

WORKSHOP AUTOMATE SOFTWARE DEVELOPMENT

ด้วย

Docker, Git, Phpunit, nginx, Jenkins

Step-By-Step ตั้งแต่ Setup ไปจน Production



docker



git



PHPUnit

NGINX



Jenkins



PRESENT

WORKSHOP AUTOMATE SOFTWARE DEVELOPMENT

ด้วย

Docker, Git, PHPUnit, nginx, Jenkins



วิทยากร : คุณ ไชยพงศ์ ลากเลี้ยงตระกูล

CTO/CEO 3DS Interactive



docker



git



PHPUnit



NGINX



Jenkins



PRESENT

WORKSHOP AUTOMATE SOFTWARE DEVELOPMENT

ด้วย

Docker, Git, PHPUnit, nginx, Jenkins



วิทยากร : คุณ Praparn Lungpoonlarp

Infrastructure Engineer, Network Engineer, System Engineer



docker



git



PHPUnit



NGINX



Jenkins



PRESENT

WORKSHOP AUTOMATE SOFTWARE DEVELOPMENT

ด้วย

Docker, Git, PHPUnit, nginx, Jenkins



วิทยากร : คุณ บาร์บ เก็จธันย์

ตำแหน่ง โปรแกรมเมอร์ บริษัท ไดมคลาวด์



docker



git



PHPUnit

NGINX



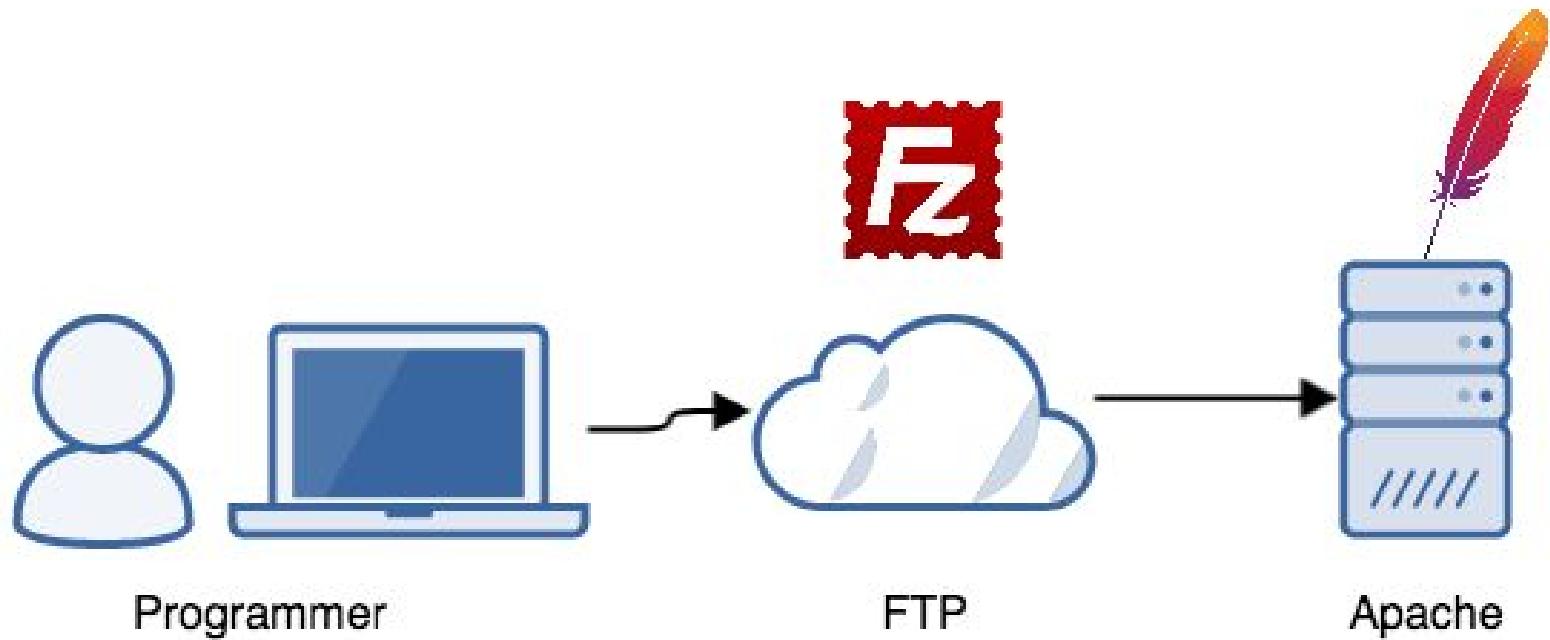
Jenkins

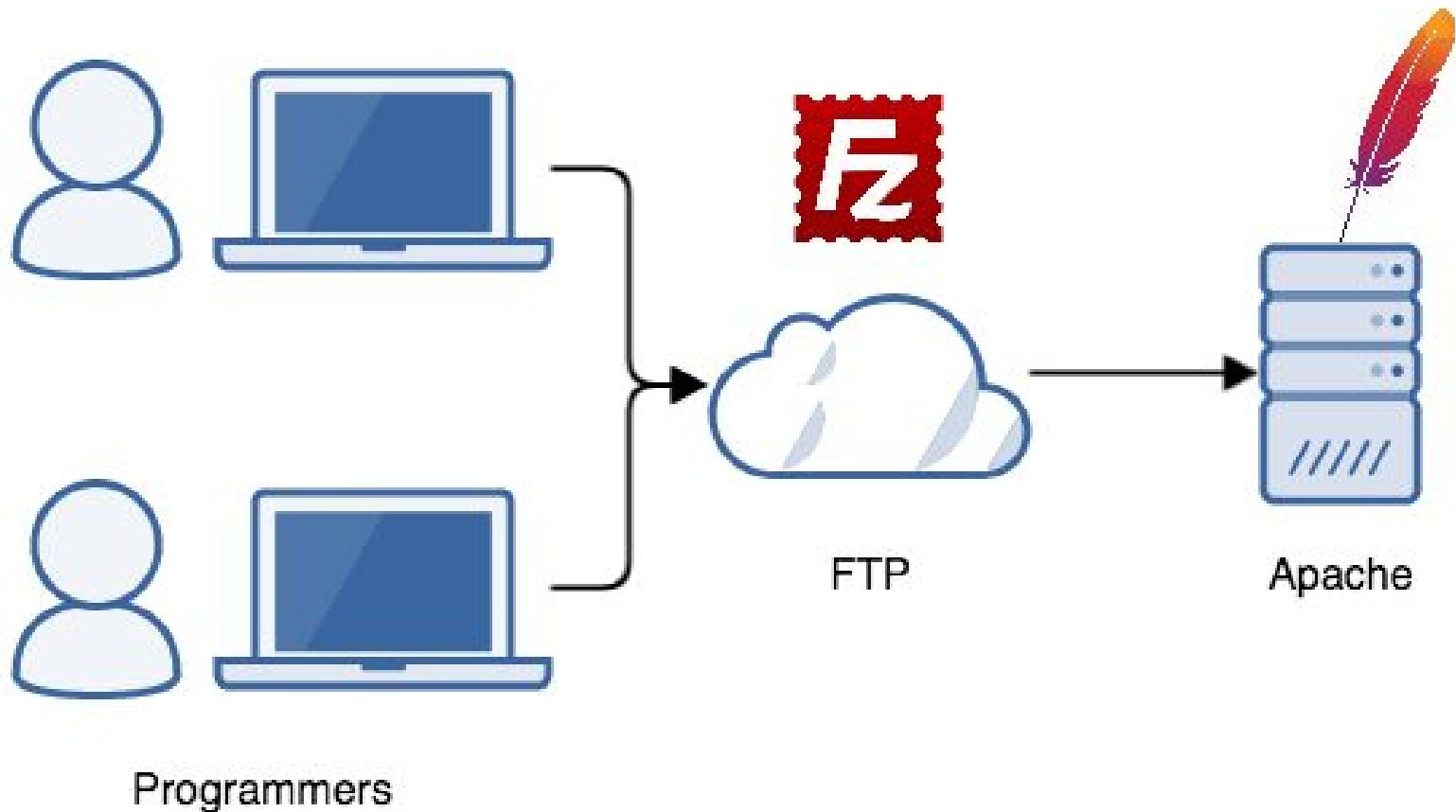
Slides online version is here!

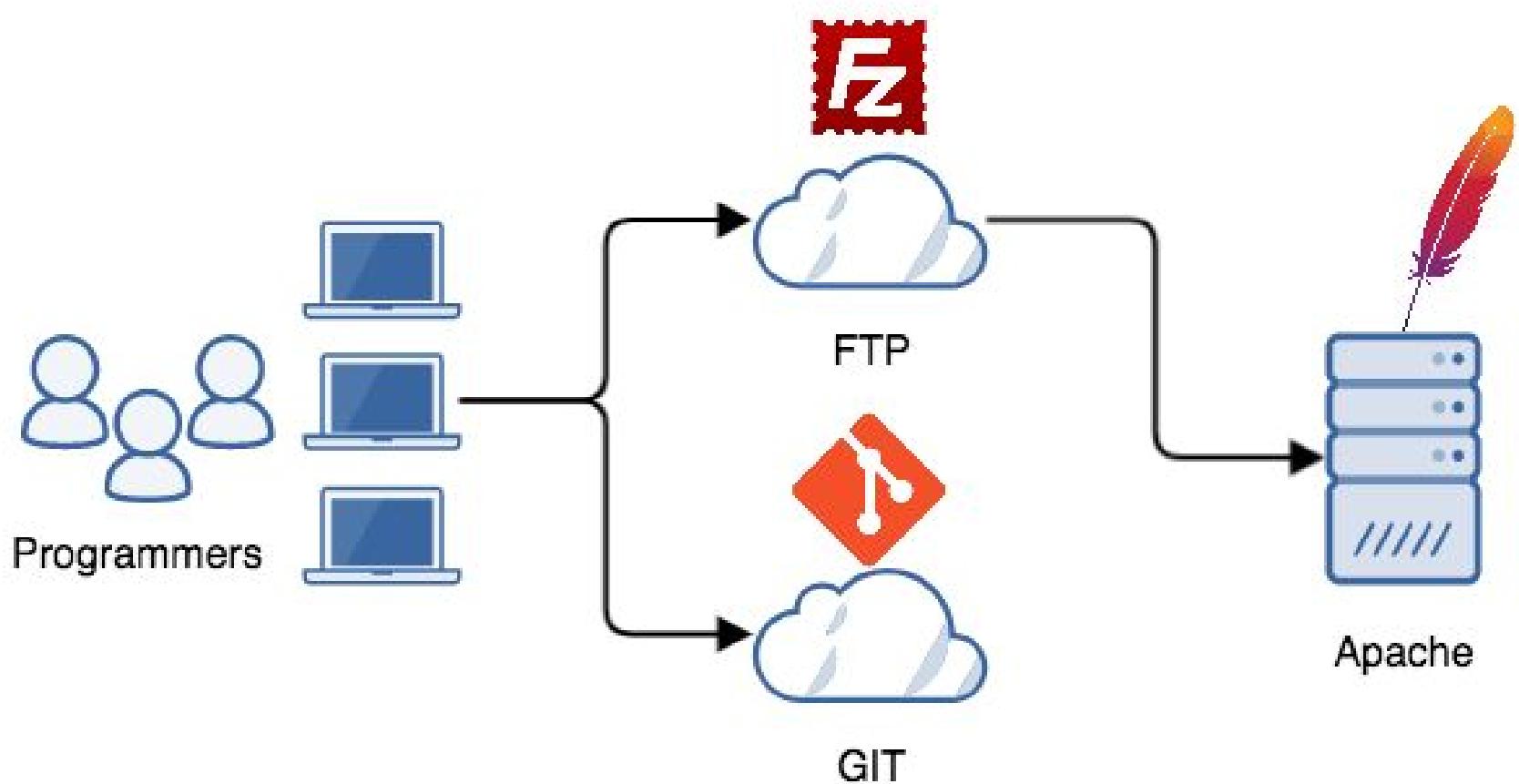
<http://bit.ly/automate-dev-workshop>

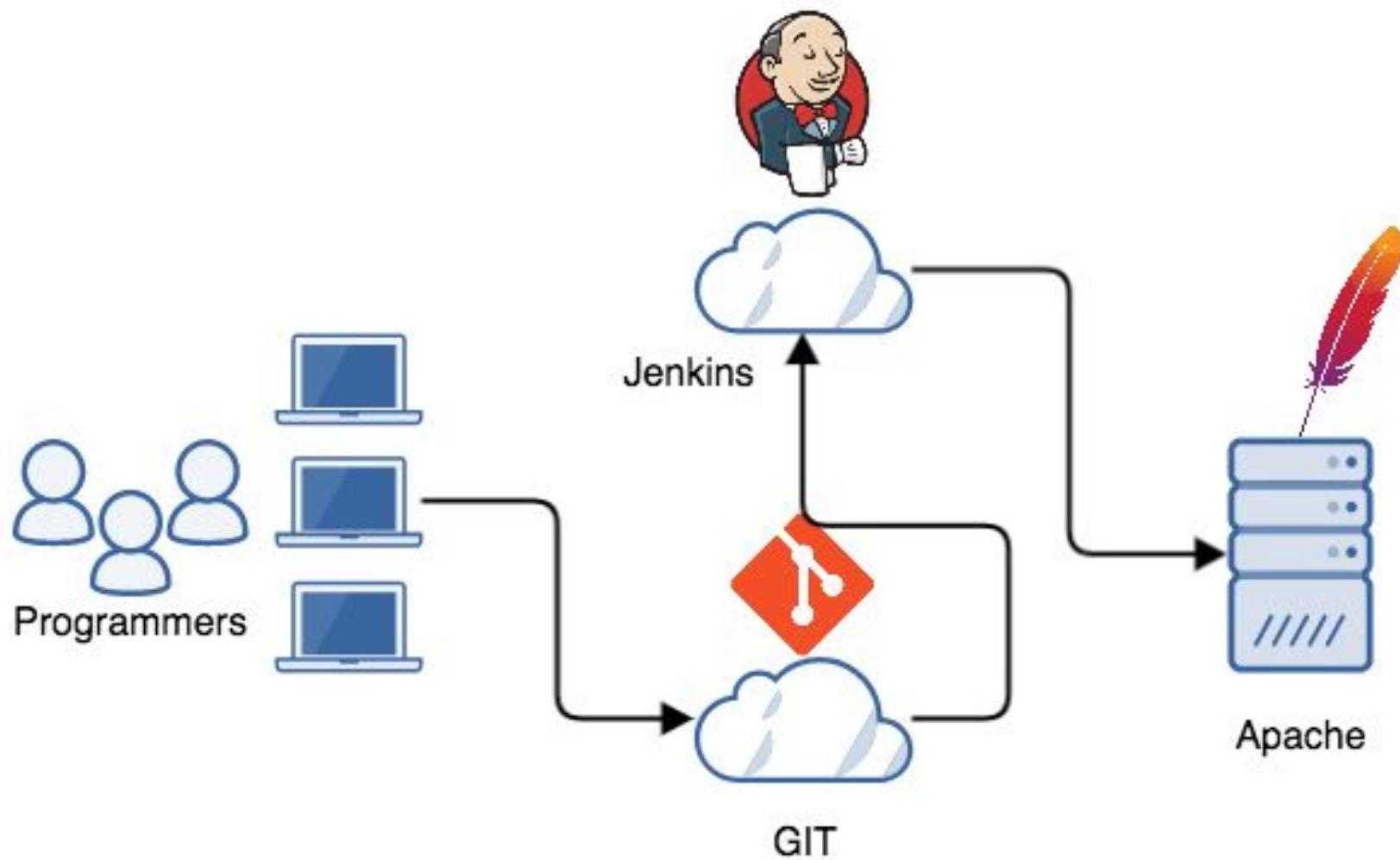
WHY WE AUTOMATE?

