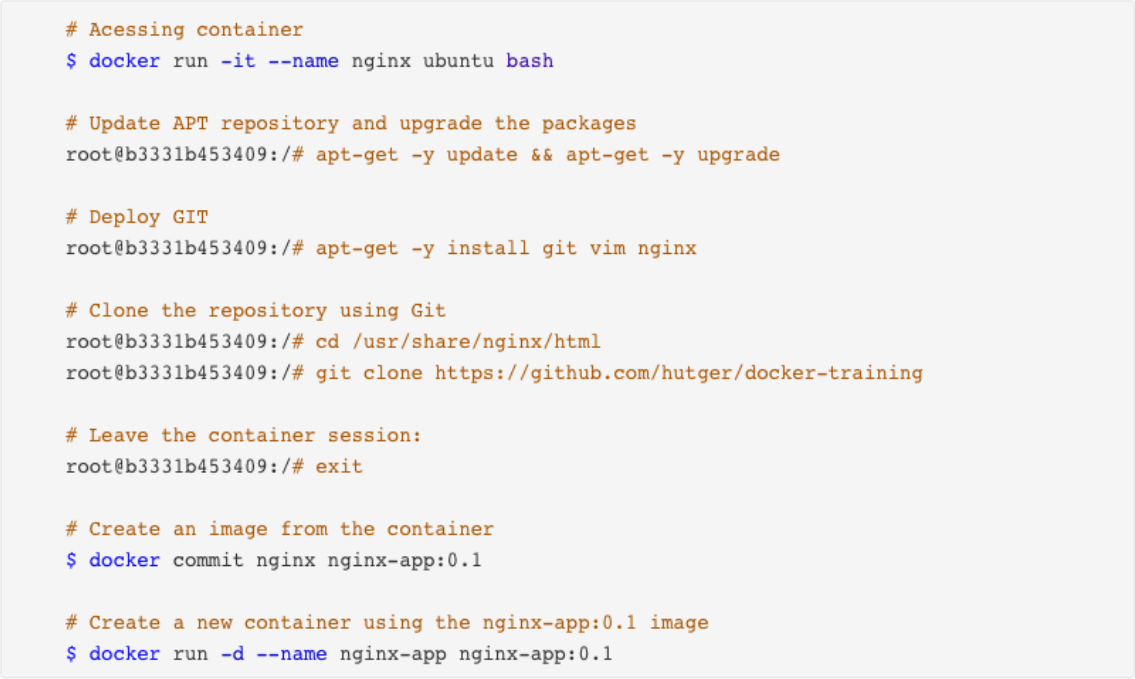
Docker for Beginners

Day 3 – Dockerfiles, Images and Registries

- Lucas Albuquerque -
lucas.albuquerque@nutanix.com

Dockerfiles

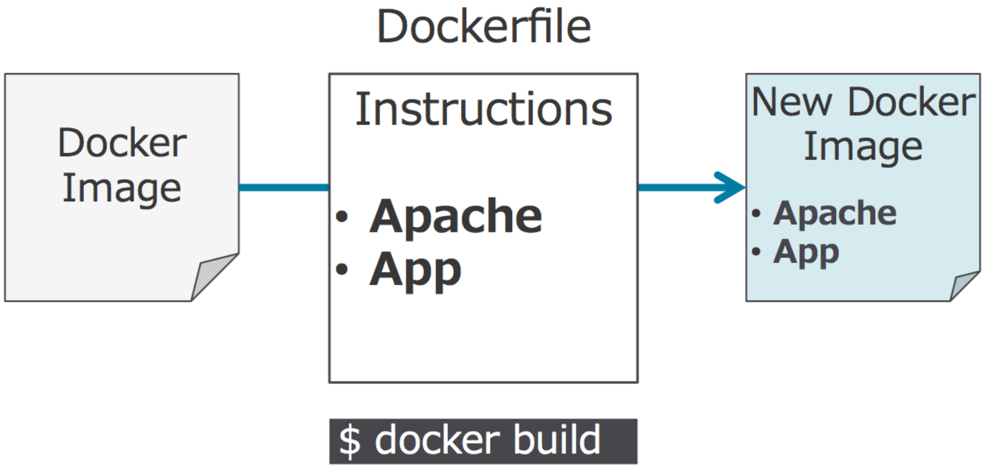
Let's suppose you want to deploy an Webserver using Ubuntu image



```
# Acessing container
$ docker run -it --name nginx ubuntu bash

# Update APT repository and upgrade the packages
root@b3331b453409:/# apt-get -y update && apt-get -y upgrade

# Deploy GIT
root@b3331b453409:/# apt-get -y install git vim nginx

# Clone the repository using Git
root@b3331b453409:/# cd /usr/share/nginx/html
root@b3331b453409:/# git clone https://github.com/hutger/docker-training

# Leave the container session:
root@b3331b453409:/# exit

# Create an image from the container
$ docker commit nginx nginx-app:0.1

# Create a new container using the nginx-app:0.1 image
$ docker run -d --name nginx-app nginx-app:0.1
```

