

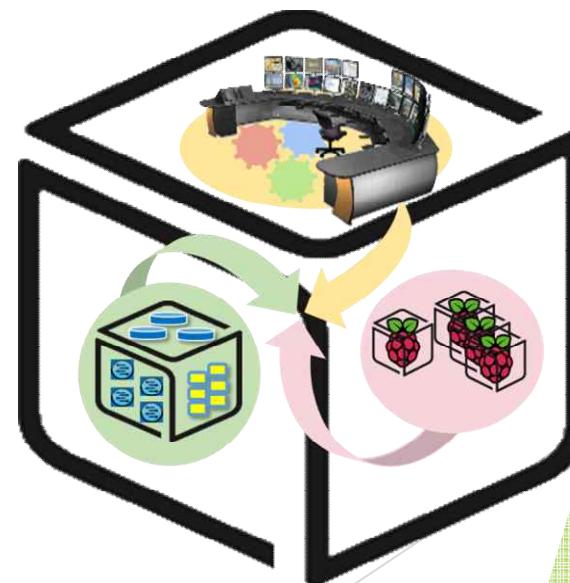
SmartX Labs for Computer Systems

Functions Lab

v1.1

(2016, Spring)

NetCS Lab



History and Contributor of Functions Lab

(2016. 06. 01)

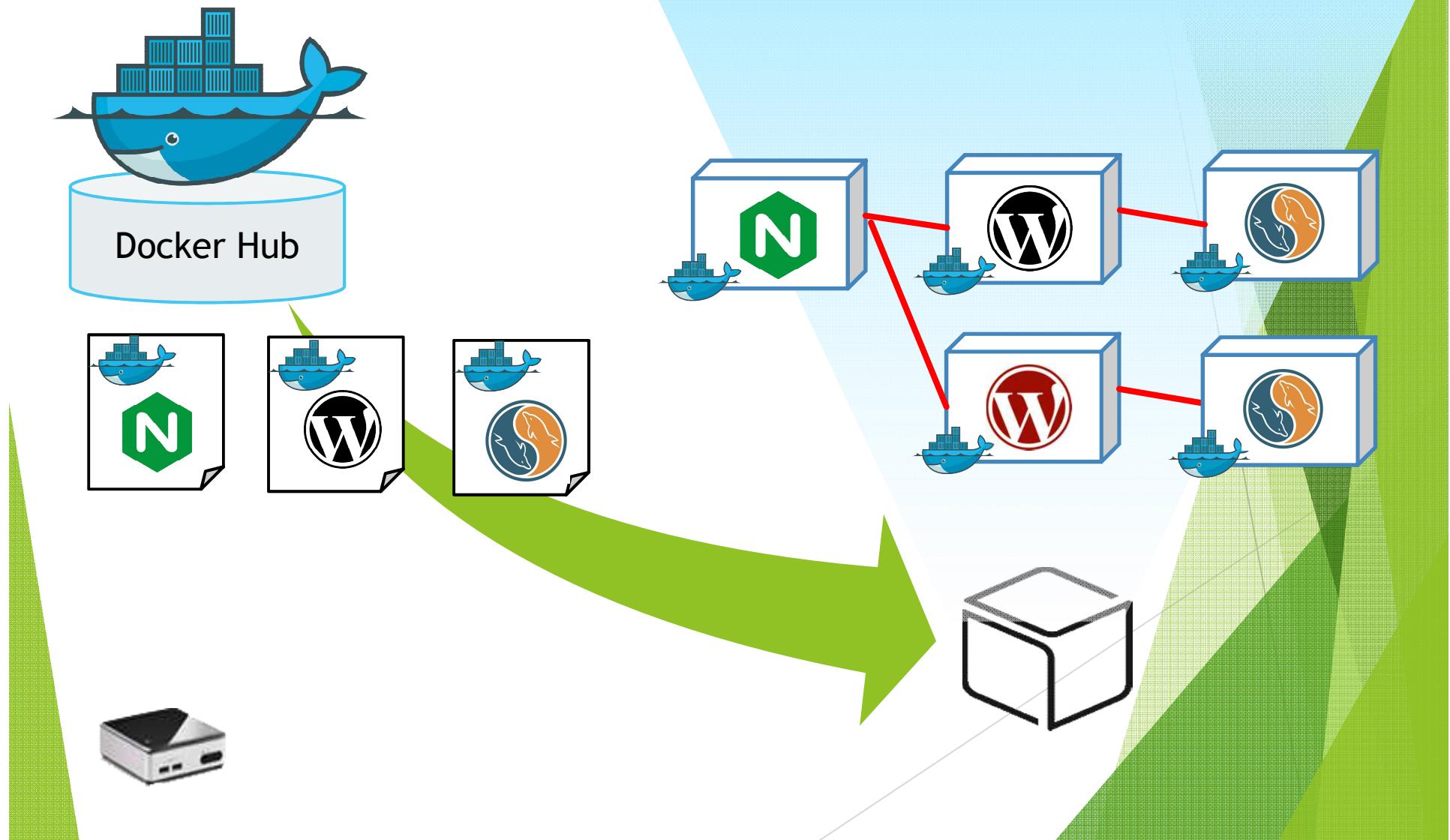
Version	Updated Date	Updated Contents	Contributor
v1.0	20160517	초고 완성	배정주
v1.1	20160601	p20, p32 wordpress url을 yourip -> localhost로 변경 (헷갈림 방지)	배정주

Functions Lab: Goal

- Strength of Docker: Docker Image
- Introduce Docker Hub
 - + Searching and Getting Image from Docker Hub
- Running 3 Tier (nginx-wordpress-mysql) Web Application
- Understanding Docker basic network
(--link option)

Functions Lab: Overall

- One of 3-Tier example



Functions Lab: Overall

- Background knowledge of goal

We will running one of web application: Wordpress

This web application is consisted with 3 containers: nginx, wordpress, mysql



nginx

: A http server which has following features.

- Reverse proxying
- SSL TLS SNI support... and etc

Usually, It is compared by Apache.



wordpress

: It is web software to create website, blog, or application.