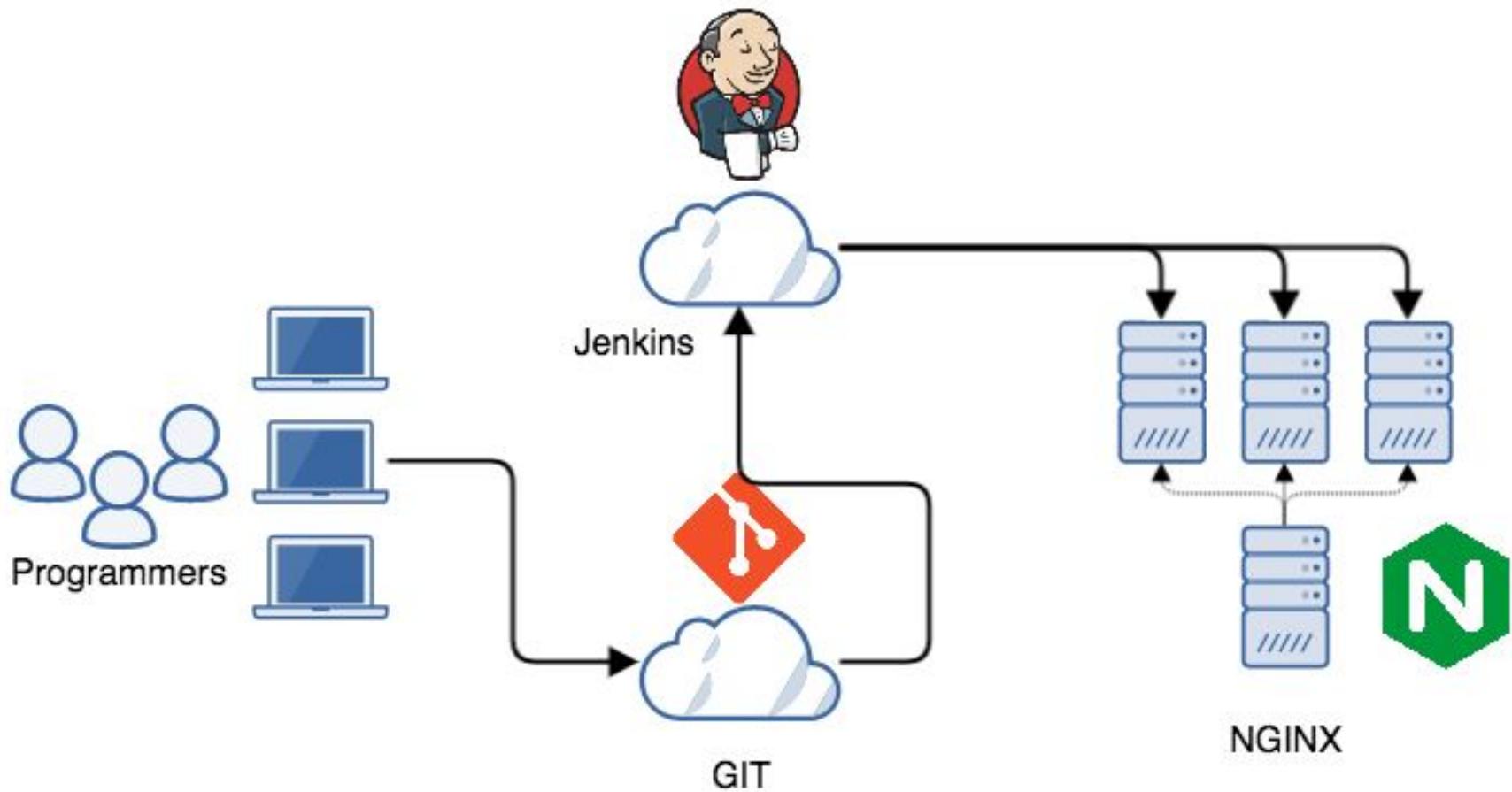
10 Years ago

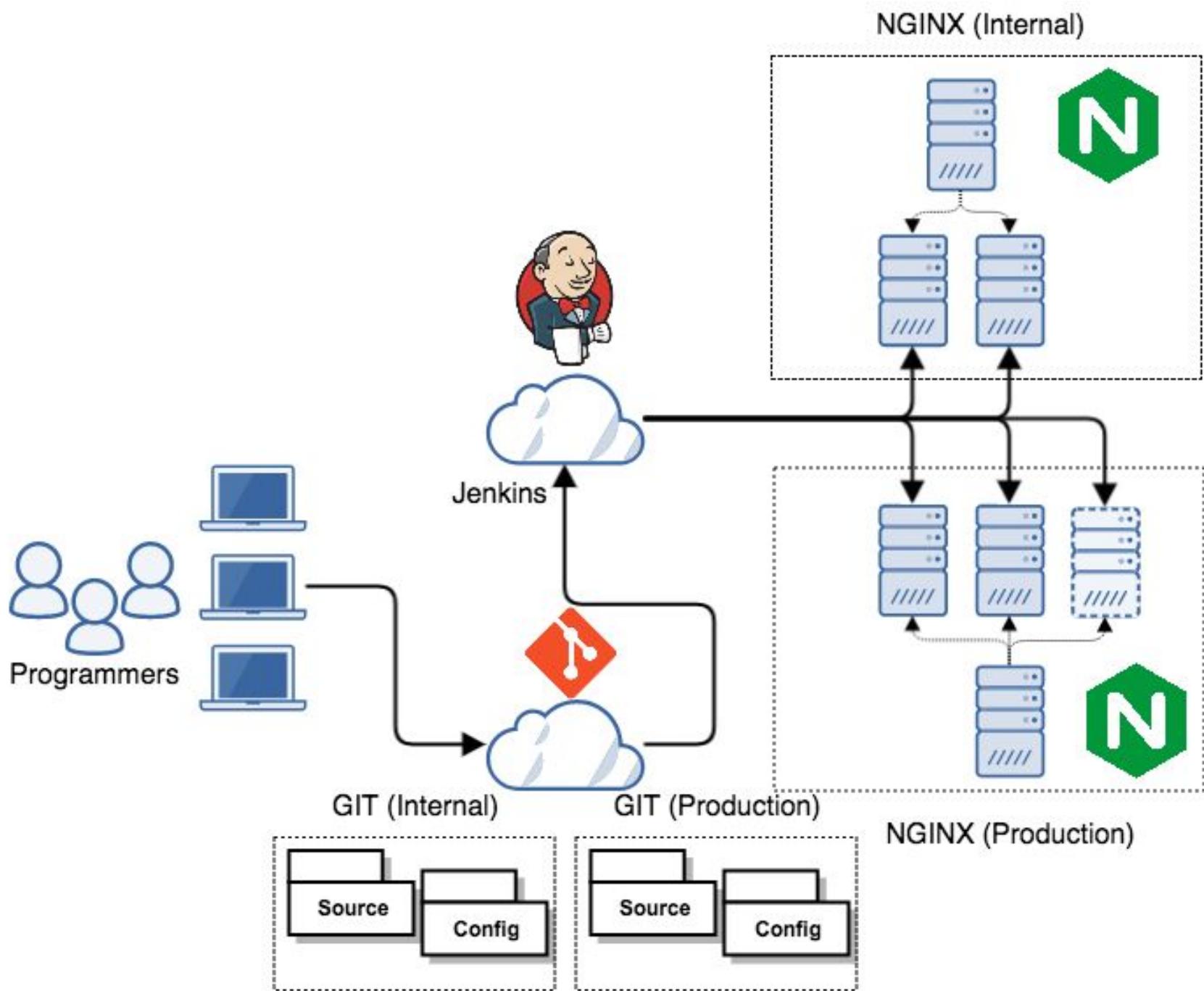


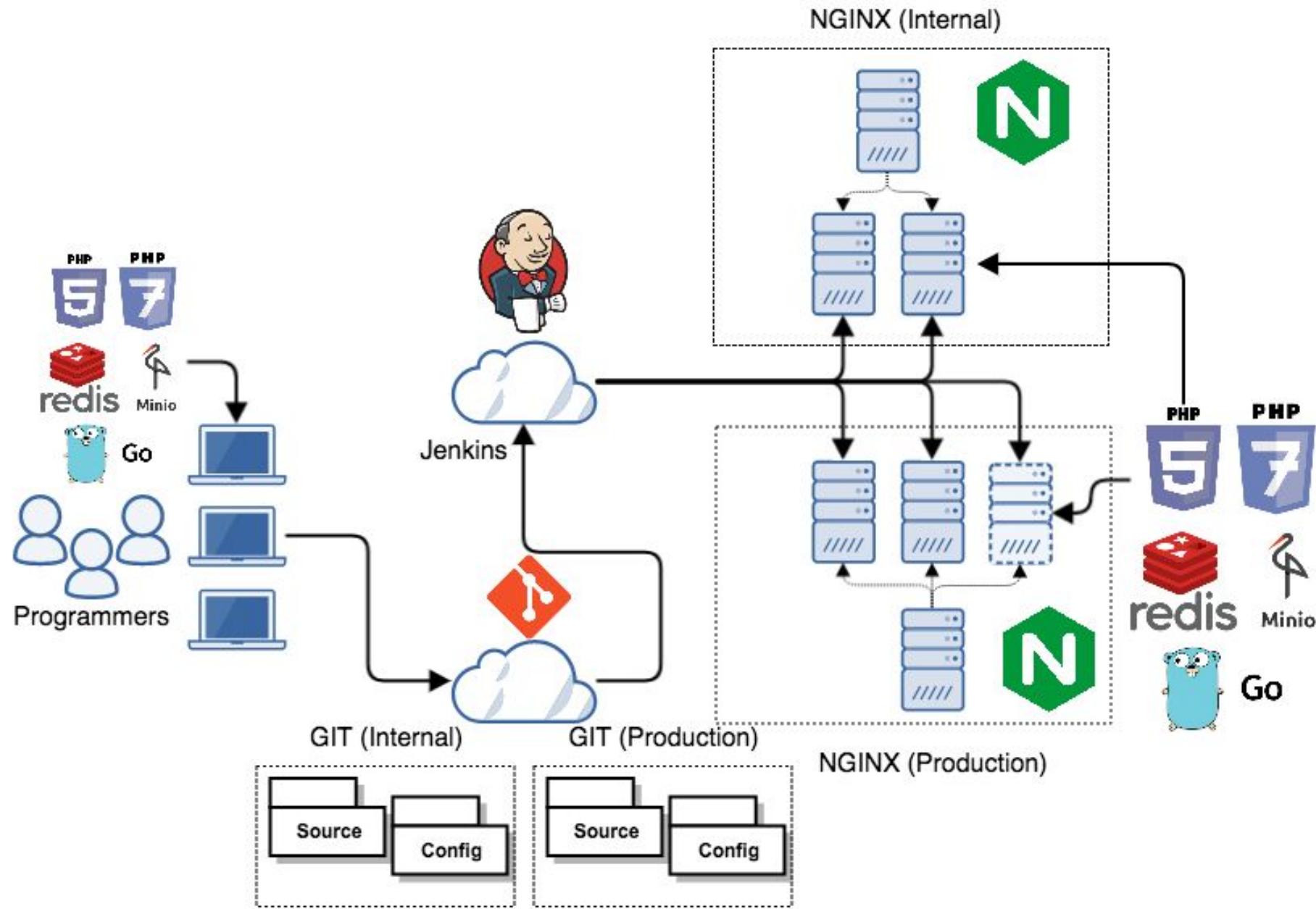


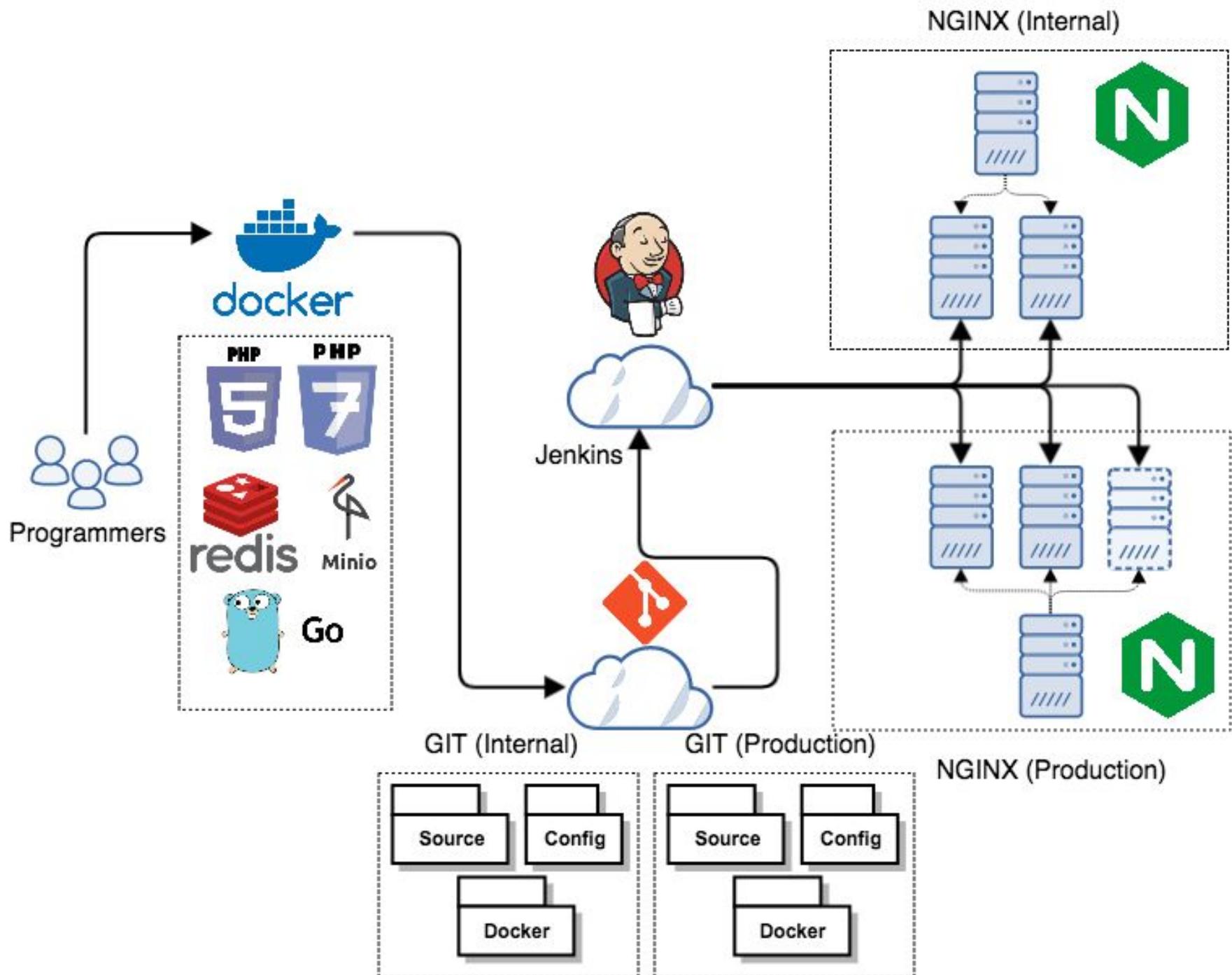


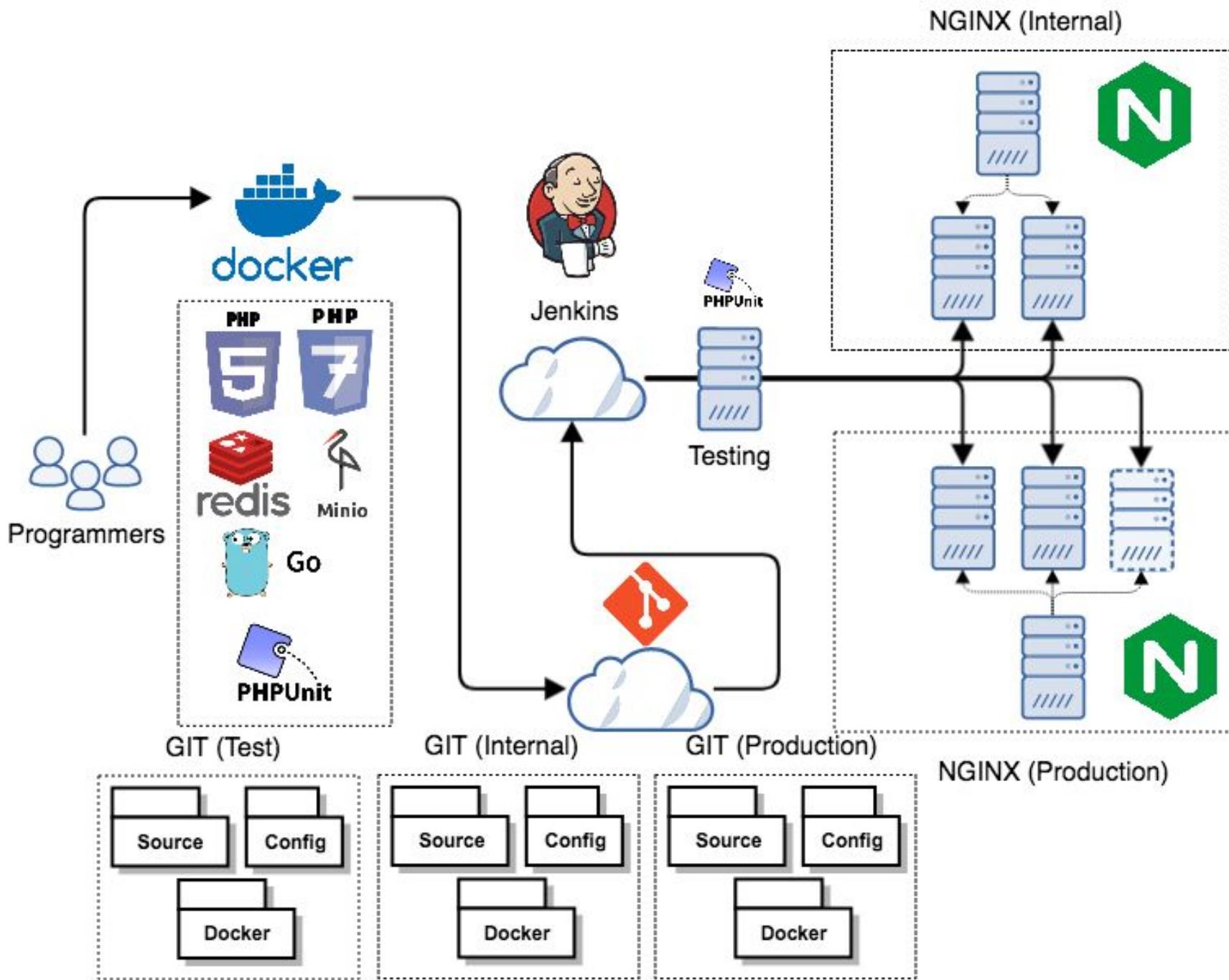




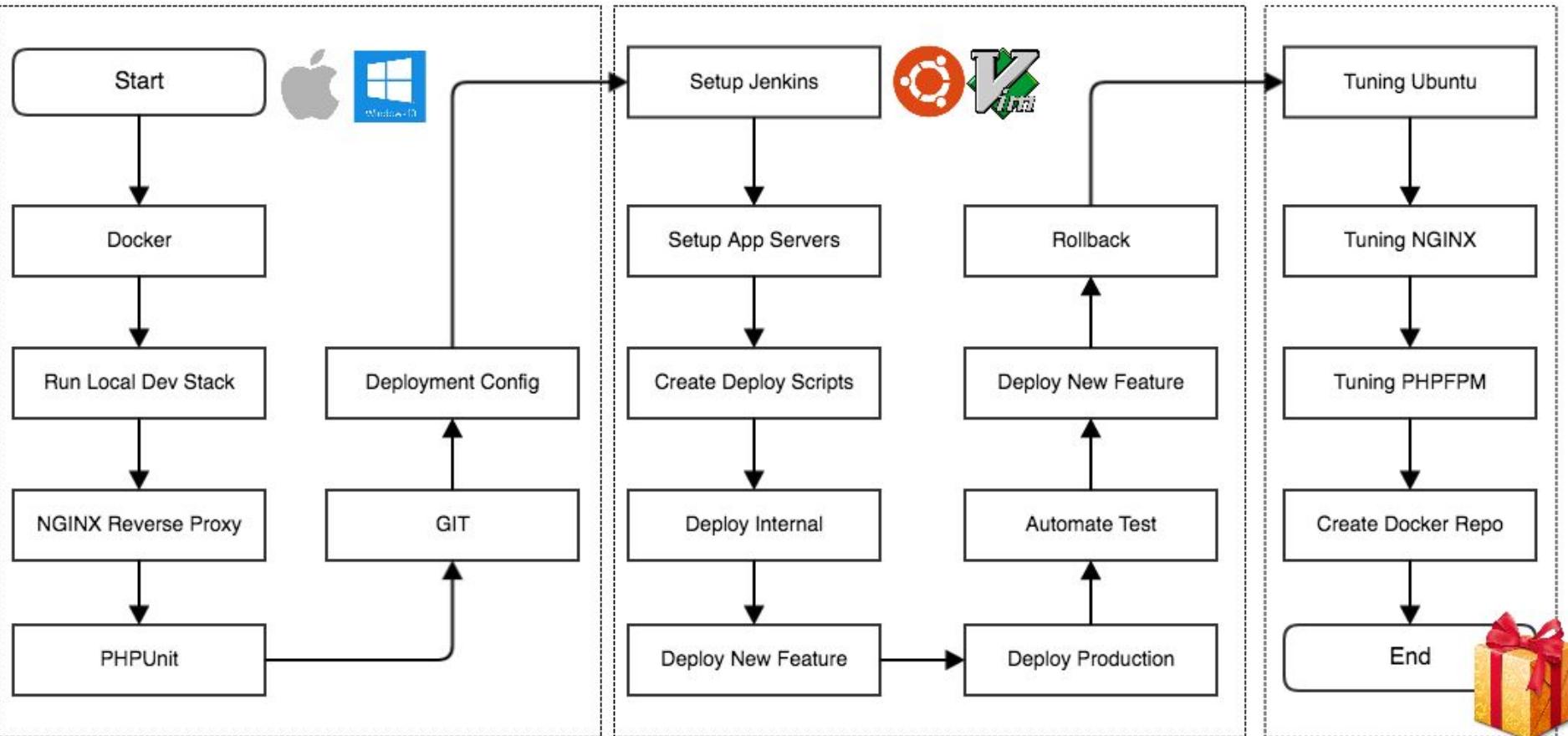








AGENDA



DOCKER

Outline

- Docker principle
- Docker machine
- Image, Repository & Tag, container
- Volume
- Compose (Application Stack Concept)

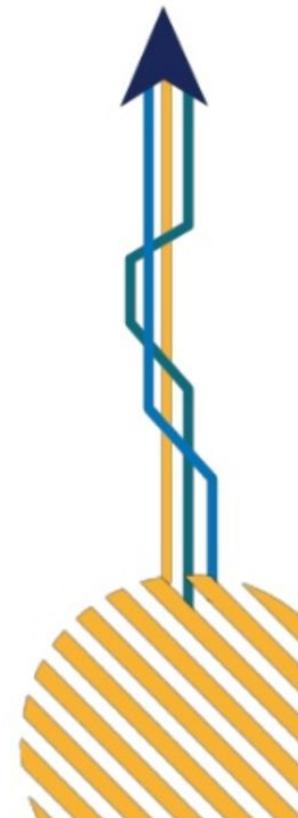
Prerequisite

- Windows (64 bit) / Mac / Linux (64 bit) machine (ubuntu / alpine prefer)
- 1 email address (For register “hub.docker.com”) / hub.docker.com account
- Tool for editor (vscode etc)
- Tool for shell (putty / terminal etc)
- Tool for transfer file (winscp / scp)
- Basic understand for linux operate
- Basic text editor skill (vim prefer) and linux structure
- Internet for download / upload image

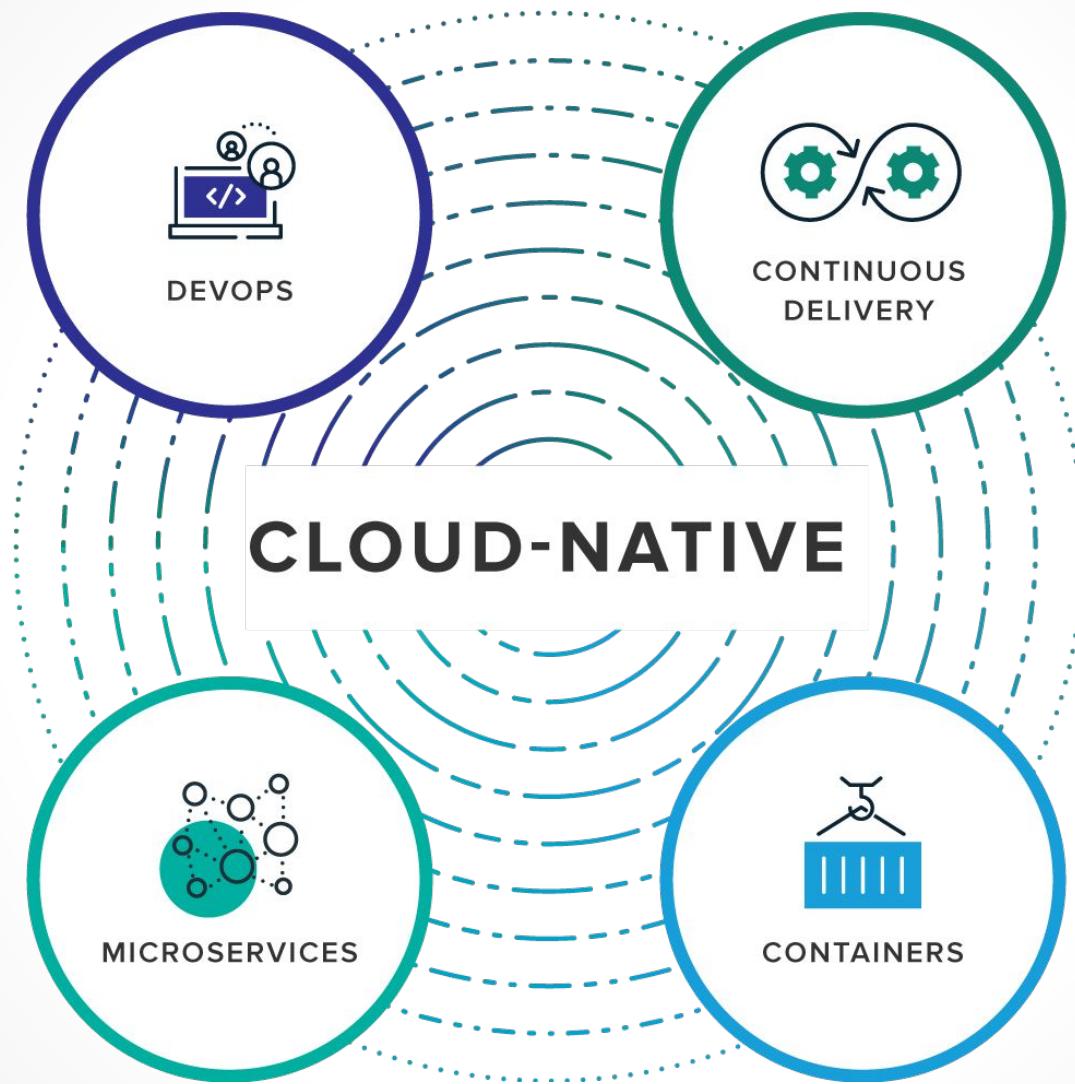
Landscape of the world now



Imagine how the
world **should** work



Landscape of the world now



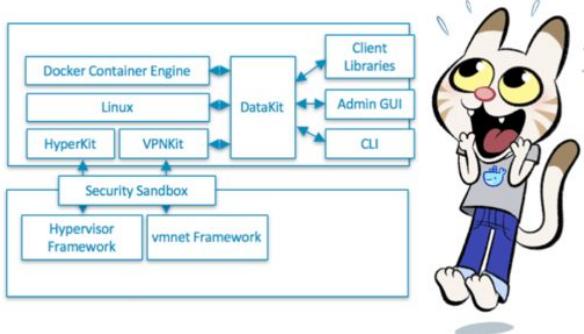
Docker Principle

• • •

Workshop 1: Download and Install

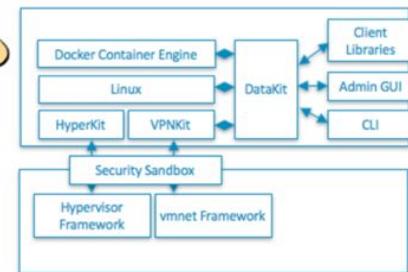
Docker for Mac

Docker for Mac is an easy-to-install desktop app for building, debugging and testing Dockerized apps on a Mac. Docker for Mac is a complete development environment deeply integrated with the MacOS Hypervisor framework, networking and filesystem. Docker for Mac is the fastest and most reliable way to run Docker on a Mac.



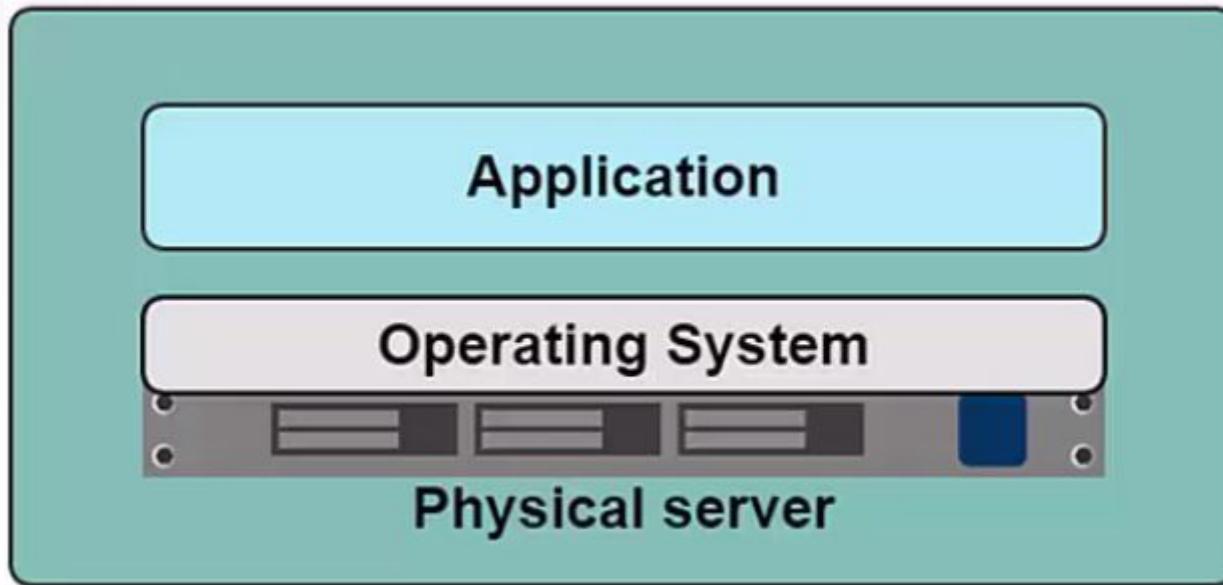
Docker for Windows

An integrated, easy-to-deploy development environment for building, debugging and testing Docker apps on a Windows PC. Docker for Windows is a native Windows app deeply integrated with Hyper-V virtualization, networking and file system, making it the fastest and most reliable Docker environment for Windows.

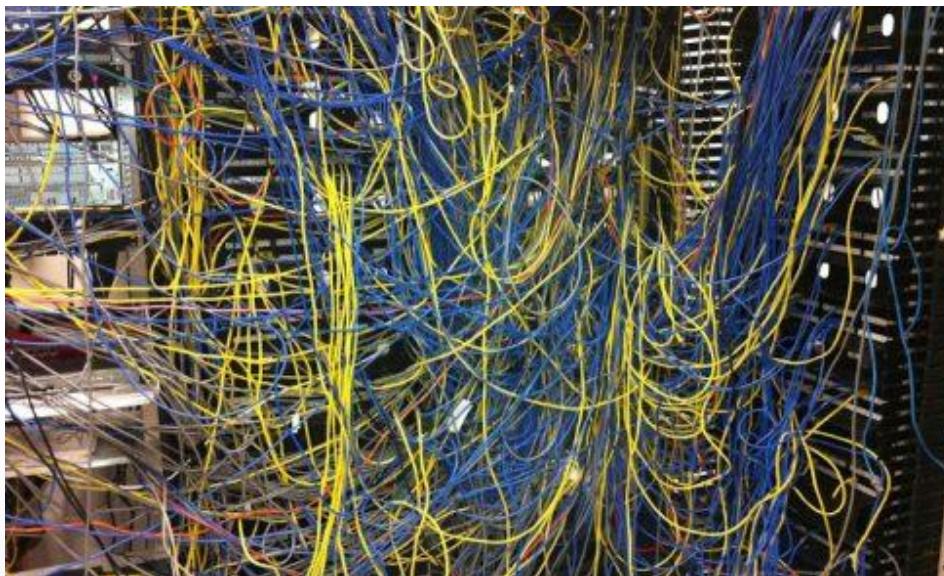
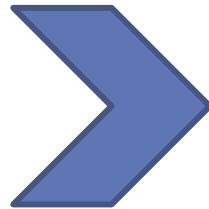


- <https://www.docker.com/docker-mac>
- <https://www.docker.com/docker-windows>
- Install the software
 - Windows : Manual_Docker_For_Windows
 - Mac OS: Manual_Docker_For_Mac.pdf

What is docker ?



Existing Technology

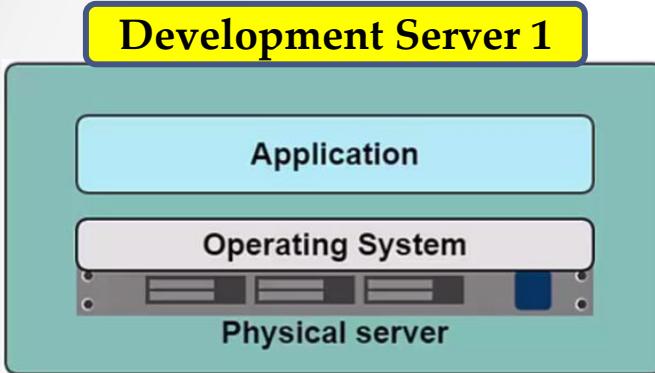


Workshop Automate Software Development

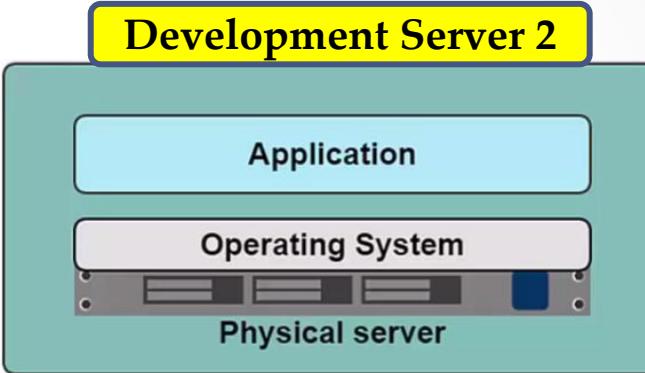


Existing Technology

- Development Environment (Mix everything possible)



- Apache 2.20 Web Server
 - PHP 5.5 Engine
 - Laravel 4.1 Framework
 - MariaDB 5.1
 - Etc
- (All-in-One Server)

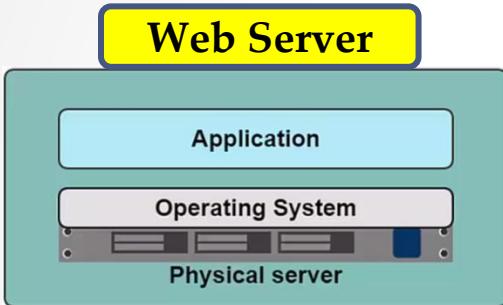


- IIS 8.0
- .Net Framework 3.5
- ASP.NET
- Etc

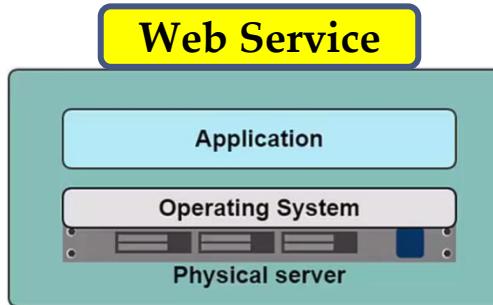
- Need concern about conflict component
- Survive among legacy dependency
- Lack for environment for fulfill develop & test (module test/integrate test/ UAT test / MOT test etc)
- Unexpected software conflict frequently occur
- Incomplete software's integrated test

Existing Technology

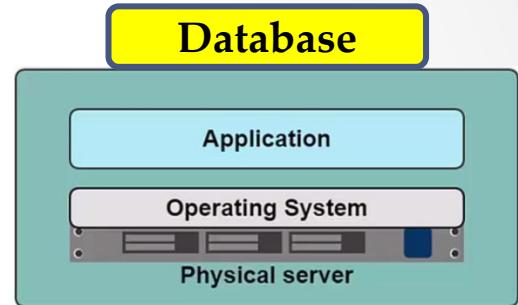
- Production Environment (Best design)
- Day 1: Application 1: Implement



- Apache 2.20 Web Server
- PHP 5.5 Engine
- Laravel 4.1 Framework



- IIS 8
- .Net FrameWork 3.5



- MariaDB 5.1

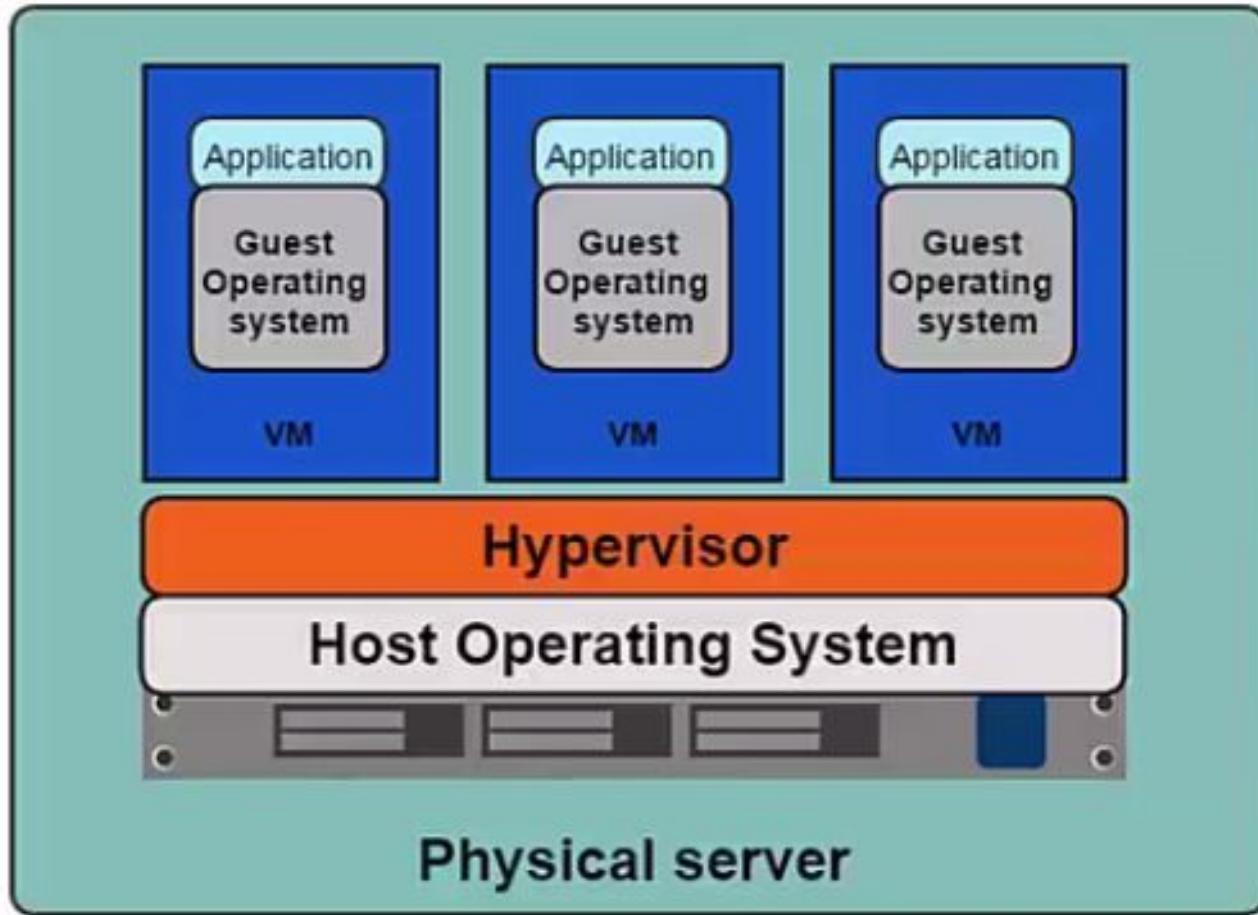
- Day 2: Application 2: Need to implement
 - Need PHP 7.0 ?
 - MariaDB 10.1.14 (Need search feature on 10.1)
- Problem ?
 - Possible to upgrade PHP to 7.0 ? / How to test existing application ?
 - What effect to MariaDB upgrade ?

