i-bitz company limited
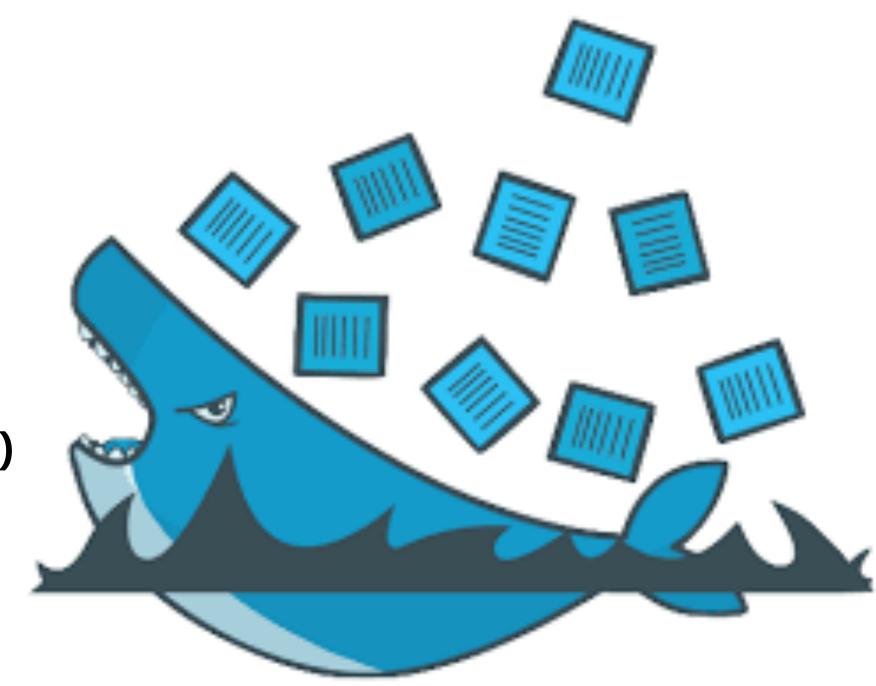THE GEOMATICS COMPANY

Docker

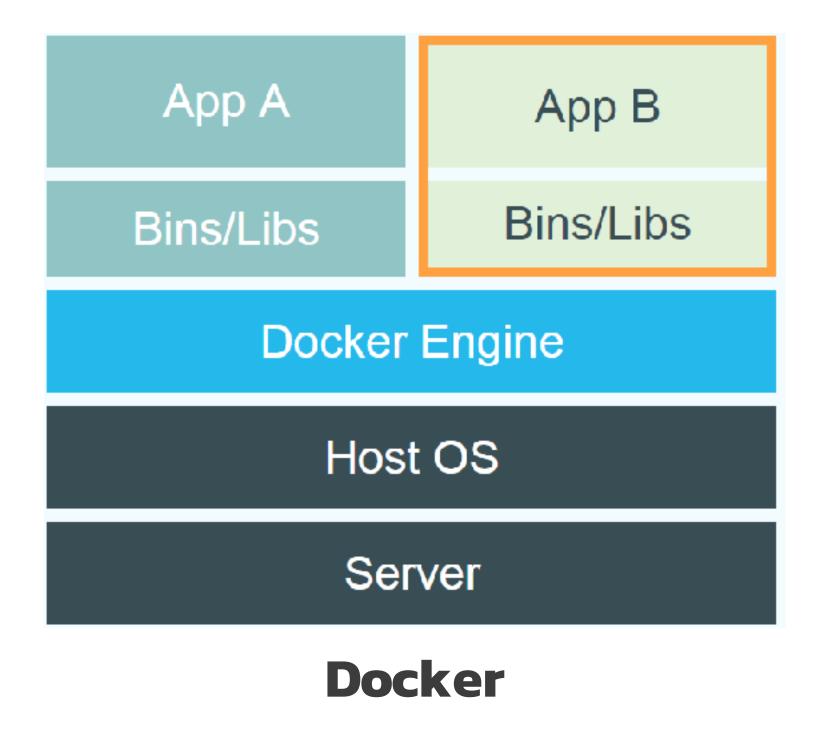~~From Chick to Dragon~~

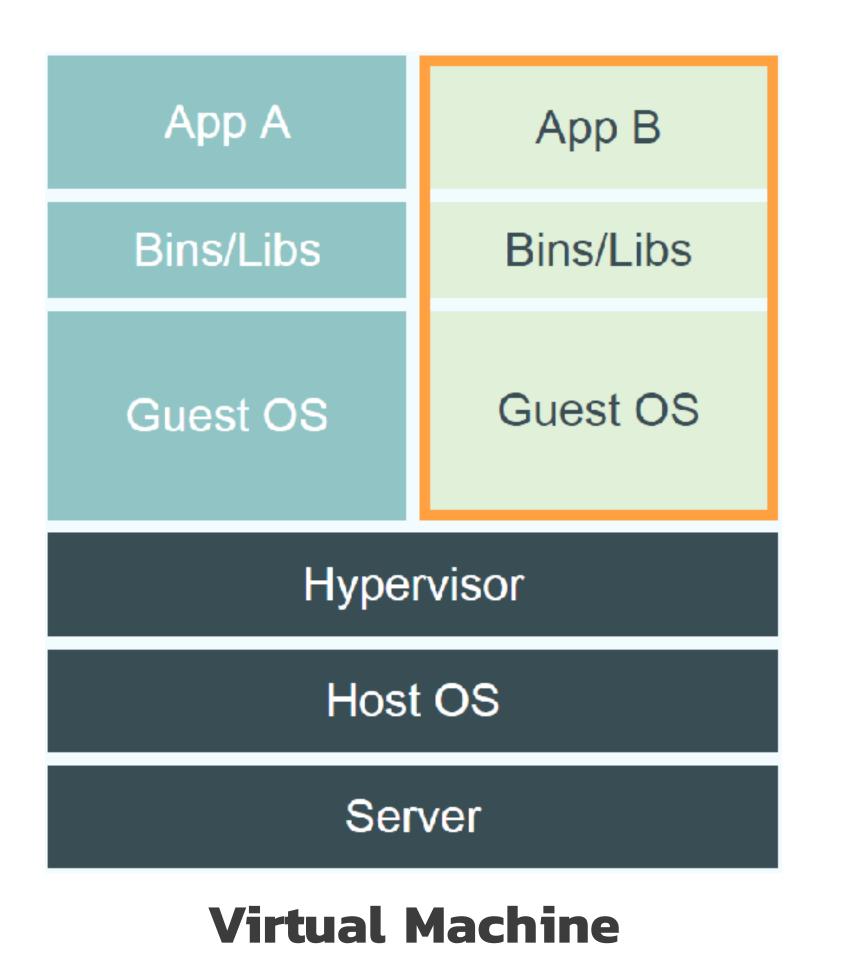From Zero to Hero (boss write)

What is Docker?

Docker is an open-source project that automates the deployment of applications inside software containers, by providing an additional layer of abstraction and automation of operating system-level virtualization on Linux.

[en.wikipedia.org]

i-bitz company limited
THE GEOMATICS COMPANY

Docker vs. Virtual Machine

Docker

Virtual Machine

" *Container ≠ Virtual Machine* "
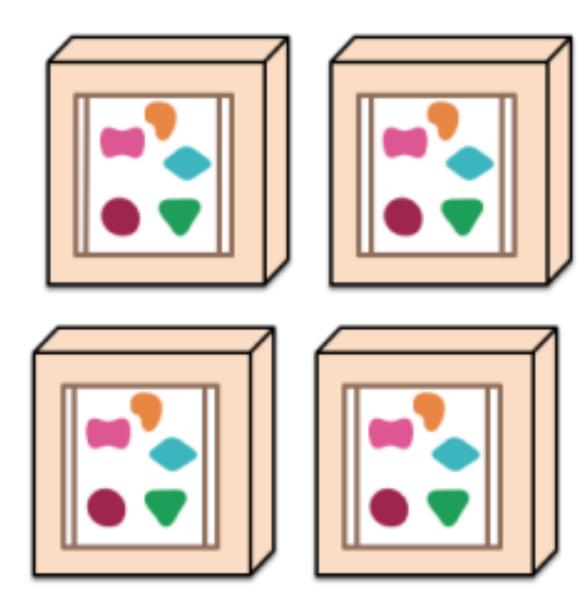
MicroService Concept.

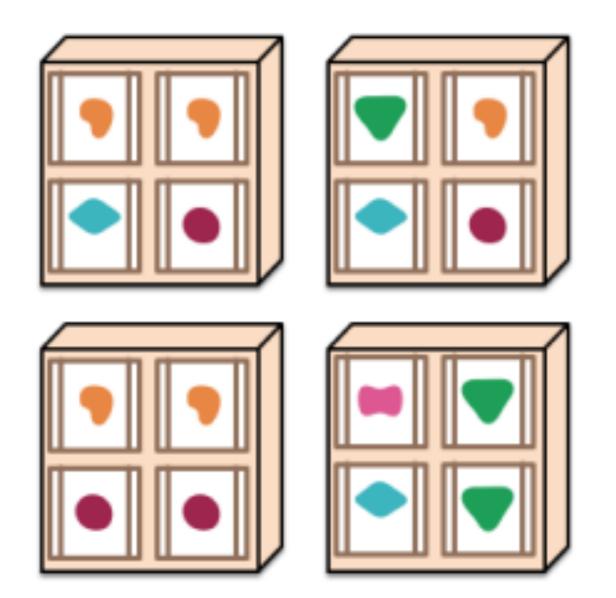A monolithic application puts all its functionality into a single process...

A microservices architecture puts each element of functionality into a separate service...

