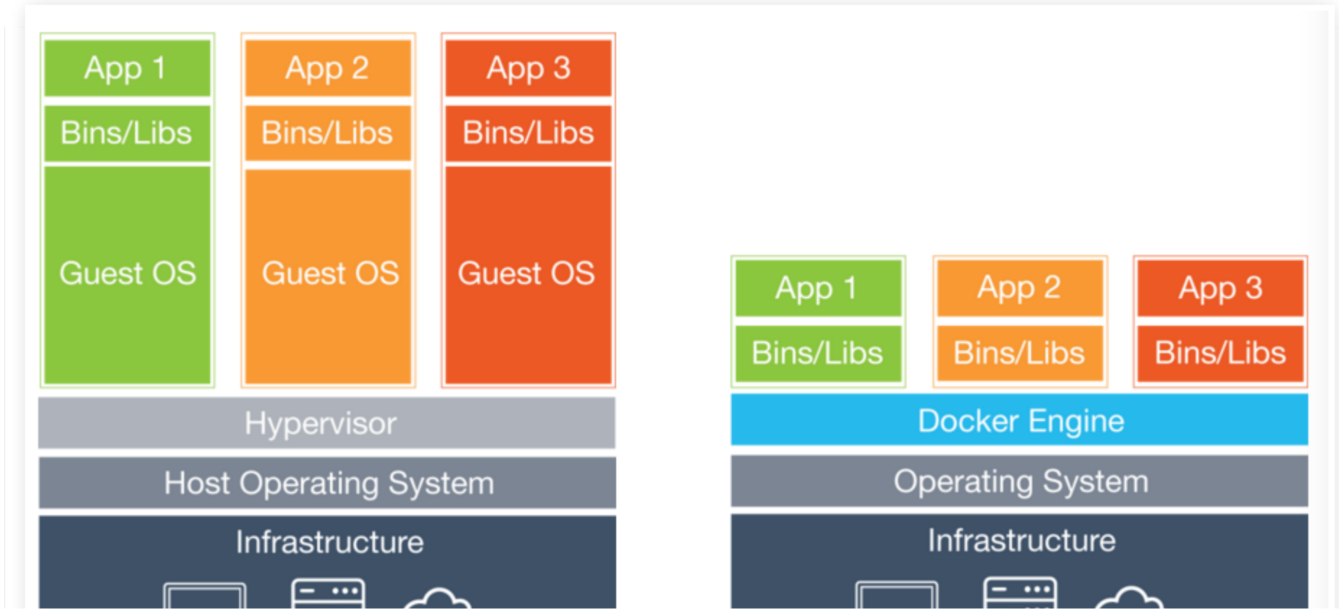
Virtual machines (left) vs Docker (right)



Software container = operating-system-level

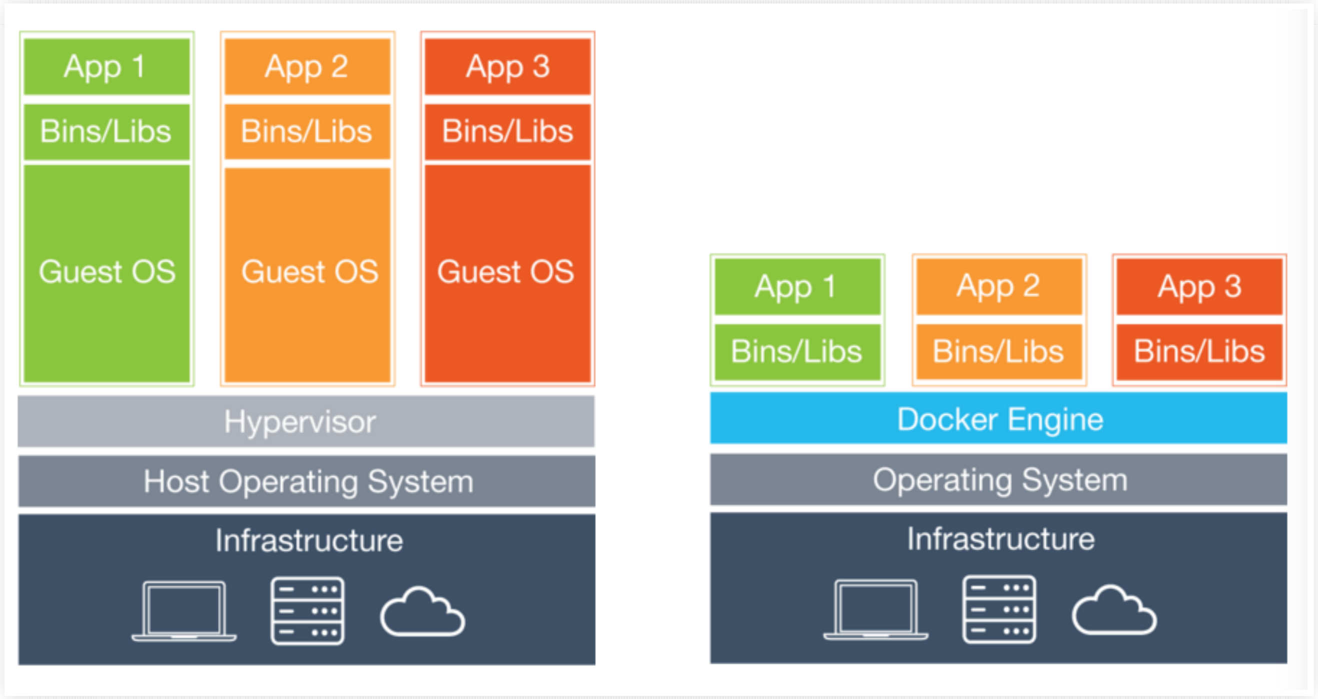
Virtual machines (left) vs Docker (right)

