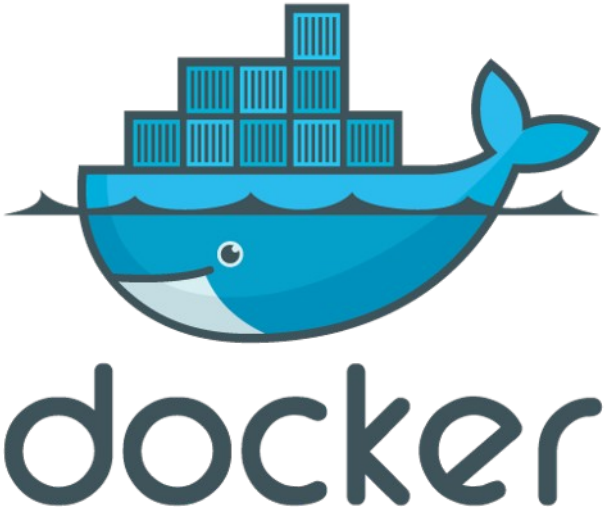


Workshop

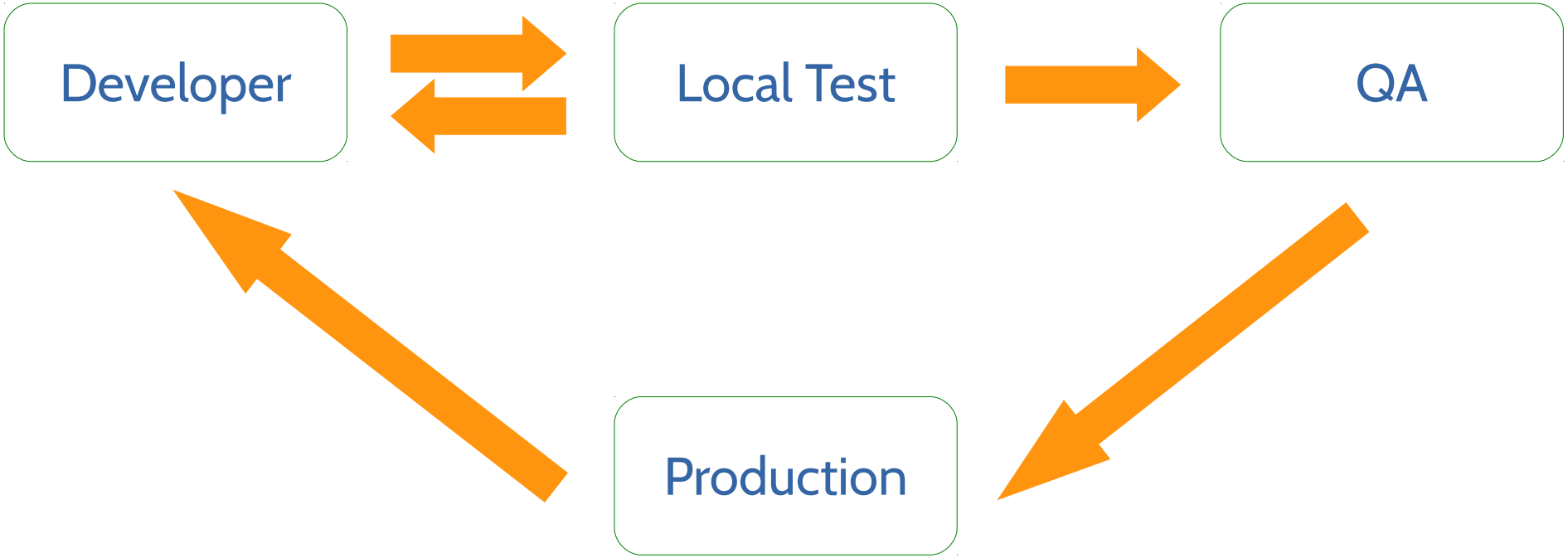
Jirayut Nimsaeng (Dear)

Docker Workshop for beginner
February 20, 2016 @ Hangar DTAC

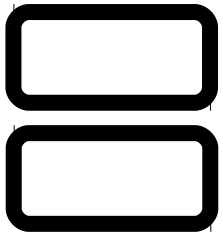
#whoami

- **Jirayut Nimsaeng (Dear)**
- The Builder at Kaidee.com
- Interested in Cloud and Open Source Technology
- Agile Practitioner and ScrumMaster with DevOps Driven Development