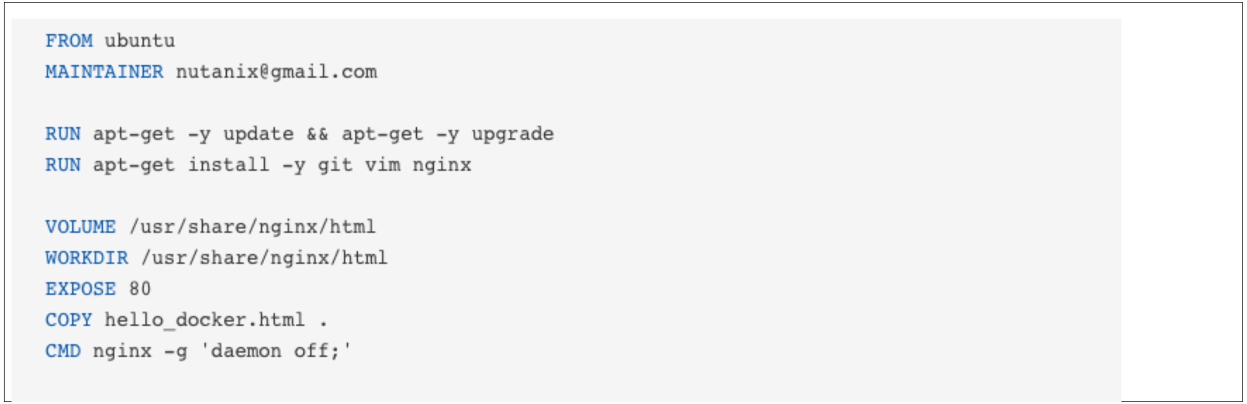
Dockerfiles

- A Dockerfile is a text document that contains all the commands a user could call on the command line to assemble an image
- Very useful for automating image builds

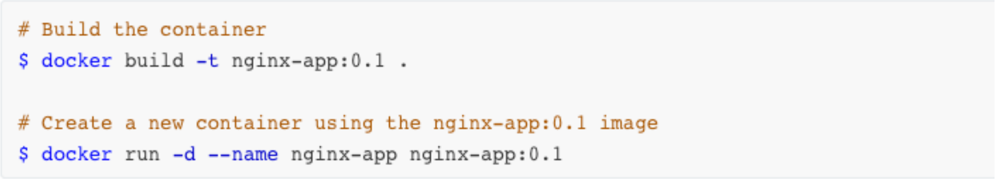


Dockerfiles

- Building an NGINX web server container



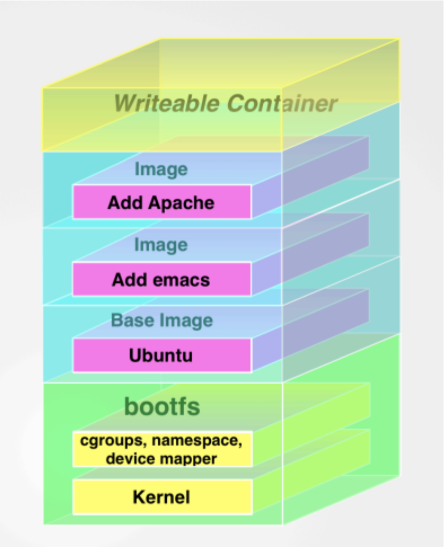
```
FROM ubuntu
MAINTAINER nutanix@gmail.com

RUN apt-get -y update && apt-get -y upgrade
RUN apt-get install -y git vim nginx

VOLUME /usr/share/nginx/html
WORKDIR /usr/share/nginx/html
EXPOSE 80
COPY hello_docker.html .
CMD nginx -g 'daemon off;'
```

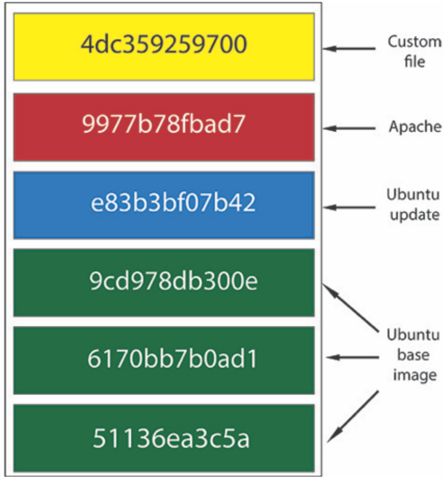


```
# Build the container
$ docker build -t nginx-app:0.1 .

# Create a new container using the nginx-app:0.1 image
$ docker run -d --name nginx-app nginx-app:0.1
```

Docker Images

- A Docker image is a file, comprised of multiple layers, used to execute code in a Docker container.
- An image is essentially built from instructions previously provided (e.g. Dockerfile)





Docker Images

- Docker images are managed using **docker** command:

  $ docker **build**— Build an image.