... and scales by replicating the monolith on multiple servers

... and scales by distributing these services across servers, replicating as needed.
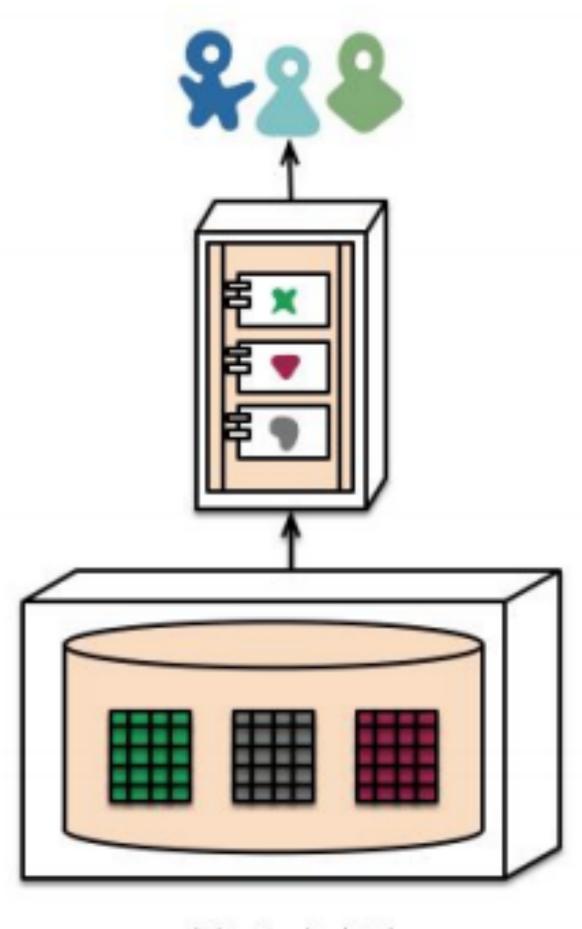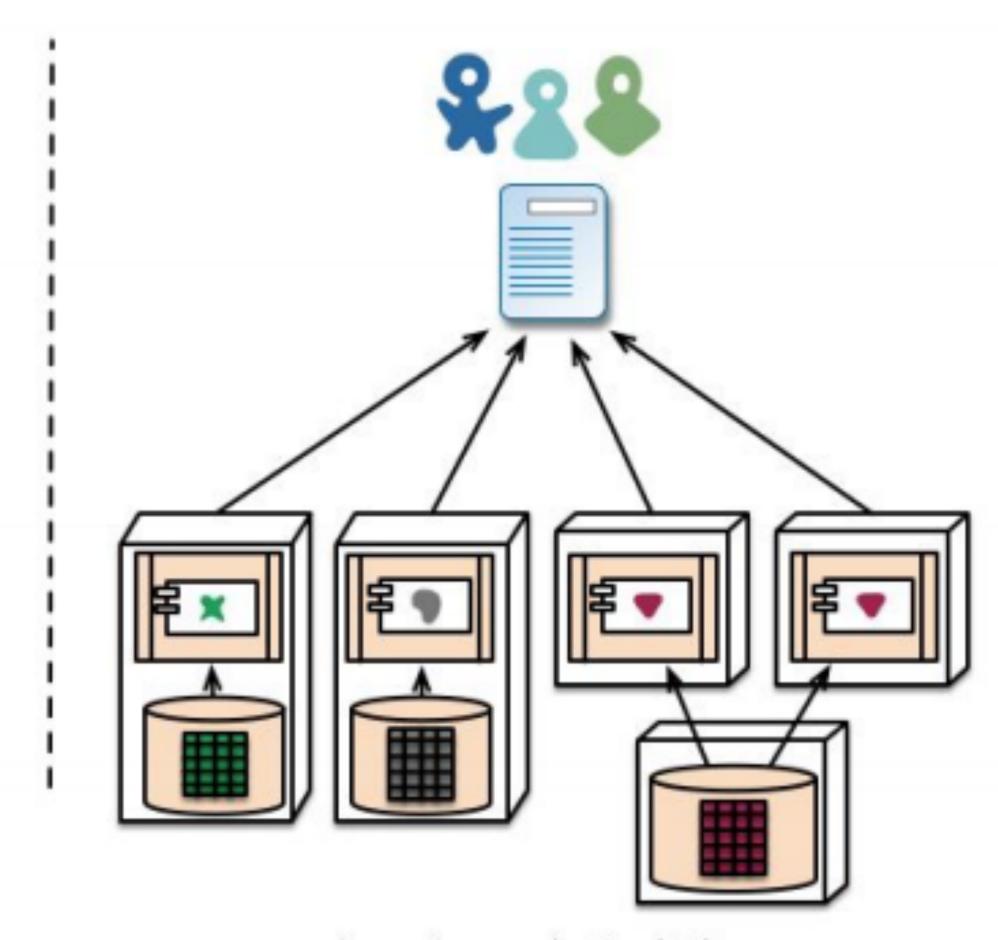
i-bitz company limited
THE GEOMATICS COMPANY

MicroService Concept.

STACK (Web Server)

- Apache + PHP

- MySQL

- PhpMyAdmin

monolith - single database

microservices - application databases

i-bitz company limited
THE GEOMATICS COMPANY

Installations.

https://www.docker.com/community-edition#/

download

Let's Go.
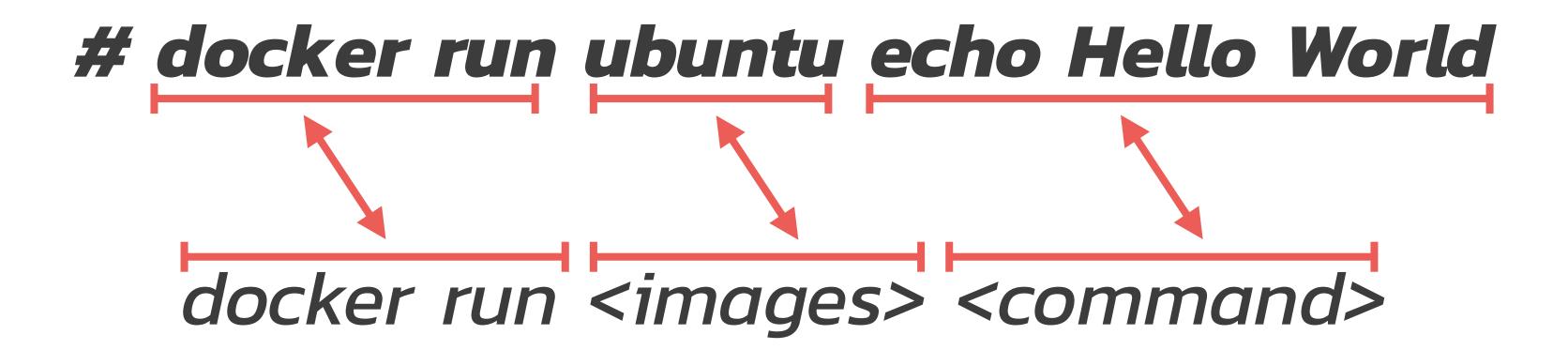
" We cannot maintain the broken container "

" Container born to die. "

[ME]

Hello world.

 # docker run ubuntu echo Hello World

docker run <images> <command>

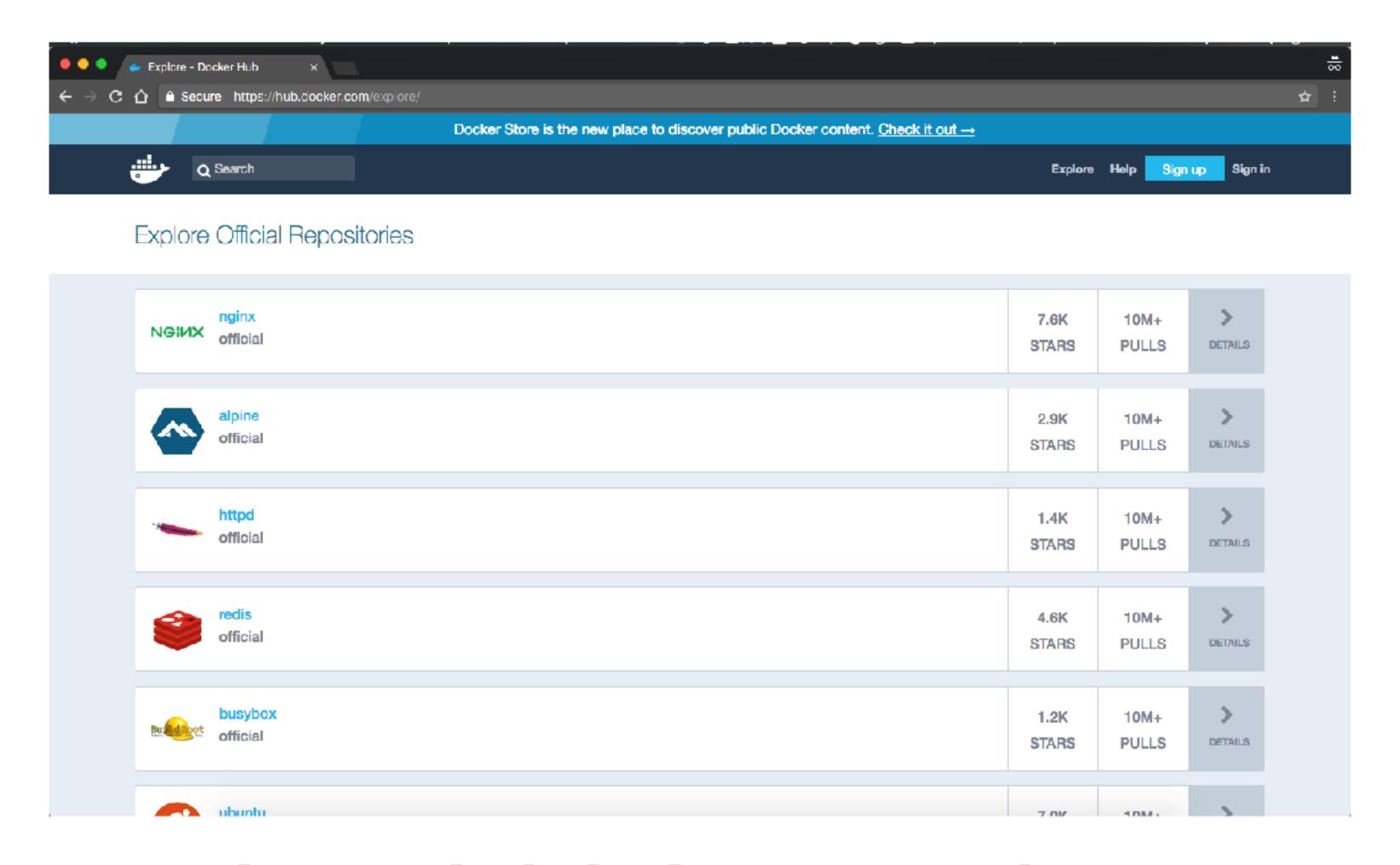i-bitz company limited
THE GEOMATICS COMPANY