VM has more overhead



What is Docker?

- Containers start up faster (~500 ms)



Basic Docker terminologies

Dockerhub

an online registry where you can pull and host publicly available Docker images

```
$ docker pull karenyng/hackerwithin_dockerfile
```

searches Dockerhub for an image file

Docker image

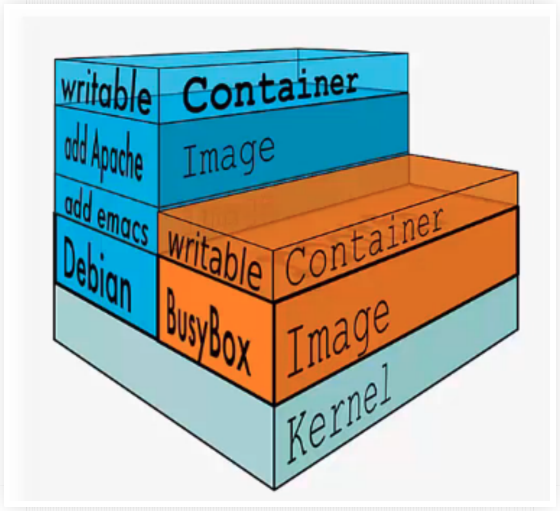
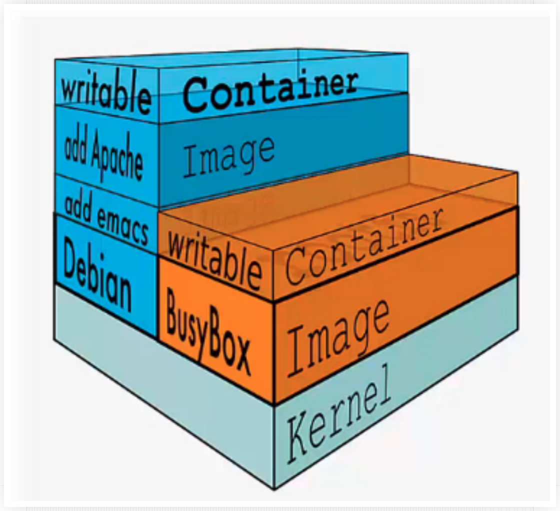


Fig. Docker infrastructure if host machine is Linux. (fig stolen from Docker)

- static image that packages Linux system runtime / library files along with the software
- can have parent / base image
- immutable