  $ docker **push**— Push an image to a remote registry.

  $ docker **pull**— Pull an image to a remote registry.

  $ docker **ls**— List images.

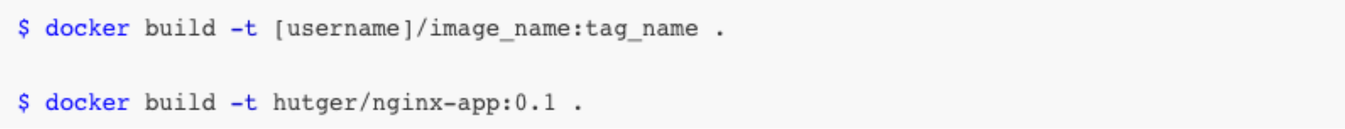
  $ docker **history**— See intermediate image info.

  $ docker **inspect**— See lots of info about an image, including the layers.

  $ docker **rm**— Delete an image.

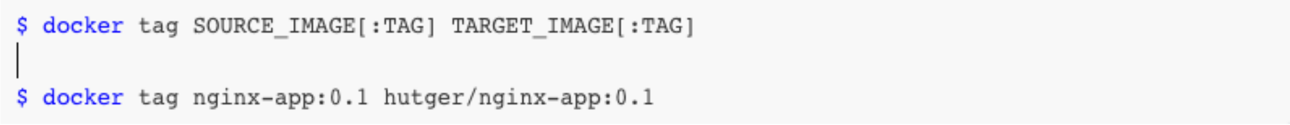
Docker Images

- Docker tags convey useful information about a specific **image** version/variant
- They are aliases to the ID of your **image** which often look like this: f1477ec11d12
- It's just a way of referring to your image



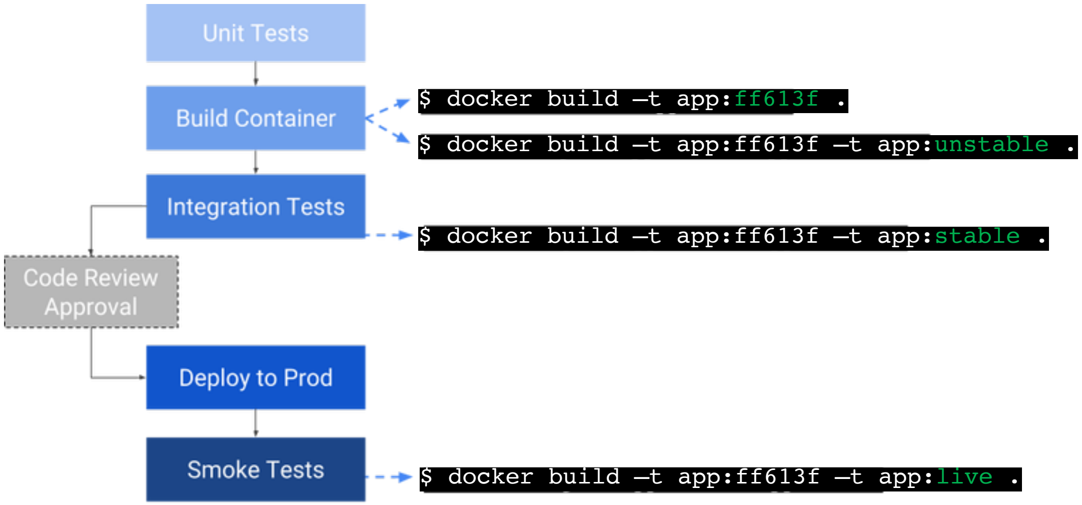
```
$ docker build -t [username]/image_name:tag_name .

$ docker build -t hutger/nginx-app:0.1 .
```

- Whenever an image is tagged without an explicit tag, it’s given the **latest** tag by default
- To re-tag an image



```
$ docker tag SOURCE_IMAGE[:TAG] TARGET_IMAGE[:TAG]

$ docker tag nginx-app:0.1 hutger/nginx-app:0.1
```