Hello world.

```
SMJTHEOFF-MBP:TMP smjtheoff$ docker run ubuntu echo Hello World    Docker Run command.
Unable to find image 'ubuntu:latest' locally
latest: Pulling from library/ubuntu
50aff78429b1: Pull complete
f6d82e297bce: Pull complete                                        Docker engine can automatic pull image if not found.
275abb2c8a6f: Pull complete
9f15a39356d6: Pull complete
fc0342a94c89: Pull complete
Digest: sha256:ec0e4e8bf2c1178e025099eed57c566959bb408c6b478c284c1683bc4298b683    Container ID
Status: Downloaded newer image for ubuntu:latest
Hello World    Result
SMJTHEOFF-MBP:TMP smjtheoff$
```

i-bitz company limited
THE GEOMATICS COMPANY

How to find docker images.

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

Component of Docker Images.

PUBLIC REPOSITORY

Repos ← ibitz/geoserver ☆ → Images

Last pushed: 5 days ago

Repo Info   Tags

Short Description

Geoserver with extension

Full Description

Geoserver

Tags

Geoserver Version	Tags
2.12.1	2.12.1 , stable , latest
2.11.4	2.11.4 , lts
2.11.1	2.11.1
2.9.4	2.9.4

PUBLIC REPOSITORY

ibitz/geoserver ☆

Last pushed: 5 days ago

Repo Info   Tags   Tags

Tag Name	Compressed Size	Last Updated
stable	1 GB	5 days ago
lts	1 GB	5 days ago
latest	1 GB	5 days ago

Owner

ibitz

https://hub.docker.com/r/ibitz/geoserver/

i-bitz company limited
THE GEOMATICS COMPANY

STACK (Web Server)

- Apache + PHP
- MySQL
- PhpMyAdmin

Case #1 Web Server

Client

Apache + PHP

MySQL

PhpMyAdmin

i-bitz company limited
THE GEOMATICS COMPANY

Case #1 Web Server

STACK (Web Server)

- **Apache + PHP**

- *MySQL*

- *PhpMyAdmin*
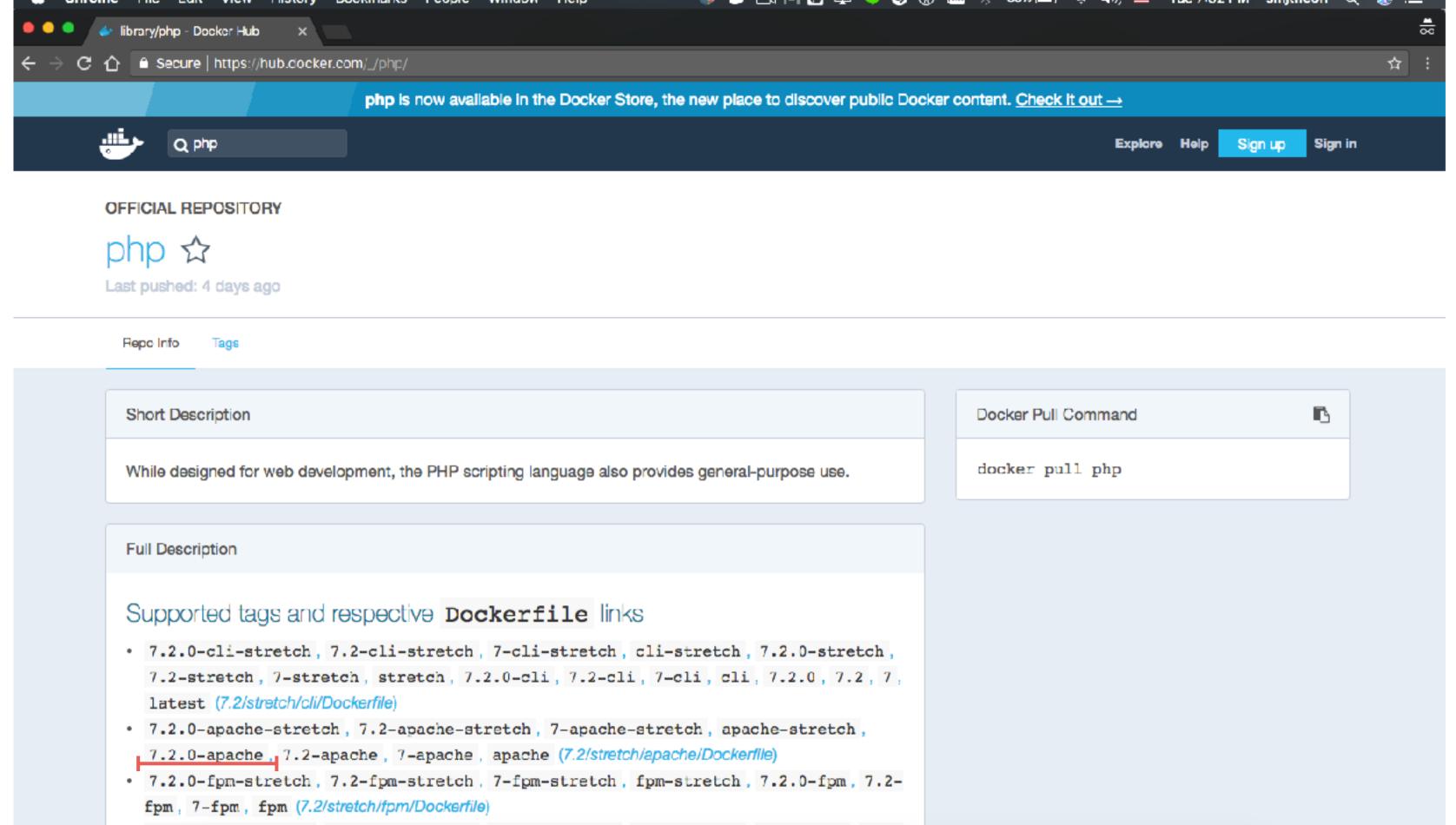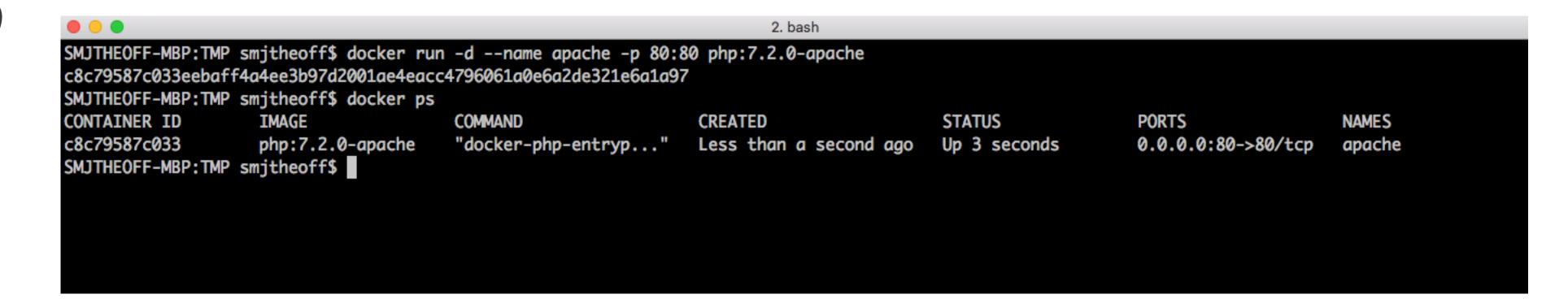
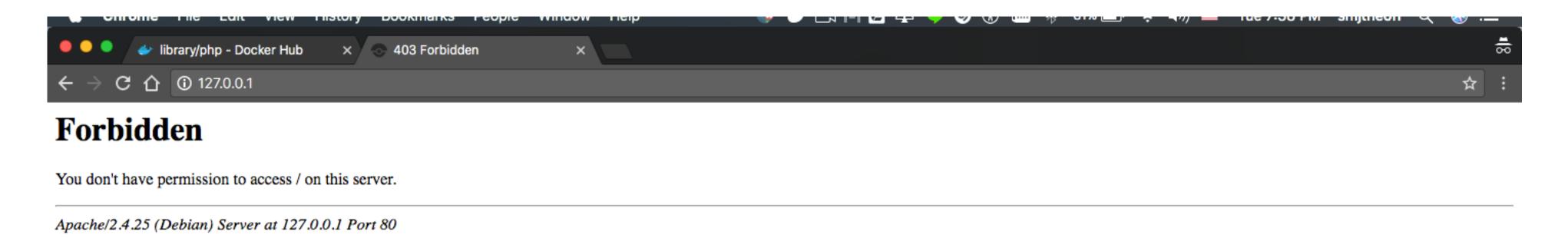**https://hub.docker.com/_/php/**

Case #1 Web Server

text The First Container

STACK (Web Server)

- **Apache + PHP**

- *MySQL*

- *PhpMyAdmin*

docker run –d --name apache –p 80:80 php:7.2.5–apache

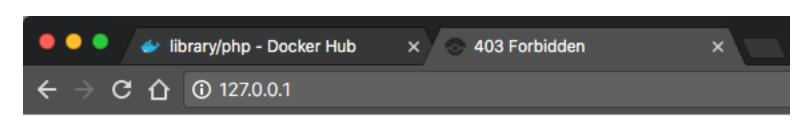** if not specific tag docker can choose tag "latest" automatic **

http://127.0.0.1

STACK (Web Server)