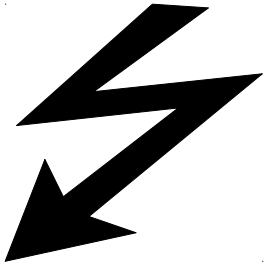
Developer Problems



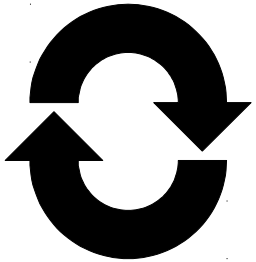
What Developer needs



Production-like

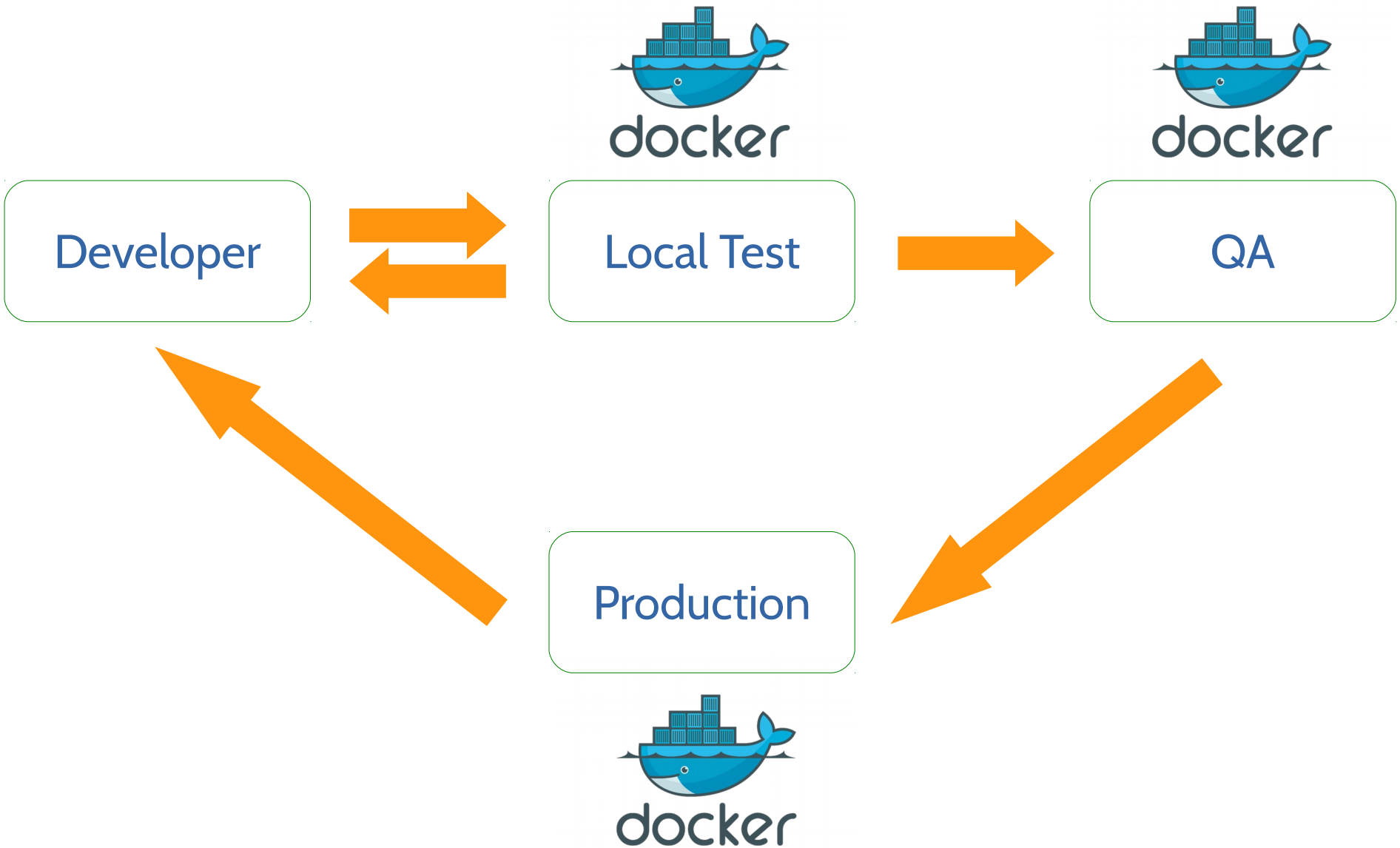


Quick



Repeatable

Docker can solve this problem



Docker can

- Deploy (almost) everything

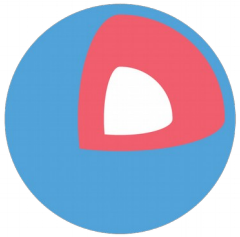
WordPress

Docker can

- Deploy (almost) everywhere

Native

ubuntu

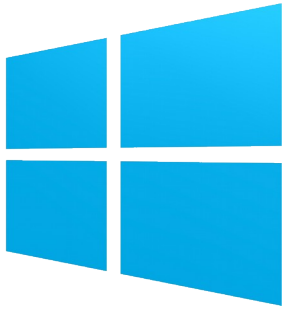


CoreOS

CentOS

redhat LINUX

On VMs



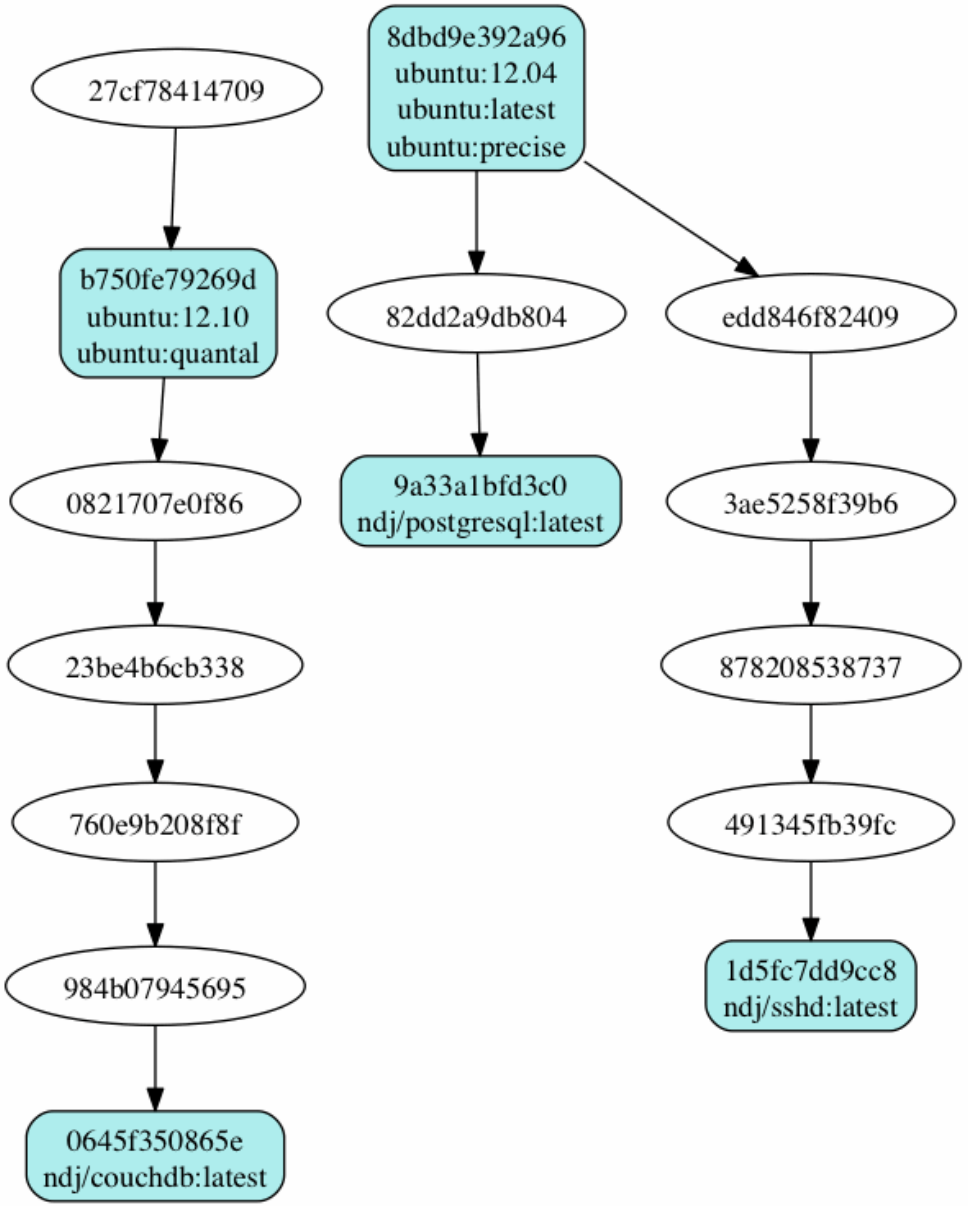
What is Docker?

