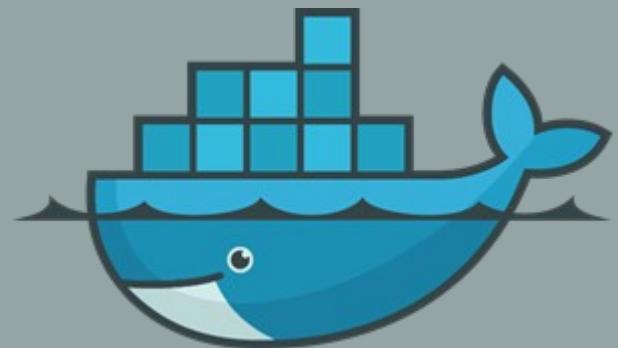


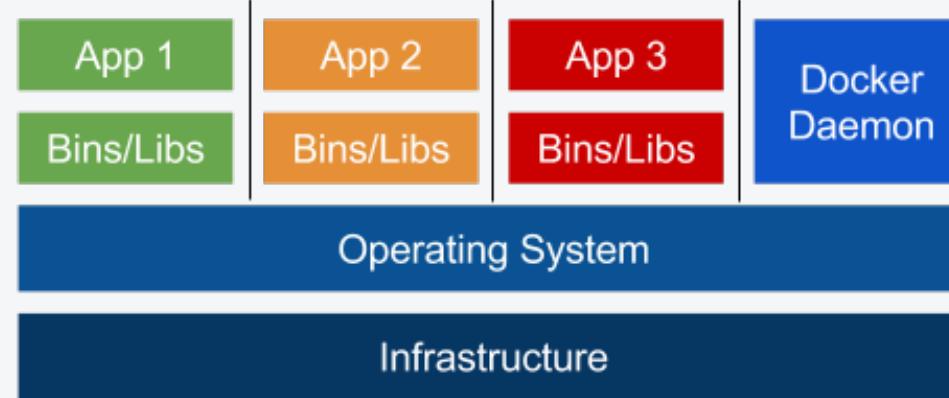
DOCKER: GET STARTED

BY VLAD GREGURCO

 Pentalog
Software Factory



WHAT IS DOCKER?



CONTAINER-BASED VIRTUALIZATION SCHEMA

WHAT WAS BEFORE?

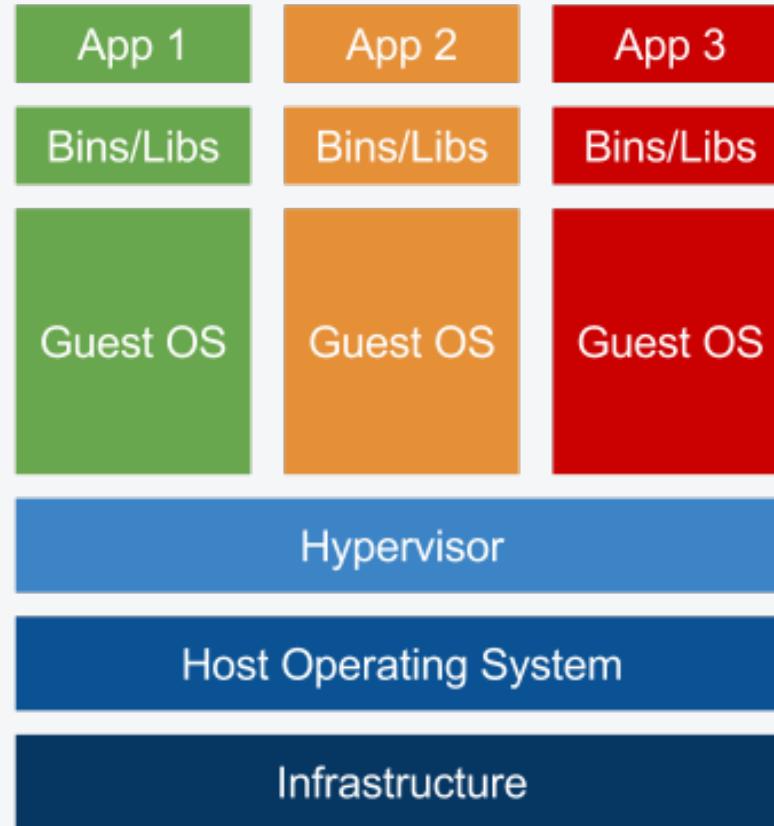


|| Parallels™

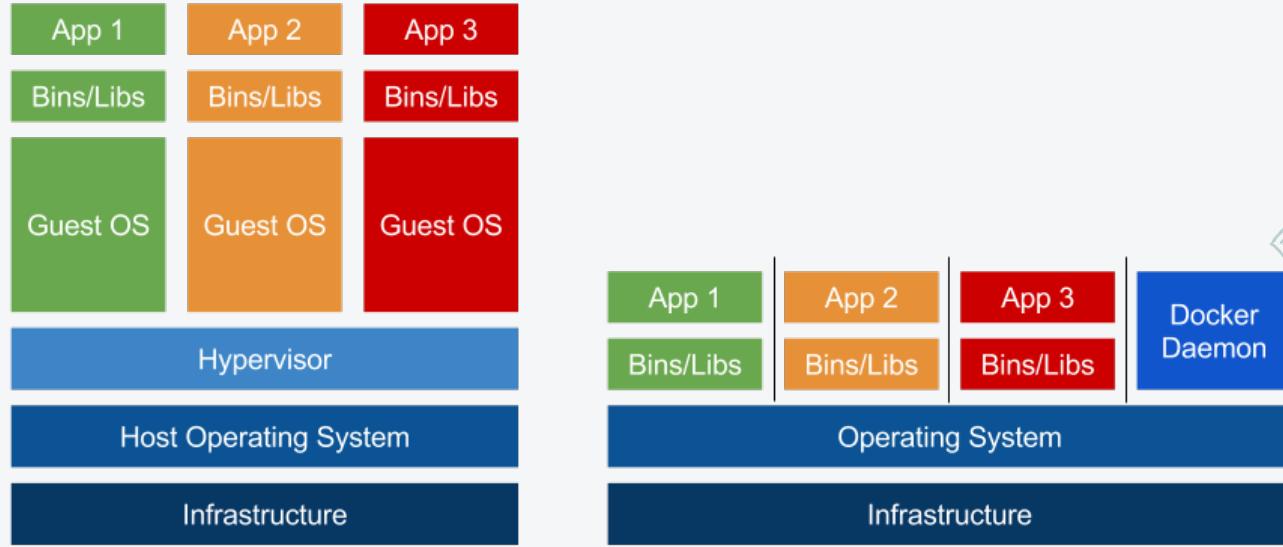


HYPERVERSOR-BASED VIRTUALIZATION

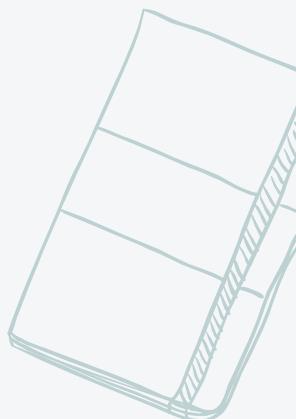




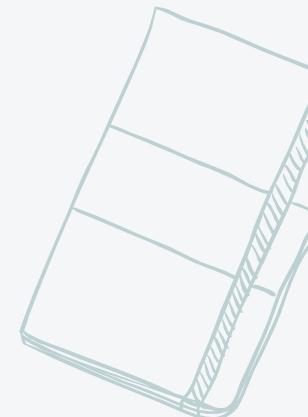
HYPERVERISOR-BASED VIRTUALIZATION SCHEMA



COMPARISON



YES



ADVANTAGES

- Simple setup
- Possibility to develop different application with different dependencies
- Prepared official/community images
- Common environment for devs
- Common setup for dev/test/prod
- *and many others...*