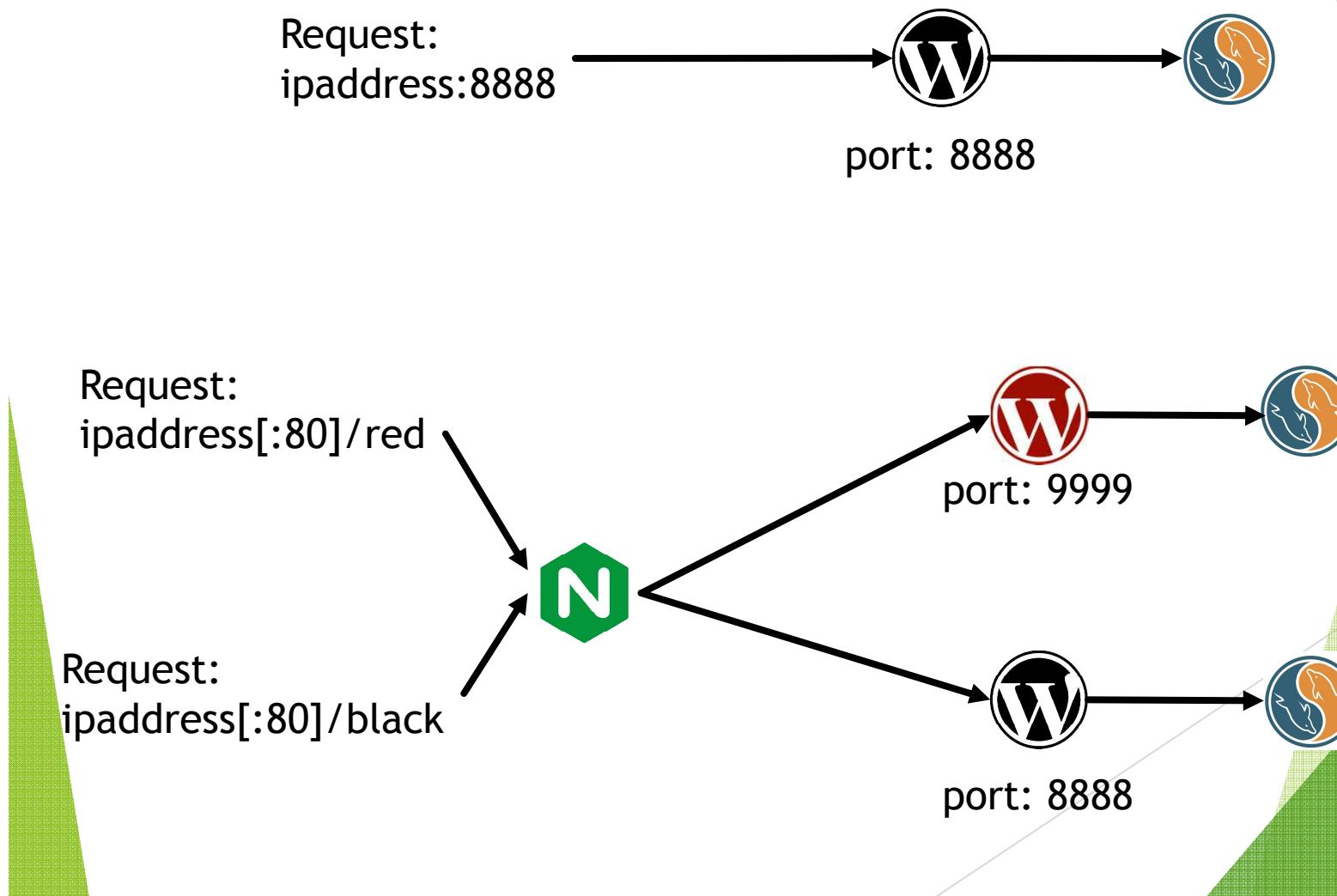


mysql

: Relational Database Management System(RDBMS)

Functions Lab: Overall

- Scenario



Prerequisite for Functions lab

Functions lab focus on (Docker container) functions of NUC



docker-engine (version: 1.10 or above)

Docker Background Knowledge

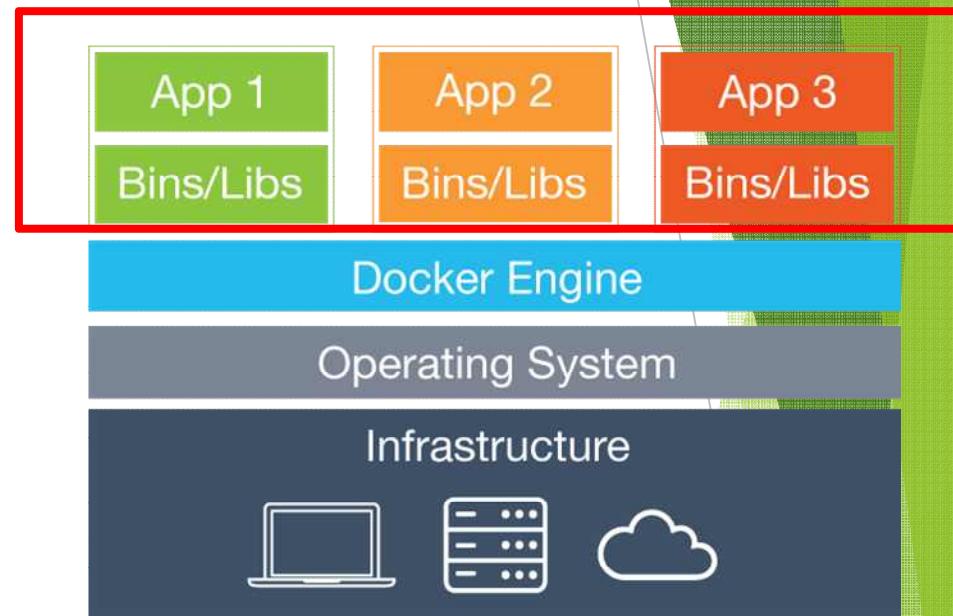
- Reminder: Docker Image

Docker image: A file which contains status of Docker container.

Similar to snapshot of VM.

It can be branched and versioned...

Docker image can be shared easily.
(It is important feature of Docker.)