- **Apache + PHP**

- *MySQL*

- *PhpMyAdmin*

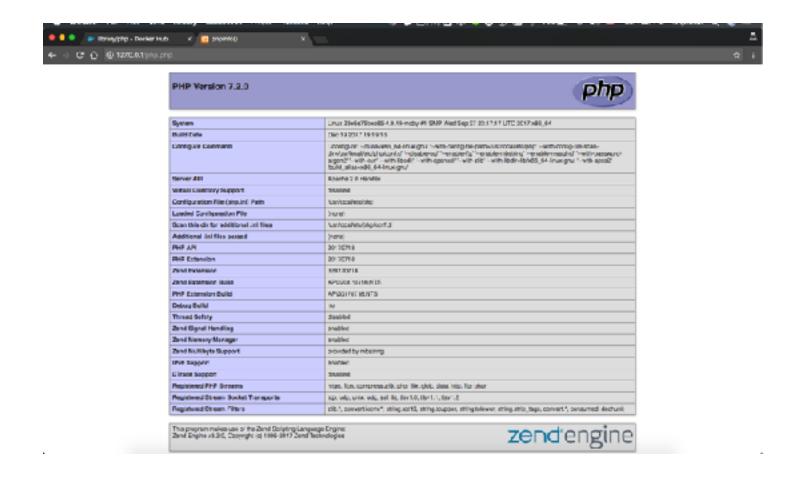Remove old container.
docker rm –f apache

Run Docker with **persistence data** in local directory
docker run –d ––name apache –p 80:80 –v $PWD/web:/var/www/html php:7.2.5–apache

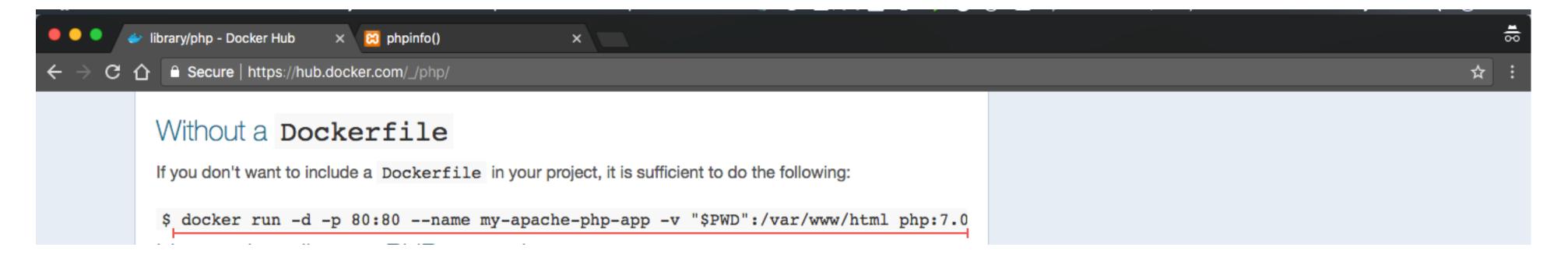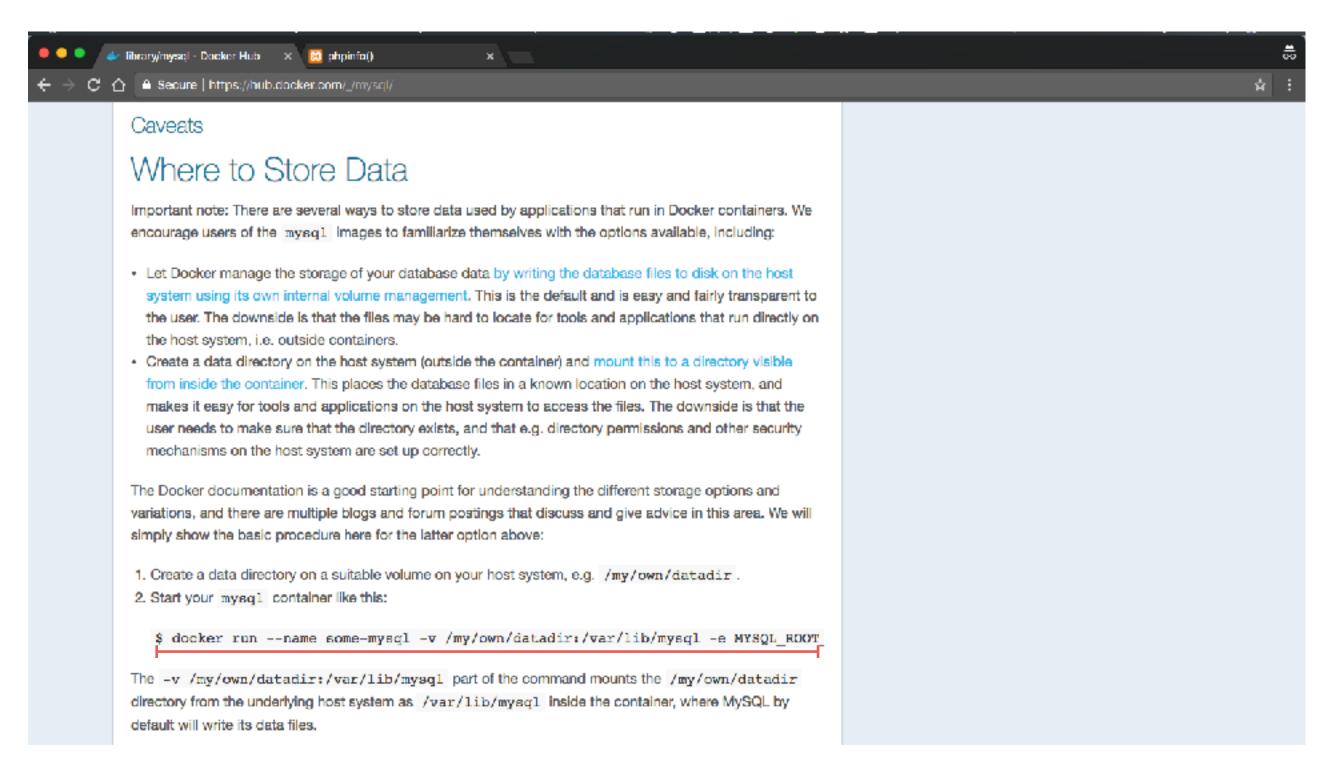**i-bitz X Docker**

http://127.0.0.1

http://127.0.0.1/php.php

i-bitz company limited
THE GEOMATICS COMPANY

How to Find Container Configurations.

Without a Dockerfile

If you don't want to include a `Dockerfile` in your project, it is sufficient to do the following:

```
$ docker run -d -p 80:80 --name my-apache-php-app -v "$PWD":/var/www/html php:7.0
```


Caveats
Where to Store Data

Important note: There are several ways to store data used by applications that run in Docker containers. We encourage users of the `mysql` images to familiarize themselves with the options available, including:

- Let Docker manage the storage of your database data by writing the database files to disk on the host system using its own internal volume management. This is the default and is easy and fairly transparent to the user. The downside is that the files may be hard to locate for tools and applications that run directly on the host system, i.e. outside containers.
- Create a data directory on the host system (outside the container) and mount this to a directory visible from inside the container. This places the database files in a known location on the host system, and makes it easy for tools and applications on the host system to access the files. The downside is that the user needs to make sure that the directory exists, and that e.g. directory permissions and other security mechanisms on the host system are set up correctly.

The Docker documentation is a good starting point for understanding the different storage options and variations, and there are multiple blogs and forum postings that discuss and give advice in this area. We will simply show the basic procedure here for the latter option above:

1. Create a data directory on a suitable volume on your host system, e.g. `/my/own/datadir`.
2. Start your `mysql` container like this:

```
$ docker run --name some-mysql -v /my/own/datadir:/var/lib/mysql -e MYSQL_ROOT
```

The `-v /my/own/datadir:/var/lib/mysql` part of the command mounts the `/my/own/datadir` directory from the underlying host system as `/var/lib/mysql` inside the container, where MySQL by default will write its data files.

Please Read Images Description.

i-bitz company limited
THE GEOMATICS COMPANY

Case #1 Web Server

STACK (Web Server)

- Apache + PHP

- **MySQL**

- *PhpMyAdmin*

https://hub.docker.com/_/mysql/

Case #1 Web Server

STACK (Web Server)

- *Apache + PHP*

- **MySQL**

- *PhpMyAdmin*

Run MySQL Container with environment variable.
docker run –d ––name mysql –p 3306:3306 –e MYSQL_ROOT_PASSWORD=P@ssw0rd mysql:8.0