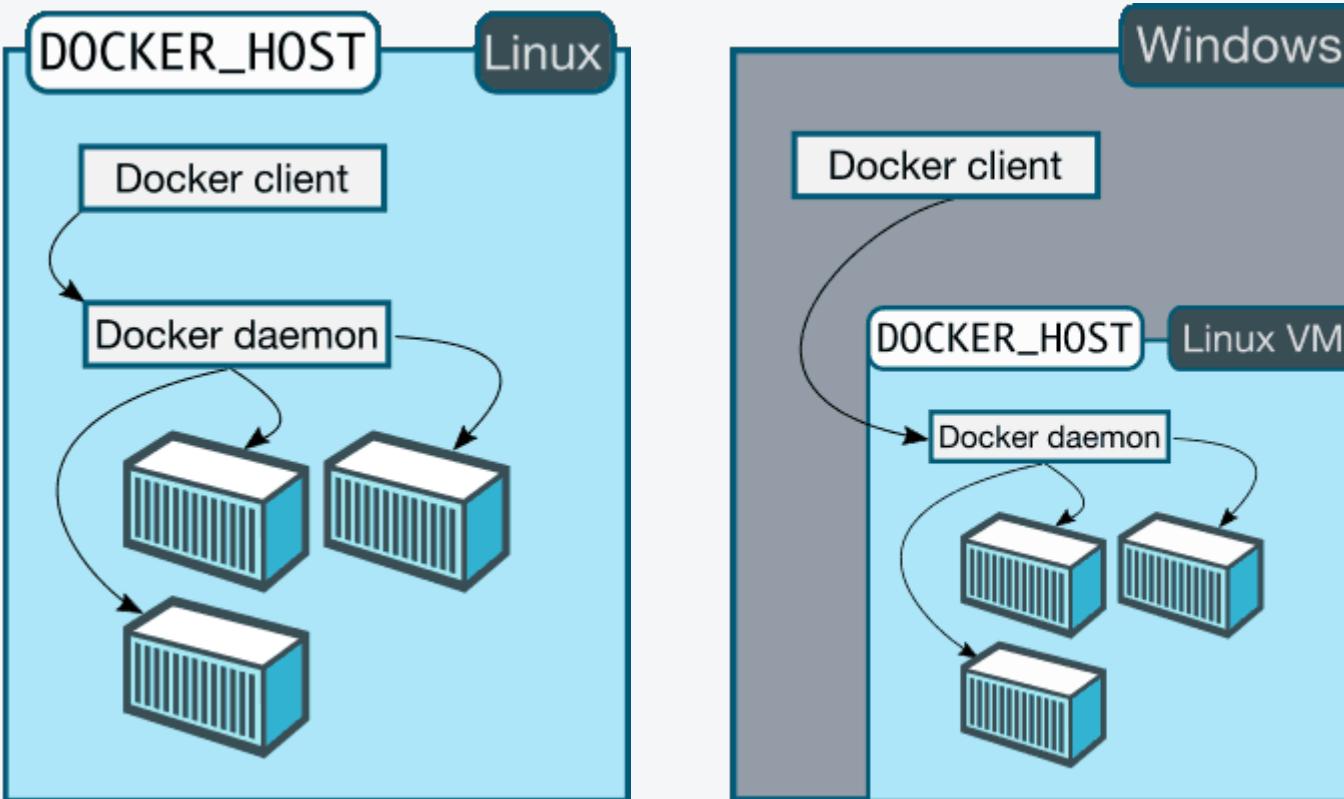
SETUP



SETUP

- Install docker
 - Ubuntu: <https://docs.docker.com/install/linux/docker-ce/ubuntu/>
 - Windows: <https://docs.docker.com/docker-for-windows/install/>
 - MacOS: <https://docs.docker.com/docker-for-mac/install/>
- Search for images:
 - <https://hub.docker.com/>

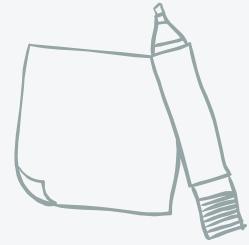
DOCKER ON LINUX AND WINDOWS



MAIN DOCKER CONCEPTS

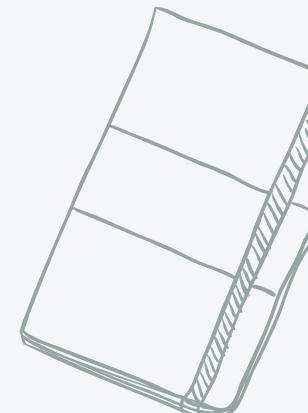
Images

- Images are read only templates used to create containers.
- Images are created with the *docker build* command.
- Images are composed of layers of other images.
- Images are stored in a Docker registry.



Containers

- If an image is a class, then a container is an instance of a class - a runtime object.
- Containers are created from images. Inside a container, it has all the binaries and dependencies needed to run the application.
- Containers are lightweight and portable encapsulations of an environment in which to run applications.



Registries

- A registry is where we store our *images*.
- You can host your own registry, or you can use Docker's public registry which is called DockerHub.

RUN FIRST CONTAINER

Linux commands

- echo "*message*" - display a line of text
- touch *fileName.txt* - create file
- ls - list directory contents
- exit - terminate process