Container

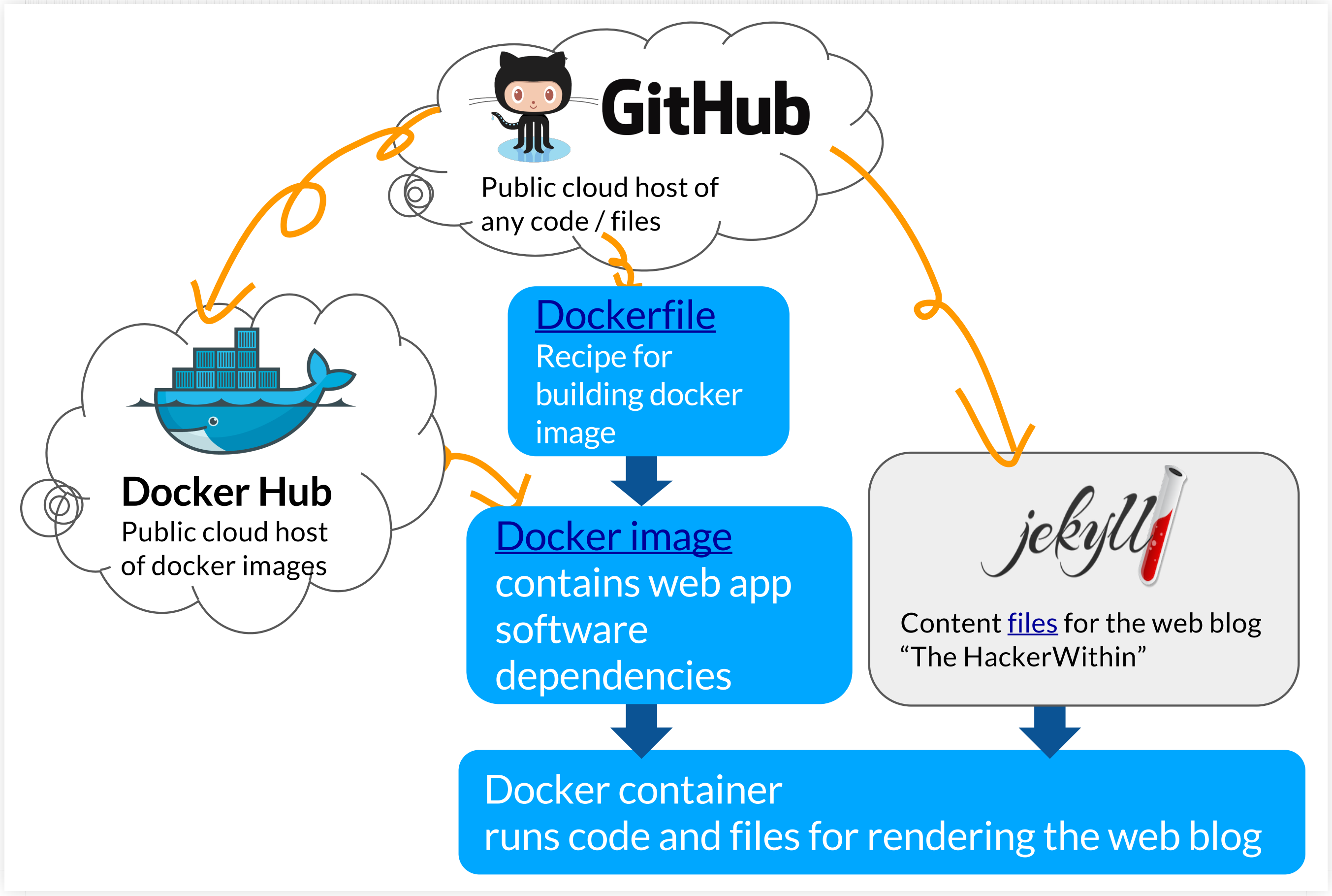


a running / stopped instance of an image

Dockerfile

- a recipe for building a Docker image
- can contain runtime configurations
- a Dockerfile example vs the original installation guide

Hands-on session


GitHub
Public cloud host of any code / files
Docker Hub
Public cloud host of docker images
Dockerfile
Recipe for building docker image
Docker image
contains web app software dependencies
jekyll
Content files for the web blog "The HackerWithin"
Docker container
runs code and files for rendering the web blog

Start Docker daemon

Linux users execute:

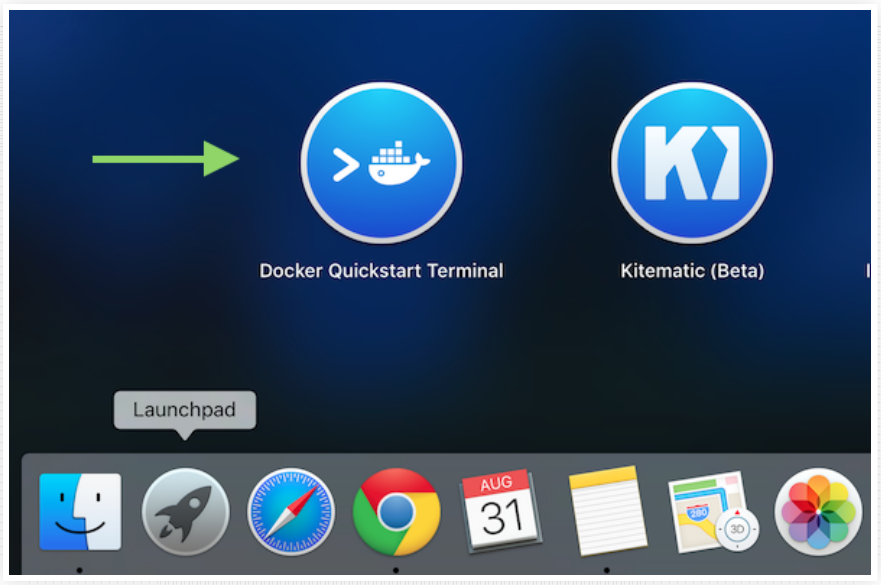
```
$ sudo docker -d &
```

Instructions for using `Docker` without `sudo` here.