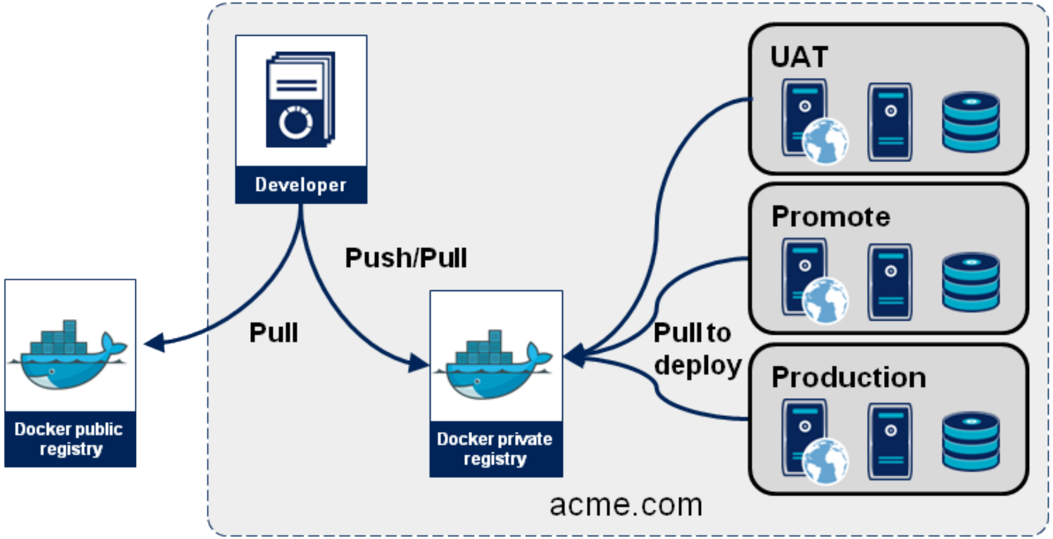
Docker Images



```
$ docker build –t app:ff613f .
$ docker build –t app:ff613f –t app:unstable .
$ docker build –t app:ff613f –t app:stable .
$ docker build –t app:ff613f –t app:live .
```

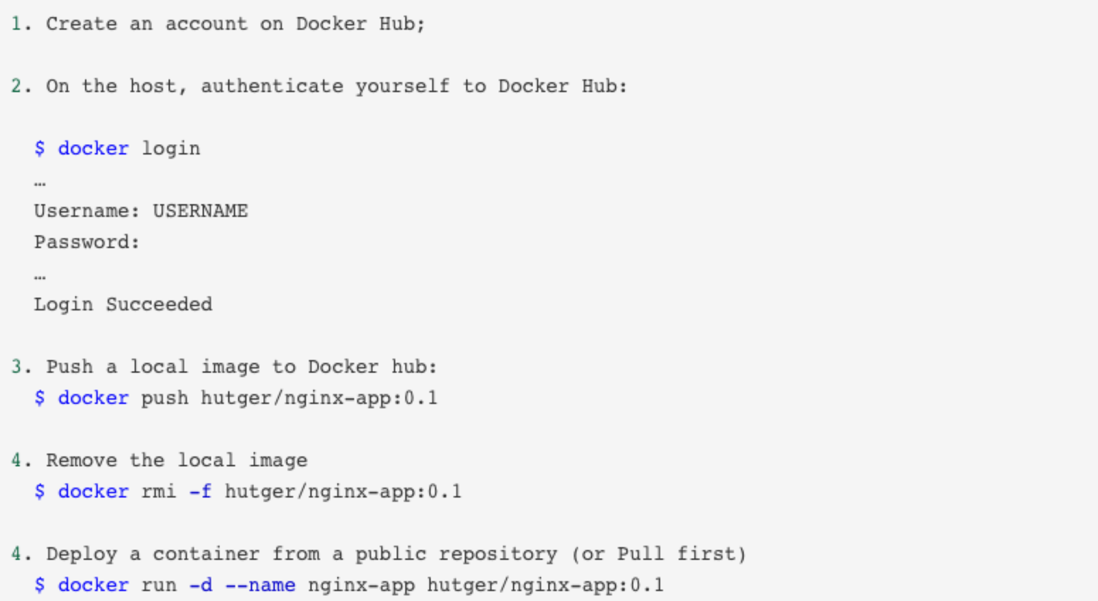
Docker Registry

- An application that manages, stores and delivery Docker container images.
- Registries centralize container images and the easiest way to share them;
- Docker default is Docker Hub (but you can deploy your own registry)



Docker Registry

- Publishing a new image on Docker Hub



```
1. Create an account on Docker Hub;

2. On the host, authenticate yourself to Docker Hub:

  $ docker login
  …
  Username: USERNAME
  Password:
  …
  Login Succeeded

3. Push a local image to Docker hub:
  $ docker push hutger/nginx-app:0.1

4. Remove the local image
  $ docker rmi -f hutger/nginx-app:0.1

4. Deploy a container from a public repository (or Pull first)
  $ docker run -d --name nginx-app hutger/nginx-app:0.1
```

Dockerfiles, Images and Registries

See you tomorrow…

- Lucas Albuquerque -
lucas.albuquerque@nutanix.com

NUTANIX®

@nutanix

Thank You