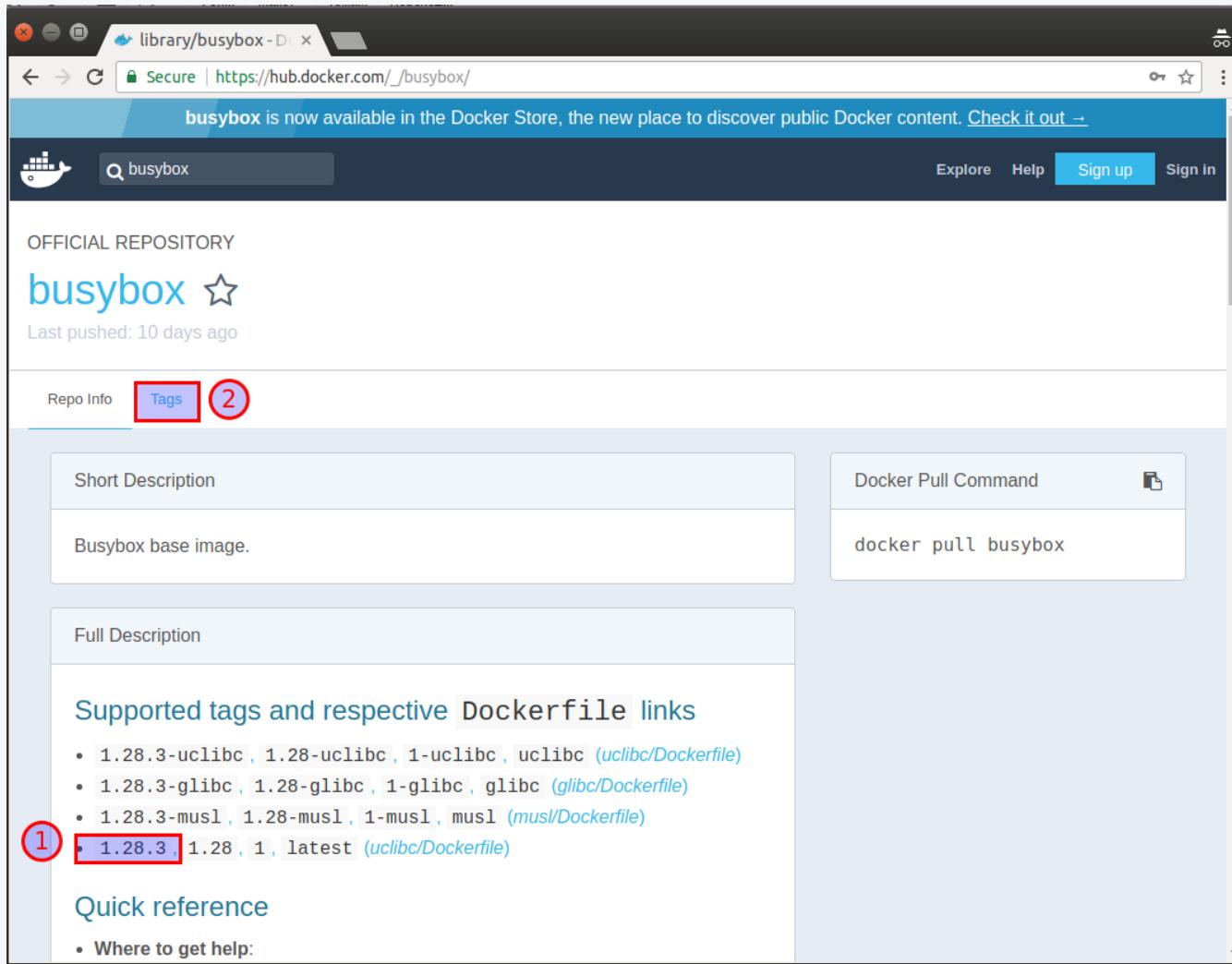
FIND BUSYBOX IMAGE

The screenshot shows a web browser window with the URL <https://hub.docker.com/search/?isAutomated=0&isOfficial=0&page=1&pullCount=0&q=busybox&starCount=0>. The search bar contains the query "busybox". The results page displays 1882 repositories. The first result, "busybox official", is highlighted with a red box and circled number 3. This repository has 1.2K stars and over 10M pulls. Other visible repositories include "arm64v8/busybox public", "s390x/busybox public", "radial/busyboxplus public | automated build", and "hypriot/rpi-busybox-httdp".

Repository	Type	Stars	Pulls	Actions
busybox official	official	1.2K	10M+	DETAILS
arm64v8/busybox	public	0	500K+	DETAILS
s390x/busybox	public	2	100K+	DETAILS
radial/busyboxplus	public automated build	19	500K+	DETAILS
hypriot/rpi-busybox-httdp		40	1M+	>



FIND LAST BUSYBOX VERSION



library/busybox - D x

Secure | https://hub.docker.com/_/busybox/

busybox is now available in the Docker Store, the new place to discover public Docker content. [Check it out →](#)

busybox 

Last pushed: 10 days ago

OFFICIAL REPOSITORY

Repo Info **Tags** ②

Short Description

Busybox base image.

Docker Pull Command

`docker pull busybox`

Full Description

Supported tags and respective Dockerfile links

- 1.28.3-uclibc, 1.28-uclibc, 1-uclibc, uclibc ([uclibc/Dockerfile](#))
- 1.28.3-glibc, 1.28-glibc, 1-glibc, glibc ([glibc/Dockerfile](#))
- 1.28.3-musl, 1.28-musl, 1-musl, musl ([musl/Dockerfile](#))
- 1.28.3, 1.28, 1, latest ([uclibc/Dockerfile](#))

Quick reference

- Where to get help:



RUN BUSYBOX CONTAINER



```
[node1] (local) root@192.168.0.23 ~
$ docker run busybox:1.28 echo "Hello World"
Unable to find image 'busybox:1.28' locally
1.28: Pulling from library/busybox
f70adabe43c0: Pull complete
Digest: sha256:58ac43b2cc92c687a32c8be6278e50a063579655fe3090125dcb2af0ff9e1a64
Status: Downloaded newer image for busybox:1.28
Hello World
```

Docker command parts:

- **run** – run a command in a new container
- **busybox** – image name
- **1.28** – image version
- **echo “hello World”** – command to execute in container

* To run commands it's possible to use <https://labs.play-with-docker.com/> , but better is to do it locally

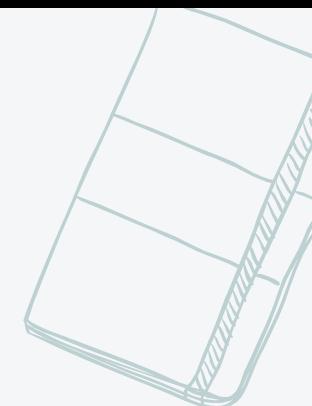




EXAMPLE #1



```
[node1] (local) root@192.168.0.18 ~  
$ docker run -it busybox:1.28█
```

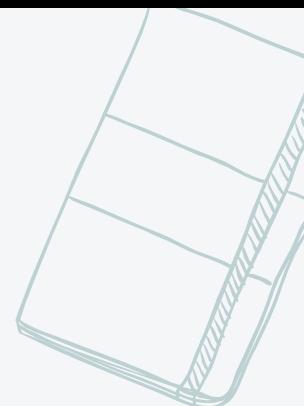




EXAMPLE #1



```
[node1] (local) root@192.168.0.18 ~
$ docker run -it busybox:1.28
Unable to find image 'busybox:1.28' locally
1.28: Pulling from library/busybox
f70adabe43c0: Pull complete
Digest: sha256:58ac43b2cc92c687a32c8be6278e50a063579655fe3090125dcb2af0ff9e1a64
Status: Downloaded newer image for busybox:1.28
/ # █
```

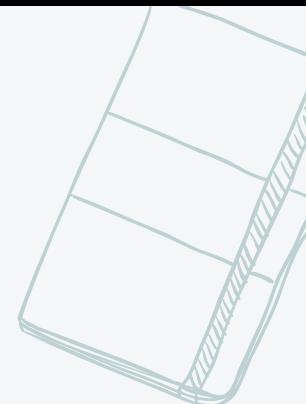




EXAMPLE #1



```
[node1] (local) root@192.168.0.18 ~  
$ docker run -it busybox:1.28  
Unable to find image 'busybox:1.28' locally  
1.28: Pulling from library/busybox  
f70adabe43c0: Pull complete  
Digest: sha256:58ac43b2cc92c687a32c8be6278e50a063579655fe3090125dcb2af0ff9e1a64  
Status: Downloaded newer image for busybox:1.28  
/ # touch file.txt  
/ # █
```



EXAMPLE #1

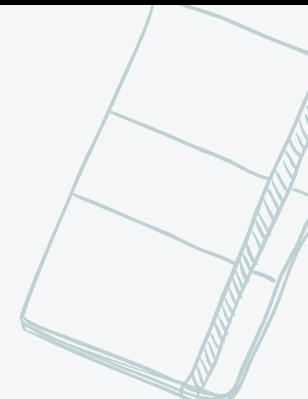
```
[node1] (local) root@192.168.0.18 ~
$ docker run -it busybox:1.28
Unable to find image 'busybox:1.28' locally
1.28: Pulling from library/busybox
f70adabe43c0: Pull complete
Digest: sha256:58ac43b2cc92c687a32c8be6278e50a063579655fe3090125dcb2af0ff9e1a64
Status: Downloaded newer image for busybox:1.28
/ # touch file.txt
/ # ls
bin      dev      etc      file.txt  home      proc      root      sys      tmp      usr      var
/ #
```



EXAMPLE #1



```
[node1] (local) root@192.168.0.18 ~
$ docker run -it busybox:1.28
Unable to find image 'busybox:1.28' locally
1.28: Pulling from library/busybox
f70adabe43c0: Pull complete
Digest: sha256:58ac43b2cc92c687a32c8be6278e50a063579655fe3090125dcb2af0ff9e1a64
Status: Downloaded newer image for busybox:1.28
/ # touch file.txt
/ # ls
bin      dev      etc      file.txt  home      proc      root      sys      tmp      usr      var
/ #
```