Existing Technology

- What operation handle in production environment ?
 - Aware about huge of difference software component between development / production machine
 - Need to know in deep all application dependency (Wow !!!)
 - Take time for discussion and find solution to implement.
 - Possible to confusion and effect to other application that existing on share server.
 - Many unexpected problem & software bug.
 - Hard control software's quality assurance (QA)

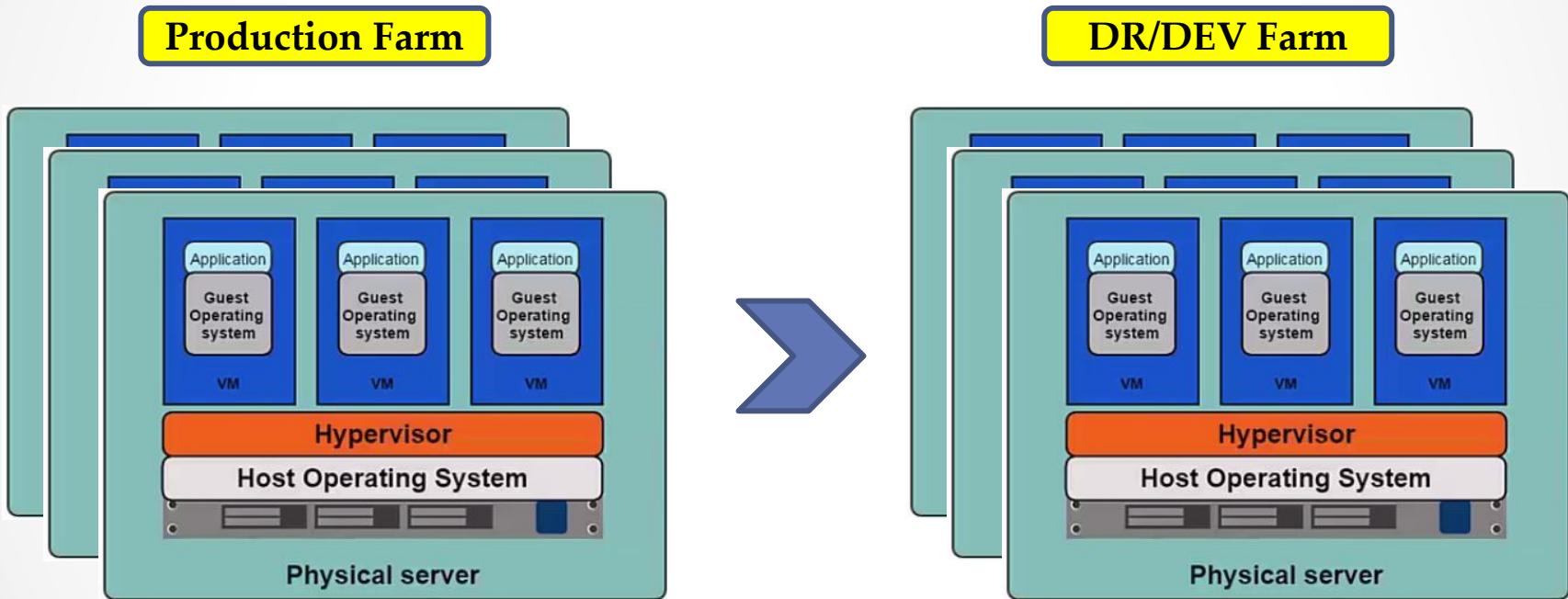


What is docker ?



Existing Technology

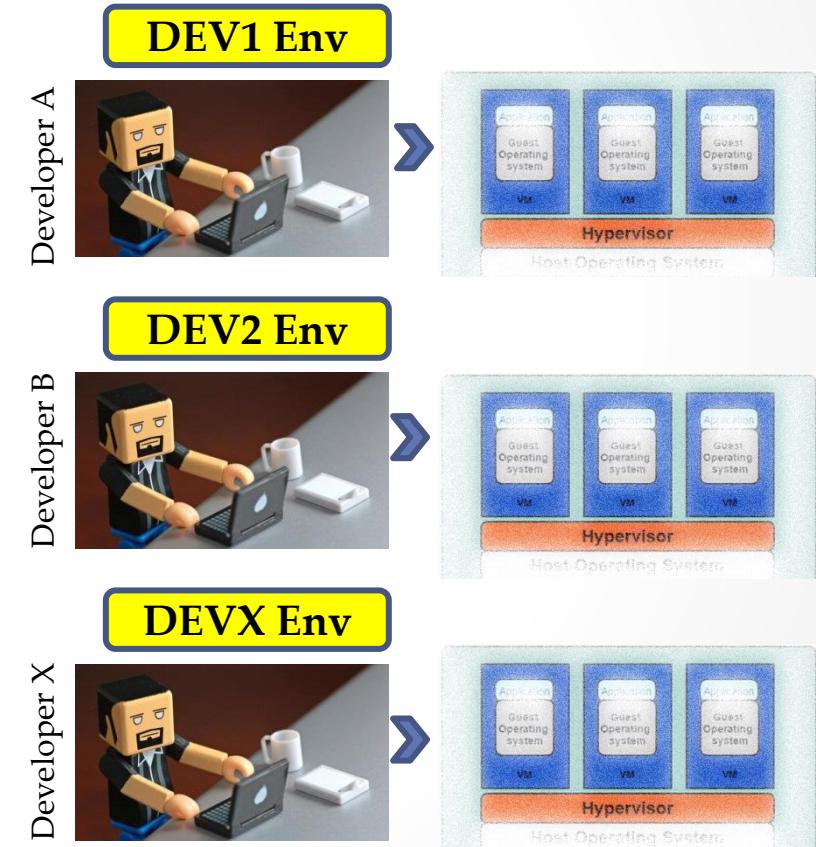
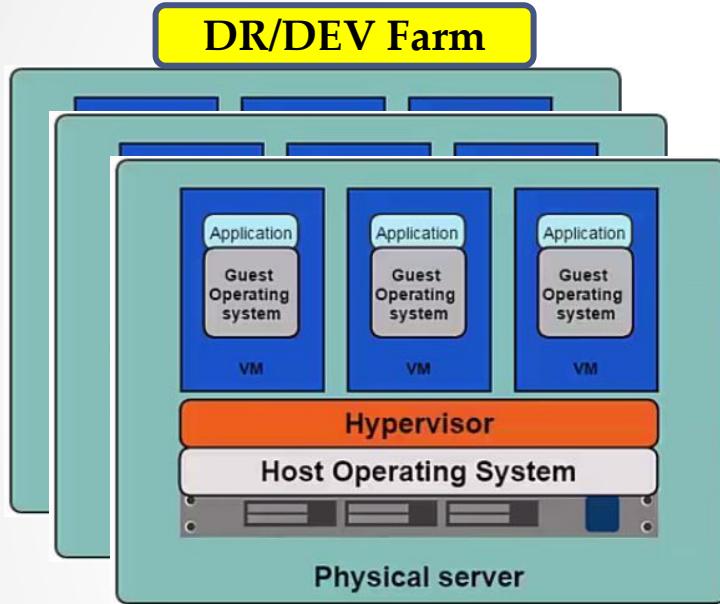
- 1 Physical server : 1 – N VMWare (&OS)
- Virtualize Hardware (CPU, Memory, Disk, Network etc)



- Kernel-based virtual machine (KVM), Vmware, Virtualbox etc

Existing Technology

- Development Environment (Clone from Production)



Virtual Machine reach limit:

- Resource insufficient (Almost from disk)
- Conflict version
- Huge of disk duplication
- Conflict & dependency still exist

Existing Technology

- Production Environment
- Day 1: Application 1: Implement



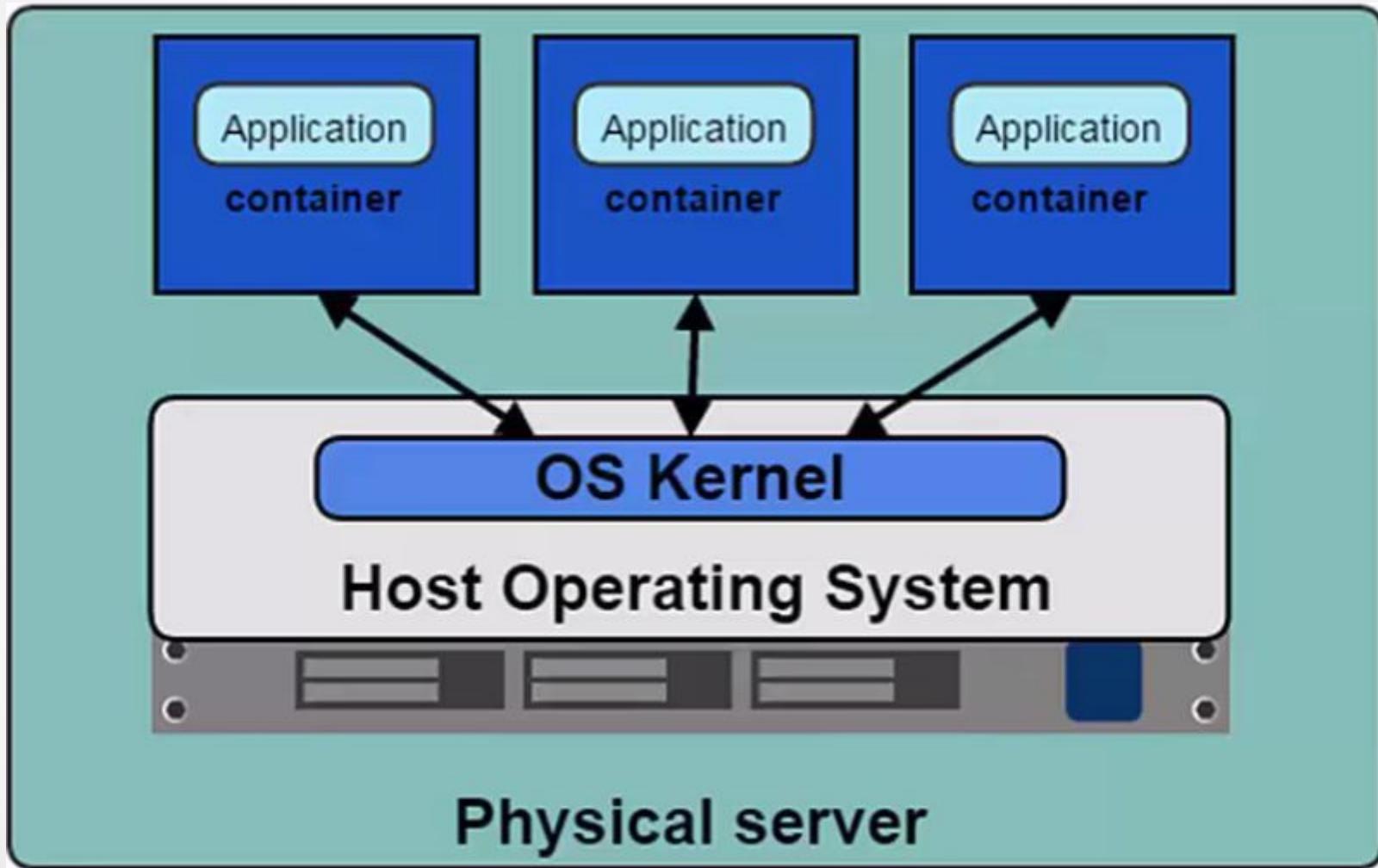
- Apache 2.20 Web Server
- PHP 5.5 Engine
- Laravel 4.1 Framework

- IIS 8
- .Net FrameWork 3.5

- MariaDB 5.1

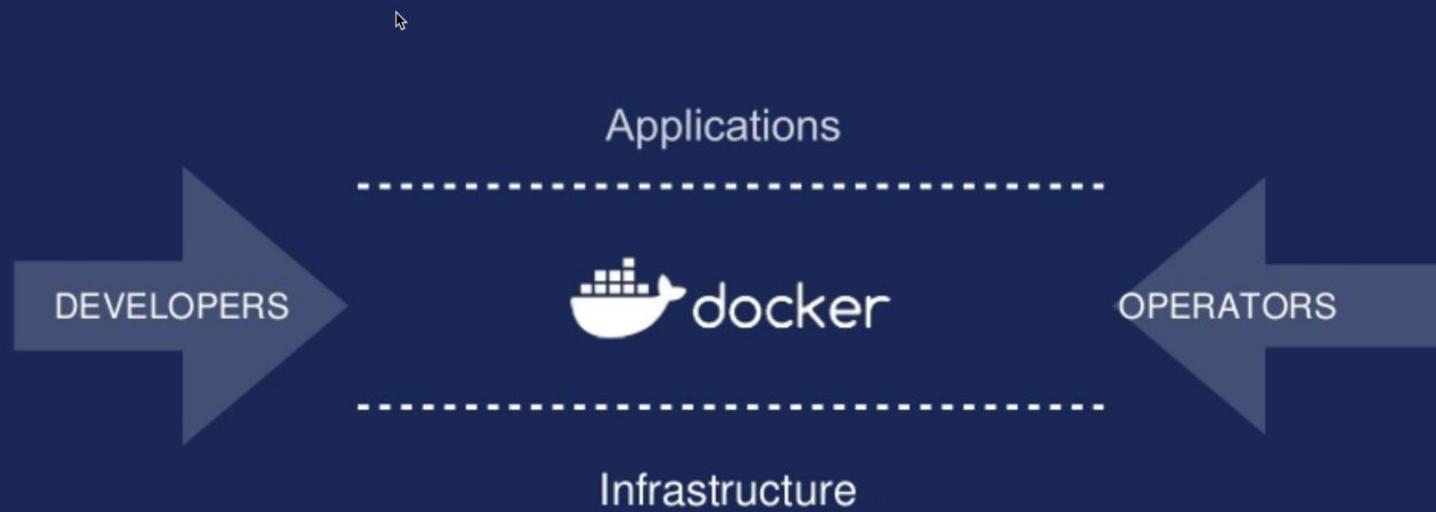
- Day 2: Application 2: Need to implement
 - Need PHP 7.0 ?
 - MariaDB 10.1.14 (Need search feature on 10.1)
- So... The problem still exist.

What is docker ?



What is docker ?

The Docker Platform in a nutshell



What is docker ?

Core Principles of the Docker Platform

INDEPENDENCE



OPENNESS



SIMPLICITY



 dockercon¹⁷ EU

What is docker ?

- Docker คือ open platform solution ที่ทำงานภายใต้คอนเซ็ปต์ของ container virtualize technology (operating -system –level virtualization)
- ผู้ใช้สามารถสร้างสภาพแวดล้อมเพื่อใช้ในพัฒนาโปรแกรมและส่งมอบเพื่อใช้งานในสภาพเดียวกัน
- Build, Ship, Run
- เหมาะสำหรับ Developer, DevOps, Architecture, Engineer
- เขียนด้วยภาษา Go (1.8.3) (Now)
- รองรับการติดตั้งบนบนลินุกซ์ 64 บิต (kernel 3.1.0) (Official)
 - Ubuntu
 - Debian
 - Red Hat Enterprise Linux
 - CentOS
 - Fedora
 - Microsoft Windows Server 2016
 - Suse Linux Enterprise Server
 - Raspberry PI

What is docker ?

Desktop

Platform	Docker CE x86_64	Docker CE ARM	Docker EE
Docker for Mac (macOS)	✓		
Docker for Windows (Microsoft Windows 10)	✓		

Cloud

Platform	Docker CE x86_64	Docker CE ARM	Docker EE
Amazon Web Services	✓		✓
Microsoft Azure	✓		✓

See also [Docker Cloud](#) for setup instructions for Digital Ocean, Packet, SoftLink, or Bring Your Own Cloud.

Server

Platform	Docker CE x86_64	Docker CE ARM	Docker CE System Z (s390x)	Docker EE
CentOS	✓			✓
Debian	✓	✓		
Fedora	✓			
Microsoft Windows Server 2016				✓
Oracle Linux				✓
Red Hat Enterprise Linux				✓
SUSE Linux Enterprise Server				✓
Ubuntu	✓	✓	✓	✓

What is docker ?

The Docker Innovation Model

9,149 Open Source Contributors

8800 PRs/Year



What is docker ?



Docker Enterprise Edition (EE) and Community Edition (CE)

Enterprise Edition (EE)

- CaaS enabled platform subscription (integrated container orchestration, management and security)
- Enterprise class support
- Quarterly releases, supported for one year each with backported patches and hotfixes.
- Certified Infrastructure, Plugins, Containers

Community Edition (CE)

- Free Docker platform for "do it yourself" dev and ops
- Monthly Edge release with latest features for developers
- Quarterly release with maintenance for ops

Lifecycle

Squaring the circle: Faster releases and better stability



Docker EE Availability

From Docker



OEM: Direct L2 / L3 Support Included



Cloud Marketplaces



Ref: <https://blog.docker.com/2017/03/docker-online-meetup-recap-docker-enterprise-edition-ee-community-edition-ce/>

What is docker ?



Docker Machine

Automated Docker provisioning



Docker Swarm

Host clustering and container scheduling



Docker Compose

Define multi-container applications



Docker Registry

Open source Docker image distribution (not included)



Docker Engine

Creates and runs Docker containers



Kitematic

Desktop GUI for Docker

Operating-system-level virtualization

Mechanism	Operating system	License	Available since/between	Features									
				File system isolation	Copy on Write	Disk quotas	I/O rate limiting	Memory limits	CPU quotas	Network isolation	Nested virtualization	Partition checkpointing and live migration	Root privilege isolation
chroot	most UNIX-like operating systems	varies by operating system	1982	Partial ^[3]	No	No	No	No	No	No	Yes	No	No
Docker Linux-VServer (security context)	Linux ^[7]	Apache License 2.0	2013	Yes	Yes	Not directly	Not directly	Yes	Yes	Yes	Yes	No	No
	Linux	GNU GPLv2	2001	Yes	Yes	Yes	Yes ^[8]	Yes	Yes	Partial ^[9]	?	No	Partial ^[9]
lxc LXC	Linux	Apache License 2.0	2013	Yes	Yes	Yes	Yes ^[9]	Yes	Yes	Partial ^[9]	?	No	Partial ^[9]
	Linux	GNU GPLv2	2008	Yes ^[9]	Yes	Partial ^[9]	Partial ^[9]	Yes	Yes	Yes	Yes	No	Yes ^[9]
LXD	Linux	Apache License 2.0	2015	Yes	Yes	Partial(see LXC)	Partial(see LXC)	Yes	Yes	Yes	Yes	Partial ^[9]	Yes
OpenVZ	Linux	GNU GPLv2	2005	Yes	No	Yes	Yes ^[10]	Yes	Yes	Yes ^[10]	Partial ^[10]	Yes	Yes ^[10]
Virtuozzo	Linux, Windows	Proprietary	2000 ^[14]	Yes	Yes	Yes	Yes ^[11]	Yes	Yes	Yes ^[11]	Partial ^[11]	Yes	Yes
Solaris Containers (Zones)	illumos (OpenSolaris), Solaris	CDDL, Proprietary	2004	Yes	Yes (ZFS)	Yes	Partial ^[12]	Yes	Yes	Yes ^{[12][13]}	Partial ^[12]	Partial ^[12]	Yes ^[8]
FreeBSD jail	FreeBSD	BSD License	2000 ^[20]	Yes	Yes (ZFS)	Yes ^[13]	No	Yes ^[21]	Yes	Yes ^[22]	Yes	No	Yes ^[23]
sysjail	OpenBSD, NetBSD	BSD License	2006–2009 (As of March 3, 2009, it is no longer supported)	Yes	No	No	No	No	No	Yes	No	No	?
WPars	AIX	Proprietary	2007	Yes	No	Yes	Yes	Yes	Yes	Yes ^[24]	No	Yes ^[25]	?
HP-UX Containers (SRP) ^[9]	HPUX	Proprietary	2007	Yes	No	Partial ^[14]	Yes	Yes	Yes	Yes	?	Yes	?
iCore Virtual Accounts	Windows XP	Proprietary/Freeware	2008	Yes	No	Yes	No	No	No	No	?	No	?
Sandboxie Spoon	Windows	Proprietary/Shareware	2004	Yes	Yes	Partial	No	No	No	Partial	No	No	Yes
	Windows	Proprietary	2012	Yes	Yes	No	No	No	No	Yes	No	No	Yes
VMware ThinApp	Windows	Proprietary	2008	Yes	Yes	No	No	No	No	Yes	No	No	Yes

Reference: https://en.wikipedia.org/wiki/Operating-system-level_virtualization



Docker History

- A dotCloud (PAAS provider) project
- Initial commit January 18, 2013
- Docker 0.1.0 released March 25, 2013
- 18,600+ github stars, 3800+ forks, 740 Contributors.... and continues
- dotCloud pivots to docker inc. October 29, 2013



Solomon Hykes

CTO and Founder

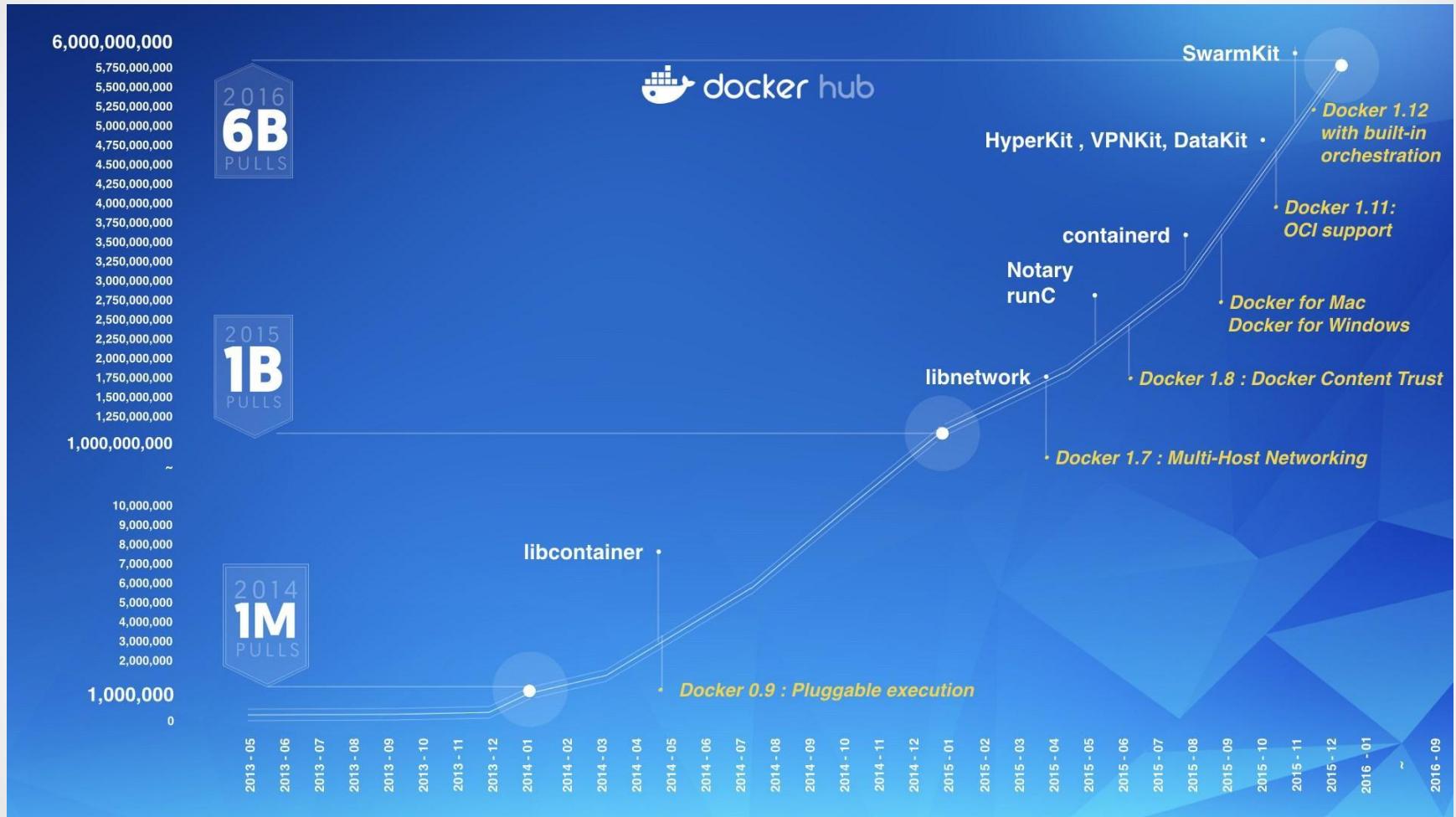
 dockercon¹⁷ EU

A portrait of Solomon Hykes, a man with dark hair and a beard, wearing a black t-shirt. He is positioned in front of a white background with a blue circular border around his head. To the right of the portrait, his name "Solomon Hykes" is written in a large, white, sans-serif font. Below his name, the text "CTO and Founder" is displayed in a smaller, white, sans-serif font. At the bottom left of the slide, there is a logo for "dockercon¹⁷ EU" featuring a stylized blue icon followed by the text.