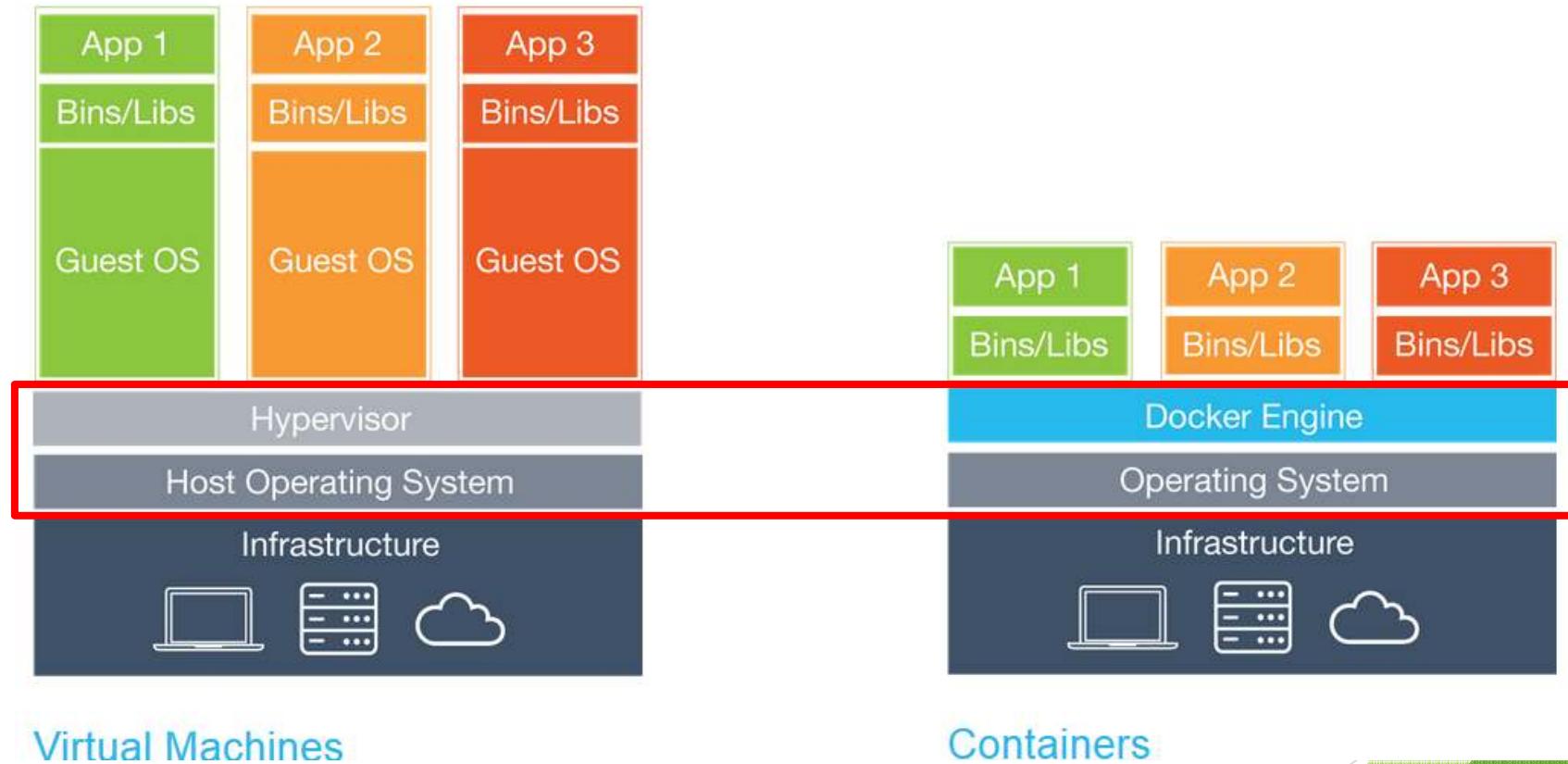


You can find Ubuntu image on your machine

REPOSITORY	CREATED	SIZE	TAG	IMAGE ID
centos	4 weeks ago	196.7 MB	latest	778a53015523
ubuntu	10 weeks ago	187.9 MB	latest	14b59d36bae0

Docker Background Knowledge

- Reminder: Container restraint



Since container uses host kernel, **OS** of host should be **Linux** distribution.

Docker Background Knowledge

- Why Docker image can be shared easily? (1)

```
Commands:
  attach      Attach to a running container
  build       Build an image from a Dockerfile
  commit      Create a new image from a container's changes
  cp          Copy files/folders between a container and the local filesystem
  create      Create a new container
  diff        Inspect changes to a container's filesystem
  pull        Pull an image or a repository from a registry
  push        Push an image or a repository to a registry
```

Docker provides related commands!

This usage is similar to code management system. (e.g, svn, git)

Docker Background Knowledge

- Why Docker image can be shared easily? (2)

```
FROM resin/rpi-raspbian:wheezy
MAINTAINER Seungryong Kim <srkim@nm.gist.ac.kr>

#Update & Install wget, vim
RUN apt-get update
RUN apt-get -y install wget
RUN apt-get -y install vim

#Timezone
RUN cp /usr/share/zoneinfo/Asia/Seoul /etc/localtime

#Install Oracle JAVA
RUN mkdir -p /opt
RUN wget --no-cookies --no-check-certificate --header "Cookie: gpw_e24=http%3A%2F%2Fwww.oracle.com%2Fjava%2Fjavase%2Findex%2Ehtml" http://www.oracle.com/java/javase/index.html
RUN tar -xzf jdk-8u111-linux-arm-vfp-hflt-jdk.tar.gz -C /opt
ENV JAVA_HOME /opt/jdk1.8.0_33
ENV PATH $PATH:/opt/jdk1.8.0_33/bin
RUN ln -s /opt/jdk1.8.0_33/bin/java /usr/bin/java

#Install Flume
RUN sudo wget --no-check-certificate http://www.apache.org/dist/flume/1.6.0/apache-flume-1.6.0-bin.tar.gz
RUN sudo mv apache-flume-1.6.0-bin /flume

ADD plugins.d /flume/plugins.d
ADD flume-conf.properties /flume/conf/

#Working directory
WORKDIR /flume
```

Docker image is built by Dockerfile which is small text file. That means sharing Docker image does not require huge bandwidth (sometimes).

Getting Docker Image

- Public Docker Image Repository: Docker Hub (1)

<https://hub.docker.com/>

The screenshot shows the Docker Hub homepage with a dark blue header. On the left, there's a logo of a white ship icon, followed by 'Explore' and 'Help' links. On the right is a search bar with a magnifying glass icon and the word 'Search'. Below the header, the main slogan 'Build, Ship, & Run Any App, Anywhere' is displayed in large white and light blue text. Underneath it, a smaller line of text reads 'Dev-test pipeline automation, 100,000+ free apps, public and private registries'. To the right of the slogan, there's a 'New to Docker?' section with a 'Create your free Docker ID to get started.' link, a 'Choose a Docker Hub ID' input field, an 'Enter your email address' input field, and a 'Choose a password' input field. At the bottom right is a large blue 'Sign Up' button. A red rectangular box highlights the bottom banner, which contains the text 'Browse Thousands of the most popular software tools in the Docker Image Library'.

Getting Docker Image

- Public Docker Image Repository: Docker Hub (2)

You can easily find application image

A screenshot of the Docker Hub website showing search results for 'mysql'. The search bar at the top has 'mysql' typed into it and is highlighted with a red box. Below the search bar, the text 'Repositories (3268)' is displayed. A list of repositories is shown, with the first entry being 'mysql' by 'official'. This repository card also has a red box around it. To the right of the repository card, there are statistics: '2.2K STARS' and '10M+ PULLS'. At the bottom of the page, there is a navigation bar with 'Explore' and 'Help' links, a search bar with 'wordpress' (also highlighted with a red box), and 'Sign up' and 'Log In' buttons.