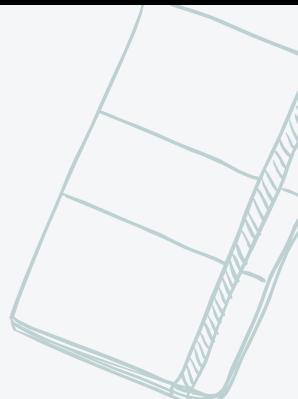




EXAMPLE #1



```
[node1] (local) root@192.168.0.18 ~
$ docker run -it busybox:1.28
Unable to find image 'busybox:1.28' locally
1.28: Pulling from library/busybox
f70adabe43c0: Pull complete
Digest: sha256:58ac43b2cc92c687a32c8be6278e50a063579655fe3090125dcb2af0ff9e1a64
Status: Downloaded newer image for busybox:1.28
/ # touch file.txt
/ # ls
bin      dev      etc      file.txt  home      proc      root      sys      tmp      usr      var
/ # exit
[node1] (local) root@192.168.0.18 ~
$ █
```

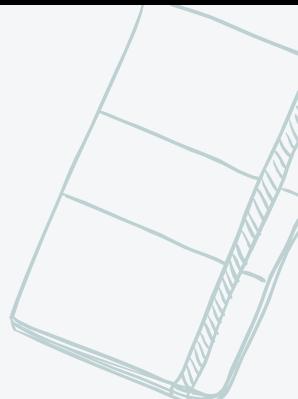




EXAMPLE #1



```
[node1] (local) root@192.168.0.18 ~
$ docker run -it busybox:1.28
Unable to find image 'busybox:1.28' locally
1.28: Pulling from library/busybox
f70adabe43c0: Pull complete
Digest: sha256:58ac43b2cc92c687a32c8be6278e50a063579655fe3090125dcb2af0ff9e1a64
Status: Downloaded newer image for busybox:1.28
/ # touch file.txt
/ # ls
bin      dev      etc      file.txt  home      proc      root      sys      tmp      usr      var
/ # exit
[node1] (local) root@192.168.0.18 ~
$ docker run -it busybox:1.28
/ # █
```

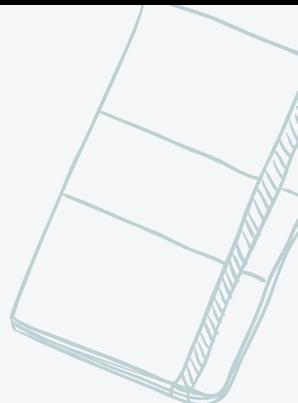




EXAMPLE #1



```
[node1] (local) root@192.168.0.18 ~
$ docker run -it busybox:1.28
Unable to find image 'busybox:1.28' locally
1.28: Pulling from library/busybox
f70adabe43c0: Pull complete
Digest: sha256:58ac43b2cc92c687a32c8be6278e50a063579655fe3090125dcb2af0ff9e1a64
Status: Downloaded newer image for busybox:1.28
/ # touch file.txt
/ # ls
bin      dev      etc      file.txt  home      proc      root      sys      tmp      usr      var
/ # exit
[node1] (local) root@192.168.0.18 ~
$ docker run -it busybox:1.28
/ # ls
bin  dev  etc  home  proc  root  sys  tmp  usr  var
/ # █
```





EXAMPLE #2

```
[node1] (local) root@192.168.0.18 ~  
$ php -v  
bash: php: command not found  
[node1] (local) root@192.168.0.18 ~  
$ █
```





EXAMPLE #2

```
[node1] (local) root@192.168.0.18 ~
$ php -v
bash: php: command not found
[node1] (local) root@192.168.0.18 ~
$ docker run \
>   -i -t \
>   -v "$PWD":/app \
>   -w /app \
>   php:7.2-alpine \
>   sh
```





EXAMPLE #2



```
[node1] (local) root@192.168.0.18 ~
$ php -v
bash: php: command not found
[node1] (local) root@192.168.0.18 ~
$ docker run \
>   -i -t \
>   -v "$PWD":/app \
>   -w /app \
>   php:7.2-alpine \
>   sh
```





EXAMPLE #2



```
[node1] (local) root@192.168.0.18 ~
$ php -v
bash: php: command not found
[node1] (local) root@192.168.0.18 ~
$ docker run \
>   -i -t \
>   -v "$PWD":/app \
>   -w /app \
>   php:7.2-alpine \
>   sh
```





EXAMPLE #2



```
[node1] (local) root@192.168.0.18 ~
$ php -v
bash: php: command not found
[node1] (local) root@192.168.0.18 ~
$ docker run \
>   -i -t \
>   -v "$PWD":/app \
>   -w /app \
>   php:7.2-alpine \
>   sh[]
```

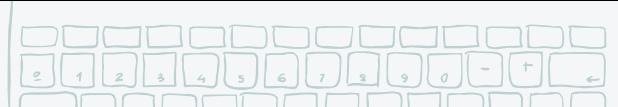




EXAMPLE #2



```
[node1] (local) root@192.168.0.18 ~
$ php -v
bash: php: command not found
[node1] (local) root@192.168.0.18 ~
$ docker run \
>   -i -t \
>   -v "$PWD":/app \
>   -w /app \
>   php:7.2-alpine \
>   sh
Unable to find image 'php:7.2-alpine' locally
7.2-alpine: Pulling from library/php
4fe2ade4980c: Pull complete
5cf60c0c0763: Pull complete
468219aa4689: Pull complete
4a8b1a3b9e51: Pull complete
f94222471f6e: Pull complete
590978259969: Pull complete
c33f0bce7578: Pull complete
e7cdec58376b: Pull complete
48a3db79f8a5: Pull complete
Digest: sha256:cdff4e3f15bd72acabeaa4e9729c48f7e98e1a8c73706ce37955cb4bb16b55b3
Status: Downloaded newer image for php:7.2-alpine
/app # █
```

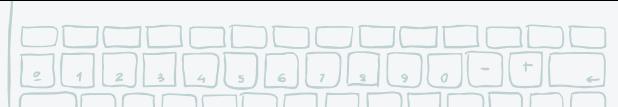




EXAMPLE #2



```
[node1] (local) root@192.168.0.18 ~
$ php -v
bash: php: command not found
[node1] (local) root@192.168.0.18 ~
$ docker run \
>   -i -t \
>   -v "$PWD":/app \
>   -w /app \
>   php:7.2-alpine \
>   sh
Unable to find image 'php:7.2-alpine' locally
7.2-alpine: Pulling from library/php
4fe2ade4980c: Pull complete
5cf60c0c0763: Pull complete
468219aa4689: Pull complete
4a8b1a3b9e51: Pull complete
f94222471f6e: Pull complete
590978259969: Pull complete
c33f0bce7578: Pull complete
e7cdec58376b: Pull complete
48a3db79f8a5: Pull complete
Digest: sha256:cdff4e3f15bd72acabeaa4e9729c48f7e98e1a8c73706ce37955cb4bb16b55b3
Status: Downloaded newer image for php:7.2-alpine
/app # █
```

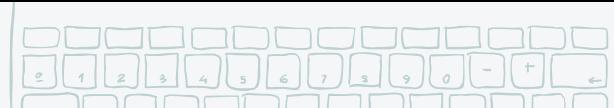




EXAMPLE #2



```
[node1] (local) root@192.168.0.18 ~
$ php -v
bash: php: command not found
[node1] (local) root@192.168.0.18 ~
$ docker run \
>   -i -t \
>   -v "$PWD":/app \
>   -w /app \
>   php:7.2-alpine \
>   sh
Unable to find image 'php:7.2-alpine' locally
7.2-alpine: Pulling from library/php
4fe2ade4980c: Pull complete
5cf60c0c0763: Pull complete
468219aa4689: Pull complete
4a8b1a3b9e51: Pull complete
f94222471f6e: Pull complete
590978259969: Pull complete
c33f0bce7578: Pull complete
e7cdec58376b: Pull complete
48a3db79f8a5: Pull complete
Digest: sha256:cdff4e3f15bd72acabeaa4e9729c48f7e98e1a8c73706ce37955cb4bb16b55b3
Status: Downloaded newer image for php:7.2-alpine
/app # php -v
PHP 7.2.11 (cli) (built: Oct 15 2018 18:50:17) ( NTS )
Copyright (c) 1997-2018 The PHP Group
Zend Engine v3.2.0, Copyright (c) 1998-2018 Zend Technologies
/app # █
```