- Platform to help **code**, **test** and **deploy** applications
- Combine with
  - Lightweight container virtualization
  - Work flows
  - Tooling

Docker image

- Docker images are read-only templates
- Each image consists of series of layers
- Docker use union file system to combine layers into single image
- Every image starts from base image

Docker image layers



Docker container

- Docker containers launched from Docker image
- Container consists of
  - Operating system
  - User-added files
  - Meta-data
- When Docker container runs, it adds a read-write layer on top of the image

Image vs Container

Docker Image is a **class**

Docker Container is a **instance of class**

Dockerfile

- Dockerfile is instructions to build Docker image
  - How to run commands
  - Add files or directories
  - Create environment variables
  - What process to run when launching container
- Result from building Dockerfile is Docker image

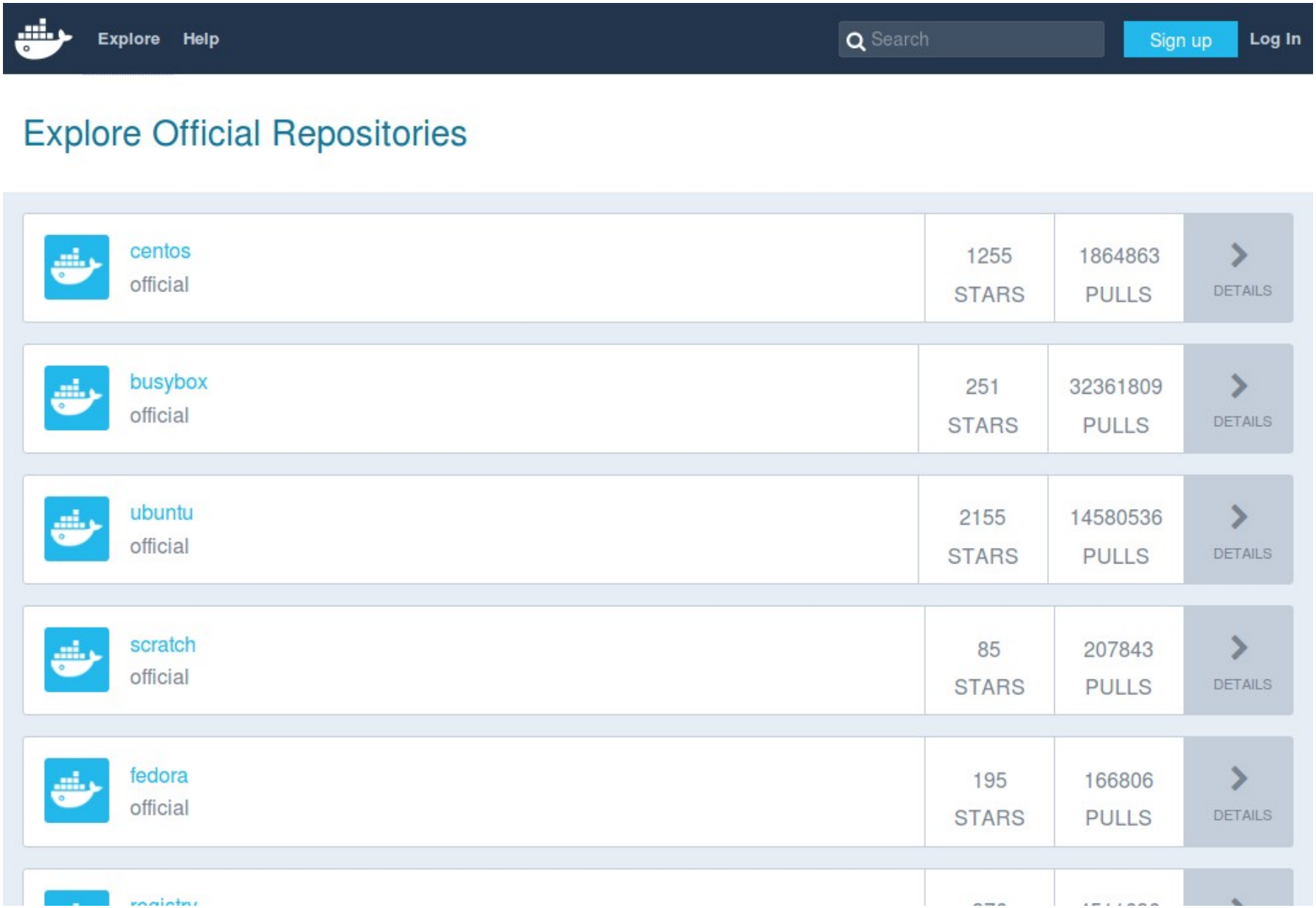
Sample Dockerfile

```
FROM ubuntu:14.04
MAINTAINER Jirayut Nimsaeng <w [at] winginfotech.net>
ADD build-files /build-files
RUN apt-get update
RUN apt-get install -y openssh-server vim tmux rsync byobu
RUN mkdir /var/run/sshd
RUN sed -i 's/required     pam_loginuid.so/optional
pam_loginuid.so/g' /etc/pam.d/sshd
CMD /start.sh
EXPOSE 22
```