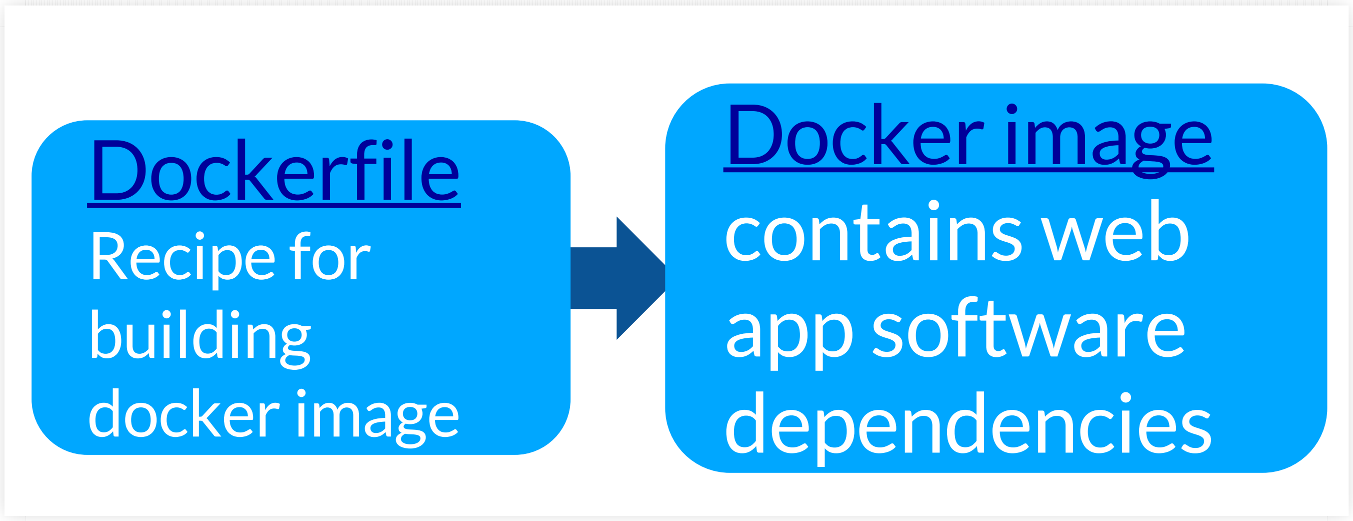
Other caveats: 64-bit Linux needed

Alternative instruction for Mac and Windows User

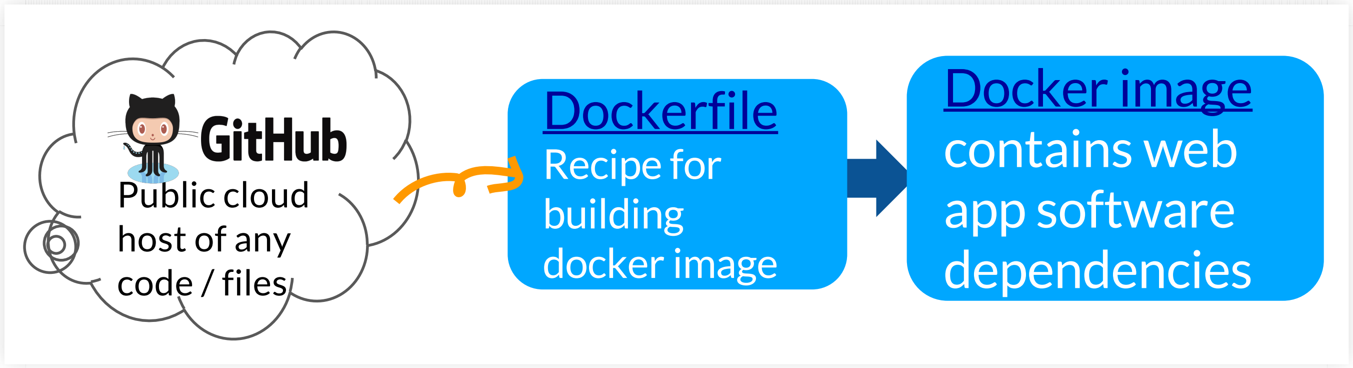
Start Docker QuickStart terminal



Ex1: Building the image from a Dockerfile


Dockerfile
Recipe for building docker image
Docker image
contains web app software dependencies

Ex1: Building the image from a Dockerfile


GitHub
Public cloud host of any code / files
Dockerfile
Recipe for building docker image
Docker image
contains web app software dependencies

```
$ git clone \
https://github.com/karenyyng/hackerwithin_docke
$ cd hackerwithin_dockerfile
$ vim Dockerfile
```

Dockerfile ref API

0

How did I build and debug the image?

switch to a problematic dockerfile first...

```
$ git fetch origin  # fetch all remote branches
$ git checkout -b failed origin/failed
$ docker build -t karenyng/silly .
```