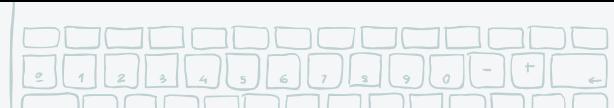




EXAMPLE #2



```
[node1] (local) root@192.168.0.18 ~
$ php -v
bash: php: command not found
[node1] (local) root@192.168.0.18 ~
$ docker run \
>   -i -t \
>   -v "$PWD":/app \
>   -w /app \
>   php:7.2-alpine \
>   sh
Unable to find image 'php:7.2-alpine' locally
7.2-alpine: Pulling from library/php
4fe2ade4980c: Pull complete
5cf60c0c0763: Pull complete
468219aa4689: Pull complete
4a8b1a3b9e51: Pull complete
f94222471f6e: Pull complete
590978259969: Pull complete
c33f0bce7578: Pull complete
e7cdec58376b: Pull complete
48a3db79f8a5: Pull complete
Digest: sha256:cdff4e3f15bd72acabeaa4e9729c48f7e98e1a8c73706ce37955cb4bb16b55b3
Status: Downloaded newer image for php:7.2-alpine
/app # php -v
PHP 7.2.11 (cli) (built: Oct 15 2018 18:50:17) ( NTS )
Copyright (c) 1997-2018 The PHP Group
Zend Engine v3.2.0, Copyright (c) 1998-2018 Zend Technologies
/app # █
```

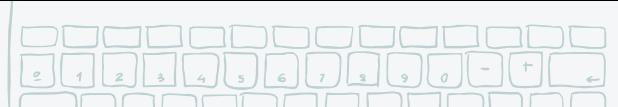




EXAMPLE #3



```
[node1] (local) root@192.168.0.18 ~
$ docker run \
>   -i -t \
>   -v "$PWD":/app \
>   -w /app \
>   php:5.6-alpine \
>   sh
Unable to find image 'php:5.6-alpine' locally
5.6-alpine: Pulling from library/php
4fe2ade4980c: Already exists
5cf60c0c0763: Already exists
468219aaa4689: Already exists
4a8b1a3b9e51: Already exists
fd34efaf813a: Pull complete
e77b5f41ad49: Pull complete
d8d95077bdf8: Pull complete
1189c427b808: Pull complete
Digest: sha256:fab0cac6c3106b105676be4cb1c0c01cd0af55ffb781b683a9e3242ed3e08def
Status: Downloaded newer image for php:5.6-alpine
/app # php -v
PHP 5.6.38 (cli) (built: Sep 15 2018 05:59:50)
Copyright (c) 1997-2016 The PHP Group
Zend Engine v2.6.0, Copyright (c) 1998-2016 Zend Technologies
/app # █
```





EXAMPLE #3



```
[node1] (local) root@192.168.0.18 ~
$ docker run \
>   -i -t \
>   -v "$PWD":/app \
>   -w /app \
>   php:5.6-alpine \
>   sh
Unable to find image 'php:5.6-alpine' locally
5.6-alpine: Pulling from library/php
4fe2ade4980c: Already exists
5cf60c0c0763: Already exists
468219aaa4689: Already exists
4a8b1a3b9e51: Already exists
fd34efaf813a: Pull complete
e77b5f41ad49: Pull complete
d8d95077bdf8: Pull complete
1189c427b808: Pull complete
Digest: sha256:fab0cac6c3106b105676be4cb1c0c01cd0af55ffb781b683a9e3242ed3e08def
Status: Downloaded newer image for php:5.6-alpine
/app # php -v
PHP 5.6.38 (cli) (built: Sep 15 2018 05:59:50)
Copyright (c) 1997-2016 The PHP Group
Zend Engine v2.6.0, Copyright (c) 1998-2016 Zend Technologies
/app # █
```





EXAMPLE #4

```
[node1] (local) root@192.168.0.43 ~  
$ docker run \  
> -i -t \  
> -v "$PWD":/app \  
> -w /app \  
> node:10.12 \  
> sh
```





EXAMPLE #4



```
[node1] (local) root@192.168.0.43 ~
$ docker run \
>   -i -t \
>   -v "$PWD":/app \
>   -w /app \
>   node:10.12 \
>   sh
Unable to find image 'node:10.12' locally
10.12: Pulling from library/node
61be48634cb9: Pull complete
fa696905a590: Pull complete
b6dd2322bbef: Pull complete
32477089adb4: Pull complete
febe7209ec28: Pull complete
4364cbe57162: Pull complete
437859acfd49: Pull complete
d8268e1e433b: Pull complete
Digest: sha256:00a7fb3df8e94ed24f42c2920f132f06e92ea5ed69b1c5e53c4bb3d20e85a3e2
Status: Downloaded newer image for node:10.12
# █
```



EXAMPLE #4



```
[node1] (local) root@192.168.0.43 ~
$ docker run \
>   -i -t \
>   -v "$PWD":/app \
>   -w /app \
>   node:10.12 \
>   sh
Unable to find image 'node:10.12' locally
10.12: Pulling from library/node
61be48634cb9: Pull complete
fa696905a590: Pull complete
b6dd2322bbef: Pull complete
32477089adb4: Pull complete
febe7209ec28: Pull complete
4364cbe57162: Pull complete
437859acfd49: Pull complete
d8268e1e433b: Pull complete
Digest: sha256:00a7fb3df8e94ed24f42c2920f132f06e92ea5ed69b1c5e53c4bb3d20e85a3e2
Status: Downloaded newer image for node:10.12
# node -v
v10.12.0
# npm -v
6.4.1
# █
```