Workshop Automate Software Development

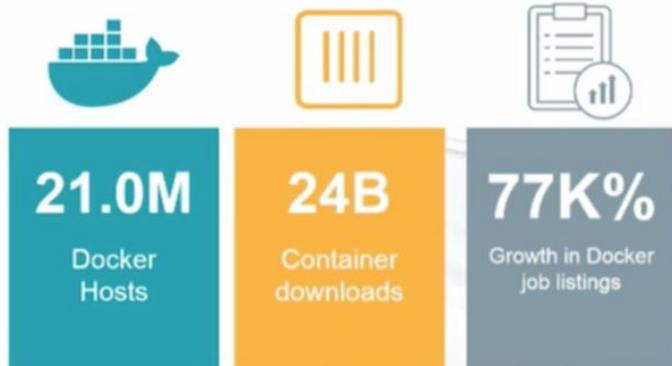


Growth of Docker (2016 stats)

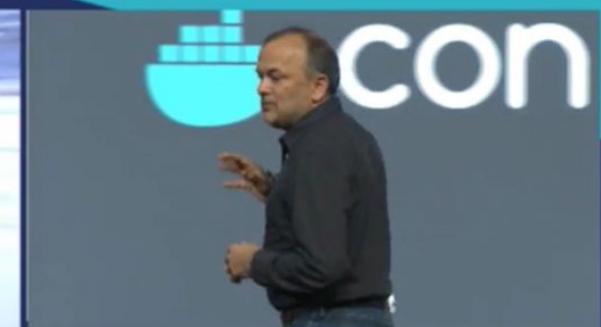


Growth of Docker (Now)

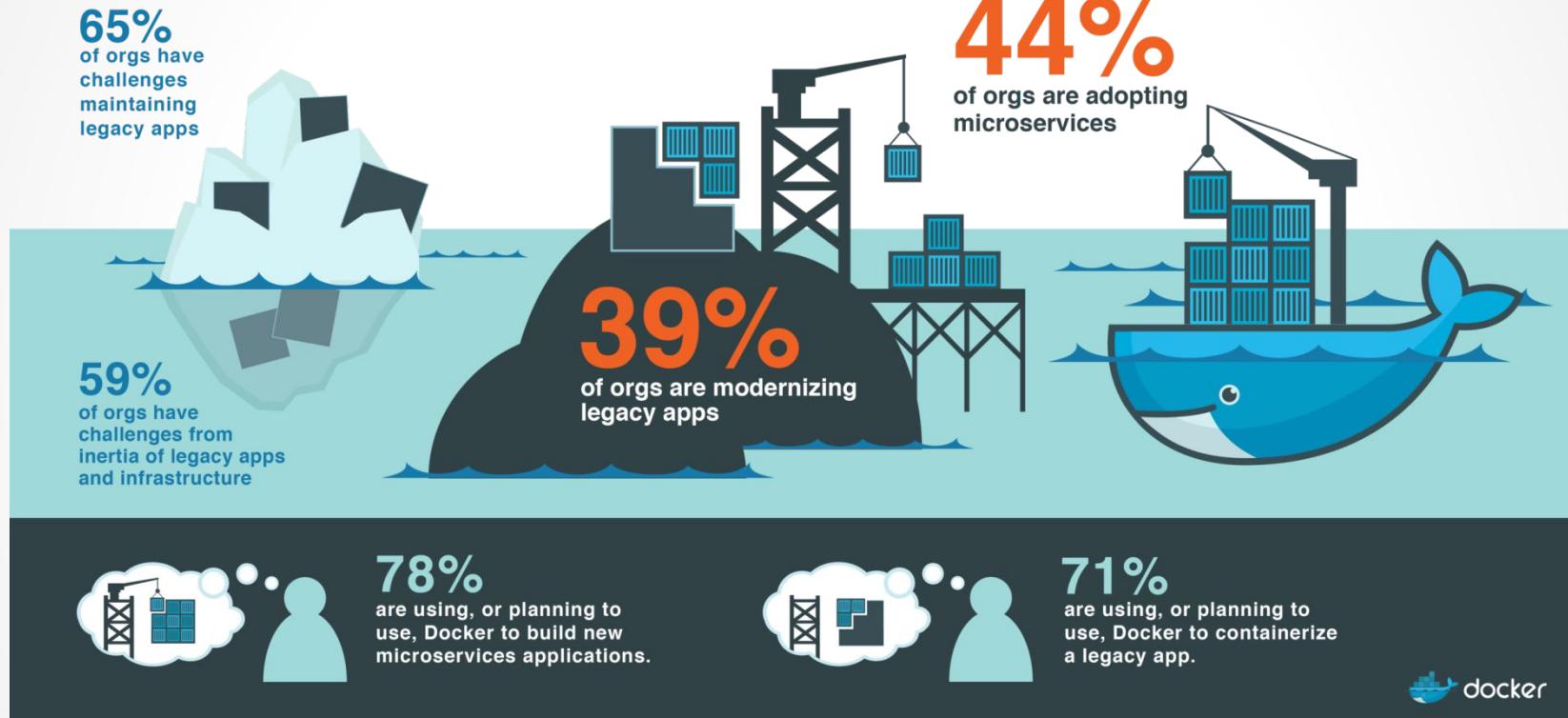
Docker Momentum



Industry Standards



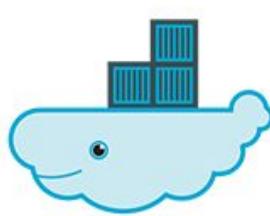
Trend of Modernize Application



Cloud Strategy Survey

80%

say Docker is part
of cloud strategy



60%

plan to use Docker to
migrate workloads to cloud



41%

want application
portability across
environments

35+%

want to avoid
cloud vendor
lock-in



Docker in Thailand

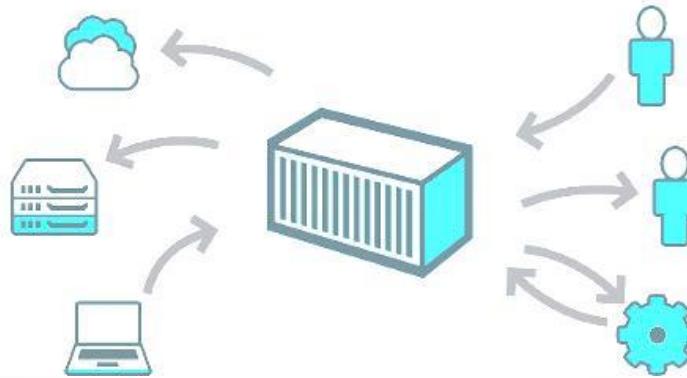
- 3 Enterprise Communication on Implement Phase
- 2 Bank on R&D

Workshop Automate Software Development

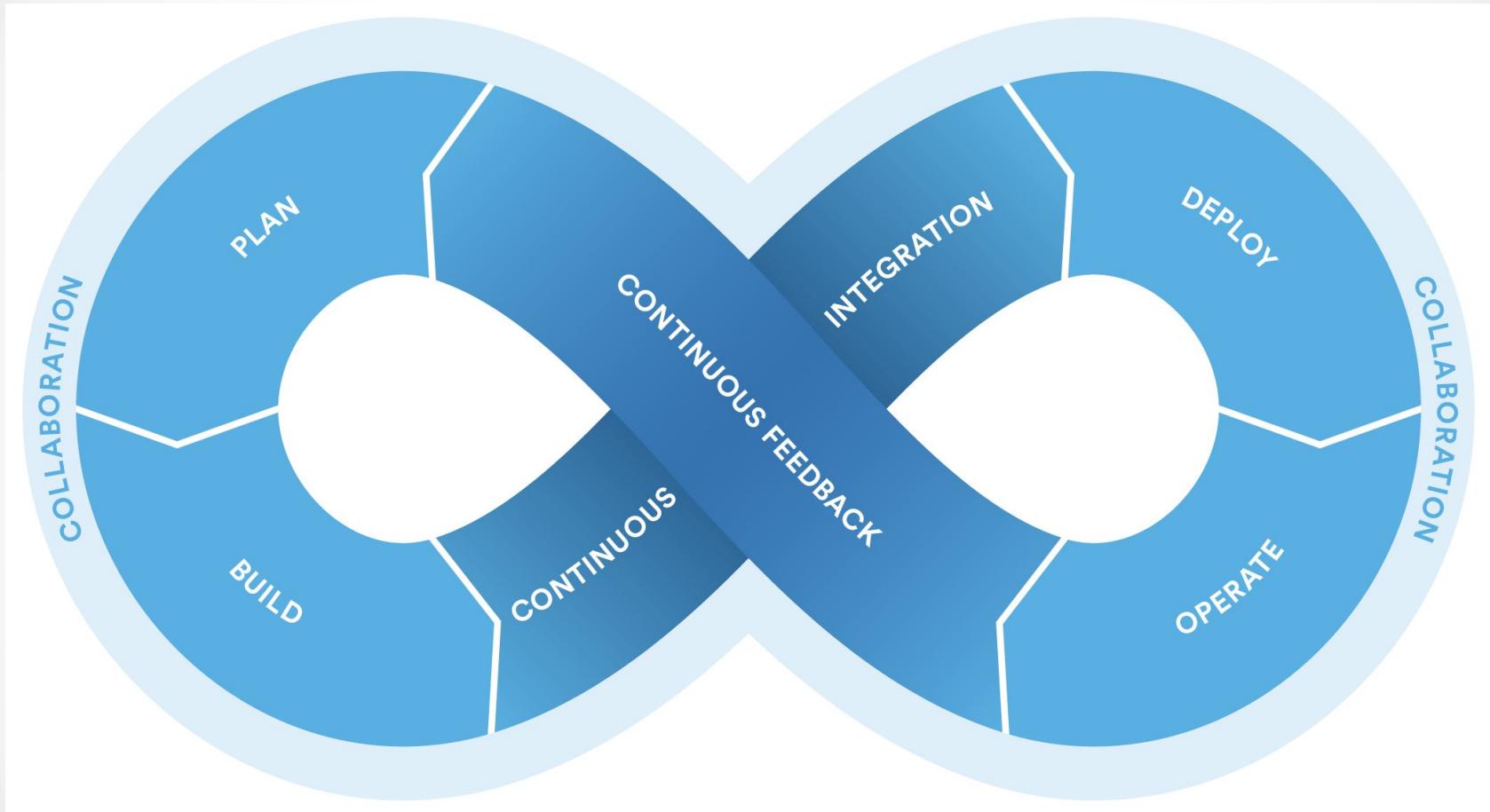


Container Life Cycle

- Developer
 - สร้าง template สำหรับโปรแกรม (build)
 - เริ่มรัน template สำหรับพัฒนาโปรแกรม (run)
 - พัฒนาเซอร์วิสเรียบเรียง สั่ง save version เพื่อเตรียมนำขึ้นใช้งาน (commit)
- Operation
 - เริ่มรัน template ที่ได้รับจาก developer (run) บนเครื่อง production
 - หยุดการให้บริการโปรแกรม (stop, kill)
 - ลบโปรแกรมออกจากเครื่อง production (rm, rmi)

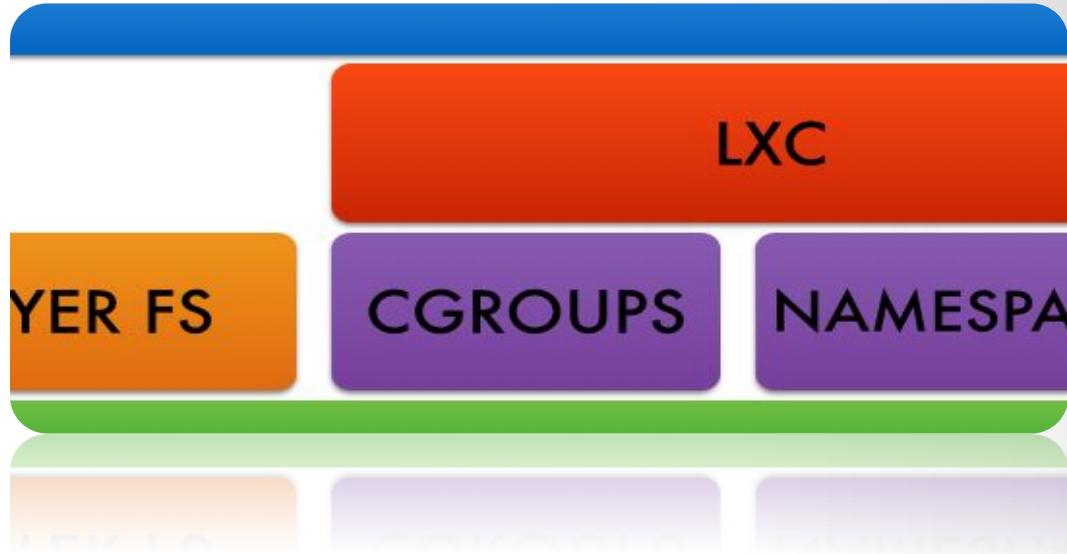


Container Life Cycle



Docker Architecture

- Docker Engine
 - Docker Client
(CLI & GUI)
 - Docker Daemon
- Docker Hub
 - Cloud service
 - Private Registry
- Docker images
- Docker containers



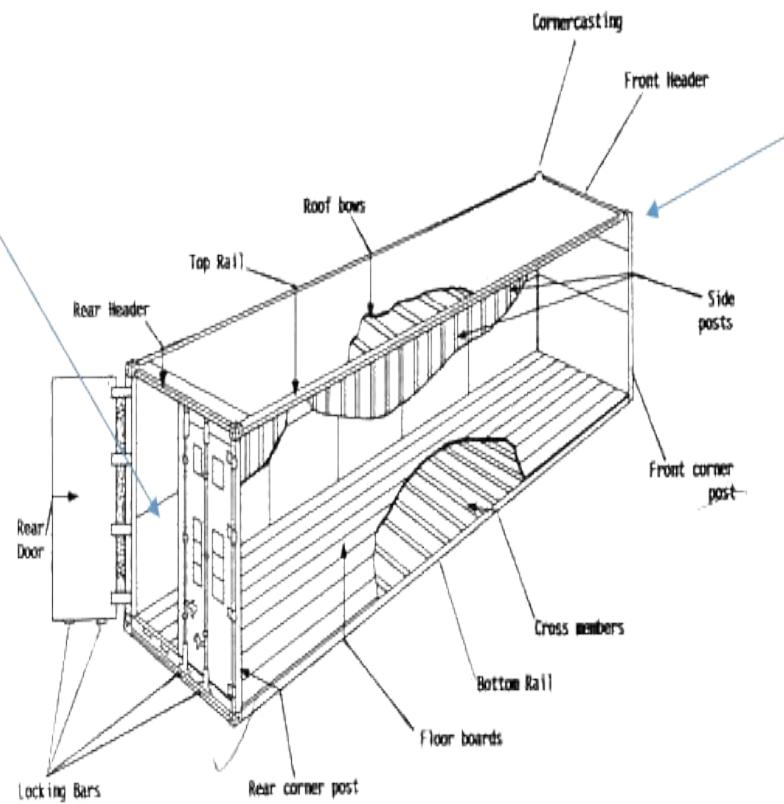
What Cool of Docker ?

- รวบรวมทุกสิ่งที่จำเป็นต้องใช้ในการรันโปรแกรมไว้ใต้ container (component, library etc)
- ขนาดไฟล์ container มีขนาดเล็กมาก (เทียบกับขนาดไฟล์ของ virtual machine หรือ os)
- มี overhead ใน การรันโปรแกรมทรัพยากรต่ำ
- ลดระยะเวลาในการติดตั้งและทดสอบโปรแกรม
- ส่งมอบโปรแกรมไปทำงานบนเครื่องแม่ข่าย production ได้โดยไม่มีความจำเป็นต้องปรับแต่งระบบใหม่ (zero configure)
- สามารถรันโปรแกรมได้บนเครื่องแม่ข่ายทุกๆระบบปฏิบัติการฯ ที่ติดตั้ง docker ได้
- สามารถ scale-out ได้ง่ายในอนาคต
- World open for docker !!!

Separate of Concern

- Dan the Developer

- Worries about what's "inside" the container
 - His code
 - His Libraries
 - His Package Manager
 - His Apps
 - His Data
- All Linux servers look the same



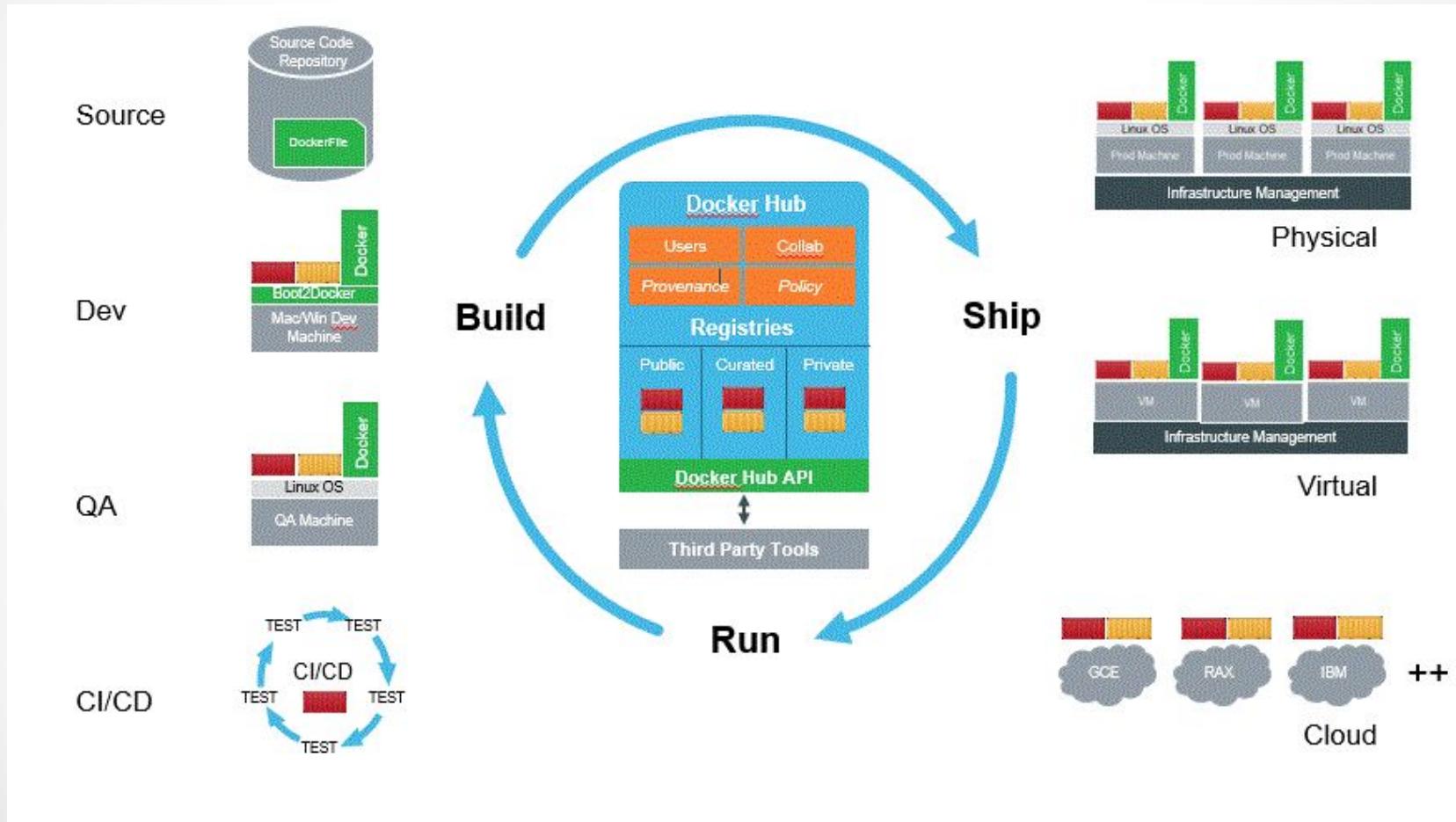
Major components of the container:

- Oscar the Ops Guy

- Worries about what's "outside" the container
 - Logging
 - Remote access
 - Monitoring
 - Network config
- All containers start, stop, copy, attach, migrate, etc. the same way

Benefit of docker for DevOps

- Build-Ship-Run



Who use docker ?

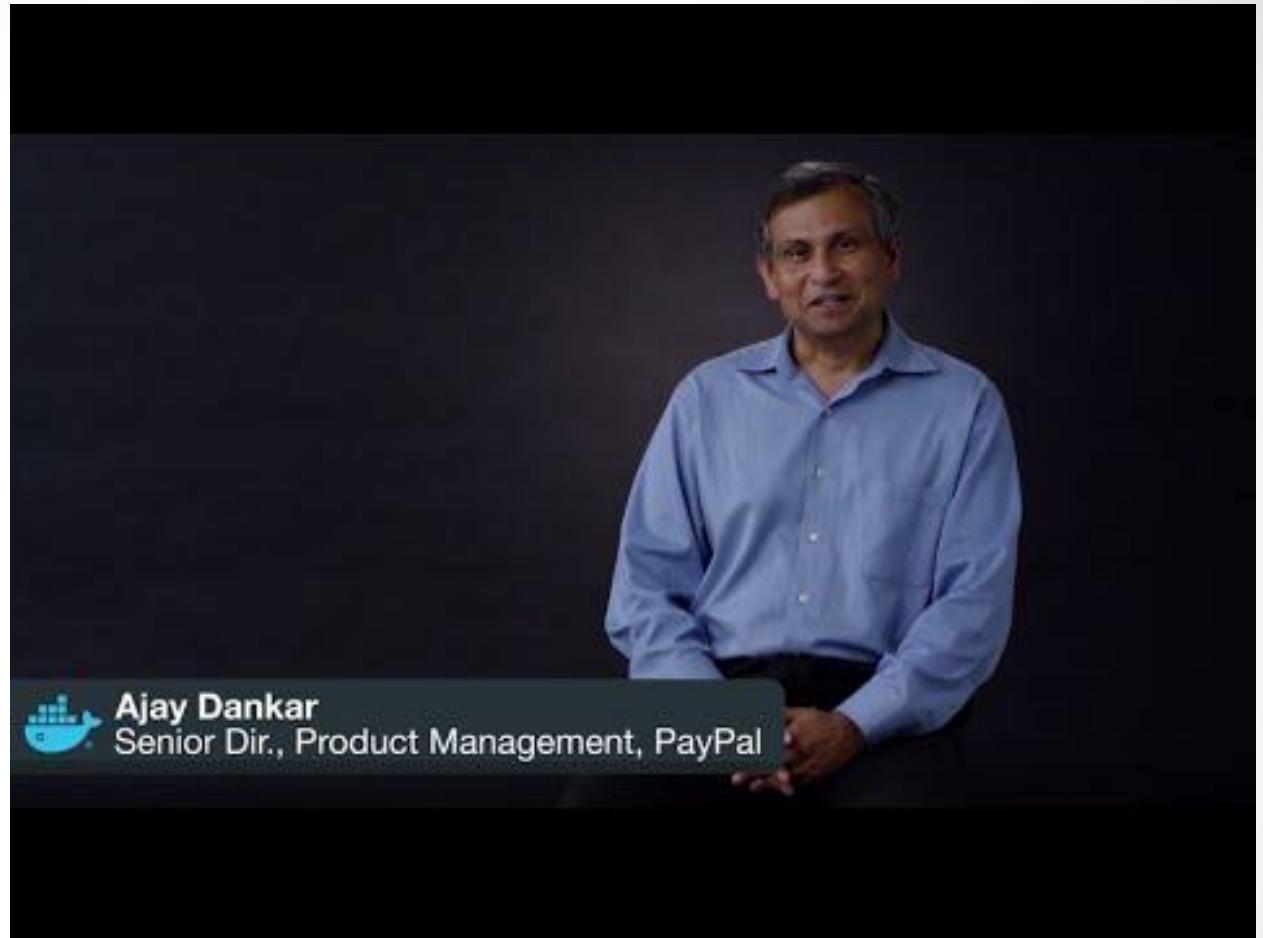


U B E R

Workshop Automate Software Development



Whow use docker ?