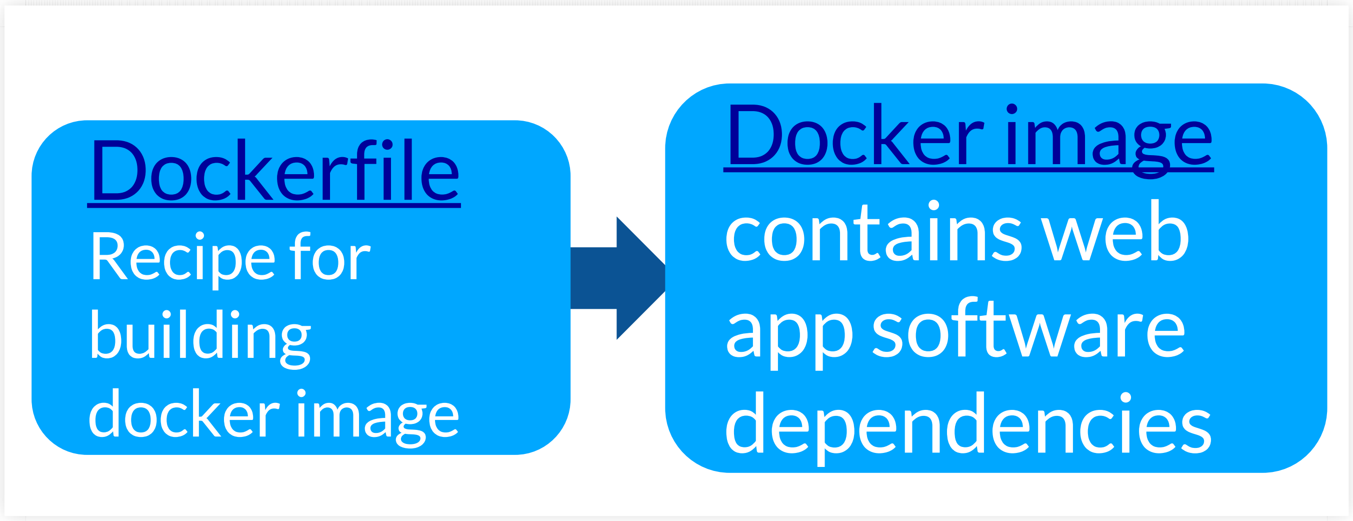
where `-t` tags the image with `REPO_NAME/IMAGE_NAME`.

Debug problematic image by running the image before problem occurs

Get the CACHED IMAGE HASH from printed message.

```
$ docker run -ti <CACHED IMAGE HASH>
```

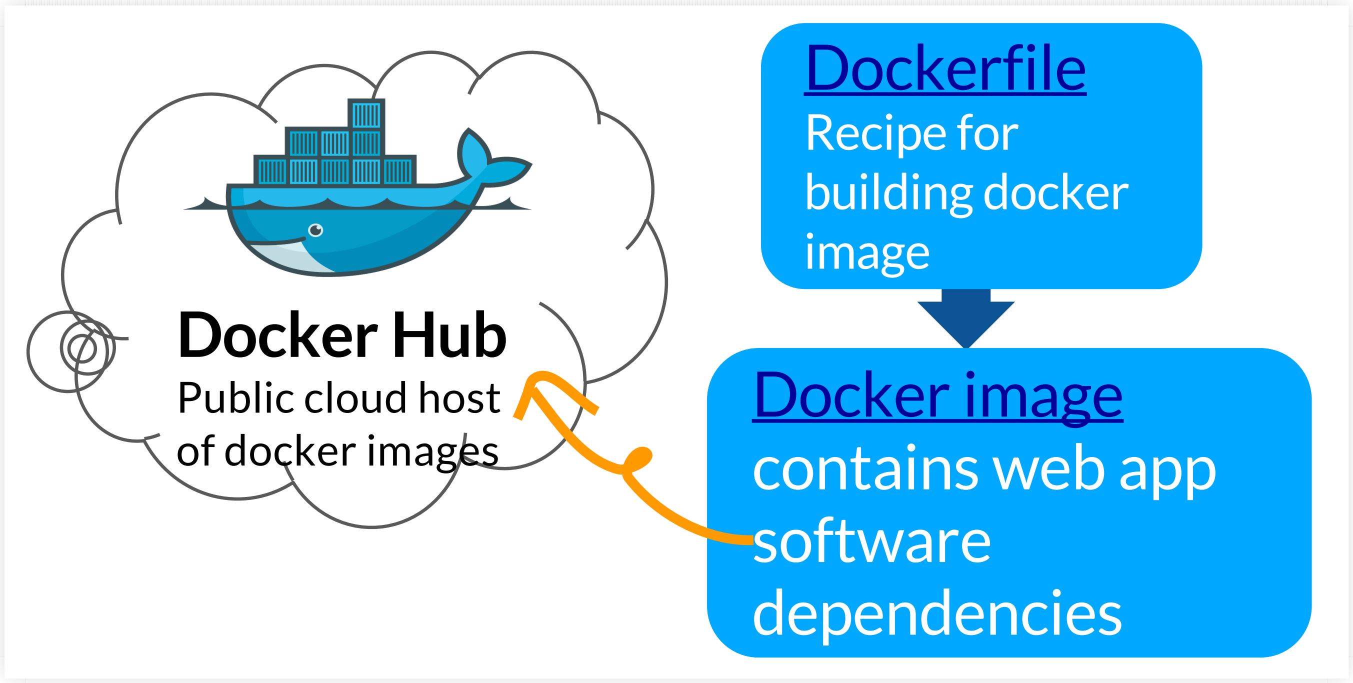
Build image then check what images are locally available


Dockerfile
Recipe for building docker image
Docker image
contains web app software dependencies

```
$ docker build -t MY_DOCKERHUB_REPONAME/silly .
$ docker images
```