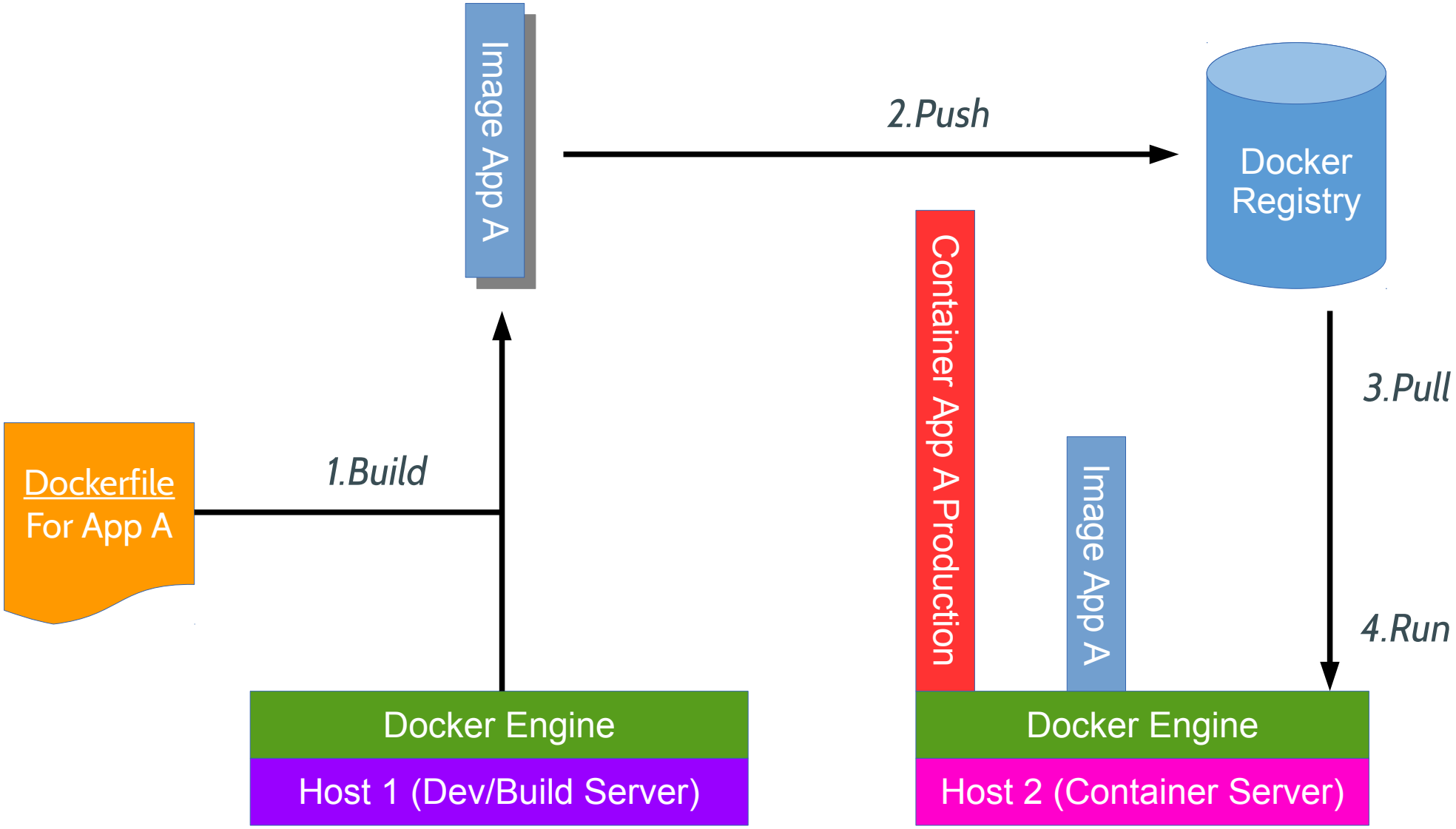
Docker Distribution

- Docker Distribution previously named Registry
- Docker Distribution is the store for Docker image
- Docker Hub is public Docker Distribution like Github
- Using Docker client to push and pull Docker image from Docker Distribution
- You can create your own Docker Distribution

Docker Hub


Explore
Help
Search
Sign up
Log In
Explore Official Repositories
centos
official
1255
STARS
1864863
PULLS
DETAILS
busybox
official
251
STARS
32361809
PULLS
DETAILS
ubuntu
official
2155
STARS
14580536
PULLS
DETAILS
scratch
official
85
STARS
207843
PULLS
DETAILS
fedora
official
195
STARS
166806
PULLS
DETAILS

Docker workflows



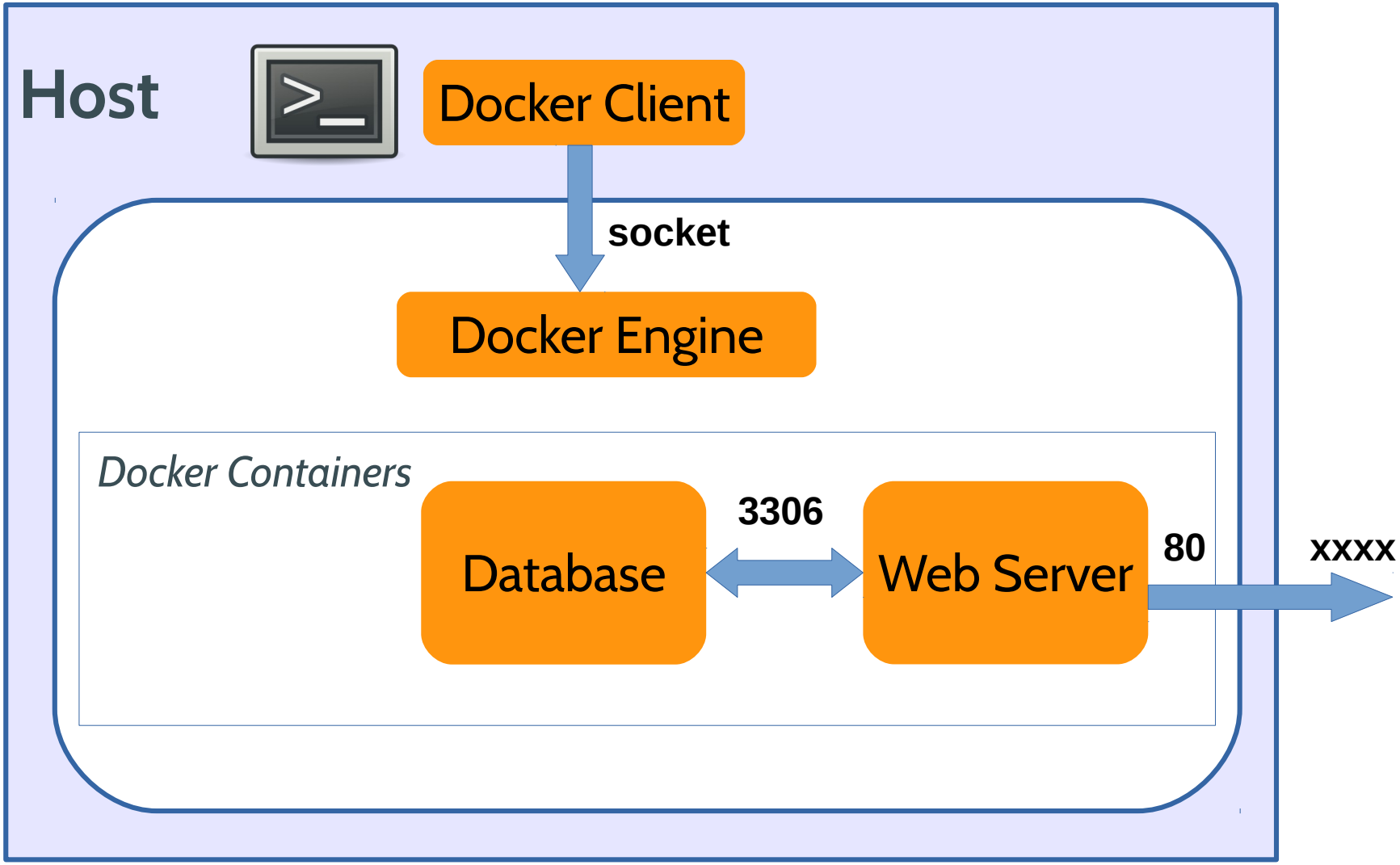
Docker Installation

- Docker Toolbox for Mac and Windows
  - https://www.docker.com/toolbox
  - All-in-one Docker installation
    - Docker Engine
    - Docker Machine
    - Docker Compose
    - Docker Kitematic
    - VirtualBox