Workshop Automate Software Development



What's New

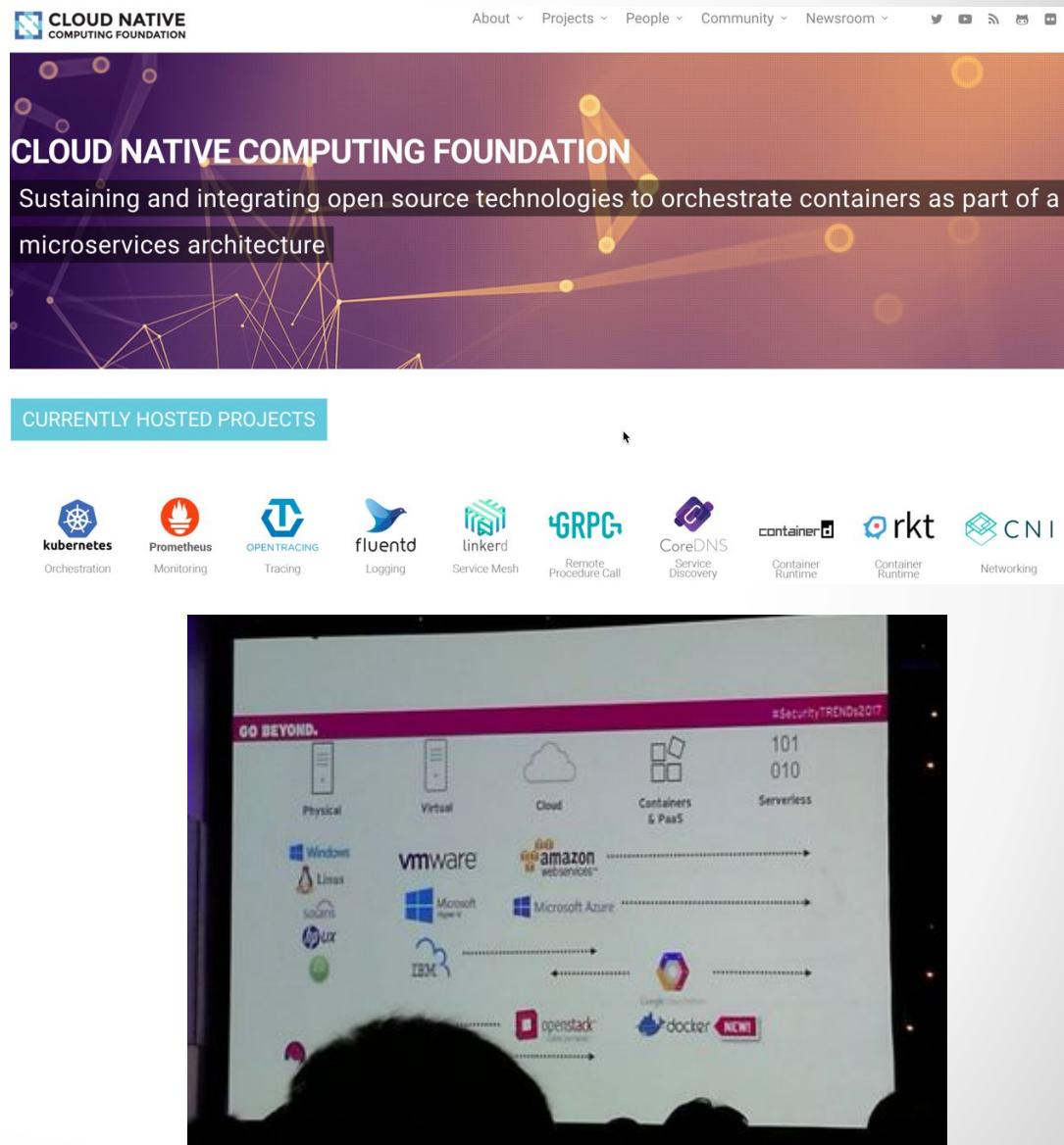


Docker's History of contribution to the OCI

Docker has lead the development of OCI from the initial commit to the donation of runc and Docker V2 Image format as a base for the image format specification.

2014	2015	2016	2017
● FEBRUARY 18 First commit of libcontainer	● APRIL Support for V2 of content addressable Docker Registry API released - groundwork for the OCI Image Specification	● JANUARY Formation of the OCI TOB - 2 Docker maintainers - Michael Crosby and Diogo Monica join effort	● MARCH Docker donates containerd to the CNCF
● FEBRUARY 20 nsinit is developed - the inspiration for runC	● MAY 1 Docker moves to donate its image format and runtime	● APRIL Schema2 of Docker's V2 Image Format support	● APRIL Docker announces project Moby
● MAR 7 Docker 0.9 ships with libcontainer as the default runtime	● MAY / JUNE Docker works with Inaugural participants and Linux Foundation to donate container format/runtime	● JULY 1.0 of OCI runtime and image format	● JULY 1.0 of OCI runtime and image format
● JUNE 22 Docker announces donation of base container format and runtime, runC*, the cornerstone for the OCI	● DECEMBER Docker spins out containerd	● APRIL Docker image format donated to the OCI	● DECEMBER Docker spins out containerd
● DECEMBER 17 Docker announces containerd, a daemon to manage runc			

Workshop Automate Software Development



The Cloud Native Computing Foundation (CNCF) homepage features a purple background with abstract network nodes and connections. The header includes the CNCF logo and navigation links for About, Projects, People, Community, and Newsroom, along with social media icons.

CLOUD NATIVE COMPUTING FOUNDATION

Sustaining and integrating open source technologies to orchestrate containers as part of a microservices architecture

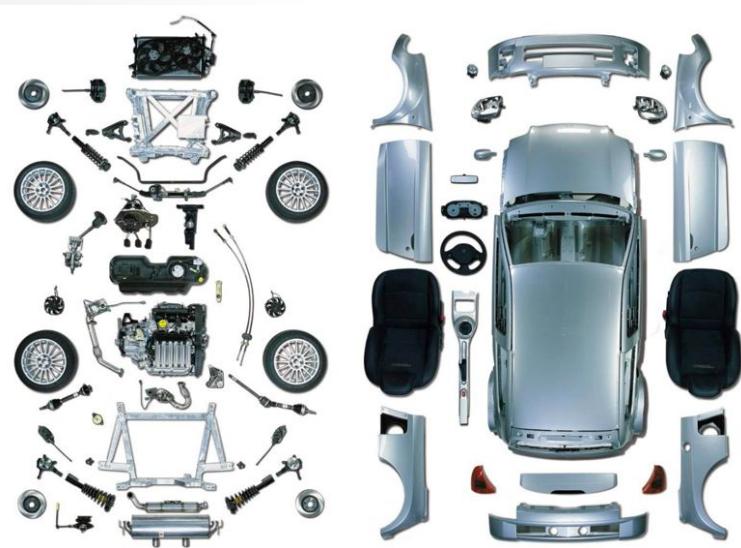
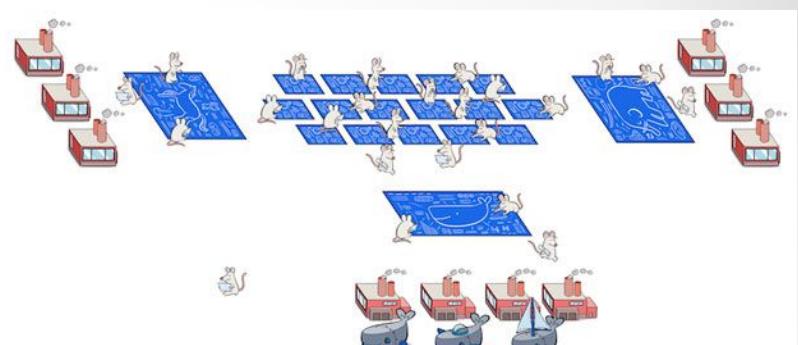
CURRENTLY HOSTED PROJECTS

- kubernetes** Orchestration
- Prometheus** Monitoring
- OPENTRACING** Tracing
- fluentd** Logging
- linkerd** Service Mesh
- gRPC** Remote Procedure Call
- CoreDNS** Service Discovery
- containerd** Container Runtime
- rkt** Container Runtime
- CNI** Networking



What's New

The screenshot shows the homepage of the Moby Project. At the top, there is a navigation bar with links for "Secure https://mobyproject.org", "Blog", "Community", and "Projects". Below the navigation bar, the Moby Project logo is displayed, which is a white circle containing a stylized blue whale. To the right of the logo, the text "Moby Project" is written in a large, bold, white sans-serif font. Below this, a subtitle reads "An open framework to assemble specialized container systems without reinventing the wheel.".



The screenshot shows the "Component Library" section of the Moby project website. The interface has a dark theme with a sidebar on the left containing a navigation menu. The menu items include "Orchestration", "Image Management", "Secret Management", "Configuration Management", "Networking", "Provisioning", and "Your Component here". Above the menu, the "Moby project" logo is visible. To the right of the sidebar, there is a large, dark, three-dimensional graphic of a ship's hull, labeled "Assemblies" and "Moby Tools". The overall design is modern and technical.

Workshop Automate Software Development

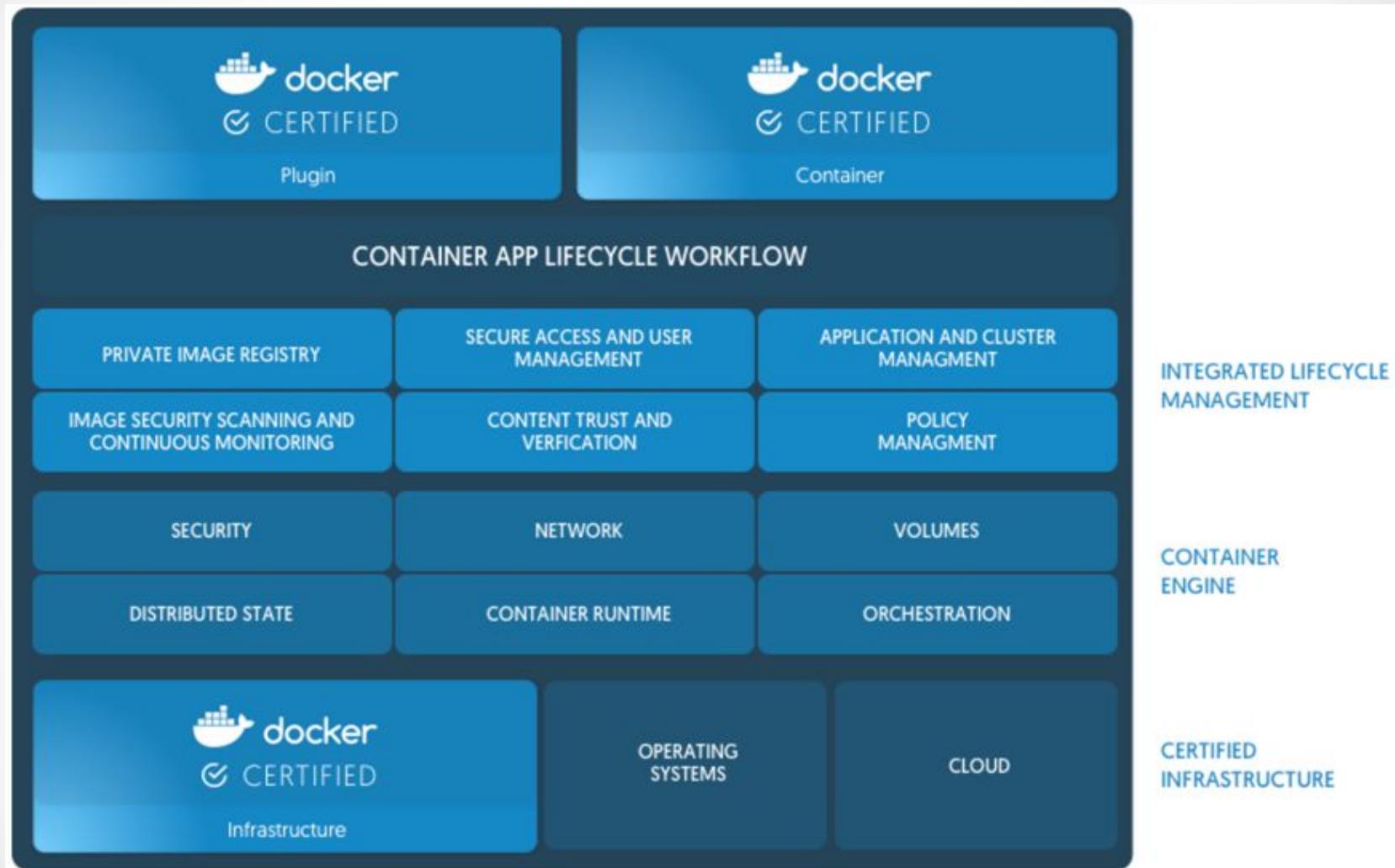


What's New

Docker Pricing Plans

	COMMUNITY EDITION	ENTERPRISE EDITION BASIC	ENTERPRISE EDITION STANDARD	ENTERPRISE EDITION ADVANCED
Container engine and built in orchestration, networking, security	✓	✓	✓	✓
Docker Certified Infrastructure, Plugins and ISV Containers		✓	✓	✓
Image Management (private registry, caching)	Cloud hosted repos		✓	✓
Docker Datacenter Integrated container app management			✓	✓
Docker Datacenter Multi-tenancy with RBAC, LDAP/AD support		↗	✓	✓
Integrated secrets mgmt, image signing policy			✓	✓
Image security scanning	Preview			✓
Support	Community Support	Business Day or Business Critical	Business Day or Business Critical	Business Day or Business Critical

What's New



What's News

17.06.0-ce (2017-06-28)

Note: Docker 17.06.0 has an issue in the image builder causing a change in the behavior of the `ADD` instruction of Dockerfile when referencing a remote `.tar.gz` file. The issue will be fixed in Docker 17.06.1.

Note: Starting with Docker CE 17.06, Ubuntu packages are also available for IBM z Systems using the s390x architecture.

Note: Docker 17.06 by default disables communication with legacy (v1) registries. If you require interaction with registries that have not yet migrated to the v2 protocol, set the `--disable-legacy-registry=false` daemon option. Interaction with v1 registries will be removed in Docker 17.12.

Server

Platform	Docker CE x86_64	Docker CE ARM	Docker CE System Z (s390x)	Docker EE
CentOS	✓			✓
Debian	✓	✓		
Fedora	✓			
Microsoft Windows Server 2016				✓
Oracle Linux				✓
Red Hat Enterprise Linux				✓
SUSE Linux Enterprise Server				✓
Ubuntu	✓	✓	✓	✓

What's News

Secure | <https://docs.microsoft.com/en-us/sql/linux/quickstart-install-connect-docker>

NMac Ked - Mac OS... Medium jenkins vagrant Mesos Vue docker NGINX Taiwan Kubernetes PWA_Progressive_W... MYSQL_Cluster GeneralKB Nodejs

 Microsoft Technologies Documentation Resources

Docs Windows Microsoft Azure Visual Studio Office More

Docs / SQL / SQL Server on Linux

Filter

About SQL Server on Linux

> Overview

Quickstarts

Install & Connect - Red Hat

Install & Connect - SUSE

Install & Connect - Ubuntu

Run & Connect - Docker

> Concepts

> Samples

> Resources

Download PDF

Run the SQL Server 2017 container image with Docker

2017-7-17 • 7 min to read • Contributors

In this quick start tutorial, you use Docker to pull and run the SQL Server 2017 RC1 container image, [mssql-server-linux](#). Then connect with **sqlcmd** to create your first database and run queries.

This image consists of SQL Server running on Linux based on Ubuntu 16.04. It can be used with the Docker Engine 1.8+ on Linux or on Docker for Mac/Windows.

 **Note**

This quick start specifically focuses on using the mssql-server-linux image. The Windows image is not covered, but you can learn more about it on the [mssql-server-windows Docker Hub page](#).

Prerequisites

- Docker Engine 1.8+ on any supported Linux distribution or Docker for Mac/Windows.
- Minimum of 4 GB of disk space
- Minimum of 4 GB of RAM
- [System requirements for SQL Server on Linux](#).

What's News

PUBLIC REPOSITORY

[microsoft/mssql-server-linux](#) 

Last pushed: 21 minutes ago

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

Tag Name	Compressed Size	Last Updated
rc2	482 MB	21 minutes ago
latest	482 MB	21 minutes ago
rc1	478 MB	16 days ago
ctp2-0	451 MB	2 months ago
ctp2-1	462 MB	2 months ago
ctp1-4	382 MB	5 months ago
ctp1-3	370 MB	6 months ago
ctp1-2	367 MB	6 months ago
ctp1-1	340 MB	6 months ago

What's News

Docker with Swarm and Kubernetes

1 →

The best enterprise
container security and
management

Docker Enterprise Edition

Docker Community Edition



3 →

Native Kubernetes
integration provides full
ecosystem
compatibility

← 2

The best container
development workflow

← 4

Industry-standard
container runtime

containerd

dockercon^{EU}17

Workshop Automate Software Development



What's News

Docker Enterprise Edition

Management for Swarm and Kubernetes

Features		Swarm Support	Kubernetes Support
100% Interoperability	<ul style="list-style-type: none">- <i>Clean upstream integration</i>- <i>Full ecosystem compatibility</i>	✓	✓
Secure Cluster Lifecycle	<ul style="list-style-type: none">- <i>Easy High Availability provisioning</i>- <i>Cryptographic node identity</i>	✓	✓
Secure Supply Chain	<ul style="list-style-type: none">- <i>Registry</i>- <i>Content Trust</i>- <i>Secure Scanning</i>	✓	✓
Secure Multi-tenancy	<ul style="list-style-type: none">- <i>Role Based Access Control</i>- <i>Authorization, Authentication</i>- <i>Node Segmentation</i>	✓	✓
Management Dashboard		✓	✓
Supported and Certified on Windows Server and Major Linux Distributions		✓	✓

What's News

An Open Source Integration 1 Year in the Making



Cloud Support Docker



The screenshot shows the Docker Cloud Cloud Settings page. It displays a sidebar with options like General, Cloud providers, Source providers, Notifications, Default Privacy, Billing, Plan, Coupon Codes, and Quotas. The main content area shows a user profile for "labdocker" (Member since Feb 08, 2016) and a table titled "Cloud providers" listing various cloud services with their status and tier information.

Cloud provider	Action	Status	Tier		
Amazon Web Services	Add new credentials			Free Tier	
Digital Ocean	Add new credentials			\$20 Code	
Microsoft Azure	Add new credentials			Global Admin	Free trial
SoftLayer	Add new credentials			Free trial	
Packet	Add new credentials			\$25 code	



DigitalOcean



SOFTLAYER®

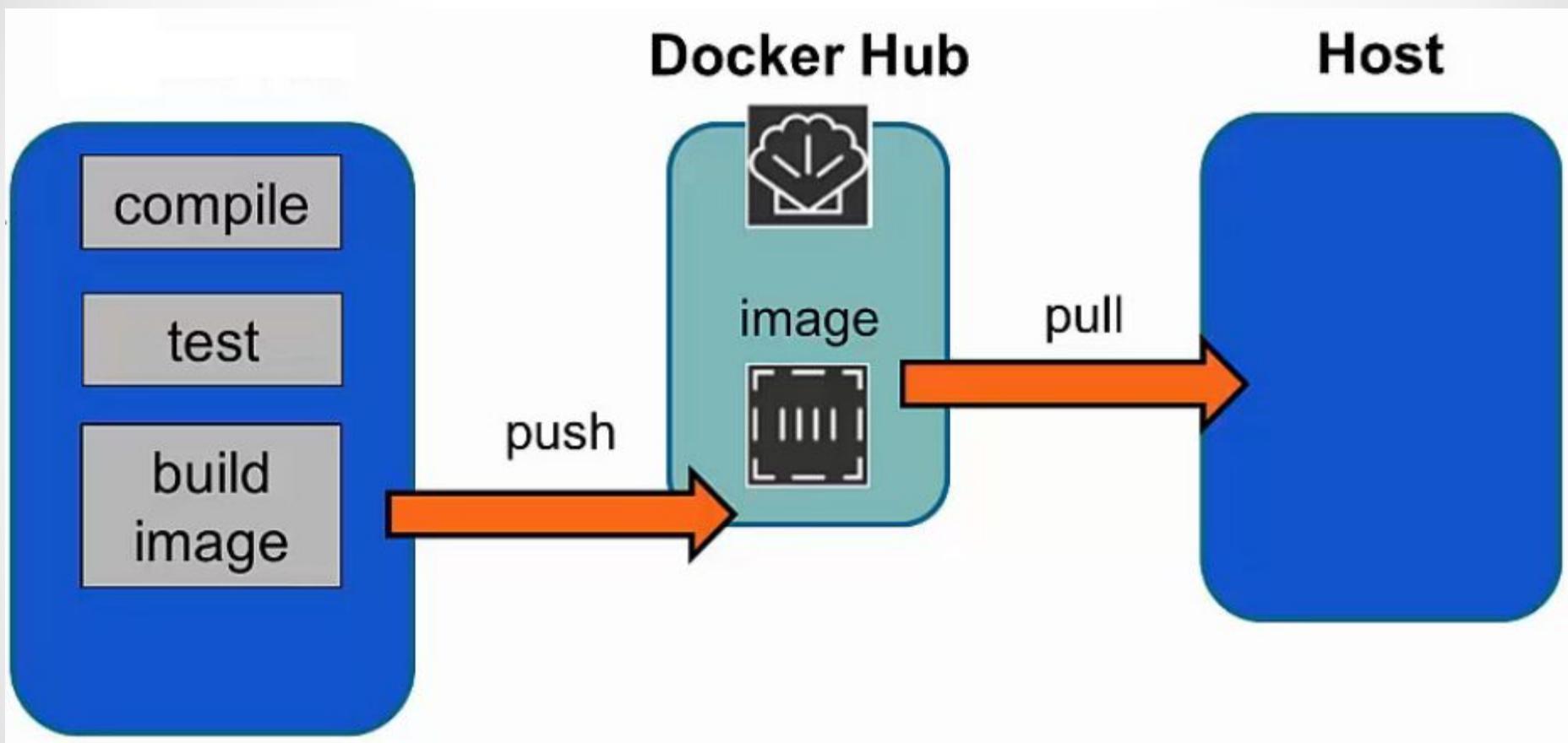


Workshop Automate Software Development

Image, Repository and Container

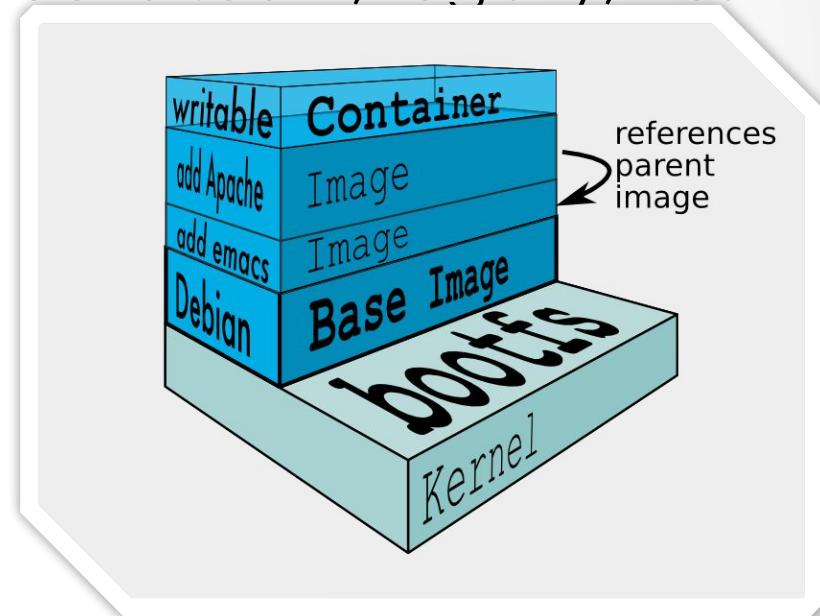
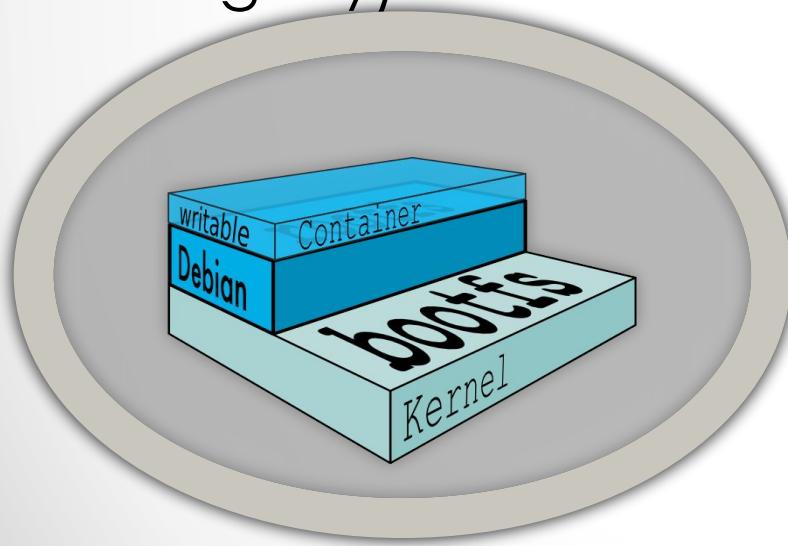
• • •

Docker Deployment



Docker Image

- Image คือ template ที่ถูกสร้างขึ้นเพื่อเตรียมใช้ในการรัน container
- เป็นไฟล์ที่อ่านได้อย่างเดียว
- ถูกสร้างโดยผู้ใช้งานเอง หรือผู้อื่น
- จัดเก็บไว้ใน repository (hub.docker.com, registry, trust registry)



Docker Image

- เรียกดู image ที่อยู่ภายในเครื่อง

```
docker images
```

```
[praparns-MacBook-Pro% docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
phpmyadmin/phpmyadmin	latest	f46764abb293	2 hours ago	103MB
3dsinteractive/mysql	5.7	a5f6e6bdba9a	10 hours ago	492MB
3dsinteractive/php-fpm	7.1	bc469a8c043d	5 days ago	227MB
3dsinteractive/composer	7.1	f09d0ce540e8	8 days ago	313MB
3dsinteractive/nginx	1.12	4fb76106ec6f	3 weeks ago	106MB

```
praparns-MacBook-Pro%
```

- tag image ใหม่เพื่อให้ตรงตามความต้องการในการใช้งาน

```
docker tag <image id> <acc name/imagename: version>
```

Workshop 2: Pull Image

ทำการ download image ลงมาที่เครื่องของเราเพื่อเตรียมความพร้อมในการทำงาน

docker images

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
phpmyadmin/phpmyadmin	latest	f46764abb293	3 hours ago	103MB
3dsinteractive/mysql	5.7	a5f6e6bdba9a	10 hours ago	492MB
3dsinteractive/php-fpm	7.1	bc469a8c043d	5 days ago	227MB
3dsinteractive/composer	7.1	f09d0ce540e8	8 days ago	313MB
3dsinteractive/nginx	1.12	4fb76106ec6f	3 weeks ago	106MB

Repository (Registry)

- Hub.docker.com

The screenshot shows the homepage of the Docker Hub website (<https://hub.docker.com>). The page has a dark blue header with a navigation bar containing 'Explore' and 'Help' links, and a search bar with the placeholder 'Search'. On the right side of the header is a 'Log In' button. Below the header, there's a large call-to-action section with the text 'Build, Ship, & Run Any App, Anywhere' in white and blue, followed by the subtext 'Dev-test pipeline automation, 100,000+ free apps, public and private registries'. To the right of this, under the heading 'New to Docker?', there are three input fields for creating a Docker ID: 'Choose a Docker Hub ID', 'Enter your email address', and 'Choose a password'. At the bottom of the page, there's a footer with the text 'Create your free Docker ID to get started.'

Repository (Registry)

Screenshot of a Docker Registry interface showing a public repository for '3dsinteractive/nginx'.

PUBLIC REPOSITORY

3dsinteractive/nginx 

Last pushed: 24 days ago

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

Short Description	 Short description is empty for this repo.
Full Description	 Full description is empty for this repo.
Docker Pull Command	 <code>docker pull 3dsinteractive/nginx</code>
Owner	 3dsinteractive

Comments (0)

[Add Comment](#)

Repository (Registry)

- download image ອອກຈາກ registry