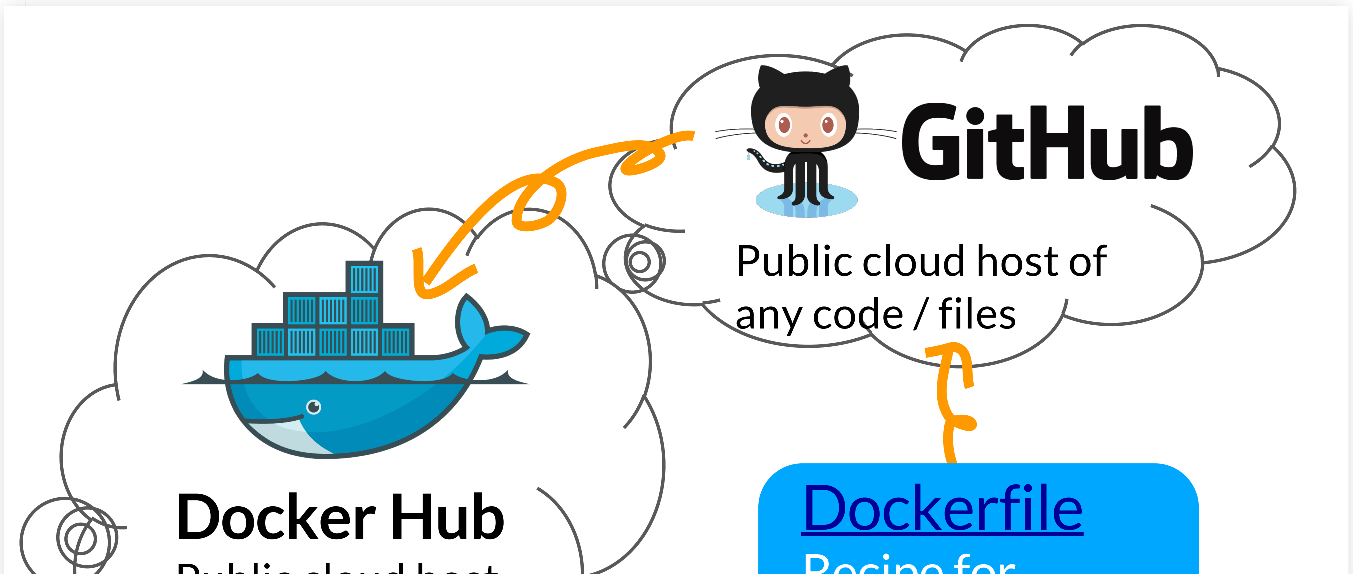
`-t` tags the `REPO/IMAGE_NAME`

Login and push image to DockerHub


Docker Hub
Public cloud host of docker images
Dockerfile
Recipe for building docker image
Docker image
contains web app software dependencies

```
$ docker login
$ docker push MY_DOCKERHUB_REPONAME/silly
```

Or commit Dockerfile to GitHub for an automated build of Docker image on DockerHub



Check what containers are running

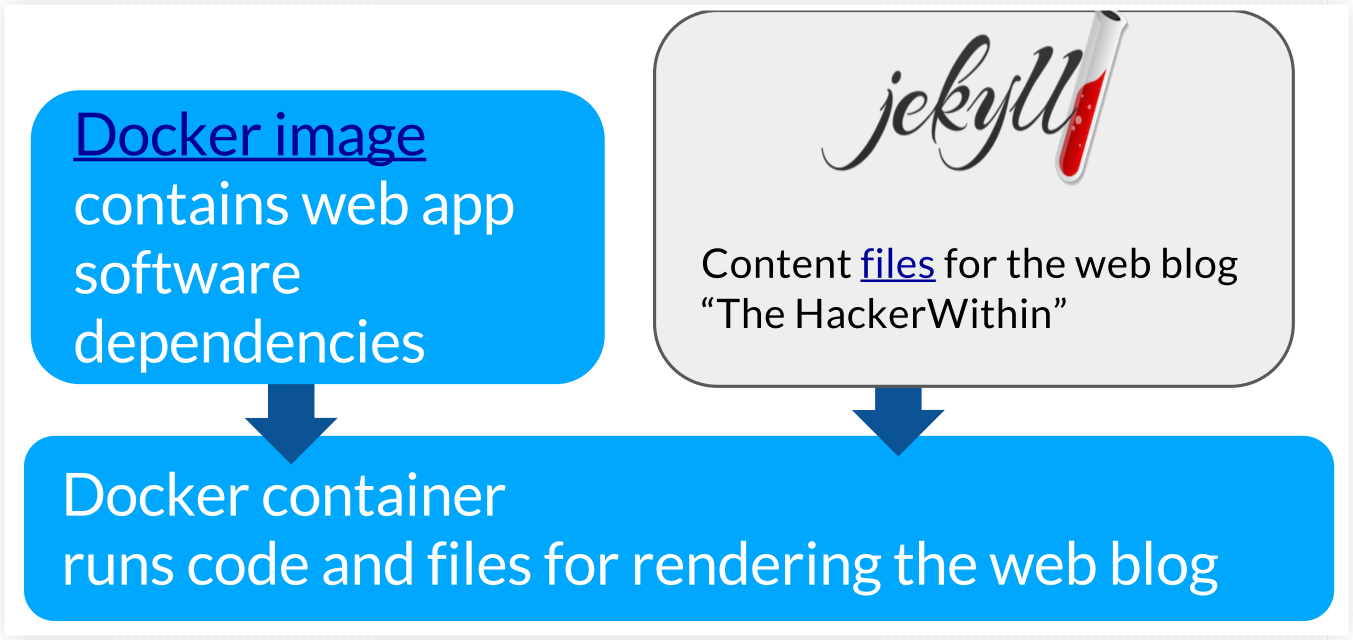
```
# only shows running containers
$ docker ps

# shows all the containers
$ docker ps -a
```