A screenshot of the Docker Hub website showing search results for 'wordpress'. The search bar at the top has 'wordpress' typed into it and is highlighted with a red box. Below the search bar, the text 'Repositories (1028)' is displayed. A list of repositories is shown, with the first entry being 'wordpress' by 'official'. This repository card also has a red box around it. To the right of the repository card, there are statistics: '950 STARS' and '5M+ PULLS'. At the bottom of the page, there is a navigation bar with 'Explore' and 'Help' links, a search bar with 'wordpress' (also highlighted with a red box), and 'Sign up' and 'Log In' buttons.

Getting Docker Image

- Be aware!

The screenshot shows the Docker Hub search interface. A red box highlights the search bar containing 'armhf'. Below it, the search results are displayed under the heading 'Repositories (2360)'. Two specific images are shown: 'container4armhf/armhf-alpine' and 'container4armhf/armhf-busybox'. The second image has a red box highlighting its name.

Repositories (41)

The screenshot shows the Docker Hub search interface again, this time with 'hypriot' in the search bar. A red box highlights the search bar. Below it, the search results are displayed under the heading 'Repositories (41)'. Two specific images are shown: 'np83/rpi2-hypriot-nodejs' and 'configman/hypriot-jenkins-slave'. The first image has a red box highlighting its name.

Getting Docker Image

- Public Docker Image Repository: Docker Hub (3)

Before using Docker image, **you should read** description.

It is very important because required option rely on image.

OFFICIAL REPOSITORY

mysql 

Last pushed: 2 days ago

Repo Info Tags

Short Description

MySQL is a widely used, open-source relational database management system (RDBMS).

Full Description

Supported tags and respective

Starting a MySQL instance is simple:

```
$ docker run --name some-mysql -e MYSQL_ROOT_PASSWORD=my-secret-pw -d mysql:tag
```

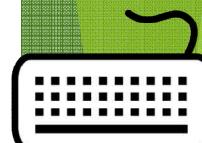
Getting Docker Image

- Public Docker Image Repository: Docker Hub (4)

OFFICIAL REPOSITORY

wordpress 

Last pushed: 2 days ago



Repo Info Tags

Short Description

The WordPress rich content management system can utilize plugins, widgets, and themes.

Full Description

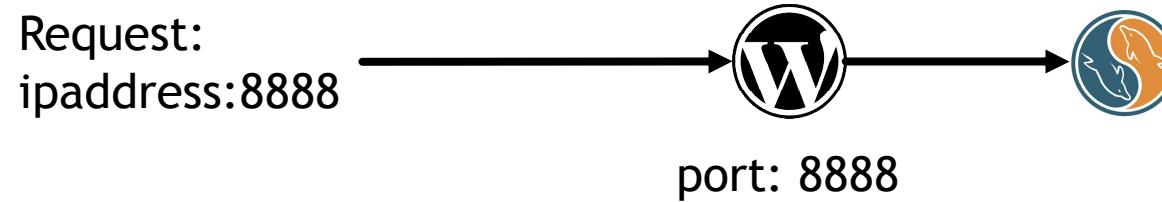
Supported tags and respective

How to use this image

```
$ docker run --name some-wordpress -link some-mysql:mysql -d wordpress
```

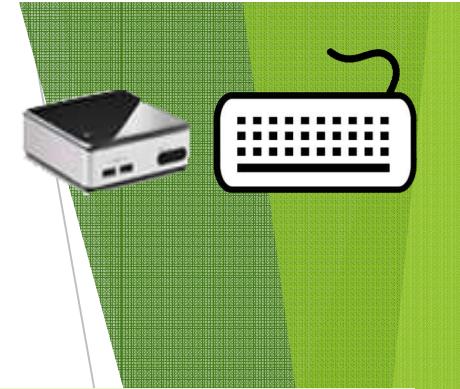
Functions Lab:

- Try: wordpress-mysql



Running Web Application

- Run mysql container



```
mkdir ~/sql  
docker run --name word_sql -v /home/[username]/sql:/var/lib/mysql -e  
MYSQL_ROOT_PASSWORD=[password] -d mysql:5.7.12
```

Tag. Default is latest (Latest version)

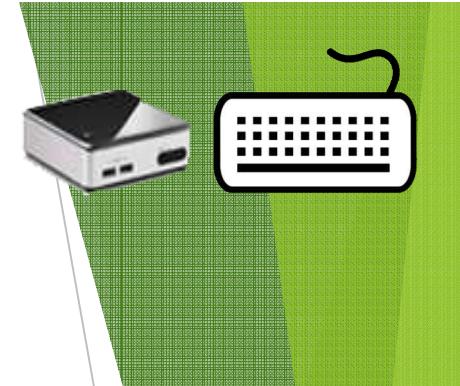
```
tein@vbox-develop:~$ docker run --name word_sql -v /home/tein/sql:/var/li  
b/mysql -e MYSQL_ROOT_PASSWORD=functions -d mysql:5.7.12  
5cba5d67f49c412ee477c8e803795d06fd477758f150678bfd526a9aeed62d7b
```

```
docker ps
```

```
tein@vbox-develop:~$ docker ps  
CONTAINER ID        IMAGE               COMMAND             CREATED            NAMES  
5cba5d67f49c        mysql:5.7.12       "docker-entrypoint.sh"   3 second          word_sql
```

Running Web Application

- Run wordpress container



```
docker run --name wordpress --link word_sql:mysql -p 8888:80 -d wordpress:4.5.1-apache
```

(Will be introduced)

```
tein@vbox-develop:~$ docker run --name wordpress --link word_sql:mysql -p 8888:80 -d wordpress:4.5.1-apache  
752dcdfdd3881a78b1bd़ee2ead1ac81837730289ac511448c19d73c696a0d743
```

```
docker ps
```