```
docker pull <account name>/image name: tags>
```

Ex: docker pull labdocker/alpine:latest

- Tag image from existing image
 - Login

```
docker tag <existing image> <new image name>
```

Ex: docker tag 3dsinteractive/nginx:1.12 praparn/nginx:1.12

Workshop 3: Tag Image

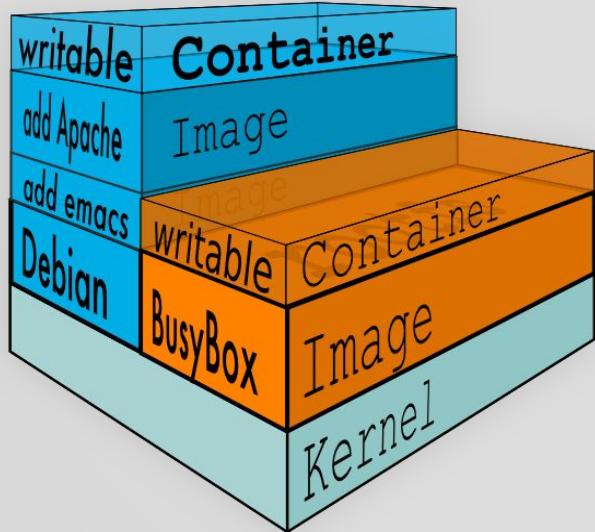
- ใน workshop นี้เราจะดำเนินการสร้าง Tag Image nginx เพื่อให้เป็นชื่อเฉพาะสำหรับงานของเรา

```
|paparns-MacBook-Pro% docker images
REPOSITORY          TAG        IMAGE ID      CREATED       SIZE
phpmyadmin/phpmyadmin  latest    f46764abb293  3 hours ago  103MB
3dsinteractive/mysql   5.7      a5f6e6bdba9a  10 hours ago 492MB
3dsinteractive/php-fpm  7.1      bc469a8c043d  5 days ago   227MB
3dsinteractive/composer 7.1      f09d0ce540e8  8 days ago   313MB
3dsinteractive/nginx    1.12     4fb76106ec6f  3 weeks ago  106MB
|paparns-MacBook-Pro% docker tag 3dsinteractive/nginx:1.12 paparn/nginx:1.12
|paparns-MacBook-Pro% docker images
REPOSITORY          TAG        IMAGE ID      CREATED       SIZE
phpmyadmin/phpmyadmin  latest    f46764abb293  3 hours ago  103MB
3dsinteractive/mysql   5.7      a5f6e6bdba9a  10 hours ago 492MB
3dsinteractive/php-fpm  7.1      bc469a8c043d  5 days ago   227MB
3dsinteractive/composer 7.1      f09d0ce540e8  8 days ago   313MB
3dsinteractive/nginx    1.12     4fb76106ec6f  3 weeks ago  106MB
paparn/nginx           1.12     4fb76106ec6f  3 weeks ago  106MB
paparns-MacBook-Pro%
```



Docker Container

- Container คือชุดของ software layer ที่รันอิสระโดยอ้างอิงจาก image (run)
- ใช้สำหรับสร้างสภาพแวดล้อมที่จำเป็นต้องใช้ในการรันหรือพัฒนาโปรแกรม
- เมื่อพัฒนาโปรแกรมเสร็จเรียบร้อยแล้ว หรือต้องการ backup container สามารถสั่งเก็บ container ชุดปัจจุบันไปเป็น image เพื่อรอการเรียกใช้งานต่อไป (commit)



Docker Container

- ลั่ง run docker เพื่อสร้าง container จาก image file

```
docker run <option> <image id/name> <command>
```

Ex: docker run -dt --name nginxtest \
-p 80:8080 -p 443:8443 3dsinteractive/nginx:1.12

- ลั่ง exec command ไปบน container ที่รันอยู่

```
docker exec -it <container id/name> <command>
```

- เริ่ม/หยุด container

```
docker <start/stop> <container id/name>
```

- ลบ container

```
docker rm <containerid/name>
```

Workshop 4: Run NGINX Container

- ลั่งรัน container NGINX โดยตั้งชื่อว่า nginxtest และ map network 80:8080 (http), 443:8443 (https)

```
docker run -dt --name nginxtest \
-p 80:8080 -p 443:8443 3dsinteractive/nginx:1.12
```

- ทดสอบเปิดหน้าเว็บบน url: <http://localhost>, https://localhost
- ตรวจสอบไฟล์ใน container ด้วยคำสั่ง

```
docker exec -it nginxtest sh
more index.html
```

Volume

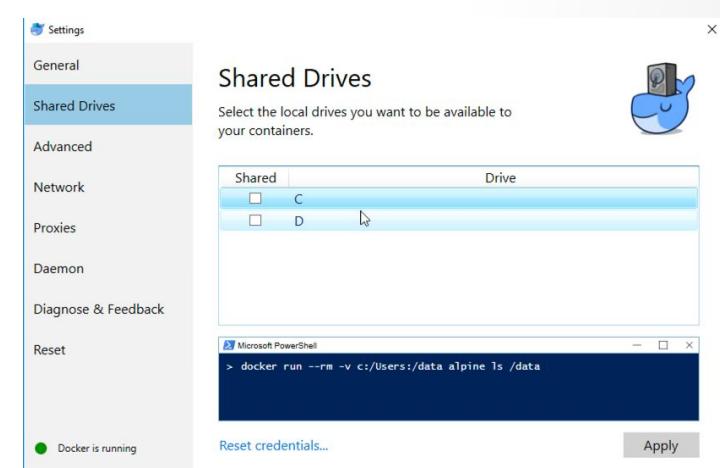
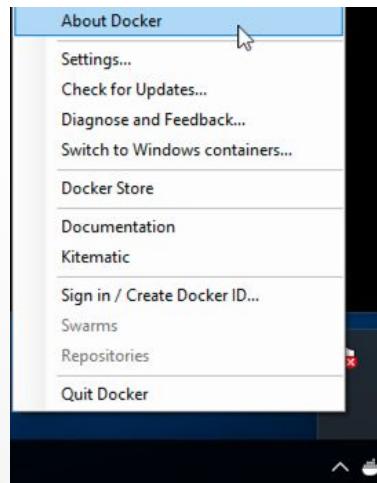
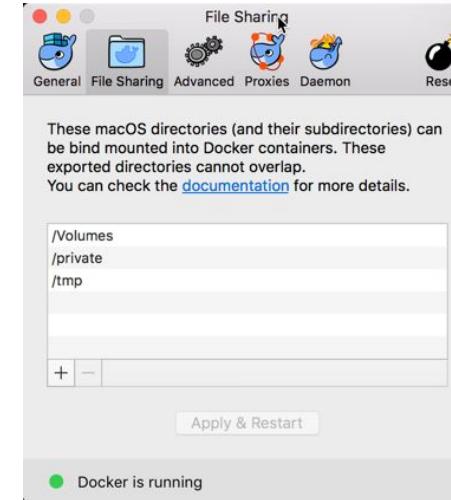
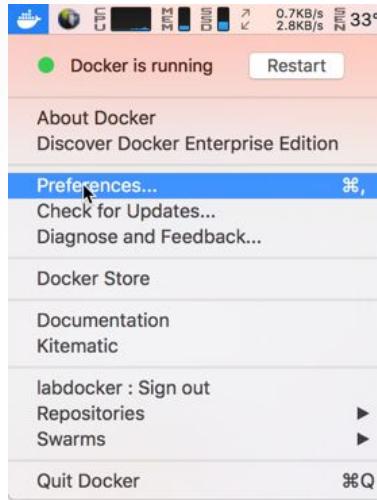
• • •

Volume

- ตามปกติแล้ว docker จะเก็บข้อมูลทุกอย่างเอาไว้ภายใน container
 - /var/log
 - /sys/data
 - /etc/nagios/
 - /etc/mysql
 - Etc.
- สำหรับการใช้งาน container ใน production environment docker ได้สร้าง tool สำหรับอำนวยความสะดวกในการใช้งานข้อมูลร่วมกันในหลายๆกรณี อาทิเช่น
 - การ share ข้อมูลร่วมกันระหว่าง container
 - การ share ข้อมูลร่วมกันระหว่าง container / host
 - จัดเก็บ configuration / log / data ของทุกๆ container ไว้ที่ศูนย์กลาง
 - data migration
 - backup / restore

Volume

- Docker for MAC/Docker for Windows (Feature)



Volume

- การ map volume ระหว่าง host directory และ container directory

```
-v /volume-host:/volume-container:(ro/rw)
```

```
Ex: -v /var/log:/var/log:rw
```

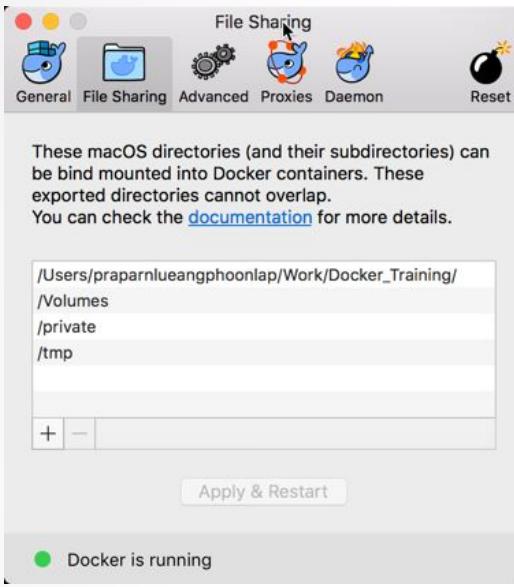
- ตรวจสอบการคอนฟิกบัน container (inspect)

```
Ex: docker run -dt -v /etc/nginx.conf:/etc/nginx.conf:ro
```

Volume

Plugin	Description
Azure File Storage plugin	Lets you mount Microsoft Azure File Storage shares to Docker containers as volumes using the SMB 3.0 protocol. Learn more.
BeeGFS Volume Plugin	An open source volume plugin to create persistent volumes in a BeeGFS parallel file system.
Blockbridge plugin	A volume plugin that provides access to an extensible set of container-based persistent storage options. It supports single and multi-host Docker environments with features that include tenant isolation, automated provisioning, encryption, secure deletion, snapshots and QoS.
Contiv Volume Plugin	An open source volume plugin that provides multi-tenant, persistent, distributed storage with intent based consumption. It has support for Ceph and NFS.
Convoy plugin	A volume plugin for a variety of storage back-ends including device mapper and NFS. It's a simple standalone executable written in Go and provides the framework to support vendor-specific extensions such as snapshots, backups and restore.
DigitalOcean Block Storage plugin	Integrates DigitalOcean's block storage solution into the Docker ecosystem by automatically attaching a given block storage volume to a DigitalOcean droplet and making the contents of the volume available to Docker containers running on that droplet.
DRBD plugin	A volume plugin that provides highly available storage replicated by DRBD . Data written to the docker volume is replicated in a cluster of DRBD nodes.
Flocker plugin	A volume plugin that provides multi-host portable volumes for Docker, enabling you to run databases and other stateful containers and move them around across a cluster of machines.
Fuxi Volume Plugin	A volume plugin that is developed as part of the OpenStack Kuryr project and implements the Docker volume plugin API by utilizing Cinder, the OpenStack block storage service.
gce-docker plugin	A volume plugin able to attach, format and mount Google Compute persistent-disks .
GlusterFS plugin	A volume plugin that provides multi-host volumes management for Docker using GlusterFS.
Horcrux Volume Plugin	A volume plugin that allows on-demand, version controlled access to your data. Horcrux is an open-source plugin, written in Go, and supports SCP, Minio and Amazon S3.
HPE 3Par Volume Plugin	A volume plugin that supports HPE 3Par and StoreVirtual iSCSI storage arrays.
IPFS Volume Plugin	An open source volume plugin that allows using an ipfs filesystem as a volume.
Keywhiz plugin	A plugin that provides credentials and secret management using Keywhiz as a central repository.
Local Persist Plugin	A volume plugin that extends the default local driver's functionality by allowing you specify a mountpoint anywhere on the host, which enables the files to <i>always persist</i> , even if the volume is removed via docker volume rm.
NetApp Plugin(nDVP)	A volume plugin that provides direct integration with the Docker ecosystem for the NetApp storage portfolio. The nDVP package supports the provisioning and management of storage resources from the storage platform to Docker hosts, with a robust framework for adding additional platforms in the future.
Netshare plugin	A volume plugin that provides volume management for NFS 3/4, AWS EFS and CIFS file systems.
Nimble Storage Volume Plugin	A volume plug-in that integrates with Nimble Storage Unified Flash Fabric arrays. The plug-in abstracts array volume capabilities to the Docker administrator to allow self-provisioning of secure multi-tenant volumes and clones.
OpenStorage Plugin	A cluster-aware volume plugin that provides volume management for file and block storage solutions. It implements a vendor neutral specification for implementing extensions such as CoS, encryption, and snapshots. It has example drivers based on FUSE, NFS, NBD and EBS to name a few.
Portworx Volume Plugin	A volume plugin that turns any server into a scale-out converged compute/storage node, providing container granular storage and highly available volumes across any node, using a shared-nothing storage backend that works with any docker scheduler.
Quobyte Volume Plugin	A volume plugin that connects Docker to Quobyte 's data center file system, a general-purpose scalable and fault-tolerant storage platform.
REX-Ray plugin	A volume plugin which is written in Go and provides advanced storage functionality for many platforms including VirtualBox, EC2, Google Compute Engine, OpenStack, and EMC.
Virtuozzo Storage and Ploop plugin	A volume plugin with support for Virtuozzo Storage distributed cloud file system as well as ploop devices.
VMware vSphere Storage Plugin	Docker Volume Driver for vSphere enables customers to address persistent storage requirements for Docker containers in vSphere environments.

Workshop 5: Mount Source Code Outside



The screenshot shows a code editor with the file 'index.html' open. The content of the file is:

```
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
body {
    width: 35em;
    margin: 0 auto;
    font-family: Tahoma, Verdana, Arial, sans-serif;
}
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and working. Further configuration is required.</p>
<p>For online documentation and support please refer to
<a href="http://nginx.org">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com">nginx.com</a>.</p>
<p><em>Thank you for using nginx.</em></p>
</body>
</html>
```

The screenshot shows a web browser window with the URL <https://localhost>. The page title is 'Welcome to nginx! WorkShop Automate'. The content of the page is:

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

Compose

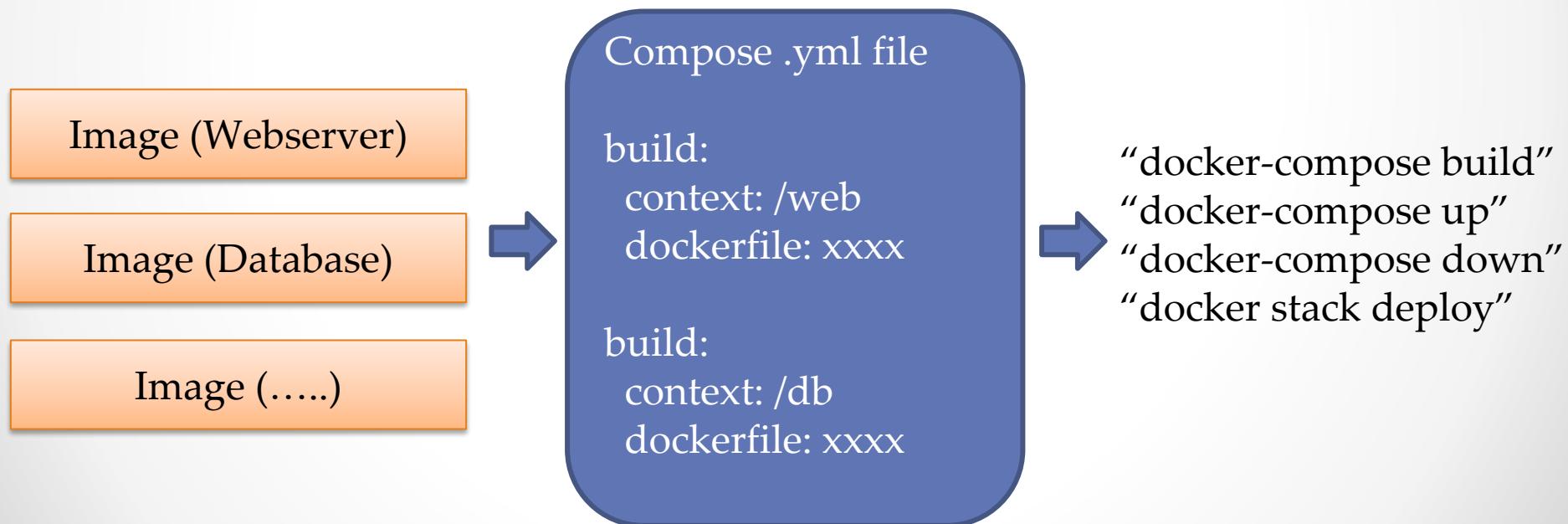
• • •

Compose

- Compose เป็นเครื่องมือที่ใช้ในการสร้าง system service ที่จะใช้ในการรันระบบงานทั้งระบบ ซึ่งตามปกติจะประกอบไปด้วยหลาย component อาทิเช่น
 - Web component service
 - Application component service
 - Database component service
 - Load balance component service
 - etc
- compose สามารถควบคุมการ start / stop / monitor การทำงานของ service ทั้งระบบเป็น single command
- เหมาะสำหรับการสร้าง development environment / test environment / automatic deploy to production (docker-machine / swam) (build one → ship to everywhere)
- *No Support --ip now

Compose

- ขั้นตอนในการสร้าง compose
 - ทำการดึง Image สำหรับแต่ละ container (component)
 - กำหนดค่า running parameter ใน .yml/.yaml file ของ compose ซึ่งจะอ้างอิงถึง docker file แต่ละตัว
 - docker-compose up



Compose

Compose file format	Docker Engine release
3.3	17.06.0+
3.2	17.04.0+
3.1	1.13.1+
3.0	1.13.0+
2.2	1.13.0+
2.1	1.12.0+
2.0	1.10.0+
1.0	1.9.1.+

Compose

- Compose syntax:
 - XXX (service name):
 - depends_on:
 - XXXX
 - XXXX
 - networks:
 - XXXX1
 - XXXX2



Ref:<https://docs.docker.com/compose/compose-file/#command>

Workshop Automate Software Development



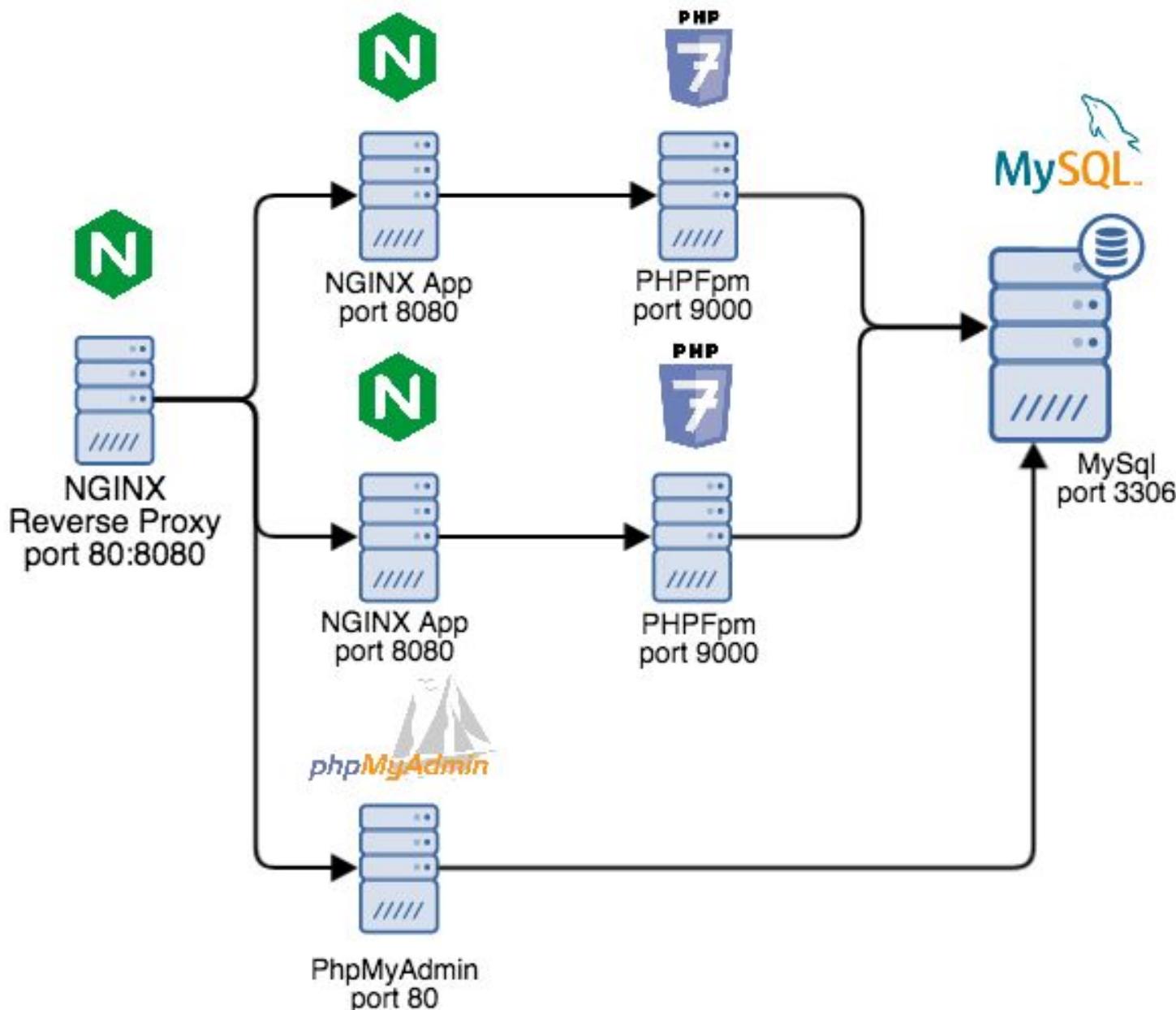
Compose

```
1 version: '2'  
2  
3 services:  
4   lb:  
5     image: 3dsinteractive/nginx:1.12  
6     ports:  
7       - 80:8080  
8     depends_on:  
9       - web1  
10      - web2  
11     volumes:  
12       - ./dockers/web/lb/logs:/opt/bitnami/nginx/logs  
13       - ./dockers/web/lb/nginx:/bitnami/nginx  
14  
15   web1:  
16     image: 3dsinteractive/nginx:1.12  
17     depends_on:  
18       - web1-phpfpm  
19     volumes:  
20       - ./dockers/web/web1/logs:/opt/bitnami/nginx/logs  
21       - ./dockers/web/web1/nginx:/bitnami/nginx  
22       - ./web:/app  
23  
24   web1-phpfpm:  
25     image: 3dsinteractive/php-fpm:7.1  
26     volumes:  
27       - ./dockers/web/web1/phpfpm:/bitnami/php-fpm  
28       - ./web:/app
```

```
27  
28   web2:  
29     image: 3dsinteractive/nginx:1.12  
30     depends_on:  
31       - web2-phpfpm  
32     volumes:  
33       - ./dockers/web/web2/logs:/opt/bitnami/nginx/logs  
34       - ./dockers/web/web2/nginx:/bitnami/nginx  
35       - ./web:/app  
36  
37   web2-phpfpm:  
38     image: 3dsinteractive/php-fpm:7.1  
39     volumes:  
40       - ./dockers/web/web2/phpfpm:/bitnami/php-fpm  
41       - ./web:/app  
42  
43   mysql:  
44     image: 3dsinteractive/mysql:5.7  
45     volumes:  
46       - ./dockers/web/db/data:/bitnami/mysql  
47     environment:  
48       - ALLOW_EMPTY_PASSWORD=yes  
49       - MYSQL_USER=my_user  
50       - MYSQL_PASSWORD=my_password  
51       - MYSQL_DATABASE=my_database  
52  
53   phpmyadmin:  
54     image: phpmyadmin/phpmyadmin  
55     depends_on:  
56       - mysql  
57     environment:  
58       - MYSQL_USERNAME=my_user  
59       - PMA_HOST=mysql
```

Basic Development Stack Workshop

Basic Development Stack.



NGINX

Nginx

อ่านแบบนี้

N-gin-x หรือ Engine-X

Nginx คืออะไร

Nginx คือ เว็บเซิฟเวอร์ [HTTP Server] และ Reverse Proxy Server ที่ออกแบบมาเพื่อแก้ปัญหา C10K

C10K Problem

Concurrent != Requests per second

....

**Concurrent = Network (socket) connections per
second**

Nginx

เป็นมากกว่าเว็บเซิฟเวอร์

Nginx ทำอะไรได้บ้าง

- Web Server [HTTP Server] **
- Reverse Proxy Server **
- Mail Proxy Server
- TCP/UDP Server e.g. LDAP, MySQL, RTMP etc.

Nginx Basic HTTP Server

- Serving Static file
- Load Balancing
- Memcached, FastCGI
- HTTP2
- etc

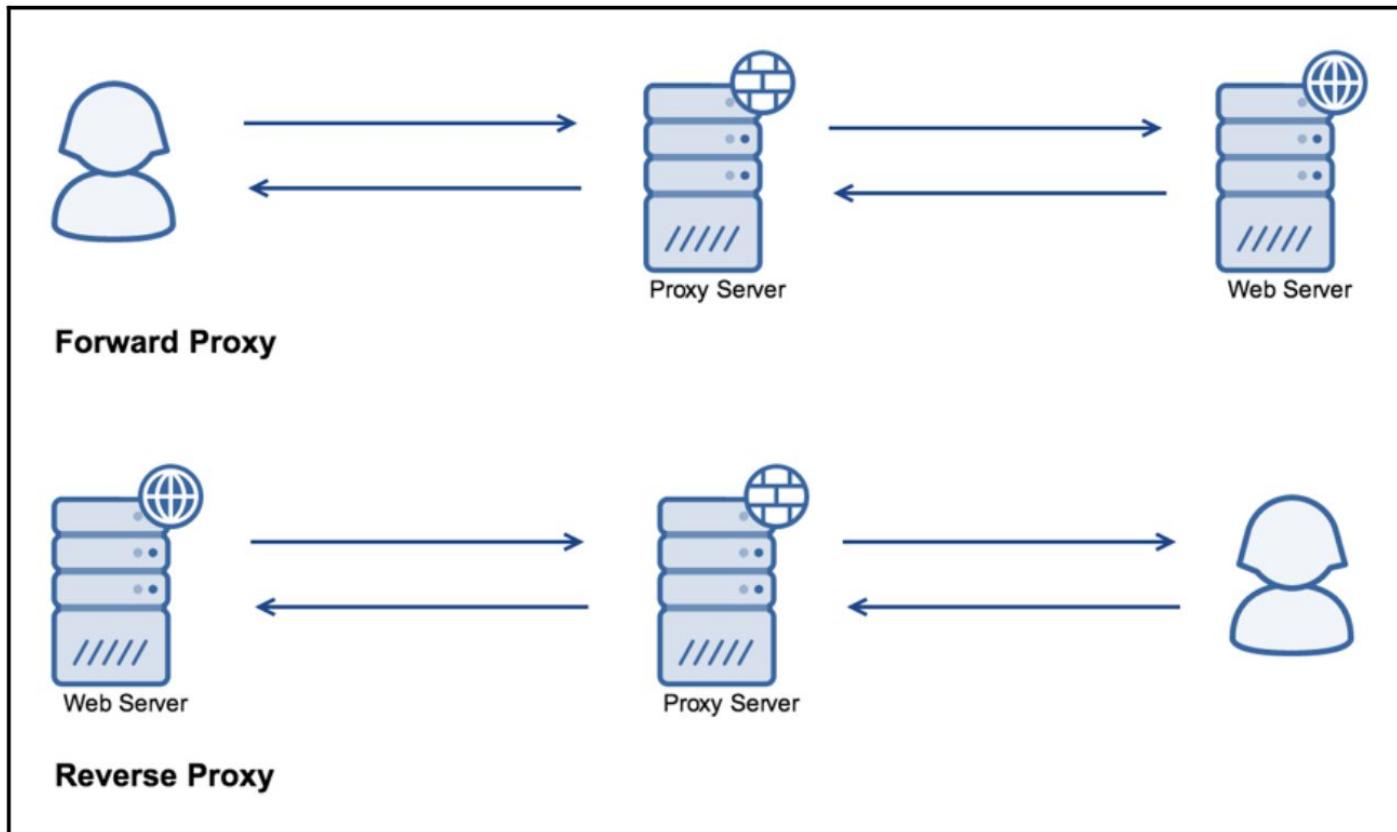
Nginx Reverse Proxy

Reverse Proxy Server

คือ

ตัวที่ทำงานตรงข้ามกับ Proxy Server

Nginx Reverse Proxy



Nginx Reverse Proxy

Example.