** If MySQL Persistence data Use internal Directory " /var/lib/mysql " **
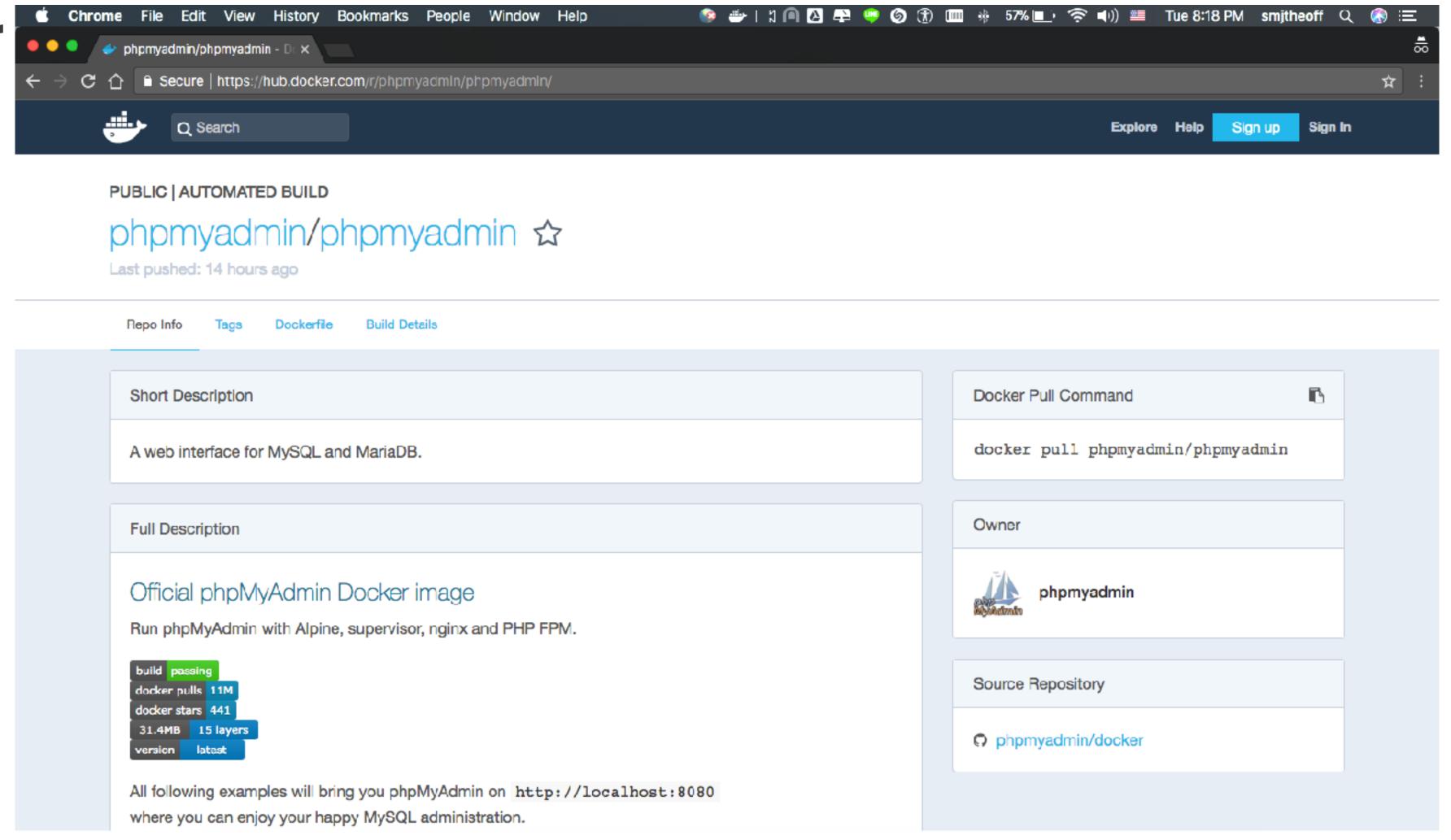eg, " –v $PWD/mysql:/var/lib/mysql "

Environment Variable


```
$ docker run -it --rm mysql:tag --verbose --help
```

Environment Variables

When you start the `mysql` image, you can adjust the configuration of the MySQL instance by passing one or more environment variables on the `docker run` command line. Do note that none of the variables below will have any effect if you start the container with a data directory that already contains a database: any pre-existing database will always be left untouched on container startup.

MYSQL_ROOT_PASSWORD

This variable is mandatory and specifies the password that will be set for the MySQL `root` superuser account. In the above example, it was set to `my-secret-pw`.

MYSQL_DATABASE

This variable is optional and allows you to specify the name of a database to be created on image startup. If a user/password was supplied (see below) then that user will be granted superuser access (corresponding to `GRANT ALL`) to this database.

MYSQL_USER, MYSQL_PASSWORD

These variables are optional, used in conjunction to create a new user and to set that user's password. This user will be granted superuser permissions (see above) for the database specified by the `MYSQL_DATABASE` variable. Both variables are required for a user to be created.

Do note that there is no need to use this mechanism to create the root superuser, that user gets created by default with the password specified by the `MYSQL_ROOT_PASSWORD` variable.

MYSQL_ALLOW_EMPTY_PASSWORD

This is an optional variable. Set to `yes` to allow the container to be started with a blank password for the root user. *NOTE*: Setting this variable to `yes` is not recommended unless you really know what you are

Case #1 Web Server

STACK (Web Server)

- Apache + PHP

- MySQL

- **PhpMyAdmin**

https://hub.docker.com/r/phpmyadmin/phpmyadmin/

Case #1 Web Server

STACK (Web Server)

- *Apache + PHP*

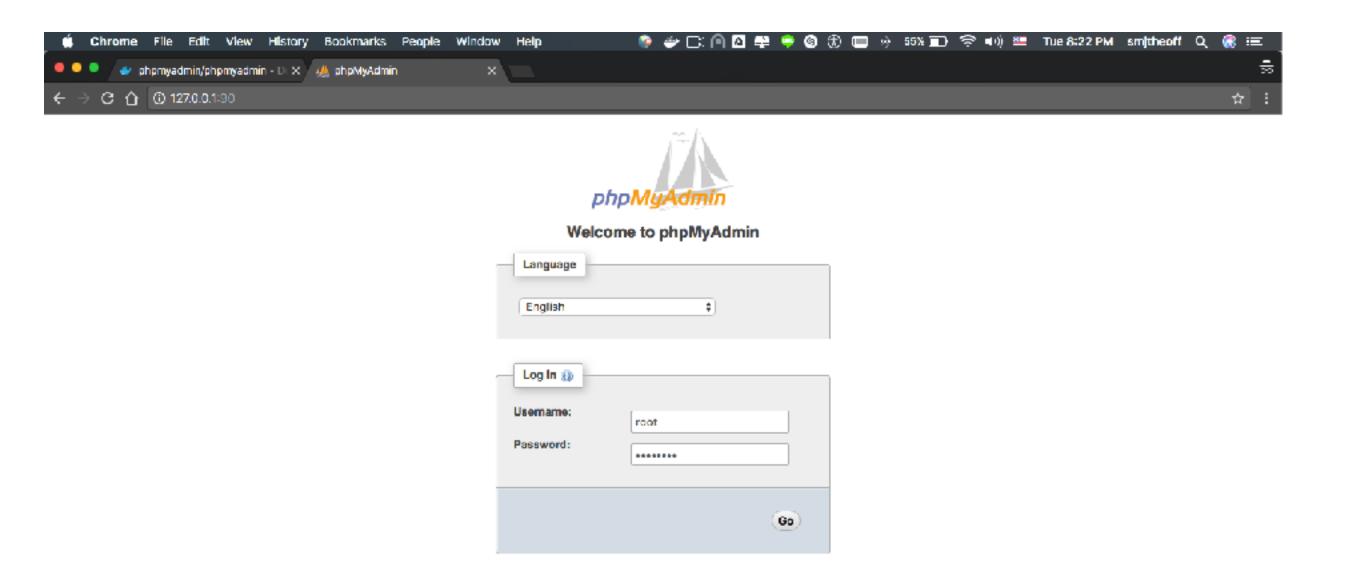- *MySQL*

- ***PhpMyAdmin***

Run PhpMyAdmin in custom port. (Port 80 duplicate apache)
docker run –d ––name phpmyadmin –p 90:80 –e PMA_HOST=192.168.10.66 phpmyadmin/phpmyadmin
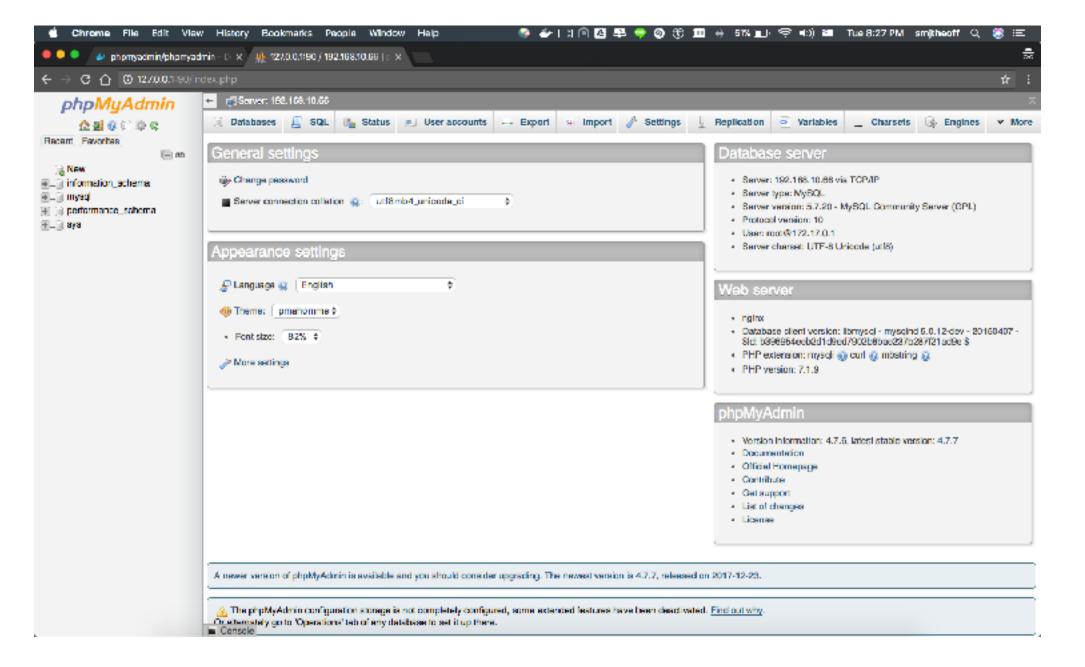
*** In "PMA_HOST" please input your IP Address. ***

Case #1 Web Server

STACK (Web Server)