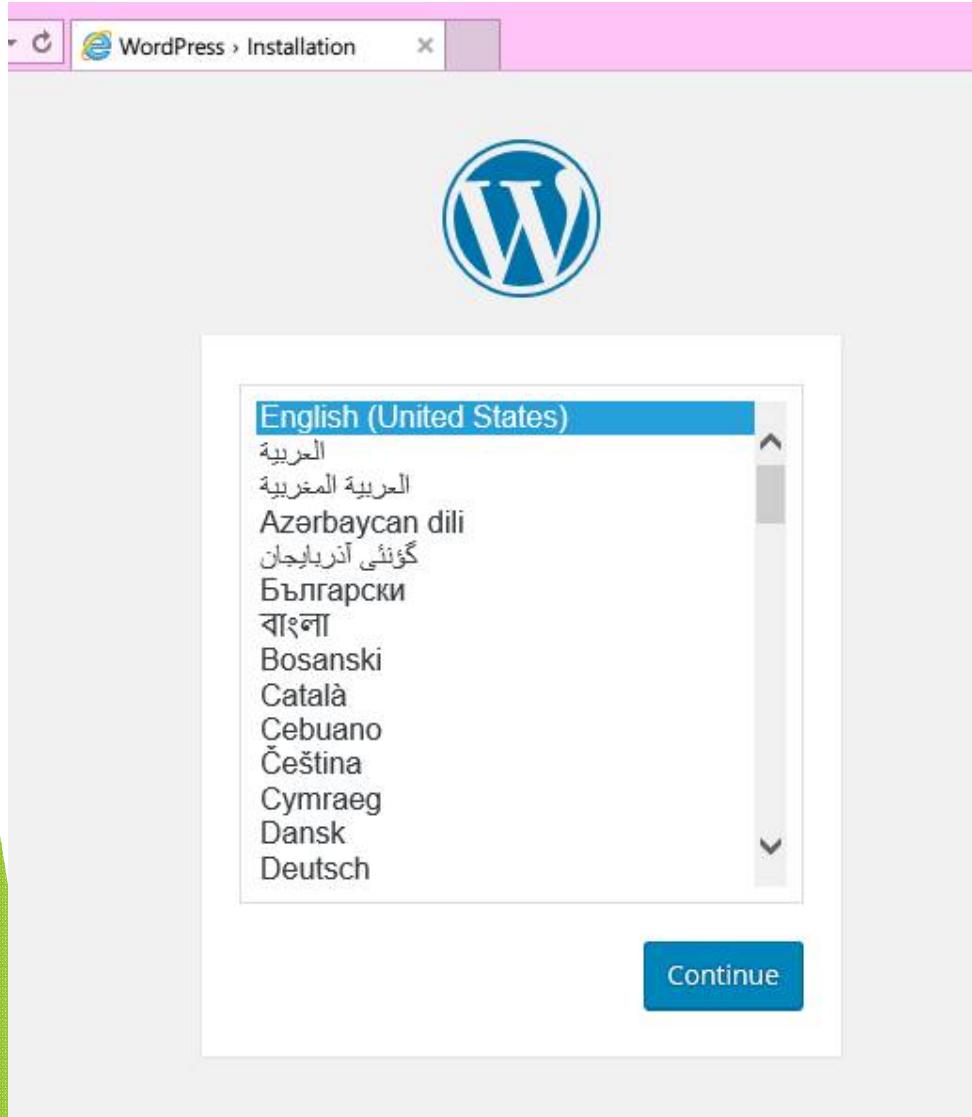
```
tein@vbox-develop:~$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CRE
STATUS	PORTS	NAMES	S
752dcdfdd3881a78b1bd़ee2ead1ac81837730289ac511448c19d73c696a0d743	wordpress:4.5.1-apache	/entrypoint.sh apache	3 s
5cba5d67f49c	mysql:5.7.12	"docker-entrypoint.sh"	5 m
inutes ago	Up 5 minutes	3306/tcp	word_sql

Running Web Application

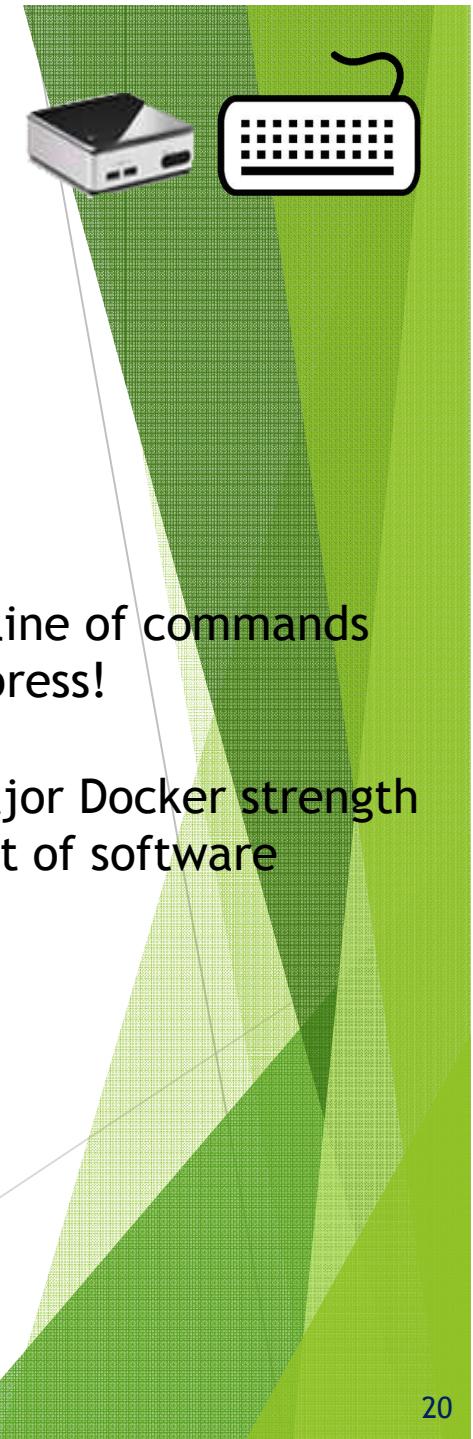
- Check wordpress!

<http://localhost:8888>



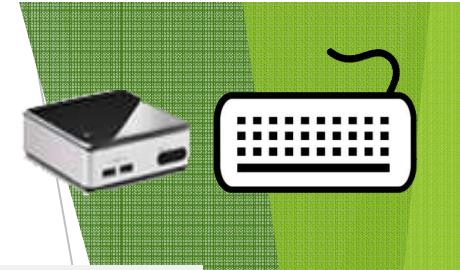
We typed just 2 line of commands
to running wordpress!

This is one of major Docker strength
:Easy deployment of software



Running Web Application

- Default configuration: Wordpress



WordPress

Gaeilge
Galego
هزاره گی
עברית
हिन्दी
Hrvatski
Magyar
Հայերեն
Bahasa Indonesia
Íslenska
Italiano
日本語
ქართული
한국어

계속

black

다른 워드프레스 사이트

WordPress

환영합니다

유명한 5분 워드프레스 설치 과정에 오신 것을 환영합니다! 아래에서 정보를 입력한 하면 세계에서 가장 확장성 높고 강력한 개인 출판 플랫폼을 사용하는 길로 들어서게 됩니다.

필요한 정보

다음 정보들을 제공해주세요. 나중에 다시 변경할 수 있으니 걱정하지 않아도 됩니다.

사이트 제목

사용자명

비밀번호
매우 약함

중요: 로그인하려면 이 비밀번호가 필요할 것입니다. 안전한 곳에 보관하십시오.

비밀번호 확인 약한 패스워드 사용 확인

이메일 주소:

계속하기 전에 이메일 주소를 한 번 더 확인하세요.

안녕하세요!

Running Web Application

- About --link option (1)

(In official docs..)

Docker also has a linking system that allows you to **link multiple containers together** and **send connection information from one to another**. When containers are linked, **information about a source container can be sent to a recipient container**. This allows the recipient to see selected data describing aspects of the source container.

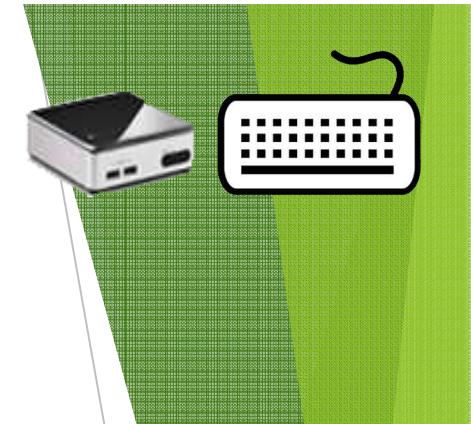
Links allow containers to discover each other and securely transfer information about one container to another container. When you set up a link, you create a conduit between a source container and a recipient container.

Usage: `--link <name or id>:alias`
`--link <name or id>`

Naming container is important!

Running Web Application

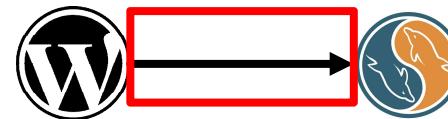
- About --link option (2)