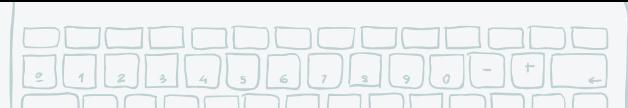




EXAMPLE #4

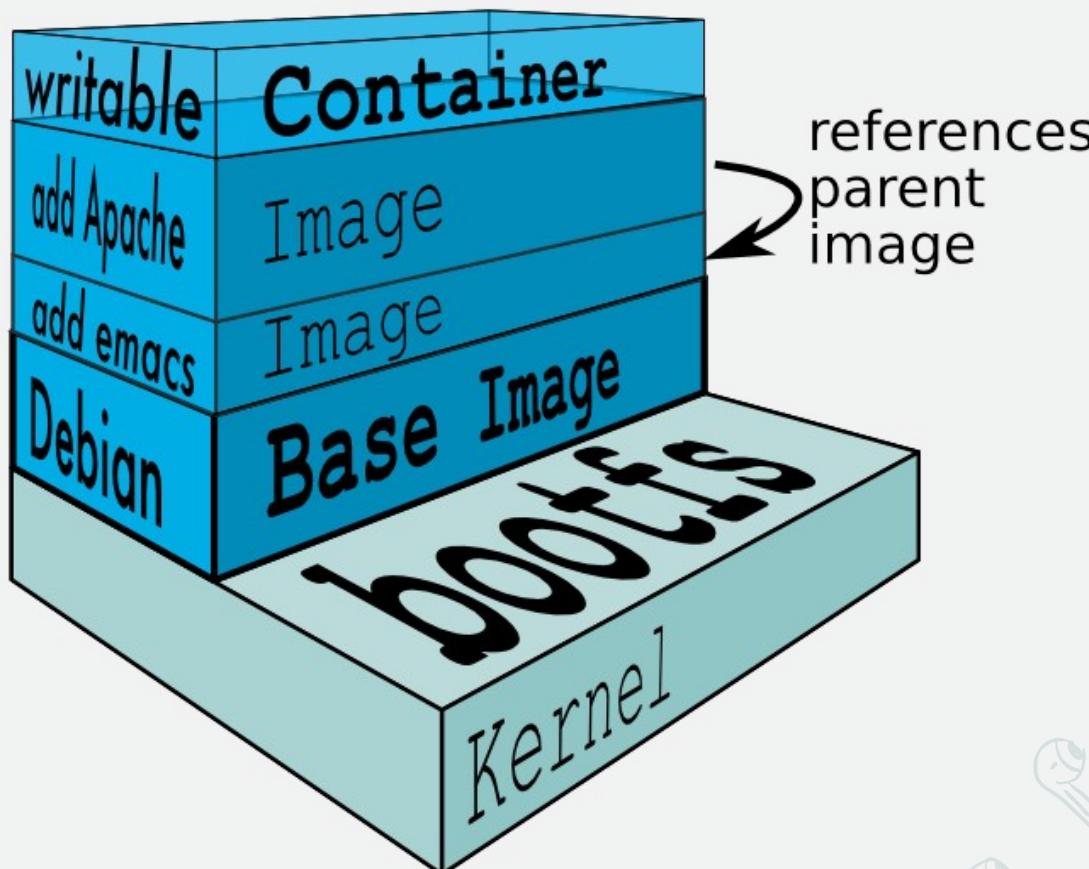


```
[node1] (local) root@192.168.0.43 ~
$ docker run \
>   -i -t \
>   -v "$PWD":/app \
>   -w /app \
>   node:10.12 \
>   sh
Unable to find image 'node:10.12' locally
10.12: Pulling from library/node
61be48634cb9: Pull complete
fa696905a590: Pull complete
b6dd2322bbef: Pull complete
32477089adb4: Pull complete
febe7209ec28: Pull complete
4364cbe57162: Pull complete
437859acfd49: Pull complete
d8268e1e433b: Pull complete
Digest: sha256:00a7fb3df8e94ed24f42c2920f132f06e92ea5ed69b1c5e53c4bb3d20e85a3e2
Status: Downloaded newer image for node:10.12
# node -v
v10.12.0
# npm -v
6.4.1
#
```

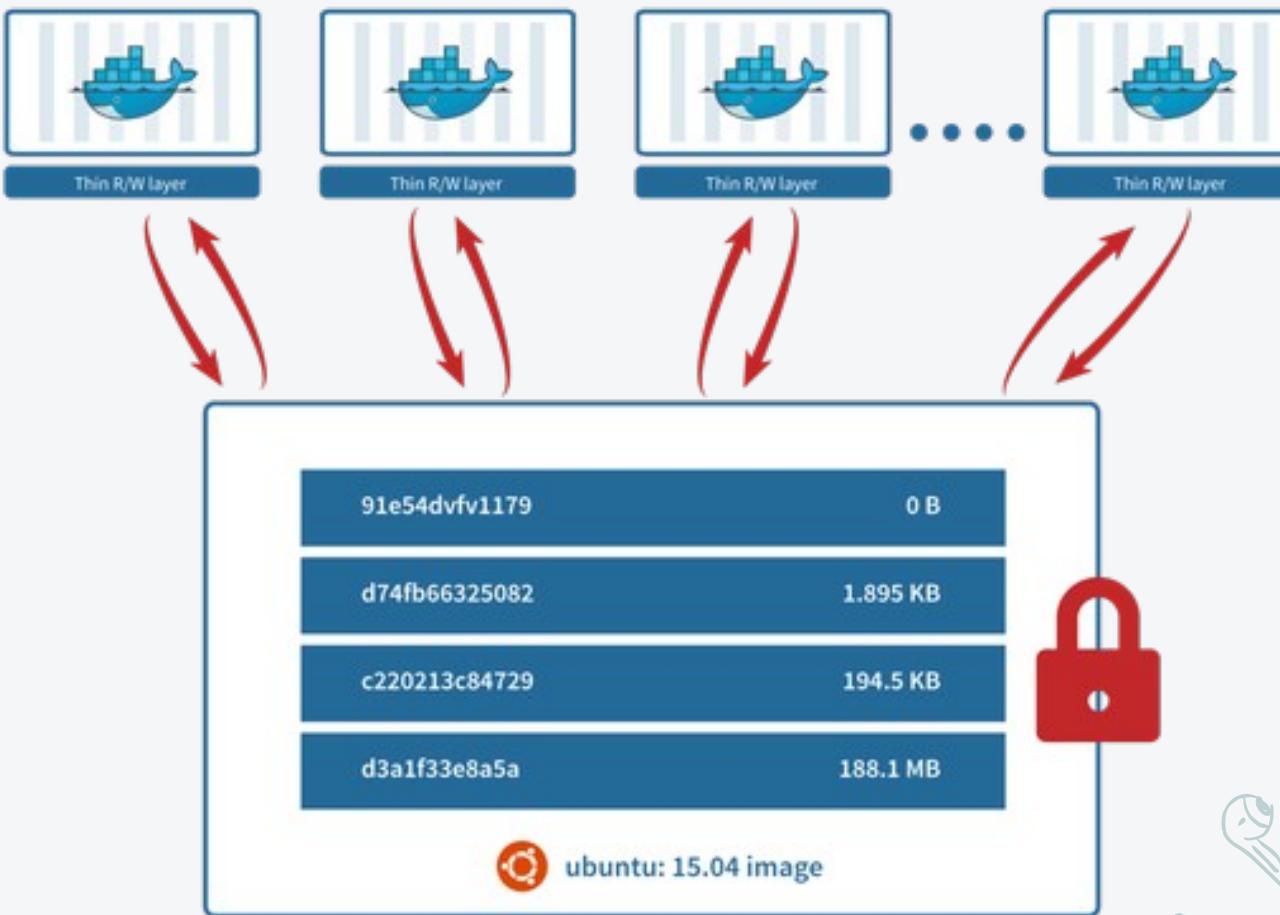


DOCKER IMAGES

DOCKER IMAGE LAYERS



DOCKER IMAGE LAYERS



BUILD DOCKER IMAGE (COMMIT COMMAND)

- Run: `docker run -it debian:jessie`
- Check that git is not installed
- Run: `apt-get update && apt-get install git`
- Check that git is installed and exit from container
- Run:
 - `docker ps -a` and find there the container id of debian
 - `docker commit container_id your_login/debian:1.0`
 - `docker images`
- Check that new image appeared
- Run: `docker history your_login/debian:1.0`

Note: Now you can push this image to registry or run containers based on this image.

BUILD DOCKER IMAGE (DOCKERFILE)

- Create new file with name *Dockerfile* and next content:

```
FROM debian:jessie  
  
RUN apt-get update && apt-get install -y git
```

- Run: *docker build -t your_login/debian_from_dockerfile .*
- Run: *docker images*
- Check that image is created
- Try to do the same build but split **run** in two commands. See the history of the image.

DOCKER COMPOSE

DOCKER COMPOSE

- Docker compose is a very handy tool to quickly get docker environment up and running.
- Docker compose uses yaml files to store the configuration of all the containers, which removes the burden to maintain our scripts for docker orchestration.