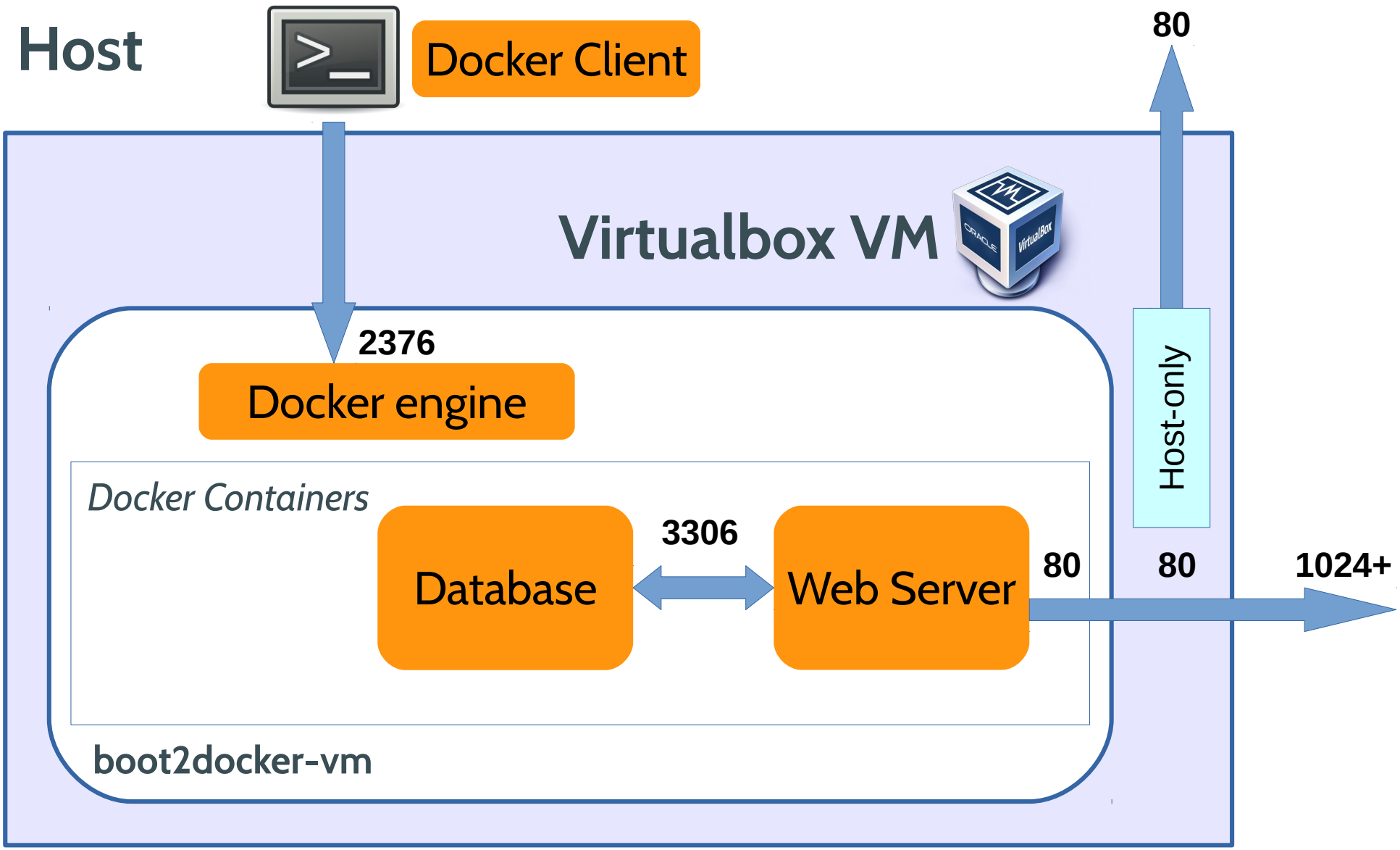
Docker installation

- Ubuntu based
  - https://docs.docker.com/installation/ubuntulinux/
  - Recommend Ubuntu 14.04 64-bit LTS or up
  - curl -sSL https://get.docker.com/ | sudo sh
- Redhat based
  - https://docs.docker.com/installation/centos/
  - Recommend CentOS 7
  - curl -sSL https://get.docker.com/ | sh

Docker architecture

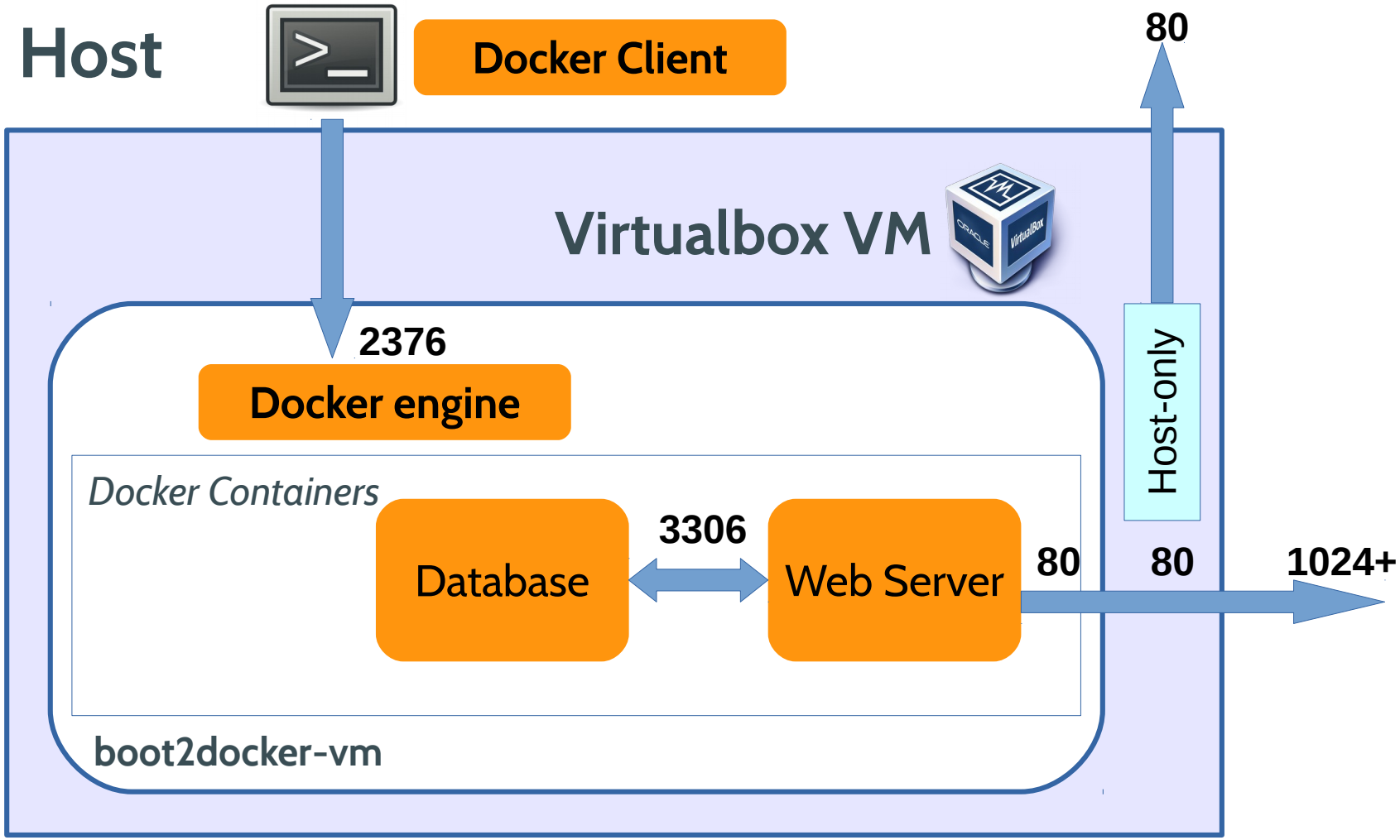


Boot2docker architecture



Know the tools (1)

- Docker Client / Engine



Know the tools (2)