```
sudo docker inspect -f "{{ .HostConfig.Links }}" wordpress
```

```
tein@vbox-develop:~$ docker inspect -f "{{ .HostConfig.Links }}" wordpress
[/word_sql:/wordpress/mysql]
```

You can see wordpress container is linked with mysql container.



Conduit which is made by --link option

Running Web Application

- About --link option (3)

(In official docs..)

Docker **creates a secure tunnel** between the containers that **doesn't need to expose any ports externally on the container**.

That's a **big benefit** of linking: **we don't need to expose the source container.**

Review our commands: Creating mysql container

→ This container doesn't expose any ports.

```
mkdir sql  
sudo docker run --name word_sql -v /home/[username]/sql:/var/lib/mysql -e  
MYSQL_ROOT_PASSWORD=[password] -d mysql
```

Also we don't need to type password of mysql when creating wordpress container

```
sudo docker run --name wordpress --link word_sql:mysql -p 80:80 -d  
wordpress
```

Running Web Application

- About --link option (4)

(In official docs..)

Functionality of this option is:

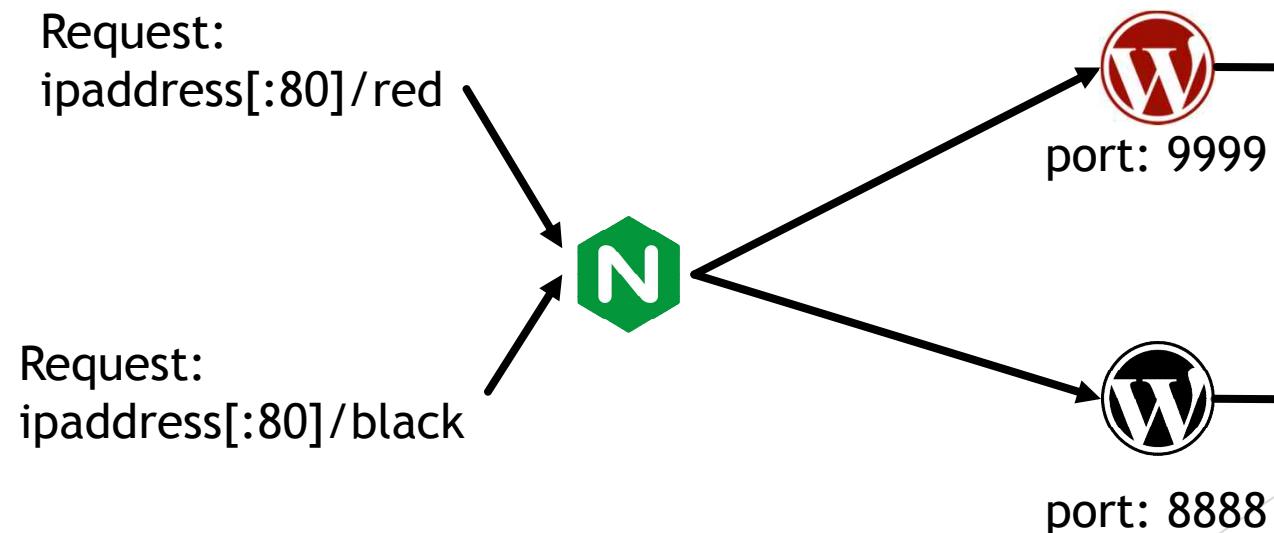
- Updating Environment variables
- Updating the /etc/hosts file