SETUP

- Install docker-compose:

<https://docs.docker.com/compose/install/>

DOCKER COMPOSE CONFIG EXAMPLE

```
1  version: "3.1"
2  services:
3    redis:
4      image: redis:alpine
5      container_name: project-redis
6
7    mysql:
8      image: mysql:5.7
9      container_name: project-mysql
10     working_dir: /application
11     volumes:
12       - ./application
13     environment:
14       - MYSQL_ROOT_PASSWORD=password
15       - MYSQL_DATABASE=project_db
16       - MYSQL_USER=user
17       - MYSQL_PASSWORD=password
18     ports:
19       - '3306:3306'
20
21    webserver:
22      image: nginx:alpine
23      container_name: project-webserver
24      working_dir: /application
25      volumes:
26        - ./application
27        - ./phpdocker/nginx/nginx.conf:/etc/nginx/conf.d/default.conf
28      ports:
29        - "8080:80"
30
31    php-fpm:
32      build: phpdocker/php-fpm
33      container_name: project-php-fpm
34      working_dir: /application
35      volumes:
36        - ./application
```

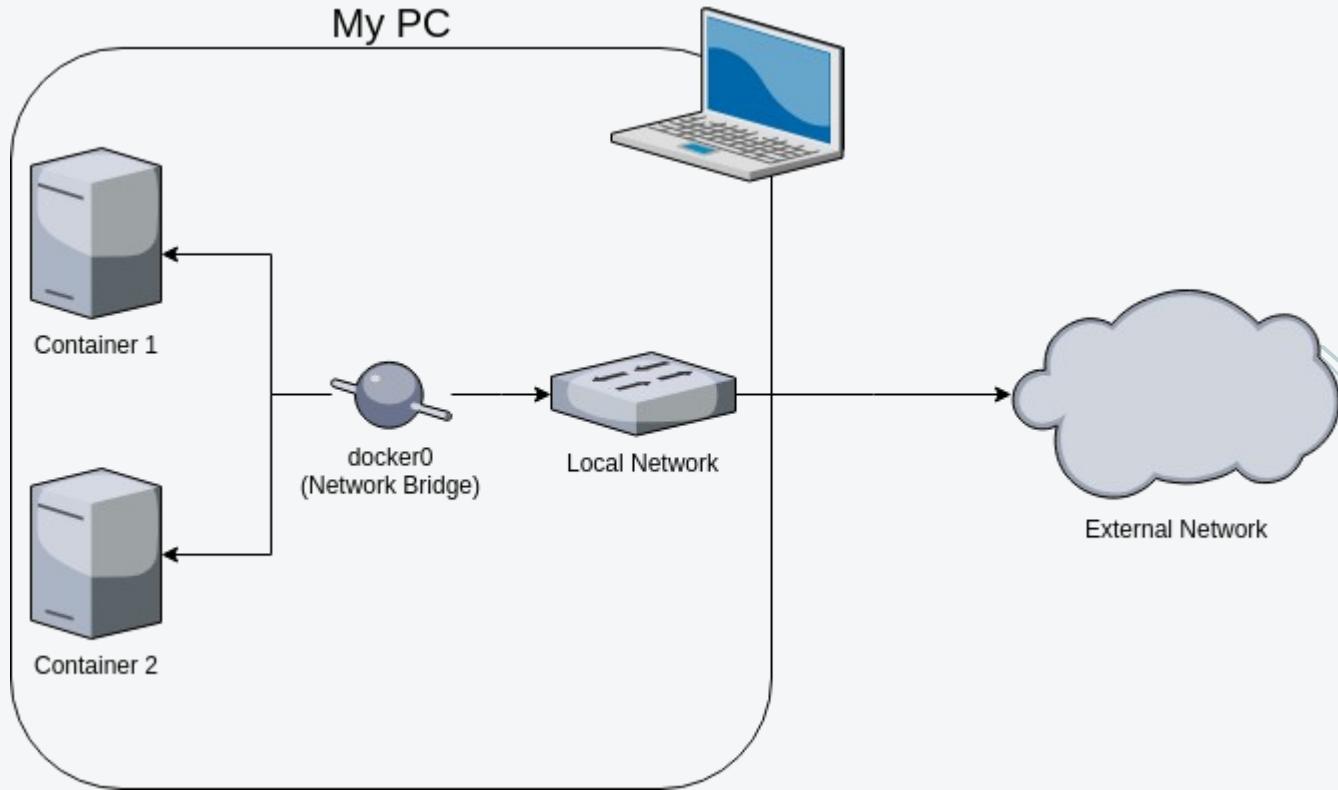
DOCKER COMPOSE COMMANDS

- **docker-compose up** – starts up all the containers.
- **docker-compose ps** – checks the status of the containers managed by docker compose.
- **docker-compose logs** – outputs colored and aggregated logs for the compose-managed containers.
- **docker-compose logs -f** – outputs appended log when the log grows.
- **docker-compose logs container_name** – outputs the logs of a specific container.
- **docker-compose stop** – stops all the running containers without removing them.
- **docker-compose rm** – removes all the containers.
- **docker-compose build** – rebuilds all the images.

DOCKER NETWORKING



DOCKER NETWORK SCHEMA



DOCKER NETWORK TYPES

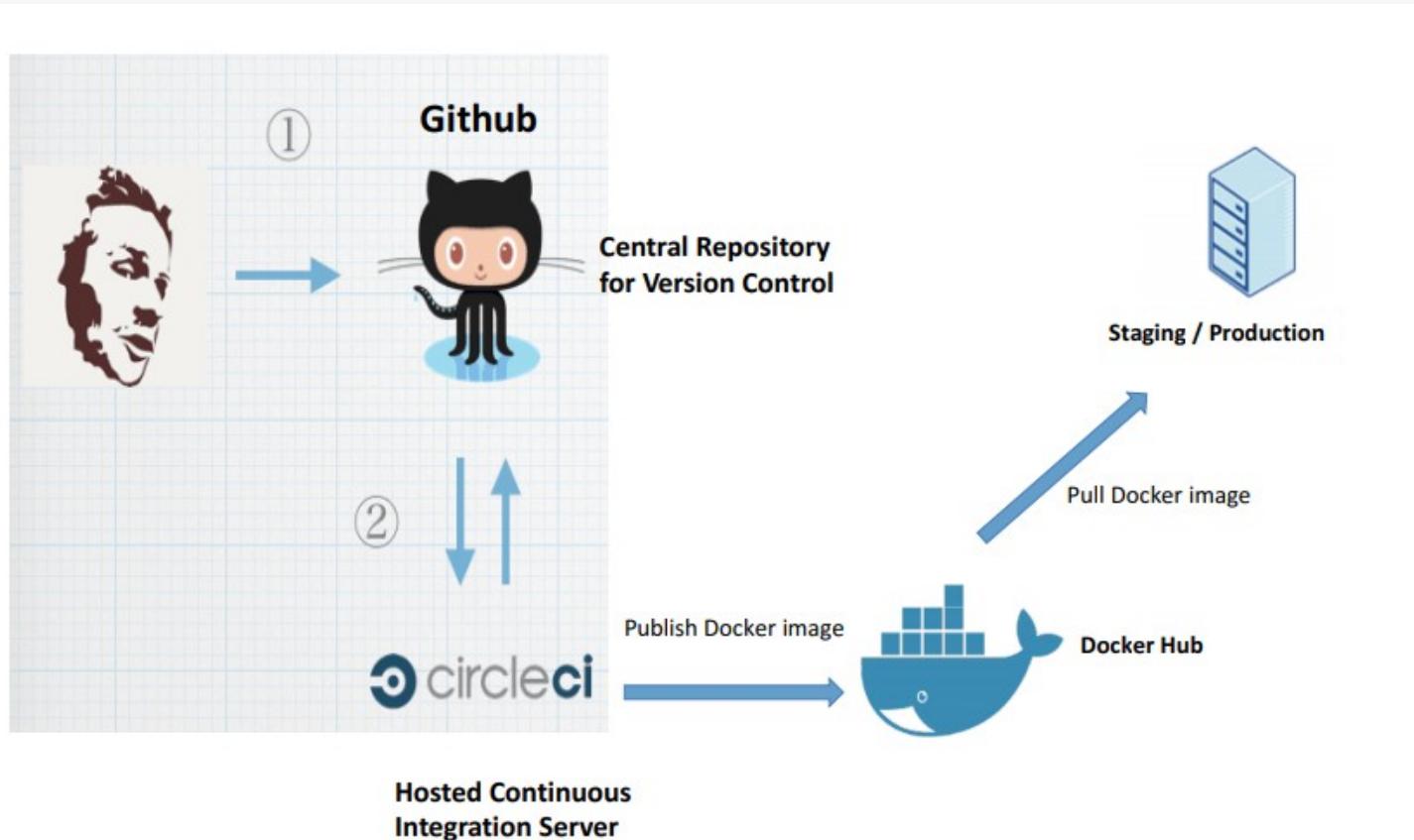
- **Closed Network / None Network** - disable all networking.
- **Bridge Network** - the default network driver which allows containers connected to the same bridge network to communicate.
- **Host Network** - adds a container on the host's network stack.