- Apache + PHP

- MySQL

- **PhpMyAdmin**

http://127.0.0.1:90/
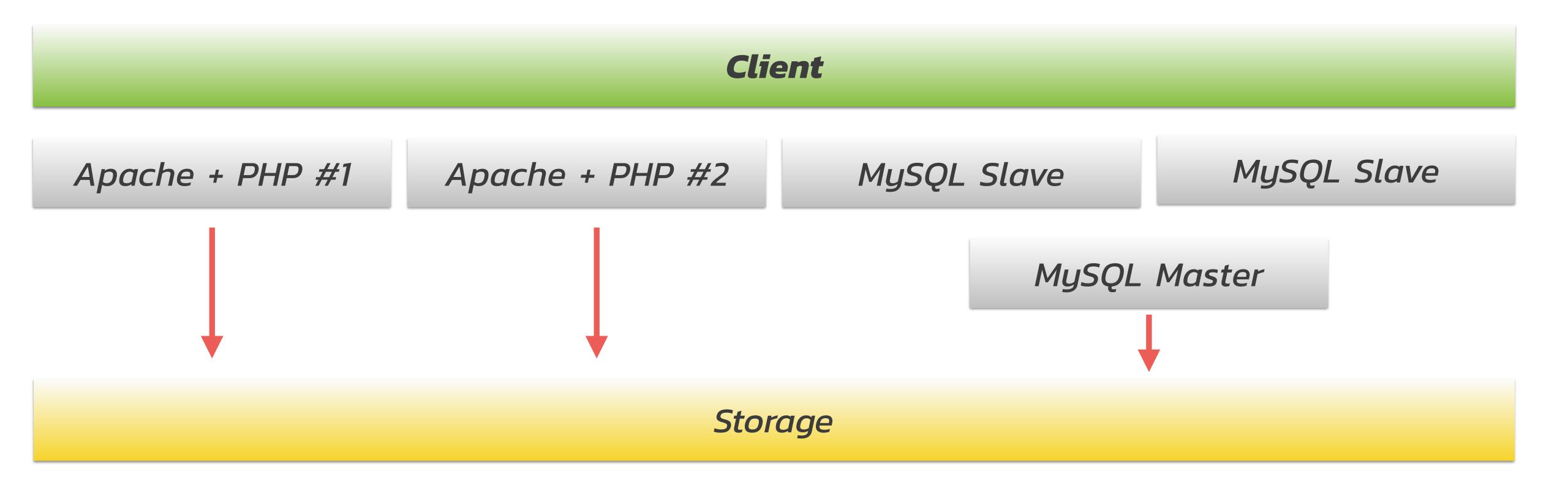Username & Password in mysql container environment variable.
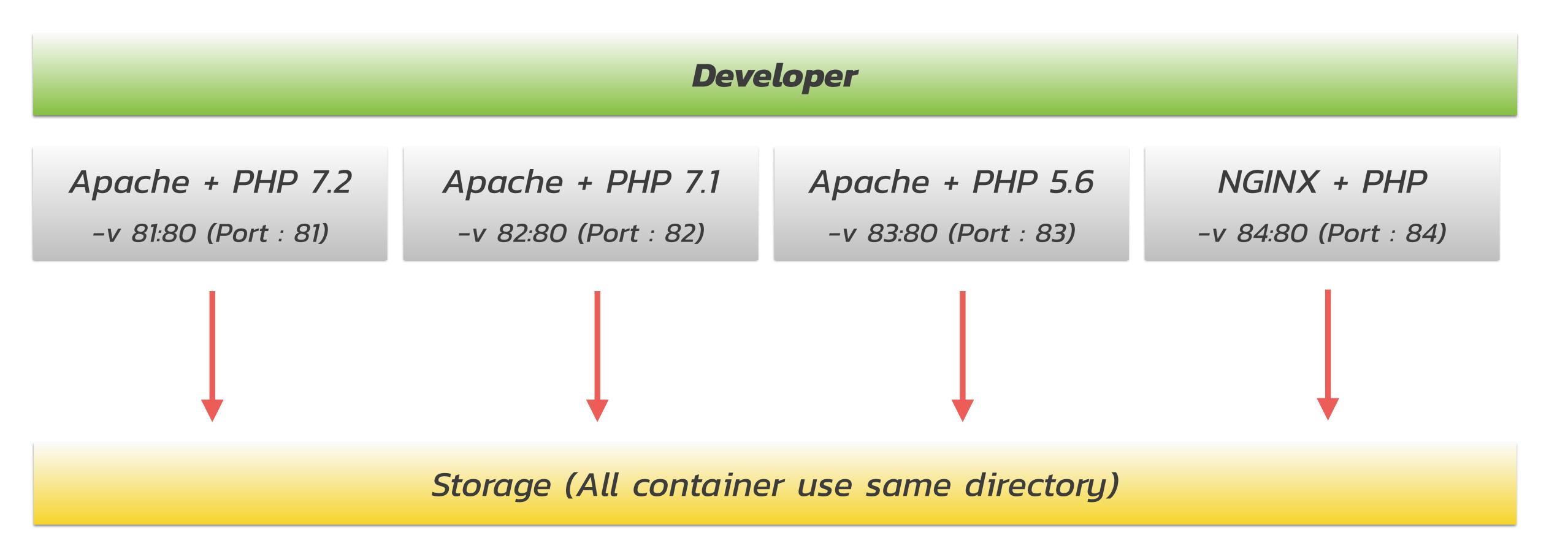
Case #1 Done!!!!

Dream of developer, Multi Env In Dev Machine.

Developer

Apache + PHP 7.2	Apache + PHP 7.1	Apache + PHP 5.6	NGINX + PHP
–v 81:80 (Port : 81)	–v 82:80 (Port : 82)	–v 83:80 (Port : 83)	–v 84:80 (Port : 84)

Storage (All container use same directory)

For Dev and Test Multiple Environment.

i-bitz company limited
THE GEOMATICS COMPANY

Why use a container in web server stack !!!!

Client

Apache + PHP #1	Apache + PHP #2	MySQL Slave	MySQL Slave

MySQL Master

Storage
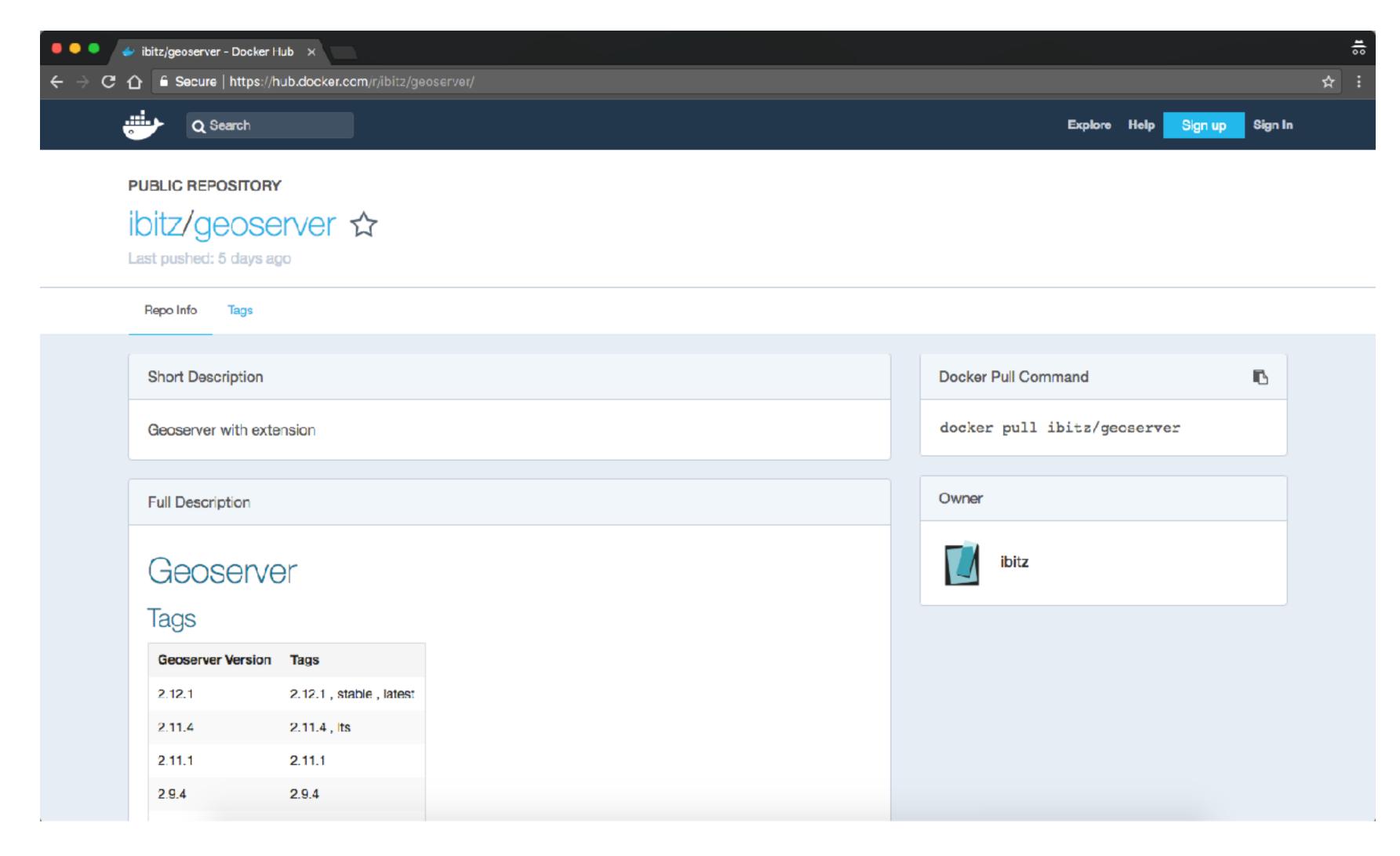
High Scale And High Availability

i-bitz company limited
THE GEOMATICS COMPANY

Case #2 GeoServer

STACK (GeoServer)

- *ibitz/geoserver*

- *Keep DATA_DIR*

- *Add context*

Case #2 GeoServer

STACK (GeoServer)

- *ibitz/geoserver*

- *Keep DATA_DIR*

- *Add context*

Tags Available.