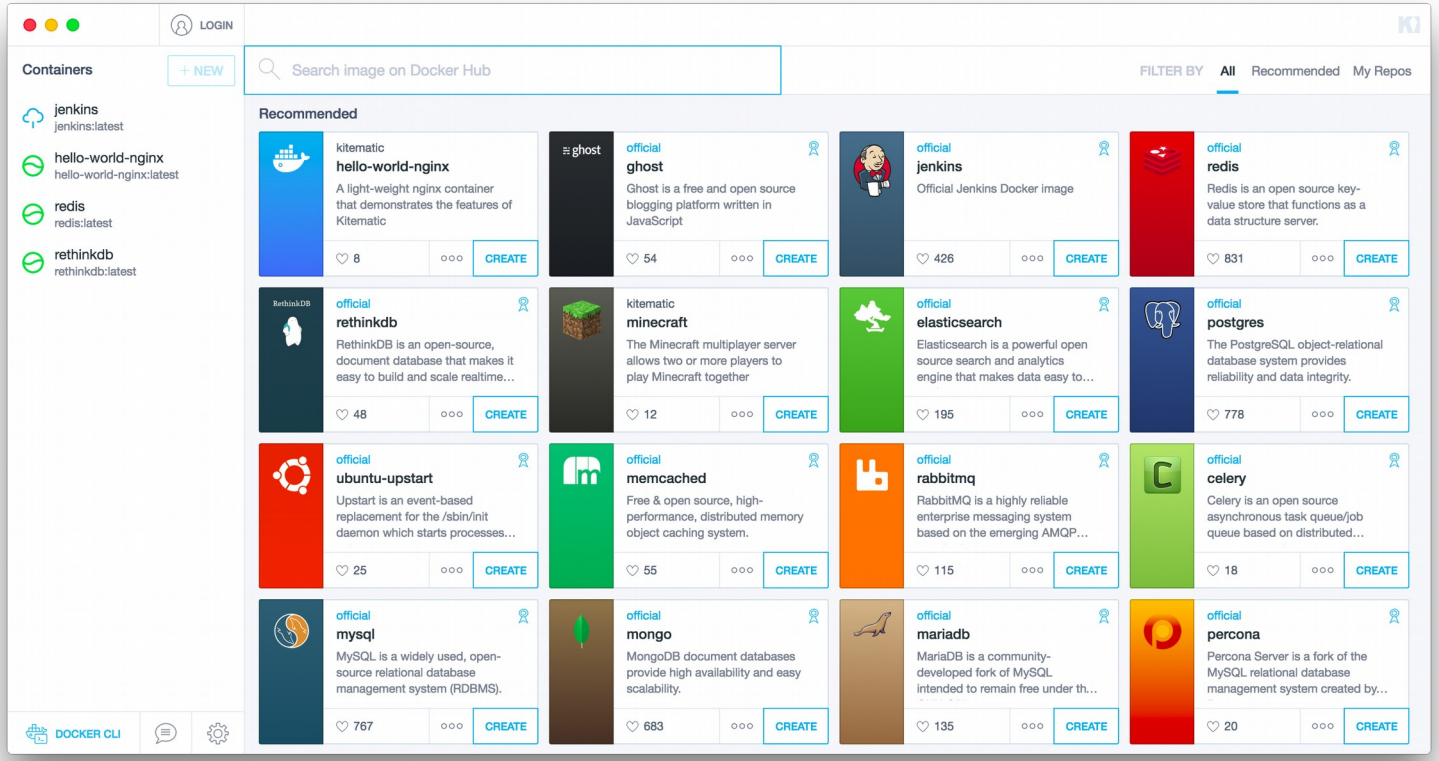
- Docker Machine
  - Lets you create Docker hosts on your computer, on cloud providers, or inside your own data center
  - Automated these steps
    - Create Docker host
    - Install Docker
    - Configure Docker client to talk with server
  - Manage Docker multiple Docker host

Know the tools (3)

- VirtualBox
  - Virtualization software to run Docker host for Mac and Windows
  - VM has been configured and managed by Docker Machine

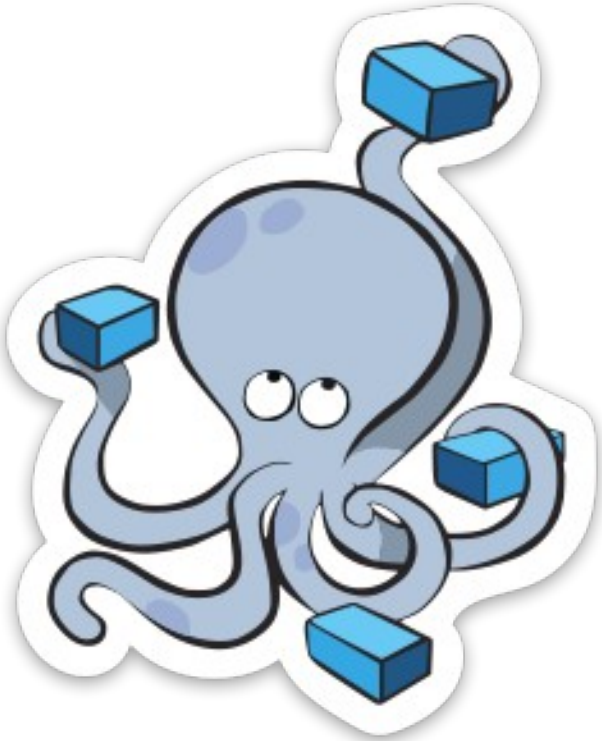
Know the tools (4)

- Docker Kitematic
  - Simple application for managing Docker containers on Mac and Windows



Know the tools (5)

- Docker Compose
  - Tool for defining and running multi-container applications with Docker in a single file



Docker Image name

- Official Docker Image
  - ubuntu:latest
  - centos:centos7
- User's Docker Image on Docker Hub
  - google/cadvisor:0.5.0
  - dockerfile/mongodb
- Docker Image on Private Docker Registry
  - r.winginfotech.net/ubuntu:14.10
  - r:5000/docker-registry

Run first Docker container

- *docker images*
- *docker pull r.winginfotech.net/ubuntu*
- *docker images*
- *docker run r.winginfotech.net/ubuntu echo "Hello World"*
- *docker run -i -t r.winginfotech.net/ubuntu bash*
  - *whoami*
  - *hostname*
  - *cat /etc/*release**
  - *exit*

Docker basic operations

- docker pull *[name[:tag]]*
  - docker pull centos
  - docker pull ubuntu:latest
- docker run [-itd] *[name[:tag]] [command]*
- docker ps
- docker ps -a
- docker rm *[name or cid]*
- docker rm *[part of cid]*
- docker images
- docker rmi *[name:tag or iid]*

Image name and tag

- docker pull r.winginfotech.net/ubuntu
- docker images
- docker pull r.winginfotech.net/ubuntu:15.10
- docker images
- docker pull r.winginfotech.net/ubuntu:14.04
- docker images

Create your first image

- docker run -it r.winginfotech.net/ubuntu bash
  - vim
  - echo 'Acquire::http::Proxy "http://192.168.30.147:3142";' > /etc/apt/apt.conf.d/11proxy
  - apt-get update
  - apt-get install -y vim
  - touch vim-installed
  - ls
  - exit
- docker ps -a
- docker commit [cid] ubuntu-vim
- docker images
- docker run -it ubuntu-vim bash
  - ls

Expose ports

- docker run -it -p 80:80 ubuntu-vim bash
  - apt-get install -y apache2
  - service apache2 start
  - Go to browser: http://ipaddress
  - exit
- Commit your apache2 container as **ubuntu-apache2** with tag **14.04** and **latest**
- Make sure that new images have apache2
- Clear your stopped containers