```
upstream web {  
    server web1:8080;  
    server web2:8080;  
}  
  
upstream phpmyadmin {  
    server phpmyadmin;  
}
```

Nginx Reverse Proxy

```
server {
    listen 8080;

    server_name laravel-quickstart.local;

    client_max_body_size 20M;
    error_log "/opt/bitnami/nginx/logs/error.log";
    access_log "/opt/bitnami/nginx/logs/access.log";

    location / {
        proxy_pass http://web;
        proxy_redirect off;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For
        $proxy_add_x_forwarded_for;
    }
}
```

Nginx Reverse Proxy

```
server {
    listen 8080;

    server_name laravel-quickstart-db.local;

    client_max_body_size 20M;
    error_log "/opt/bitnami/nginx/logs/phpmyadmin.log";
    access_log "/opt/bitnami/nginx/logs/phpmyadmin.log";

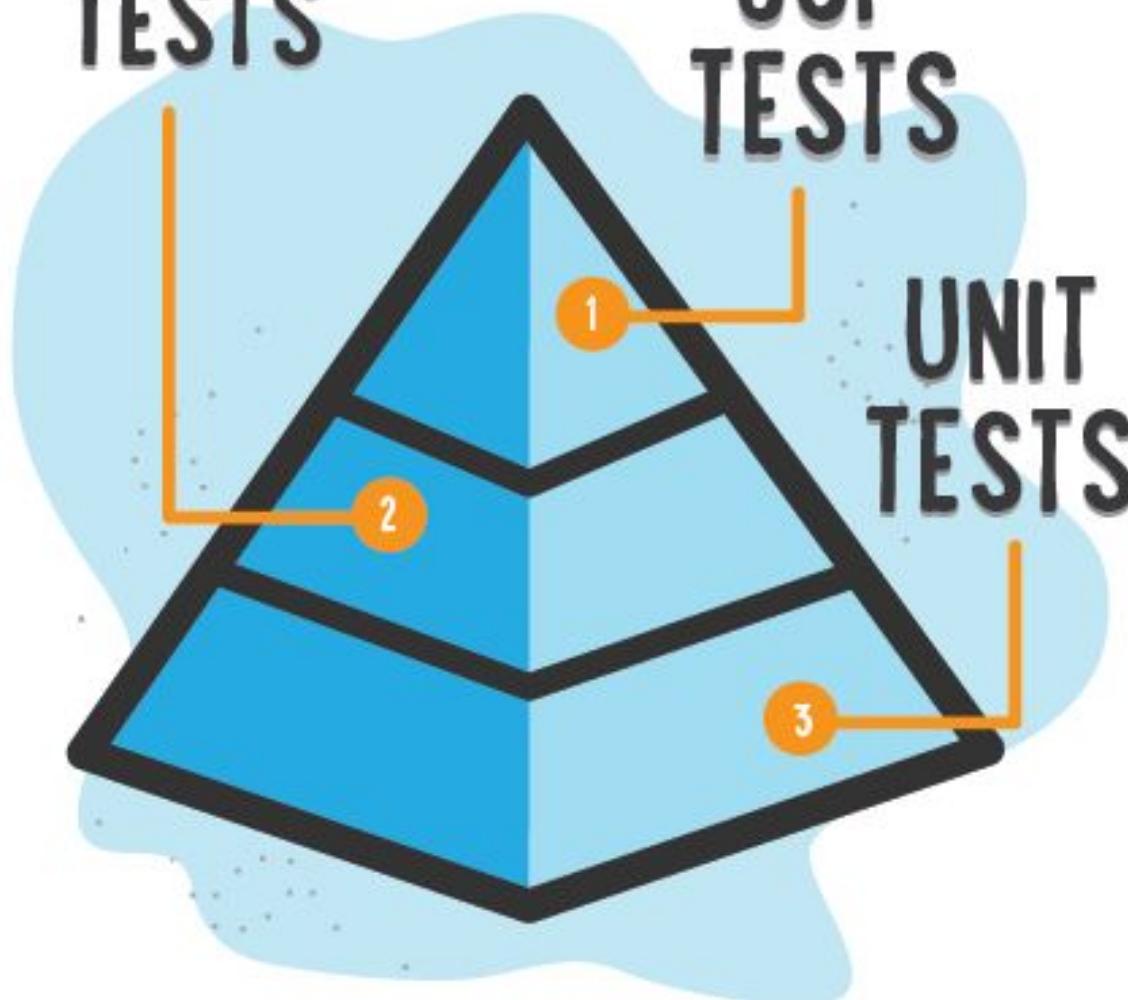
    location / {
        proxy_pass http://phpmyadmin;
        proxy_redirect off;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For
        $proxy_add_x_forwarded_for;
    }
}
```

PHPUNIT

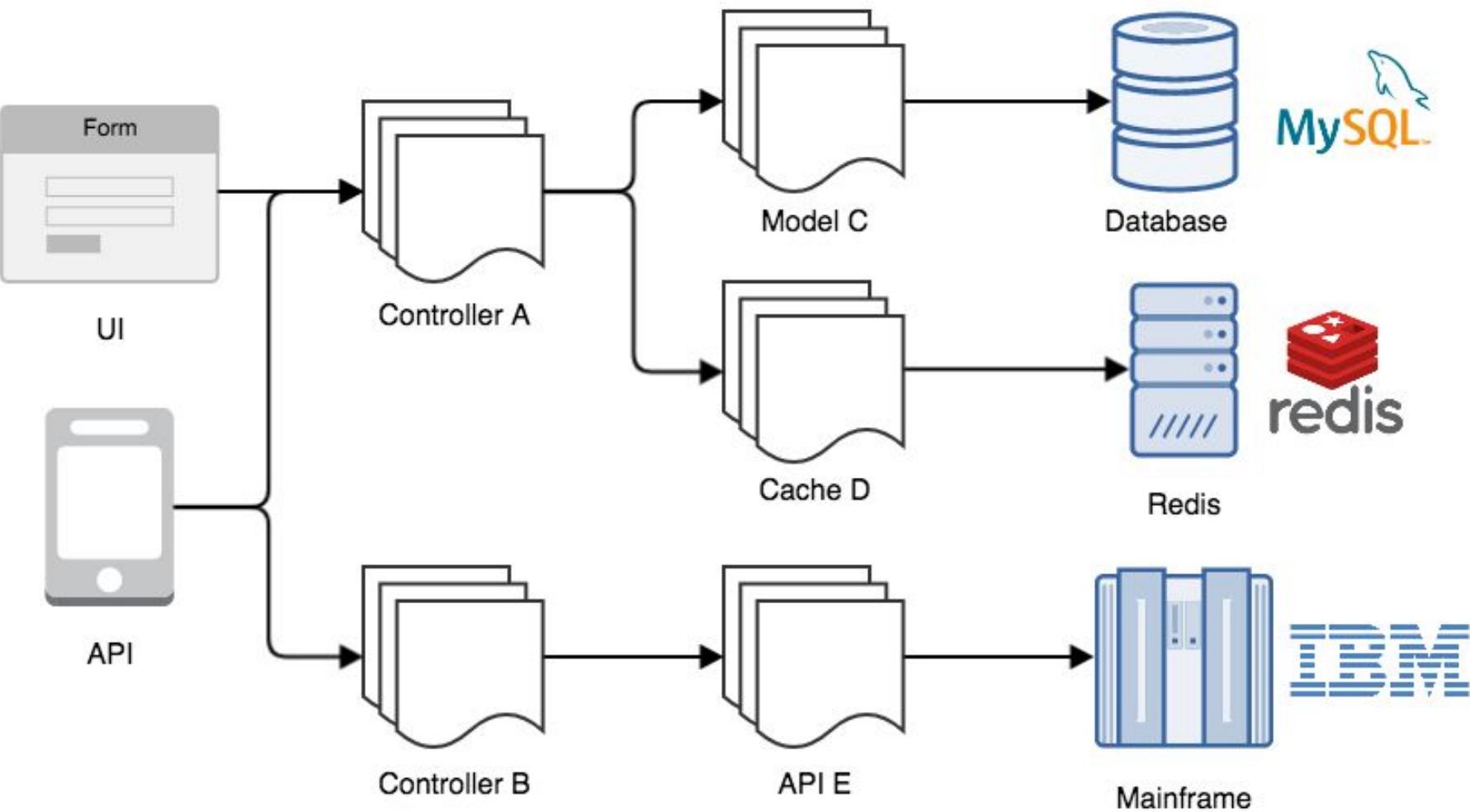
INTEGRATION TESTS

GUI TESTS

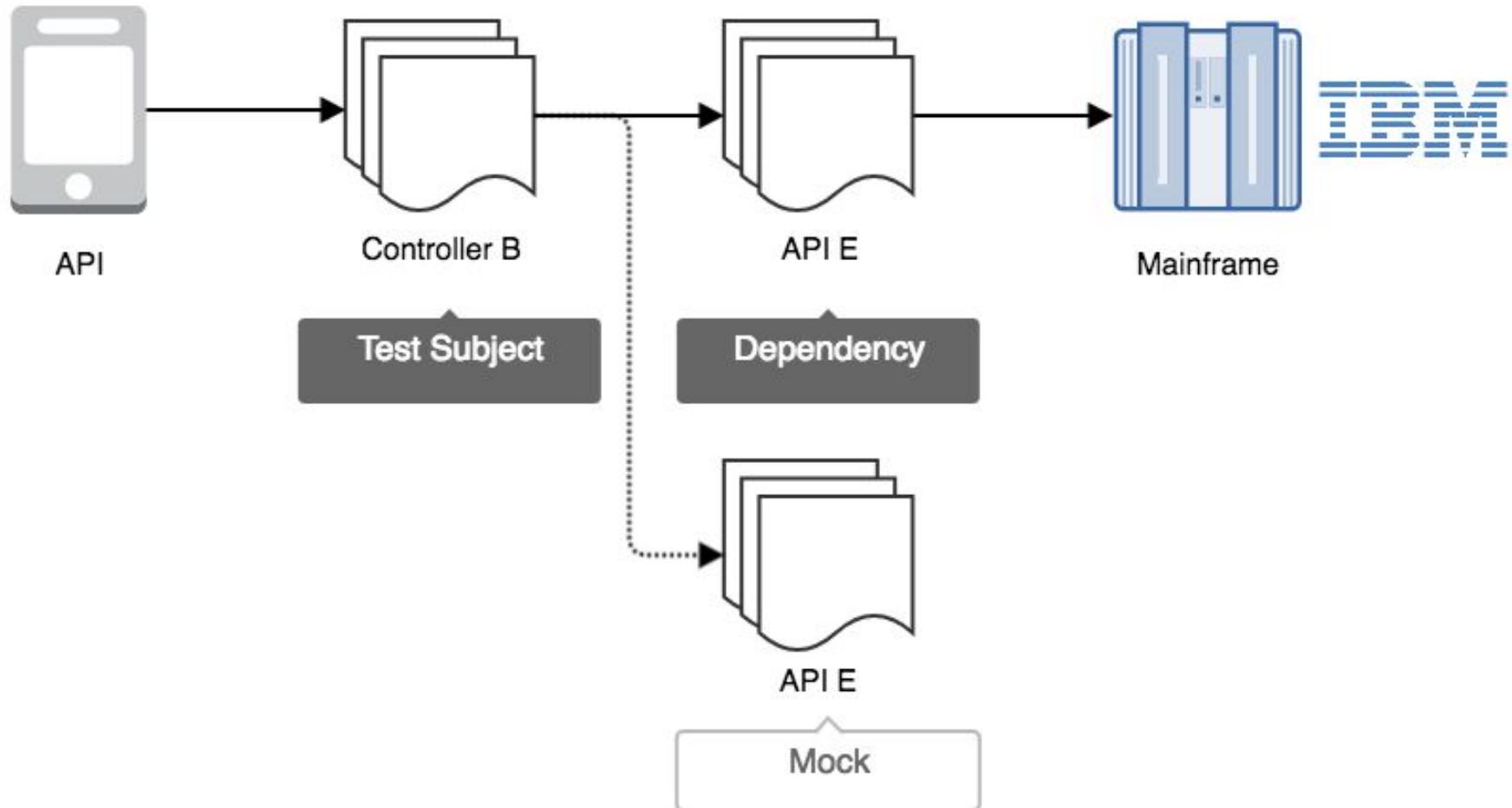
UNIT TESTS



Unit, Functional, UAT



Dependency Injection & Mock

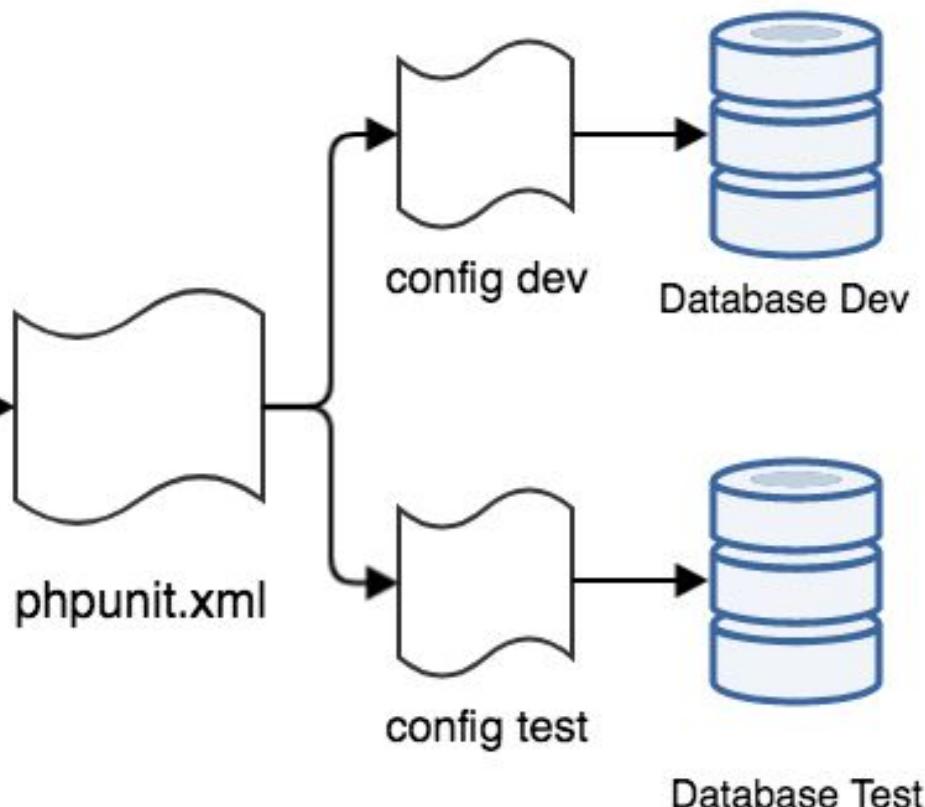


How PHPUnit work.

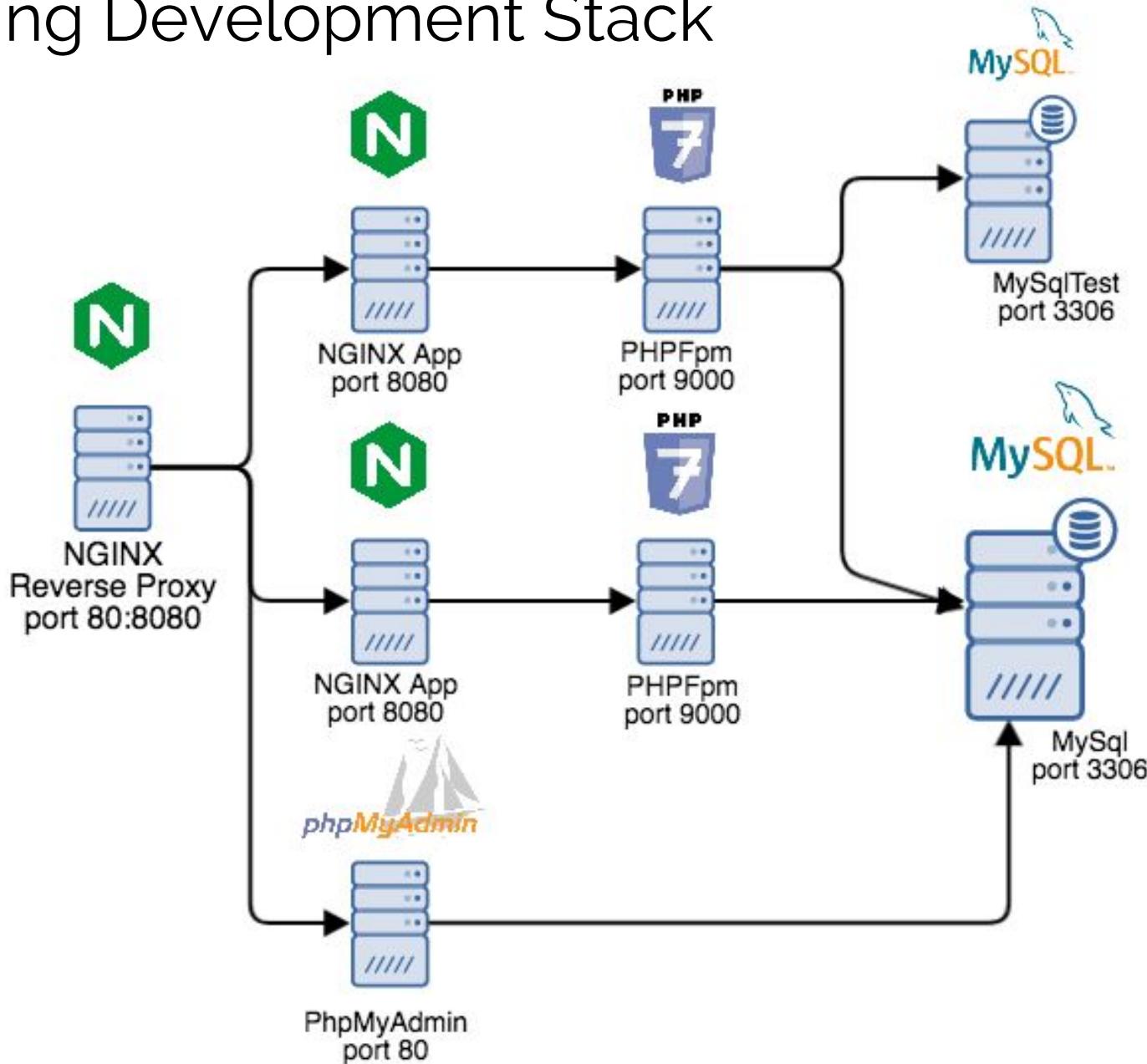
```
1. paullee@Trinitys-MacBook-Pro: ~/Desktop/automat...
PHPUnit 4.8.36 by Sebastian Bergmann and contributors.

Time: 1.3 seconds, Memory: 14.00MB
OK (7 tests, 22 assertions)
→ laravel-quickstart git:(master) x
```

phpunit



Testing Development Stack



PHPUnit Workshop

GitFlow

Getting started

Git flow needs to be initialized in order to customize your project setup.



Initialize

Start using git-flow by initializing it inside an existing git repository:

```
git flow init
```

You'll have to answer a few questions regarding the naming conventions for your branches.

It's recommended to use the default values.



Features

- ★ Develop new features for upcoming releases
- ★ Typically exist in developers repos only



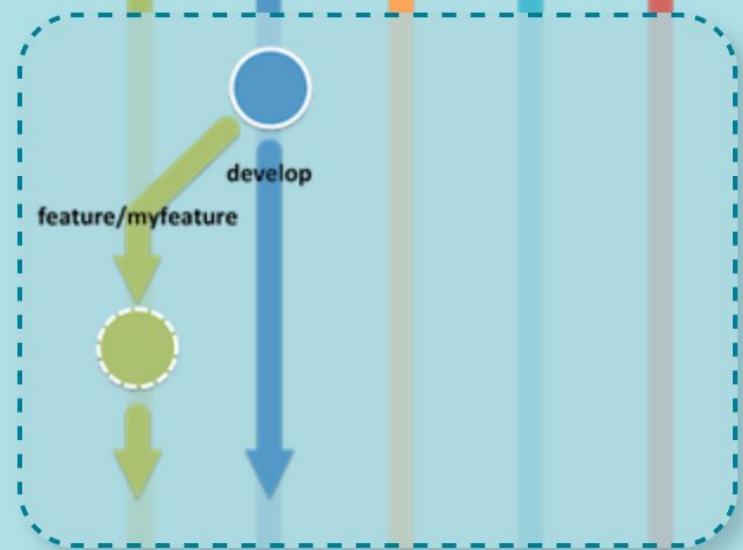
Start a new feature

Development of new features starting from the 'develop' branch.

Start developing a new feature with

```
git flow feature start  
MYFEATURE
```

This action creates a new feature branch based on 'develop' and switches to it

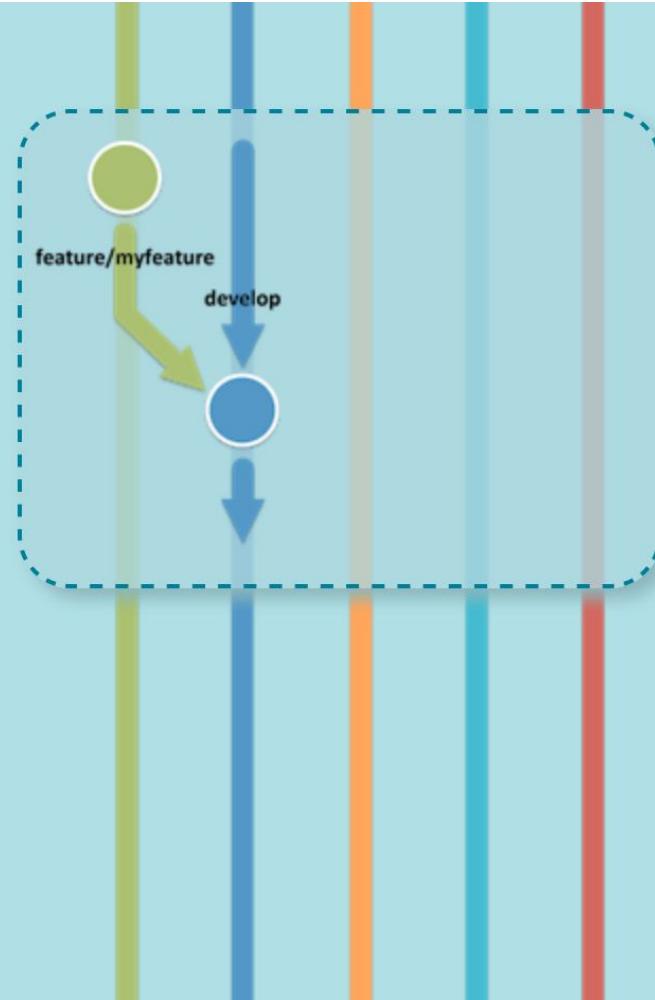


Finish up a feature

Finish the development of a feature. This action performs the following

- ★ Merges MYFEATURE into 'develop'
- ★ Removes the feature branch
- ★ Switches back to 'develop' branch

```
git flow feature finish  
MYFEATURE
```



Make a release

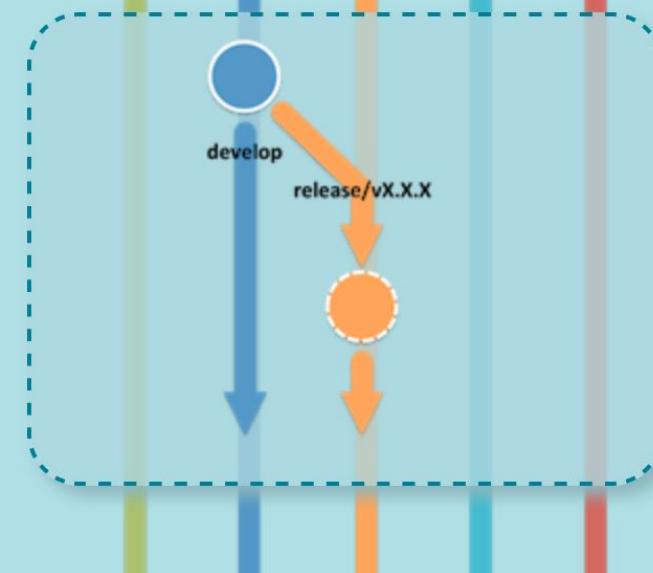
- ★ Support preparation of a new production release
- ★ Allow for minor bug fixes and preparing meta-data for a release



Start a release

To start a release, use the git flow release command. It creates a release branch created from the 'develop' branch.