STACK (GeoServer)

- *ibitz/geoserver*

- *Keep DATA_DIR*

- *Add context*


```
SMJTHEOFF-MBP:ibitz_docker smjtheoff$ docker run -d -p 8080:8080 --name geoserver ibitz/geoserver:lts
c7197c72f2a4aa81509ce71ca2f23f9038047041e32735a789e2ebd9778e16fd
SMJTHEOFF-MBP:ibitz_docker smjtheoff$ docker ps
CONTAINER ID        IMAGE                   COMMAND                 CREATED               STATUS                          PORTS
  NAMES
c7197c72f2a4        ibitz/geoserver:lts     "catalina.sh run"       Less than a second ago  Up 3 seconds (health: starting)  0.0.0.0:8080->8080/tcp
  geoserver
50c238955548        phpmyadmin/phpmyadmin   "/run.sh phpmyadmin"    4 hours ago           Up 4 hours                      0.0.0.0:90->80/tcp
  phpmyadmin
f1bd40b586ab        mysql                   "docker-entrypoint..."  4 hours ago           Up 4 hours                      0.0.0.0:3306->3306/tcp
  mysql
23e8e76bea65        php:7.2.0-apache        "docker-php-entryp..."  4 hours ago           Up 4 hours                      0.0.0.0:80->80/tcp
  apache
SMJTHEOFF-MBP:ibitz_docker smjtheoff$
```

docker run –d –p 8080:8080 ––name geoserver ibitz/geoserver:lts

http://127.0.0.1:8080/geoserver/web

STACK (GeoServer)

- *ibitz/geoserver*

- *Keep DATA_DIR*

- *Add context*

docker logs geoserver
** or use " docker logs –f geoserver " to see realtime logs
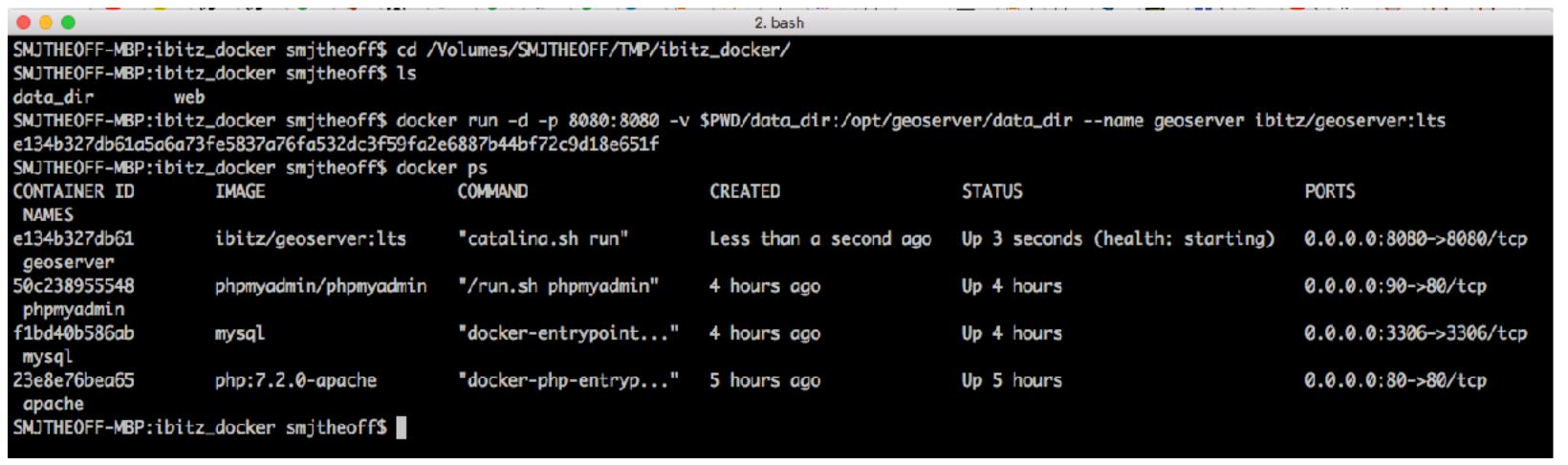
Case #2 GeoServer
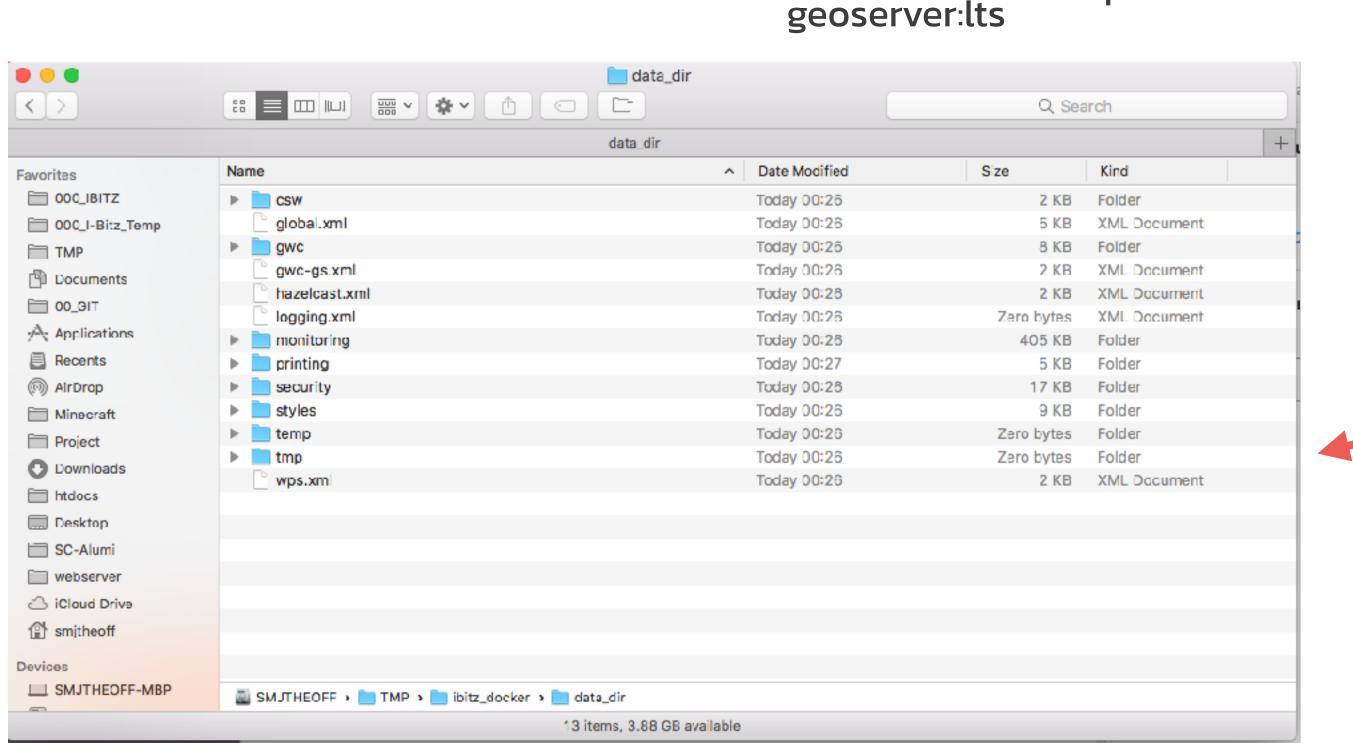
Remove old container.
 # docker rm –f geoserver

STACK (GeoServer)

- *ibitz/geoserver*

- *Keep DATA_DIR*

- *Add context*


```
SMJTHEOFF-MBP:ibitz_docker smjtheoff$ cd /Volumes/SMJTHEOFF/TMP/ibitz_docker/
SMJTHEOFF-MBP:ibitz_docker smjtheoff$ ls
data_dir        web
SMJTHEOFF-MBP:ibitz_docker smjtheoff$ docker run –d –p 8080:8080 -v $PWD/data_dir:/opt/geoserver/data_dir --name geoserver ibitz/geoserver:lts
e134b327db61a5a6a73fe5837a76fa532dc3f59fa2e6887b44bf72c9d18e651f
SMJTHEOFF-MBP:ibitz_docker smjtheoff$ docker ps
CONTAINER ID        IMAGE                    COMMAND                 CREATED               STATUS                          PORTS
  NAMES
e134b327db61        ibitz/geoserver:lts      "catalina.sh run"       Less than a second ago   Up 3 seconds (health: starting)   0.0.0.0:8080->8080/tcp
  geoserver
50c238955548        phpmyadmin/phpmyadmin    "/run.sh phpmyadmin"    4 hours ago           Up 4 hours                       0.0.0.0:90->80/tcp
  phpmyadmin
f1bd40b586ab        mysql                    "docker-entrypoint..."  4 hours ago           Up 4 hours                       0.0.0.0:3306->3306/tcp
  mysql
23e8e76bea65        php:7.2.0-apache         "docker-php-entryp..."  5 hours ago           Up 5 hours                       0.0.0.0:80->80/tcp
  apache
SMJTHEOFF-MBP:ibitz_docker smjtheoff$
```

docker run –d –p 8080:8080 –v $PWD/data_dir:/opt/geoserver/data_dir –-name geoserver ibitz/geoserver:lts

Case #2 GeoServer

Example GeoServer Context.