Run as daemon & expose port option

- docker run ubuntu-apache2
- docker run -d ubuntu-apache2 service apache2 start
- docker run -d ubuntu-apache2 apachectl -DFOREGROUND
- docker run -d -p 80:80 ubuntu-apache2 apachectl -DFOREGROUND
- docker run -d -p 8880:80 ubuntu-apache2 apachectl -DFOREGROUND
- docker run -d -p 80 ubuntu-apache2 apachectl -DFOREGROUND
- docker ps

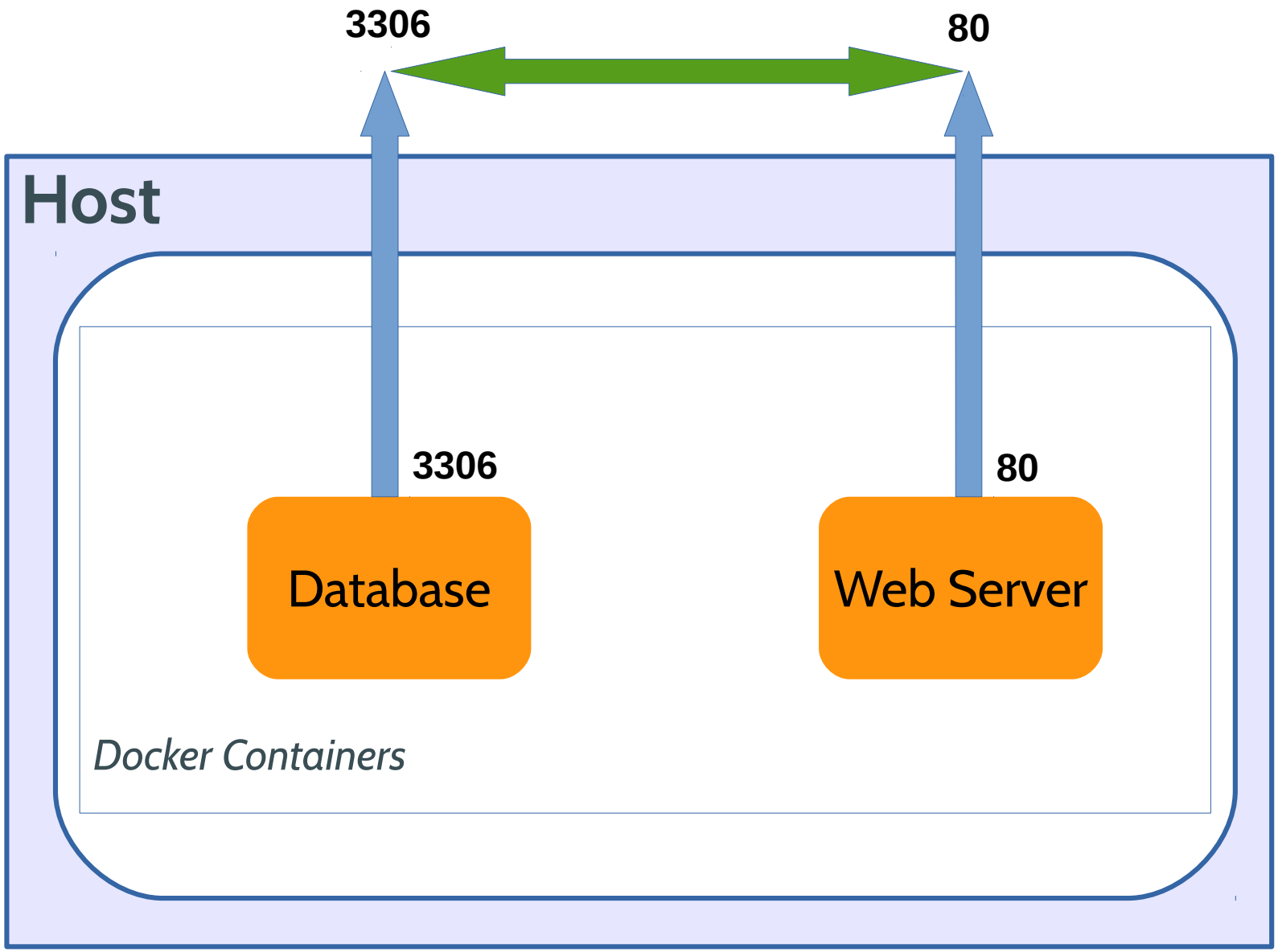
Docker container operation

- docker ps
- docker stop *[container id or name]*
- docker start *[container id or name]*
- docker kill *[container id or name]*
- docker logs *[container id or name]*
- docker diff *[container id or name]*
- docker top *[container id or name]*
- docker inspect *[container id or name]*