COMPLETE CI WORKFLOW





COMPLETE CI WORKFLOW



FINAL TIPS AND TRICKS

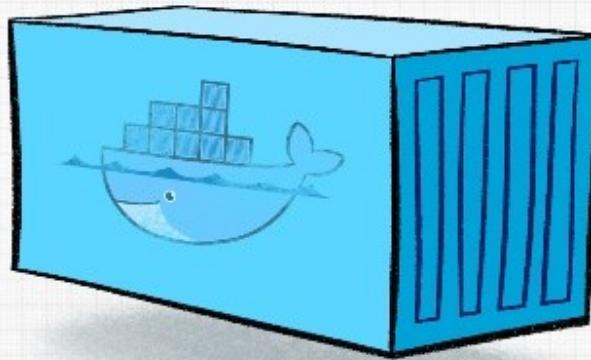
- Use Docker
- Use Docker Compose in case running multi-containers
- Whenever possible, use current Official Repositories as the basis for your image
- Use minimal base image as it's possible
- Be always up to date with last changes

MORE INFORMATION

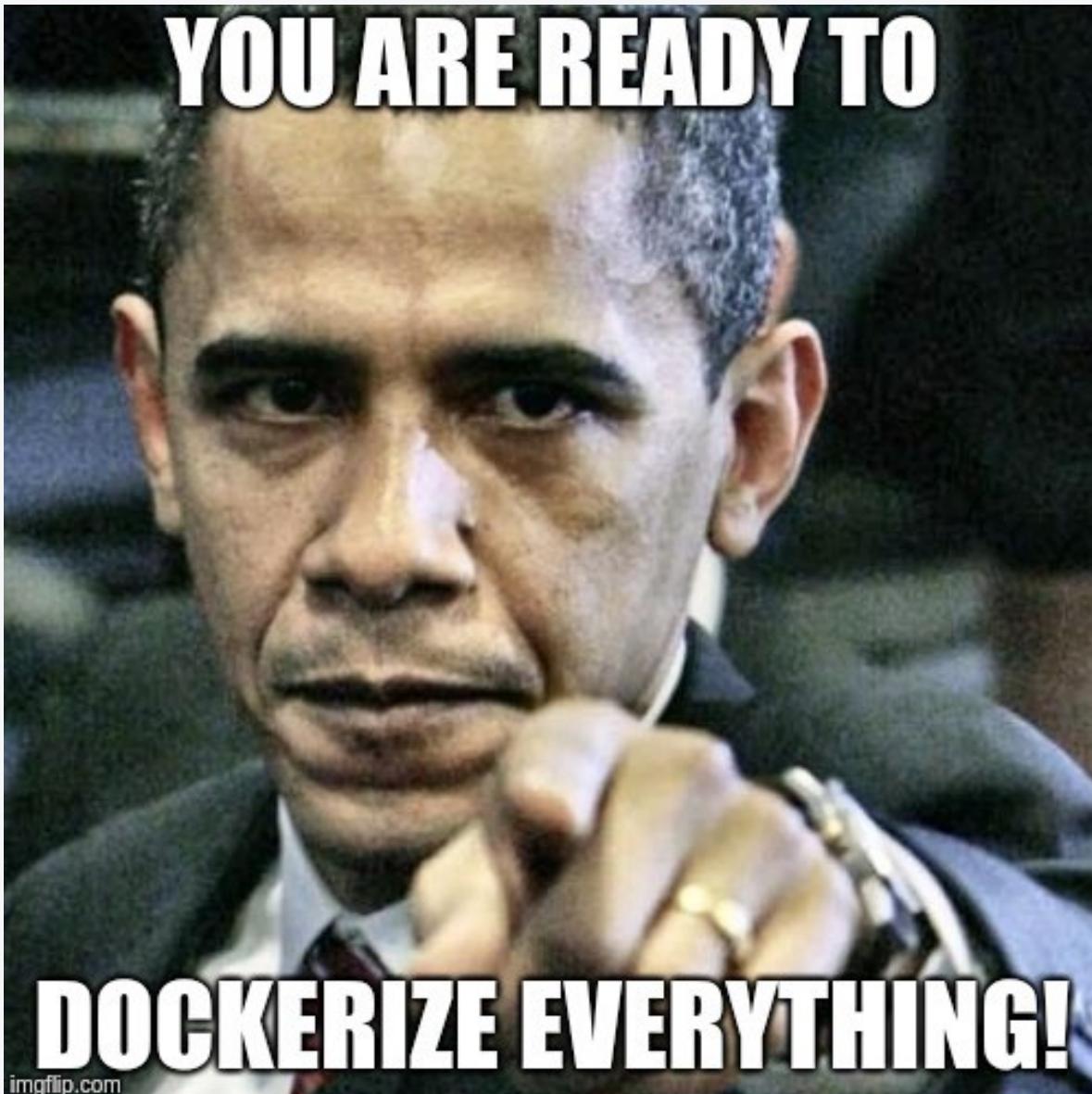
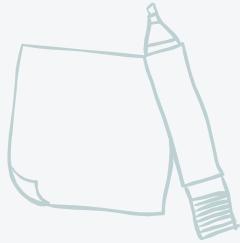
- Docker Documentation: <https://docs.docker.com>
- Docker online sandbox: <https://labs.play-with-docker.com>
- DockerCon 2017:
https://www.youtube.com/playlist?list=PLkA6oAVN3hh_nihZ1mh6cO3n-uMdF7UlV
- Docker YouTube Channel:
<https://www.youtube.com/channel/UC76AVf2JkrwjjxNKMuPpscHQ>
- Great video tutorial:
<https://www.udemy.com/docker-tutorial-for-devops-run-docker-containers>

QUOTATION

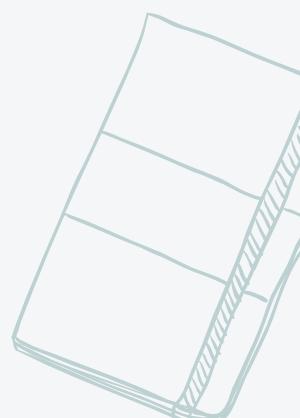
The real value of Docker is not technology



It's getting people to agree on something

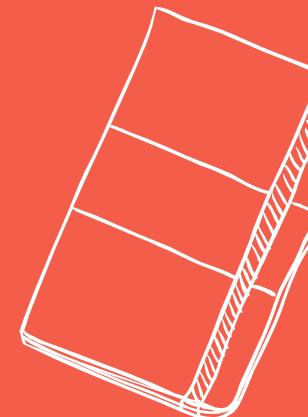


imgflip.com





QUESTIONS?





thank you!