Saving the container as an image on local machine

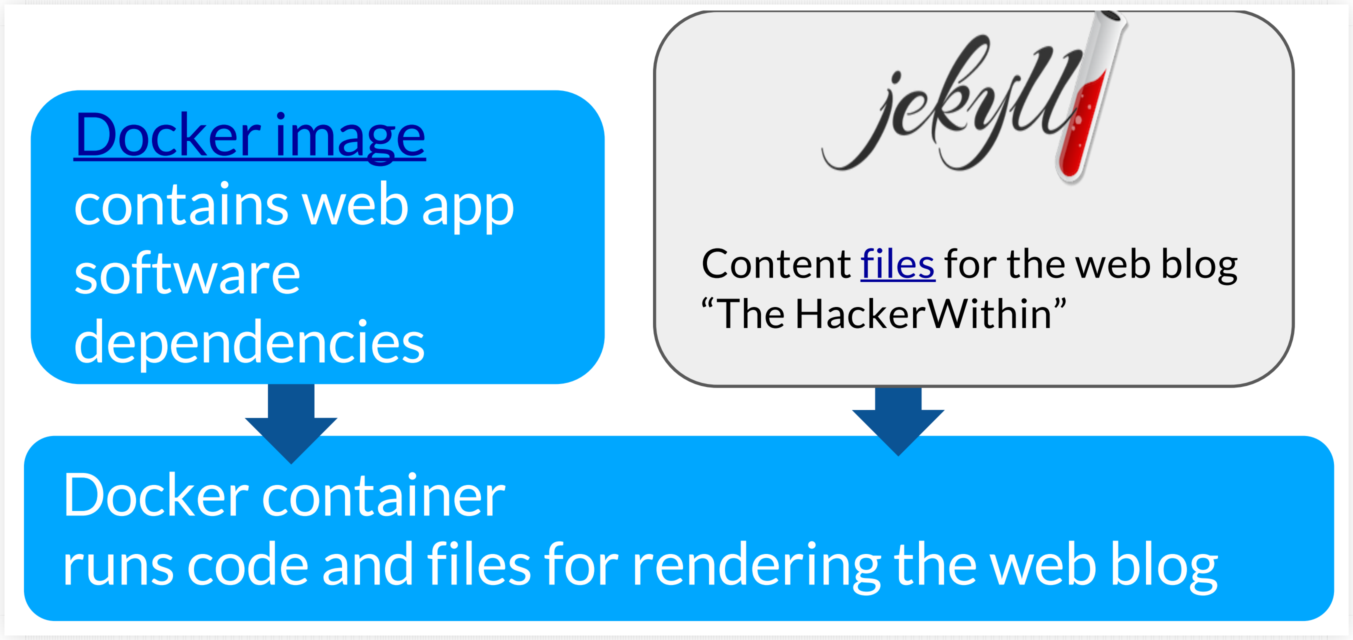
```
$ docker commit --help
$ docker commit CONTAINER_ID DOCKERHUB_REPO/IMA(
```

Ex2: Running the Jekyll web blog


Docker image
contains web app software dependencies
jekyll
Content files for the web blog "The HackerWithin"
Docker container
runs code and files for rendering the web blog

- keep `Jekyll` Hackerwithin website files (`html` / markdown) outside container

Ex2: Running the Jekyll web blog


Docker image
contains web app software dependencies
jekyll
Content files for the web blog "The HackerWithin"
Docker container
runs code and files for rendering the web blog

```
$ git clone \
https://github.com/thehackerwithin/davis
$ cd davis
```

Running the Jekyll web blog

```
$ docker run -it \
-p 4000:4000 \
-v $PATH_TO_HACKERWITHIN_DAVIS_DIR:/root \
karenyng/hackerwithin_dockerfile \
ruby -S jekyll serve \
--host=0.0.0.0 --watch --force_polling
```

Now it is up and running!