STACK (GeoServer)

- – *ibitz/geoserver*

- – *Keep DATA_DIR*

- – *Add context*
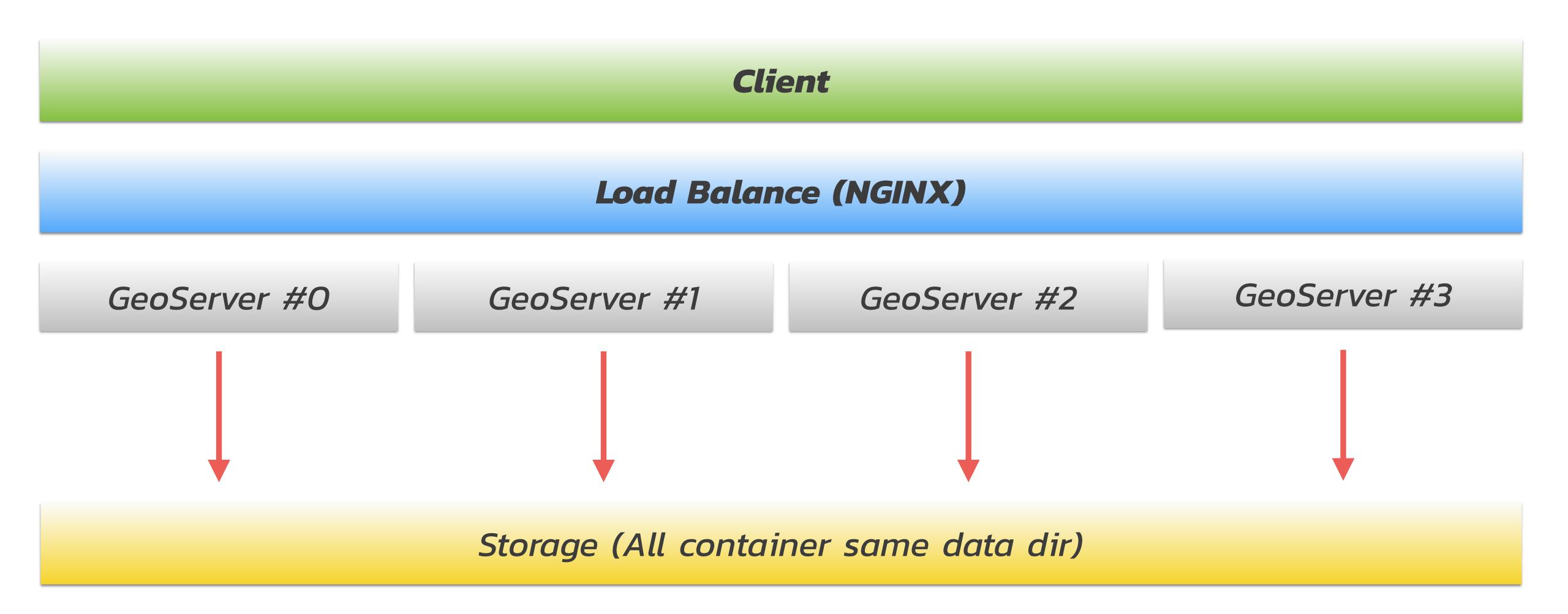
https://github.com/i-bitz/Tomcat/blob/master/conf/context.xml

docker run –d –p 8080:8080 –v $PWD/data_dir:/opt/geoserver/data_dir –v $PWD/context.xml:/usr/local/tomcat/conf/context.xml ––name geoserver ibitz/geoserver:lts

i-bitz company limited
THE GEOMATICS COMPANY

GeoServer Cluster easy ways.

Client

Load Balance (NGINX)

| GeoServer #0 | GeoServer #1 | GeoServer #2 | GeoServer #3 |

Storage (All container same data dir)

i-bitz company limited
THE GEOMATICS COMPANY

" Not Keep Important Data Inside Container "

" Container born to die. "

[ME]

" Container Must Go ON "

Production Show. <FloodV2> (Maps Server)
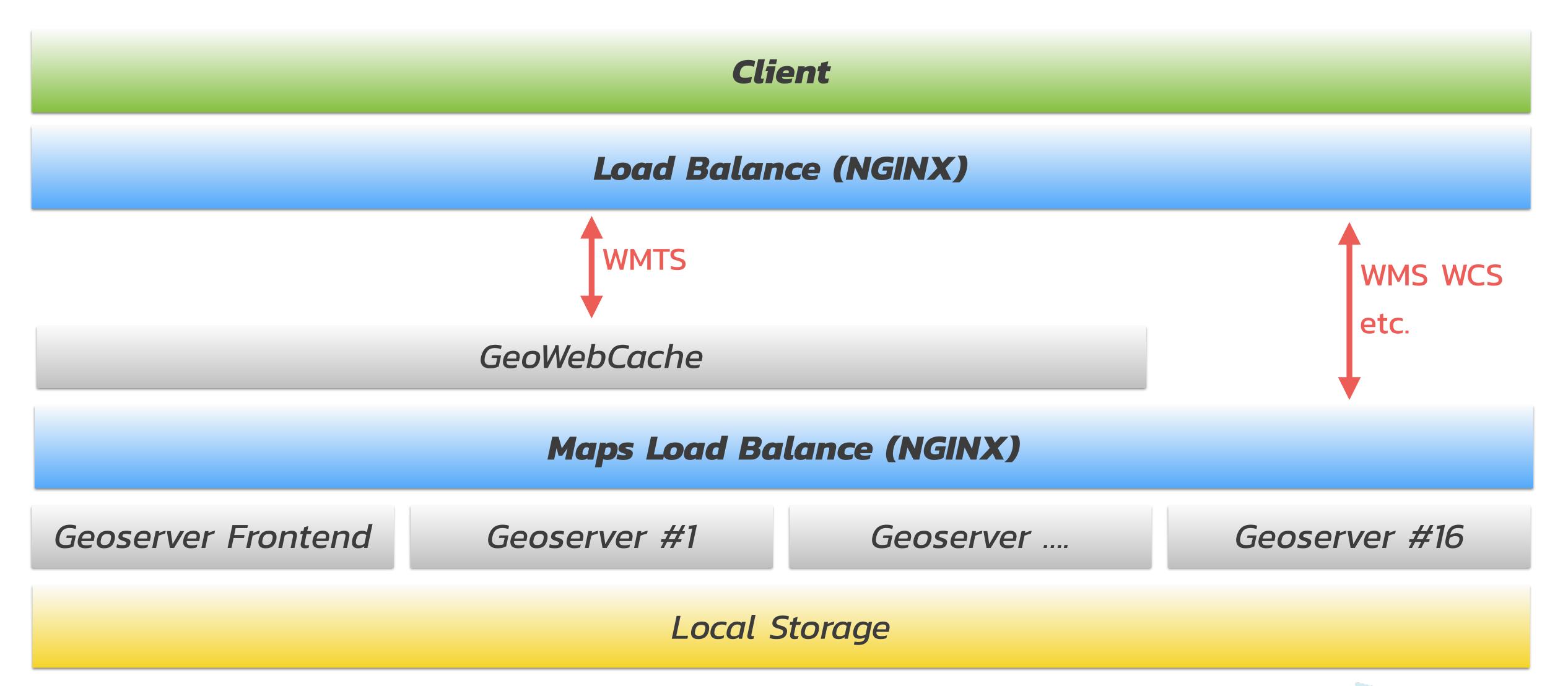
Client

Load Balance (NGINX)

WMTS

WMS WCS
etc.

GeoWebCache

Maps Load Balance (NGINX)

Geoserver Frontend   Geoserver #1   Geoserver ....   Geoserver #16

Local Storage

i-bitz company limited
THE GEOMATICS COMPANY
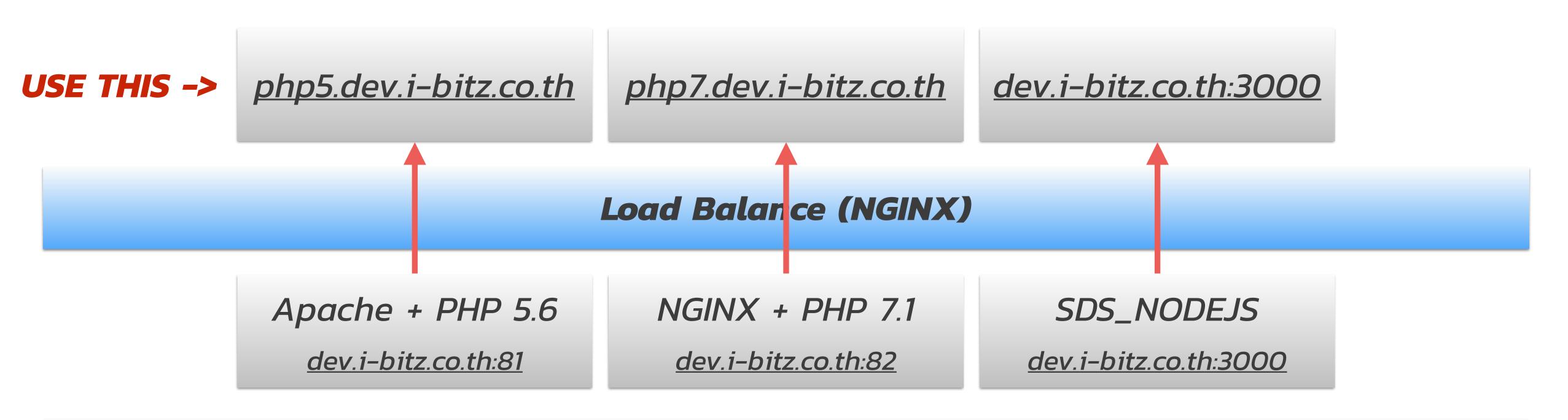
Baseapp02 (WebDocker)

uat.i-bitz.co.th

By Project (If bring for external user test)

USE THIS ->   php5.dev.i-bitz.co.th   php7.dev.i-bitz.co.th   dev.i-bitz.co.th:3000

Load Balance (NGINX)

Apache + PHP 5.6

dev.i-bitz.co.th:81

NGINX + PHP 7.1

dev.i-bitz.co.th:82

SDS_NODEJS

dev.i-bitz.co.th:3000

//dev.i-bitz.co.th/ibitzadmin (U: ibitzadmin, P: ibitz@min)

i-bitz company limited
THE GEOMATICS COMPANY

" CI & CD "

You Push >> dev.i-ibitz.co.th

You Merge >> Productions

i-bitz company limited
THE GEOMATICS COMPANY

IT TEAM Road Maps

Private Cloud >

< K8S (Container Orchestra)

CI & CD >

< Deploy Next Projects With Container

Multi Env Test >

Force Team to Container >

< Migration All to Container

i-bitz company limited
THE GEOMATICS COMPANY

" Thank You. "

i-bitz company limited
THE GEOMATICS COMPANY

Material

https://goo.gl/msmBQc