```
tein@vbox-develop:~$ docker run -it --name=container1 ubuntu /bin/bash
```

```
tein@vbox-develop:~$ docker run -it --name=container2 --link=container1:c ubuntu /bin/bash
root@9163eaff98f3:/# cat /etc/hosts
172.17.0.7      9163eaff98f3
127.0.0.1      localhost
::1      localhost ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
172.17.0.6      c 6b4246267806 container1
root@9163eaff98f3:/# ping c
PING c (172.17.0.6) 56(84) bytes of data.
64 bytes from c (172.17.0.6): icmp_seq=1 ttl=64 time=0.115 ms
64 bytes from c (172.17.0.6): icmp_seq=2 ttl=64 time=0.050 ms
^C
--- c ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 999ms
rtt min/avg/max/mdev = 0.050/0.082/0.115/0.033 ms
root@9163eaff98f3:/# ping container1
PING c (172.17.0.6) 56(84) bytes of data.
64 bytes from c (172.17.0.6): icmp_seq=1 ttl=64 time=0.072 ms
64 bytes from c (172.17.0.6): icmp_seq=2 ttl=64 time=0.051 ms
```

Functions Lab:

- Try: nginx-wordpress-mysql



Running Web Application

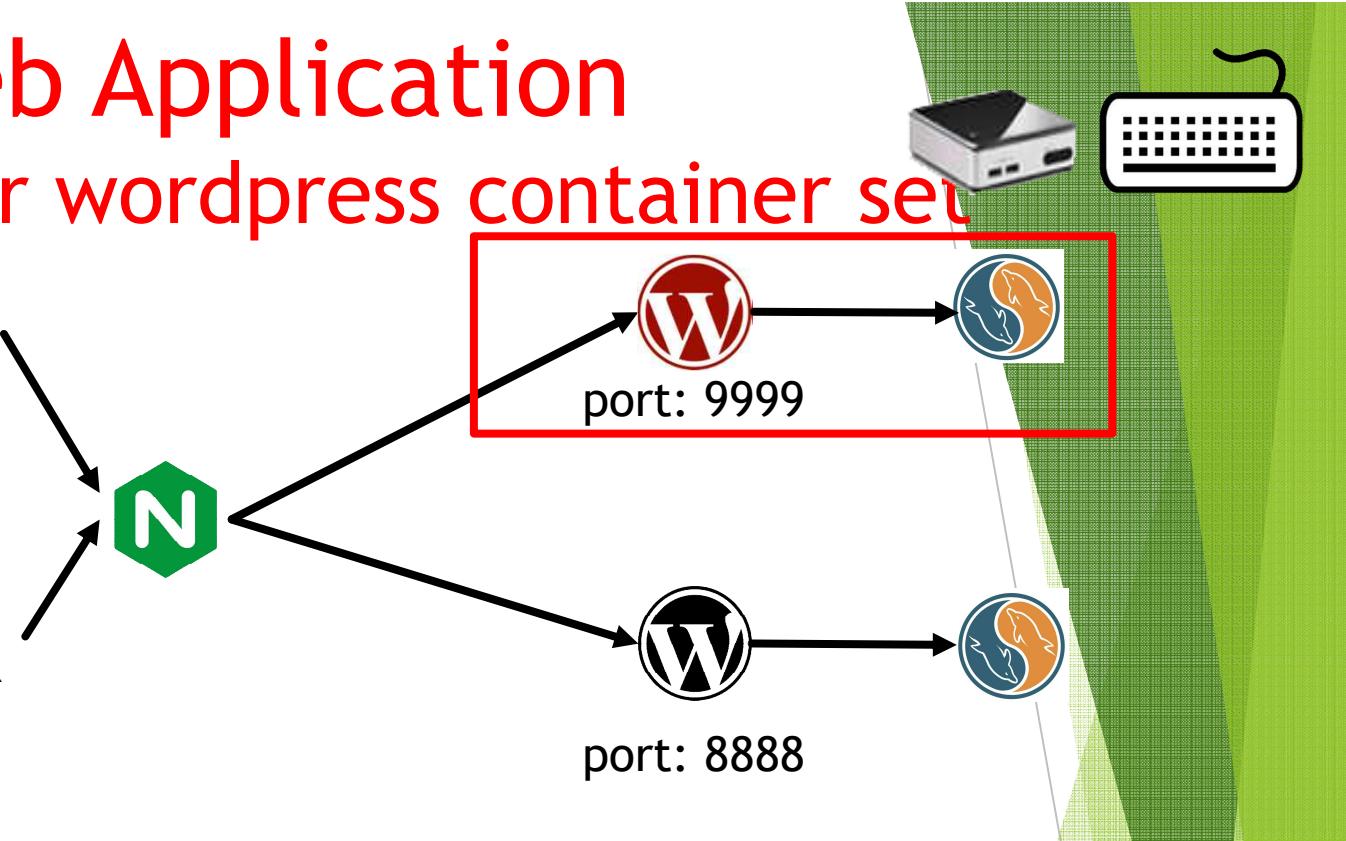
- Make another wordpress container set

Request:

ipaddress[:80]/red

Request:

ipaddress[:80]/black



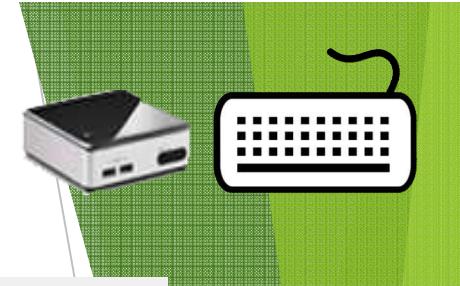
```
mkdir ~/sql2
```

```
docker run --name word_sql2 -v /home/[username]/sql2:/var/lib/mysql -e  
MYSQL_ROOT_PASSWORD=[password] -d mysql:5.7.12
```

```
docker run --name wordpress2 --link word_sql2:mysql -p 9999:80 -d  
wordpress:4.5.1-apache
```

Running Web Application

- Default configuration: Wordpress



red

다른 워드프레스 사이트

환영합니다

유명한 5분 워드프레스 설치 과정에 오신 것을 환영합니다! 아래에서 정보를 입력만 하면 세계에서 가장 확장성 높고 강력한 개인 출판 플랫폼을 사용하는 길로 들어서게 됩니다.

필요한 정보

다음 정보들을 제공해주세요. 나중에 다시 변경할 수 있으니 걱정하지 않아도 됩니다.

사이트 제목: red

사용자명: cslab

사용자명은 알파벳, 숫자, 스페이스, 밑줄, 하이픈, 마침표, @, 심볼만 가능합니다.

비밀번호: function

매우 약함

숨기기

중요: 로그인하려면 이 비밀번호가 필요할 것입니다. 안전한 곳에 보관하십시오.

비밀번호 확인: 약한 패스워드 사용 확인

이메일 주소: cslab@functions.com

계속하기 전에 이메일 주소를 한 번 더 확인하세요.

검색 엔진 접근 여부: 검색 엔진이 이 사이트 검색 차단하기
이 요청이 받아들여지는 것은 전적으로 검색 엔진에 좌우됩니다.

워드프레스 설치하기

안녕하세요!

Running Web Application

- Current status

black

다른 워드프레스 사이트

ip:8888

안녕하세요!

red

다른 워드프레스 사이트

ip:9999

안녕하세요!

Running Web Application

- Make container for reverse proxy

```
docker run -it --name=nginx -p 80:80 ubuntu:14.04
```

Forwarding host_port:container_port



Running Web Application

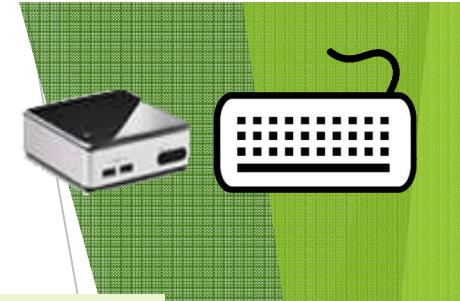
- Setting reverse proxy

```
apt-get update  
apt-get install nginx  
apt-get install vim
```

```
cd /etc/nginx/sites-enabled  
rm default  
vi default
```

```
server {  
    listen 80;  
    location /black {  
        proxy_pass http:// [your ip] :8888/;  
    }  
    location /red {  
        proxy_pass http:// [your ip] :9999/;  
    }  
}
```