Check the IP address of your Docker machine

```
$ docker-machine ip
127.0.0.1         # for Linux
192.168.99.100    # for Mac and Windows
```

Now point your browser to `DOCKER_IP:4000`

What did we just do?

```
$ docker run -it \
-p 4000:4000 \
-v $PATH_TO_HACKERWITHIN_DAVIS_DIR:/root \
karenyng/hackerwithin_dockerfile \
ruby -S jekyll serve \
--host=0.0.0.0 --watch --force_polling
```

- `docker run` runs a certain image, in this case `karenyng/hackerwithin_dockerfile` which we have pulled previously

What did we just do?

```
$ docker run -it \
-p 4000:4000 \
-v $PATH_TO_HACKERWITHIN_DAVIS_DIR:/root \
karenyng/hackerwithin_dockerfile \
ruby -S jekyll serve \
--host=0.0.0.0 --watch --force_polling
```

- `-i` means run iteractively, asking the container to read from the host's `STDIN`
- `-t` asks container to bind to a pseudo-terminal

What did we just do?

```
$ docker run -it \
-p 4000:4000 \
-v $PATH_TO_HACKERWITHIN_DAVIS_DIR:/root \
karenyng/hackerwithin_dockerfile \
ruby -S jekyll serve \
--host=0.0.0.0 --watch --force_polling
```

- `-p HOST_PORT:CONTAINER_PORT` exposes the port

What did we just do?

```
$ docker run -it \
-p 4000:4000 \
-v $PATH_TO_HACKERWITHIN_DAVIS_DIR:/root \
karenyng/hackerwithin_dockerfile \
ruby -S jekyll serve \
--host=0.0.0.0 --watch --force_polling
```

`-v $PATH_TO_HACKERWITHIN_DAVIS_DIR:/root` tells Docker to mount the directory of the Hackerwithin Davis repo to `/root` in the container

What did we just do?

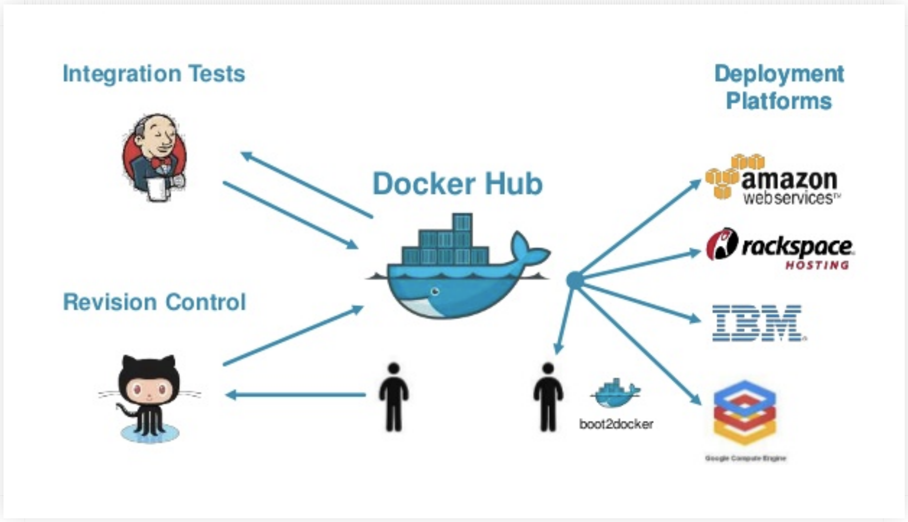
```
$ docker run -it \
-p 4000:4000 \
-v $PATH_TO_HACKERWITHIN_DAVIS_DIR:/root \
karenyng/hackerwithin_dockerfile \
ruby -S jekyll serve \
--host=0.0.0.0 --watch --force_polling
```

Last two lines tell `Jekyll` to keep on monitoring for changes.

Clearing up space

```
$ docker stop CONTAINER_ID
$ docker ps -a    # checks local containers
$ docker rm CONTAINER_ID # removes container
$ docker images   # checks local images
$ docker rmi IMAGE_NAME  # removes image
```

What is next? Use Docker for ...



- running SaaS for a startup
- running a continuous integration service
- running R Studio (NERSC HPC version) or Jupyter web client for remote machines

0

Learning resources

- Docker tutorial
- Docker online webminars
- San Francisco Docker meetups
- Docker best practises
- How to use Jekyll from Docker

Thanks for listening!

Linux resources

Guide for running Docker on Linux without `sudo` privileges.

Hands-on exercises for those who want to run docker im-ages

difficulty: easy

- run and modify the `HackerWithin`page
- run rOpenSci R studio web client
- run Jupyter Docker-stack for `Spark`/ `SparkR`/ `PySpark` / `Scipy` Jupyter notebook in Docker
- run `Tensorflow`

Building Dockerfiles / using Docker on the Cloud

difficulty: more involved

- write your own Dockerfile and build your own app
- use `Docker compose` to run the `Docker` 3rd birthday web app tutorial
- setup the `HortonWorks Data Platform` sandbox for playing with `Hadoop` and `Spark` in Docker
- challenge: write the Dockerfile(s) for Astrophysics codes, e.g.
    - Cosmosis,
    - Astrometry.net

Incomplete collection of useful commands

Restart stopped container

```
$ docker start CONTAINER_ID
$ docker attach CONTAINER_ID
```

Incomplete collection of useful commands

If your container is running in background and you want to use the command prompt inside the container

```
$ docker exec -it $(docker ps -q) bash
```