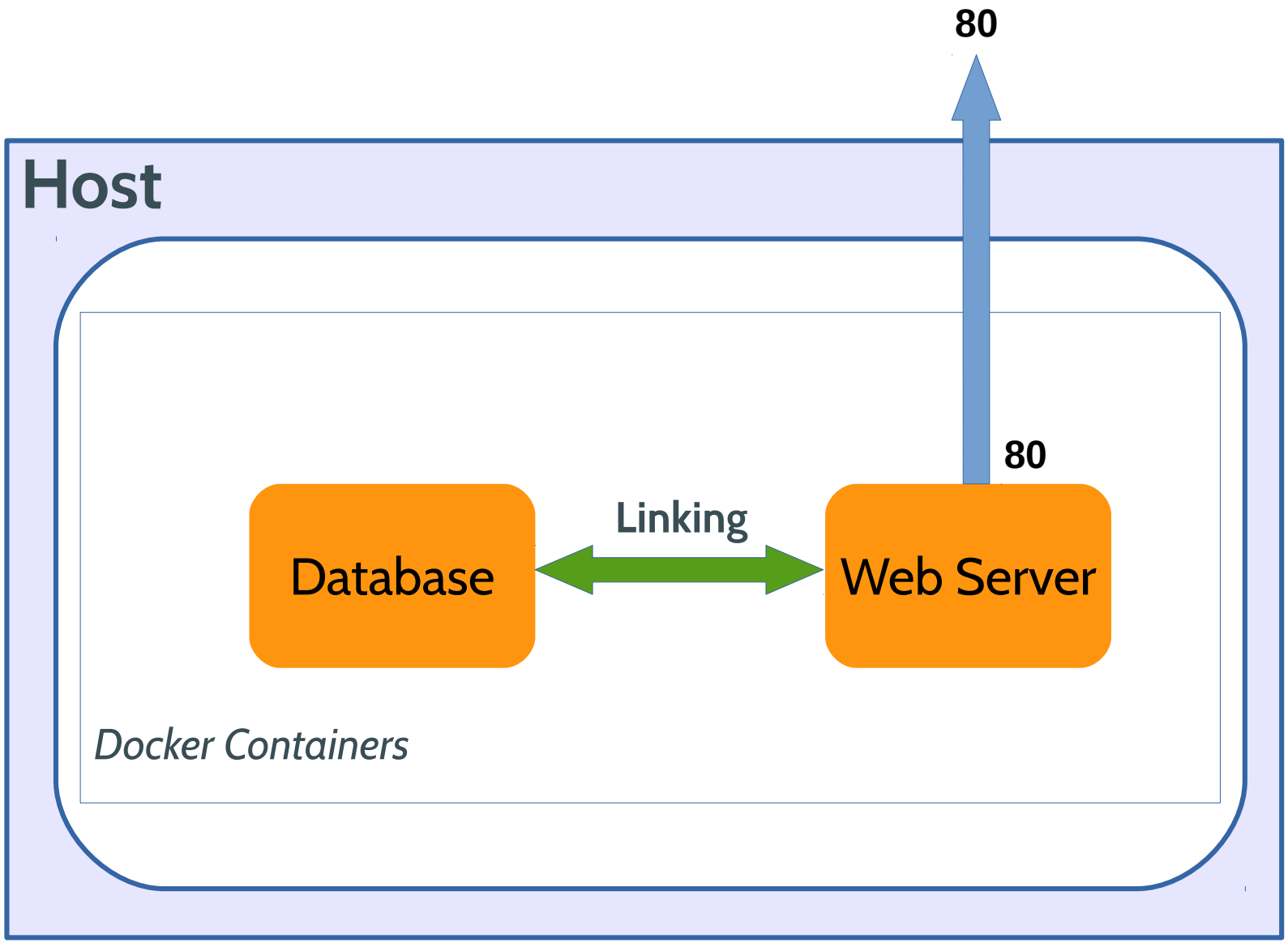
Some useful command & parameter

- *docker run --name my-nginx -d -p 80:80 r.winginfotech.net/nginx*
- *docker ps*
- *docker exec -it my-nginx /bin/bash*

Linking



Linking



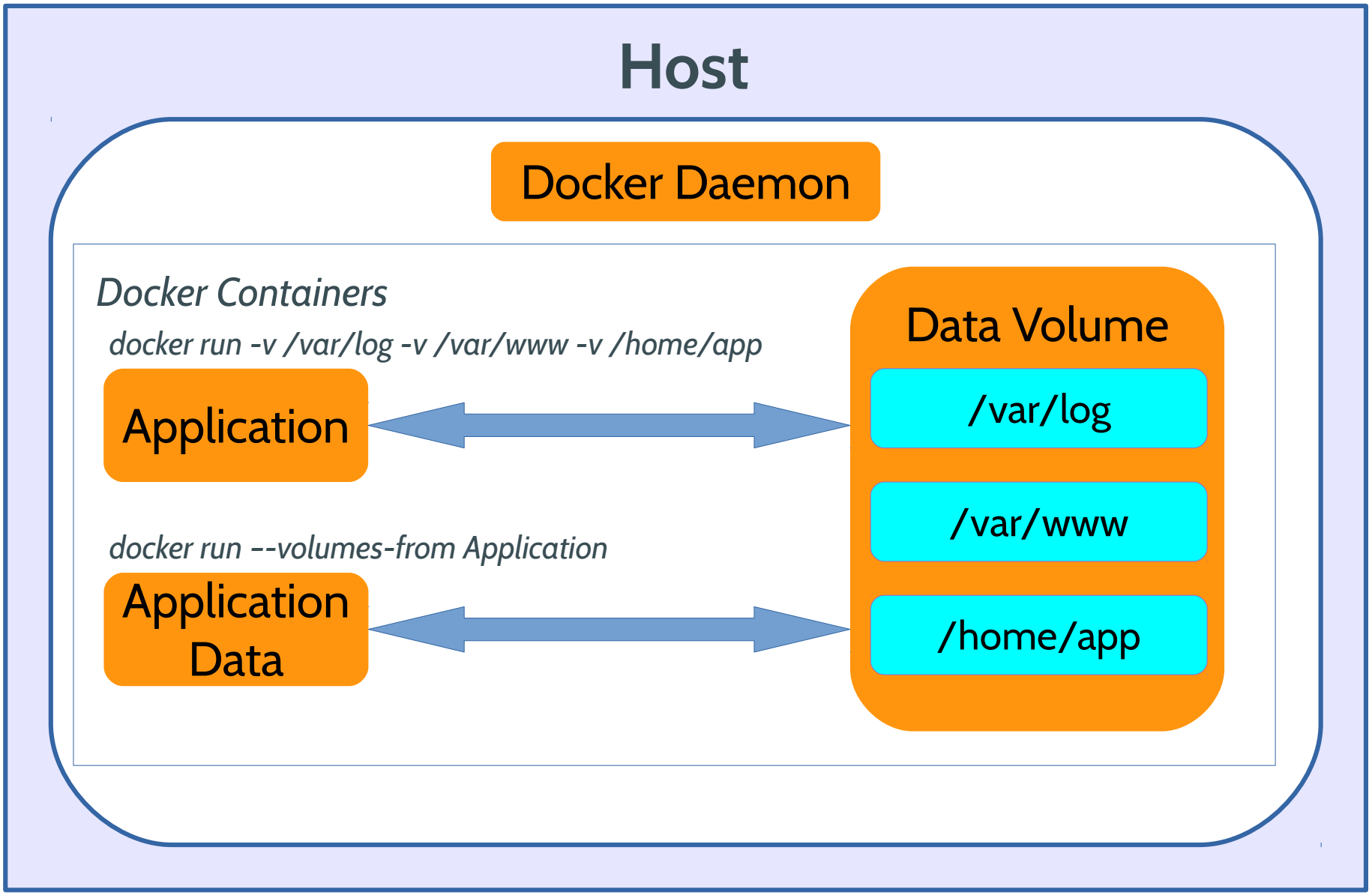
Running Wordpress without linking

- *docker run -d --name wp_mysql -p 3306:3306 *
  *-e MYSQL_ROOT_PASSWORD=mypass *
  r.winginfotech.net/mysql
- *docker run -d --name wp -p 80:80 *
  *-e WORDPRESS_DB_PASSWORD=mypass *
  *-e WORDPRESS_DB_HOST=your-ip-address:3306 *
  r.winginfotech.net/wordpress

Running Wordpress with linking

- *docker run -d --name wp_mysql *
  *-e MYSQL_ROOT_PASSWORD=mypass *
  r.winginfotech.net/mysql
- *docker run -d --name wp -p 80:80 *
  *-e WORDPRESS_DB_PASSWORD=mypass *
  ***--link wp_mysql:mysql** *
  r.winginfotech.net/wordpress

Docker data volume container



Running Wordpress with volume

- *docker run -d --name wp_mysql *
  *-e MYSQL_ROOT_PASSWORD=mypass *
  r.winginfotech.net/mysql
- *docker run -d --name wp -p 80:80 *
  *-e WORDPRESS_DB_PASSWORD=mypass *
  *--link wp_mysql:mysql *
  *--volumes $(pwd)/uploads:/var/www/html/wp-content/uploads *
  r.winginfotech.net/wordpress

Docker Compose

- Create *docker-compose.yml* file

```
wordpress:
  image: r.winginfotech.net/wordpress
  ports:
   - "80:80"
  links:
   - db:mysql
db:
  image: r.winginfotech.net/mysql
  environment:
    MYSQL_ROOT_PASSWORD: mypass
```

- *docker-compose up*

Play with Docker Compose

- docker-compose up
- docker-compose start
- docker-compose ps
- docker-compose stop
- docker-compose up -d
- docker-compose rm