```
service nginx start
```



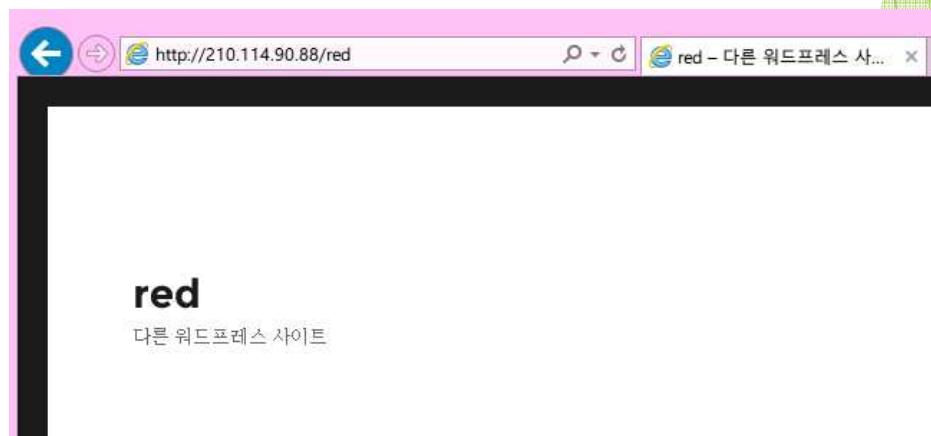
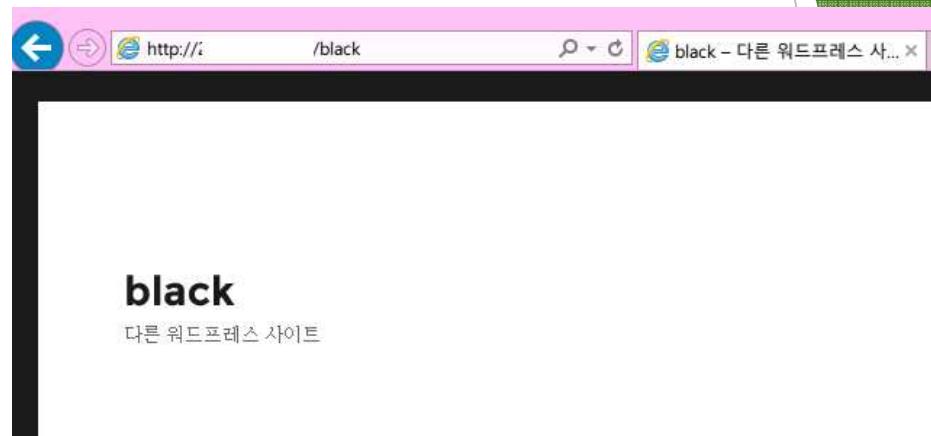
Running Web Application

- Setting reverse proxy

Try

<http://localhost/black>

<http://localhost/red>



Appendix

Control Tower: Docker Private Registry

Docker Private Registry

- About

In previous slides, Docker image is operated and maintained like source code (e.g svn, git...)

Also we used Docker public repository, Docker Hub. (which looks like github)

Surely, Docker also provides private repository: Registry.

OFFICIAL REPOSITORY

[registry](#) ☆

Last pushed: 19 days ago

[Repo Info](#) [Tags](#)

Short Description

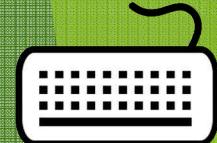
Containerized docker registry

Docker Pull Command

`docker pull registry`

Docker Private Registry

- How to use (1)



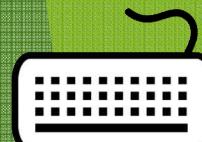
Deployment(**In control tower**):

```
docker run -d -p 5000:5000 --restart=always --name registry registry:2
```

--restart=always means this container is already running even Docker host rebooted.

Docker Private Registry

- How to use (2)



You may configure insecure-registry options on NUC and Raspberry Pi
(Basic registry runs on insecure mode.)

Setting(**In NUC and RPi**):

```
sudo vi /etc/default/docker
```

```
DOCKER_OPTS="--insecure-registry [control tower ip]:5000"
```

```
# If you need Docker to use an HTTP proxy, it can also be
# here.
#export http_proxy="http://127.0.0.1:3128/"

# This is also a handy place to tweak where Docker's temporary
# files go.
#export TMPDIR="/mnt/bigdrive/docker-tmp"
#export DOCKER_OPTS="--insecure-registry :5000"
```

If any DOCKER_OPTS is already been, just add this phrase to end of OPTS.

```
sudo service docker restart
```

Docker Private Registry

- How to push and pull image (1)

In this time, we make a image of nginx container and push to registry

Making container image and push to registry ([In NUC](#)):

```
docker commit nginx  
docker images
```

```
tein@vbox-develop:~$ docker commit nginx  
a67e74d9e4ee5dc194d7449f964e0c093643b74a0fbbebbee4d805dac34219294  
tein@vbox-develop:~$ docker images  
REPOSITORY          TAG      IMAGE ID  
CREATED             VIRTUAL SIZE  
<none>              <none>   a67e74d9e4ee  
8 seconds ago        248.3 MB  
nginx               latest   9f55a676b5c2
```

Docker Private Registry

- How to push and pull image (2)

Tagging image for pushing and push image

```
docker tag nginx [control tower ip]:5000/nginx  
docker push [control tower ip]:5000/nginx
```

```
tein@vbox-develop:~$ docker tag nginx :5000/nginx  
tein@vbox-develop:~$ docker push :5000/nginx  
The push refers to a repository [ :5000/nginx] (len:  
9f55a676b5c2: Pushed  
35a2943903f2: Pushed  
:5000/nginx]  
:5000/nginx] (len:  
:5000/nginx) (len:
```

Check!

```
curl [control tower ip]:5000/v2/_catalog
```

```
tein@vbox-develop:~$ curl :5000/v2/_catalog  
{"repositories":["arm_flume_vis_agent","arm_node_example","kafka","ka  
fka_origin","kafka_server","mktopic","nginx","u","vis2016","zookeeper  
"]}
```

You can find!

Docker Private Registry

- How to push and pull image (3)

Pulling(**In other machines**):

```
docker pull [control tower ip]:5000/nginx
```

```
bjj@orche:~$ docker pull :      :5000/nginx
Using default tag: latest
latest: Pulling from nginx
efd26ecc9548: Extracting 36.18 MB/51.34 MB
a3ed95caeb02: Download complete
a48df1751a97: Download complete
8ddc2d7beb91: Download complete
```

Thank You for
Your Attention
Any Questions?