```
git flow release start  
RELEASE [BASE]
```



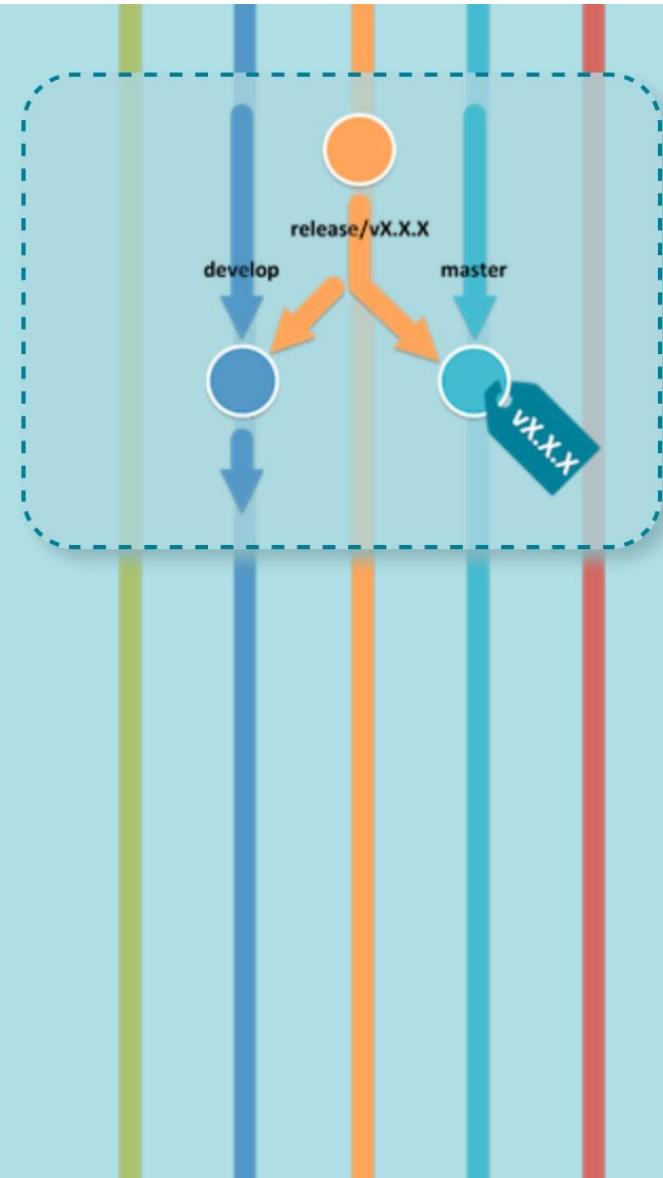
Finish up a release

Finishing a release is one of the big steps in git branching. It performs several actions:

- ★ Merges the release branch back into 'master'
- ★ Tags the release with its name
- ★ Back-merges the release into 'develop'
- ★ Removes the release branch

```
git flow release finish  
RELEASE
```

Don't forget to push your tags with `git push --tags`



Hotfixes

- ★ Hotfixes arise from the necessity to act immediately upon an undesired state of a live production version
- ★ May be branched off from the corresponding tag on the master branch that marks the production version.

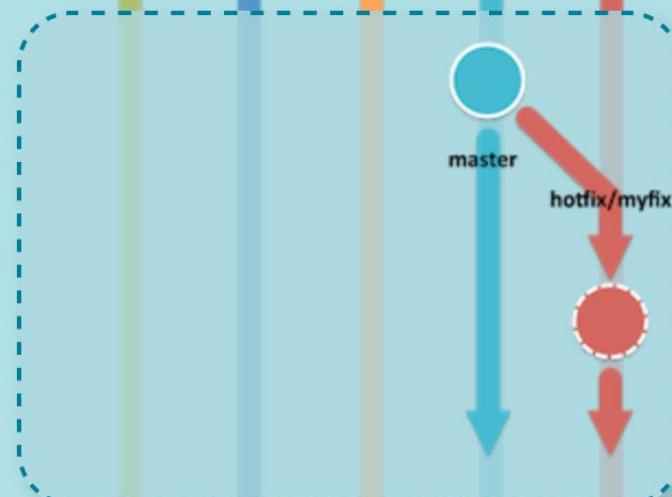
★★★

git flow hotfix start

Like the other git flow commands, a hotfix is started with

```
git flow hotfix start  
VERSION [BASENAME]
```

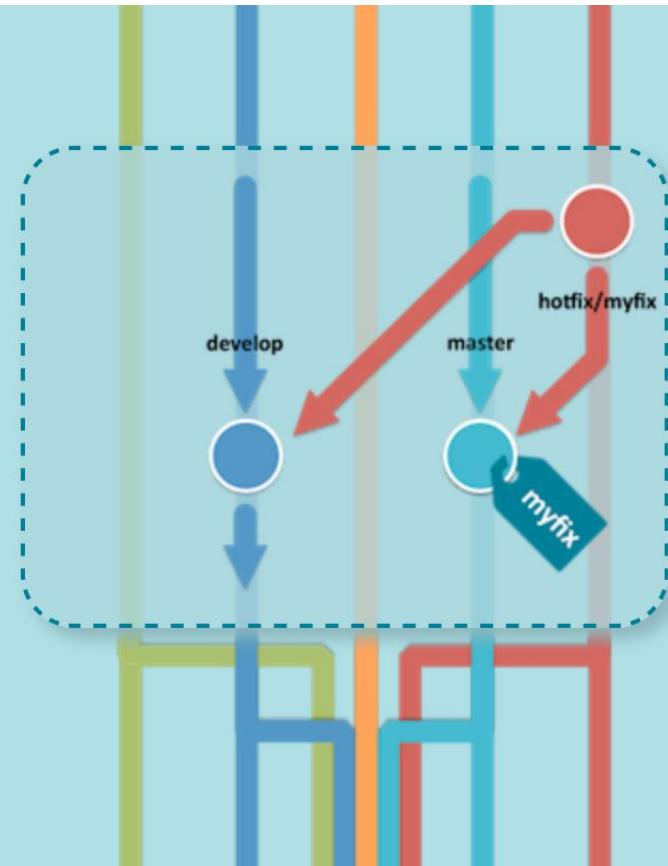
The version argument hereby marks the new hotfix release name. Optionally you can specify a basename to start from.



Finish a hotfix

By finishing a hotfix it gets merged back into develop and master. Additionally the master merge is tagged with the hotfix version.

```
git flow hotfix finish  
VERSION
```



GitFlow workshop

DEPLOY

Jenkins

What is Jenkins?

Jenkins

2 search my_username | log out

ENABLE AUTO REFRESH

New Item People Build History Project Relationship Check File Fingerprint Manage Jenkins My Views Credentials

Add description

S	W	Name ↓	Last Success	Last Failure	Last Duration
		00-update-deploy-scripts	19 hr - #34	1 day 12 hr - #5	1.4 sec
		laravel-quickstart-internal	19 hr - #28	19 hr - #26	1 min 20 sec
		laravel-quickstart-production	1 day 4 hr - #1	N/A	2 min 45 sec
		test-docker	3 days 8 hr - #1	N/A	5.7 sec

Icon: S M L Legend RSS for all RSS for failures RSS for just latest builds

Build Queue

No builds in the queue.

Build Executor Status

1 Idle
2 Idle

What is Jenkins?

Screenshot of the Jenkins web interface for the project "laravel-quickstart-internal".

The top navigation bar includes:

- Jenkins logo
- Project name: laravel-quickstart-internal
- Build count: 2
- Search bar
- User info: my_username | log out
- Enable Auto Refresh link

The left sidebar contains links:

- Back to Dashboard
- Status
- Changes
- Workspace
- Build Now
- Delete Project
- Configure

The main content area shows:

Project laravel-quickstart-internal

Recent items:

- Workspace
- Recent Changes

Permalinks

- Last build (#28), 19 hr ago
- Last stable build (#28), 19 hr ago
- Last successful build (#28), 19 hr ago
- Last failed build (#26), 19 hr ago
- Last unsuccessful build (#26), 19 hr ago
- Last completed build (#28), 19 hr ago

Build History table:

Build History		trend —
find		x
	#28	Oct 26, 2017 5:13 PM
	#27	Oct 26, 2017 5:11 PM
	#26	Oct 26, 2017 5:07 PM
	#25	Oct 26, 2017 5:06 PM
	#24	Oct 26, 2017 5:01 PM
	#23	Oct 26, 2017 4:35 PM
	#22	Oct 26, 2017 4:31 PM
	#21	Oct 26, 2017 4:30 PM
	#20	Oct 26, 2017 4:28 PM
	#19	Oct 26, 2017 4:24 PM

What is Jenkins?

Jenkins > laravel-quickstart-internal >

General Source Code Management Build Triggers Build Environment Bindings Build Post-build Actions

Secret file

Variable SERVER_PEM
Credentials Specific credentials Parameter expression automate-workshop.pem Add

Username and password (separated)

Username Variable GIT_USER
Password Variable GIT_PASS
Credentials Specific credentials Parameter expression pong3ds***** Add

Add ▾ With Ant

Build

Execute shell

Command /bitnami/jenkins/1001/deploy-scripts/laravel-quickstart/test/deploy.sh internal test develop

Save Apply

This screenshot shows the Jenkins configuration interface for a job named 'laravel-quickstart-internal'. The 'Bindings' tab is active. There are two secret file bindings: 'SERVER_PEM' (using specific credentials 'automate-workshop.pem') and 'GIT_USER/GIT_PASS' (using specific credentials 'pong3ds*****'). Below the bindings is a 'Build' section containing an 'Execute shell' step with the command '/bitnami/jenkins/1001/deploy-scripts/laravel-quickstart/test/deploy.sh internal test develop'. The 'Save' and 'Apply' buttons are visible at the bottom of the build step.

What is Jenkins?

Jenkins

2 search my_username log out

Jenkins > laravel-quickstart-internal > #28

Back to Project Status Changes Console Output View as plain text Edit Build Information Delete Build Previous Build

Console Output

```
Started by user my_username
Building in workspace /bitnami/jenkins/jobs/laravel-quickstart-internal/workspace
[workspace] $ /bin/sh -xe /opt/bitnami/tomcat/temp/jenkins4663442070630625013.sh
+ /bitnami/jenkins/1001/deploy-scripts/laravel-quickstart/test/deploy.sh internal test develop
Already up-to-date.
Already on 'test'
Your branch is up-to-date with 'origin/test'.
Already up-to-date.
Already on 'internal'
Your branch is up-to-date with 'origin/internal'.
Already up-to-date.
Already on 'develop'
Your branch is up-to-date with 'origin/develop'.
sending incremental file list
deleting .env
config/database.php

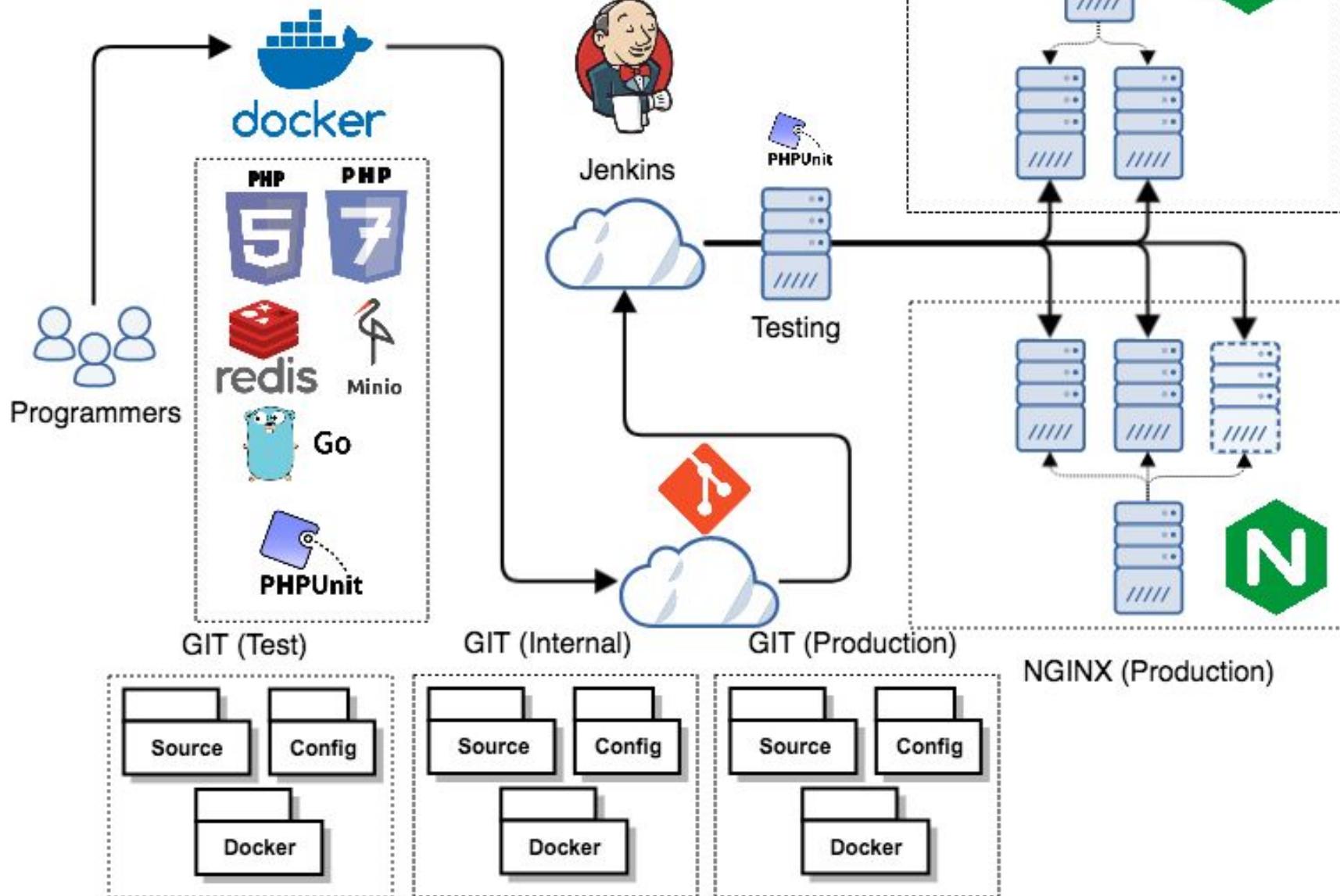
        4,217 100%    0.00kB/s   0:00:00
        4,217 100%    0.00kB/s   0:00:00 (xfr#1, ir-chk=1007/1075)
sending incremental file list
.env

        541 100%    0.00kB/s   0:00:00
        541 100%    0.00kB/s   0:00:00 (xfr#1, to-chk=2/4)
config/database.php

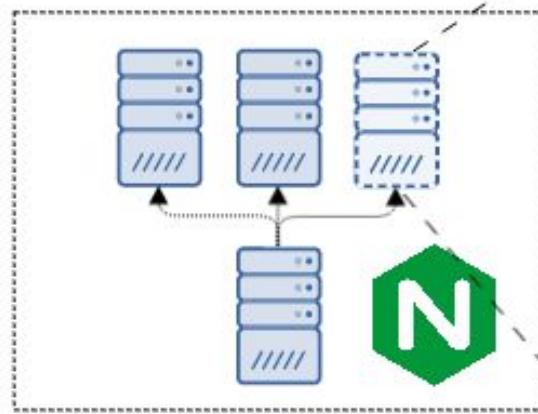
        1,440 100%  1.37MB/s   0:00:00
        1,440 100%  1.37MB/s   0:00:00 (xfr#2, to-chk=0/4)
/bitnami/jenkins/jobs/laravel-quickstart-internal/workspace
```

Setup Jenkins Workshop

Where are we?



App Servers



Docker-Compose

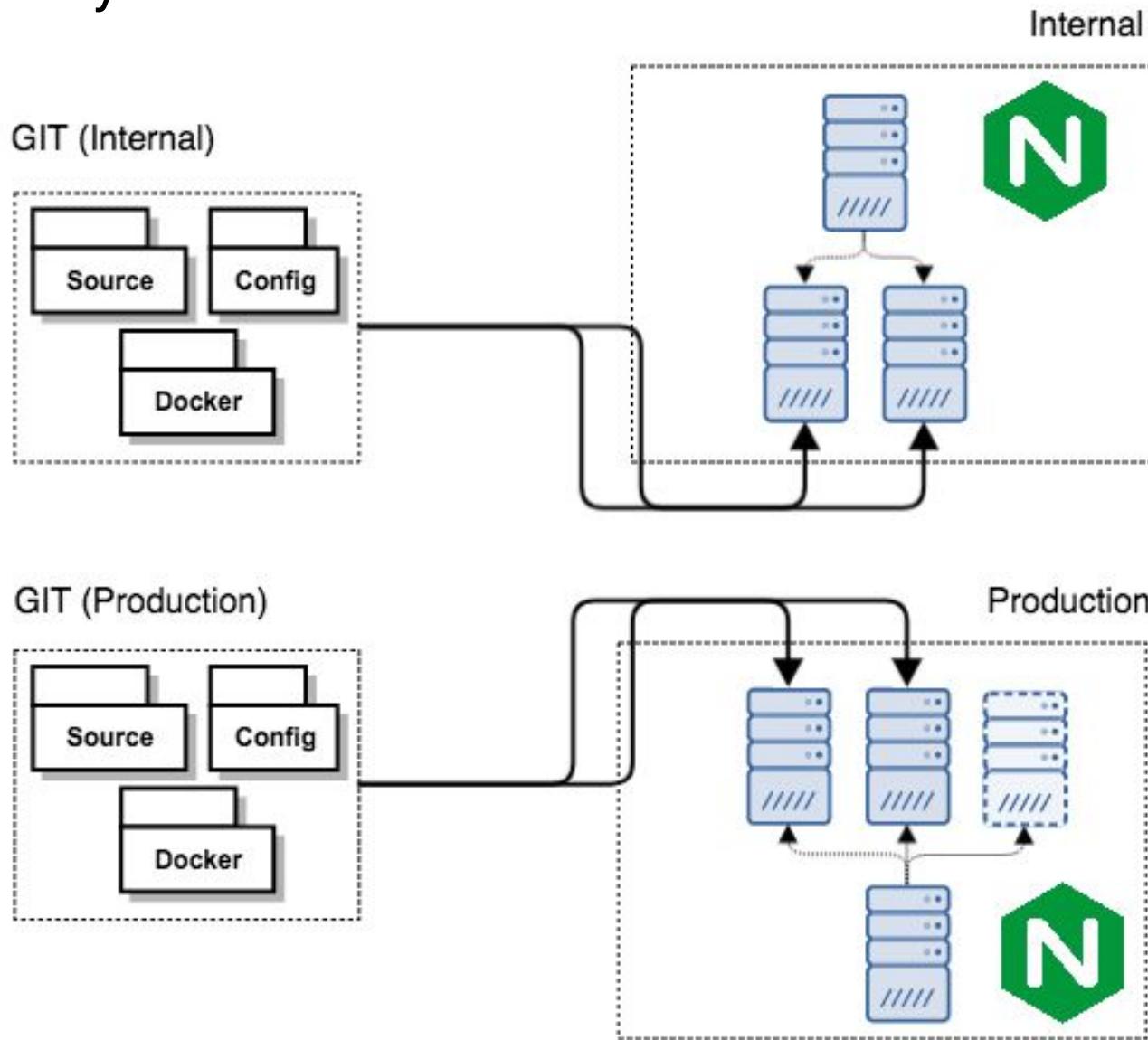


/var/www/
/var/www/laravel-quickstart.com
/var/www/laravel-quickstart-internal.com

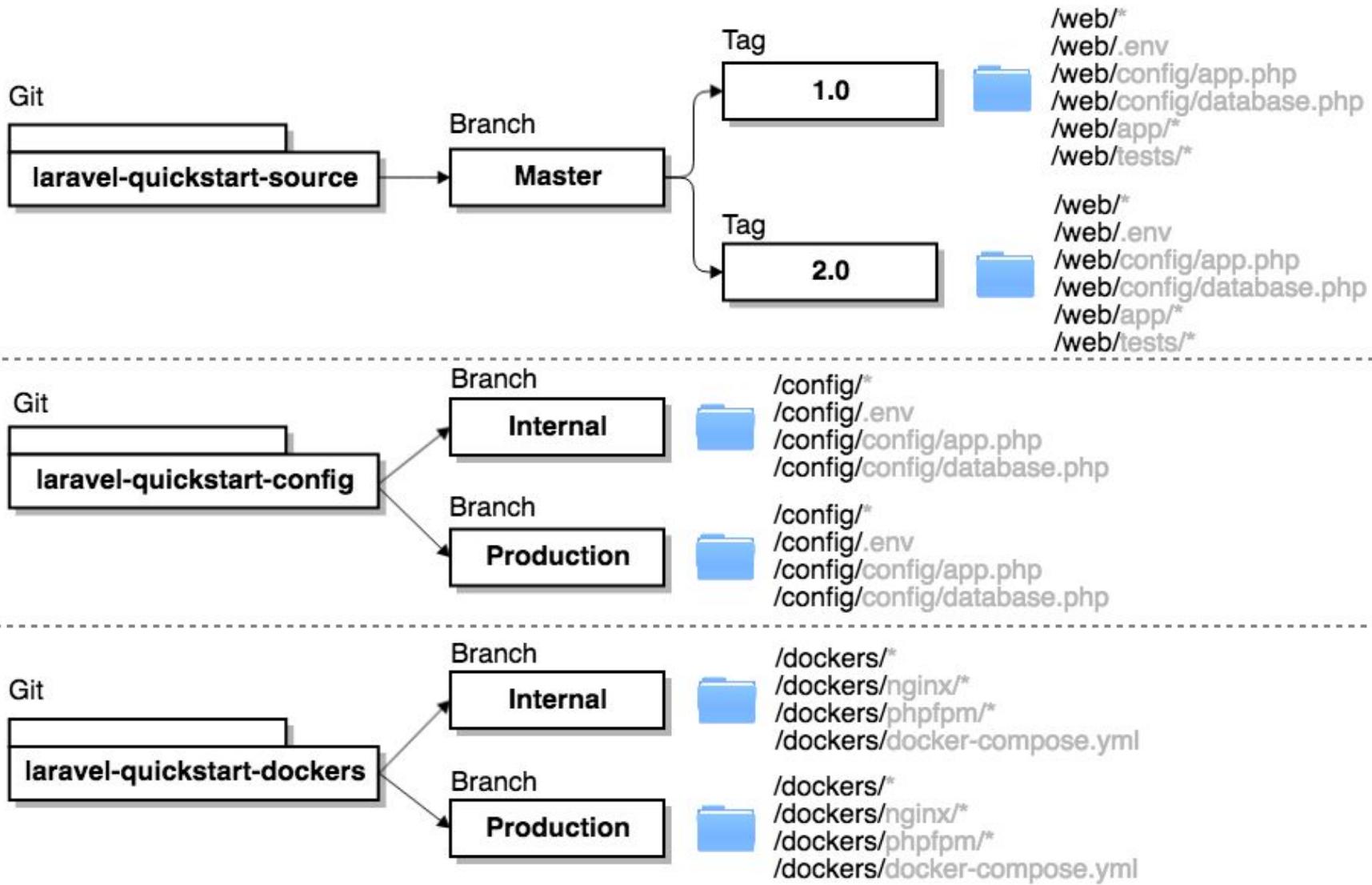
/var/dockers/
/var/dockers/laravel-quickstart.com
/var/dockers/laravel-quickstart-internal.com

Setup App Servers Workshop

Git Deploy Environment



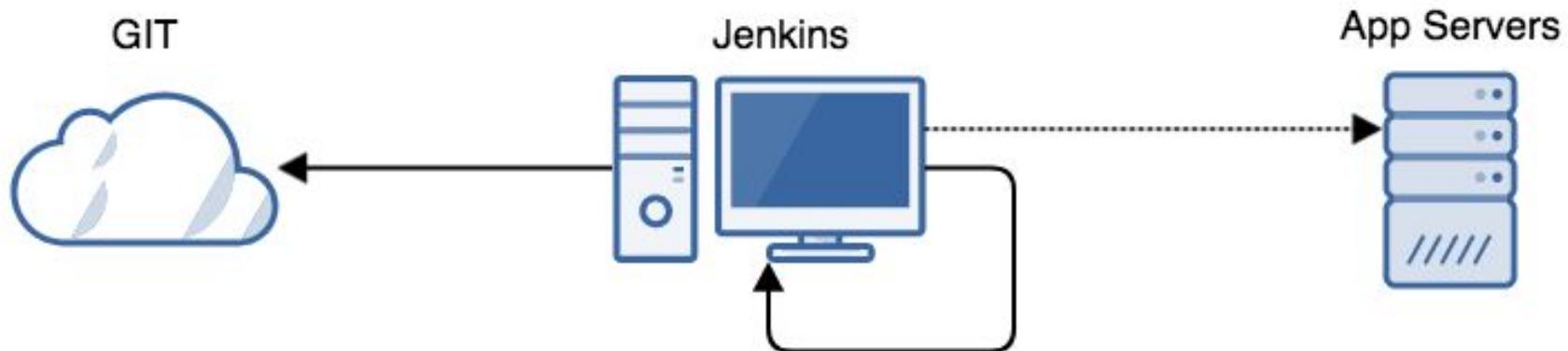
Inside Git Deploy Environment



Setup Git Deploy Config Setup Git Deploy Docker Workshop

Deployment Flow

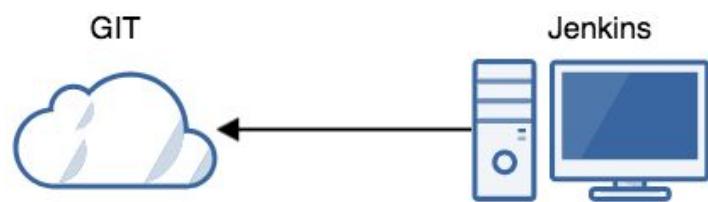
10. Download dockers to /var/dockers
11. Download source to /var/source
12. Run docker-compose pull
13. Run docker-compose down
14. Run docker-compose up -d



1. Git pull "source" to /source
2. Git pull "config" to /config
3. Git pull "docker" to /dockers

4. Sync file from /source to /build
5. Sync file from /config to /build
6. Zip /build and save in /finish
7. Zip /dockers and save in /finish
8. Copy /finish/* to "sourcemaster"
9. Ask App Servers to run re-deploy.sh

Step 0 (zero) Pull Git deploy-scripts

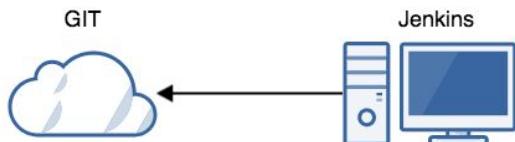


o. Pull Deploy Script to /deploy-scripts

▼	base
➤	deploy-base.sh
➤	re-deploy.sh
➤	runtest.sh
▼	laravel-quickstart
▼	internal
➤	deploy.sh
▼	production
➤	deploy.sh
▼	test
➤	deploy.sh

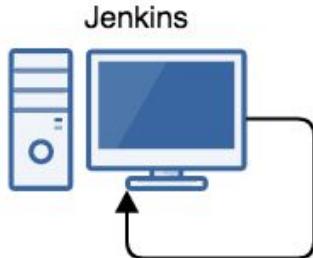
Deployment Flow deploy-base.sh

1. Git pull "source" to /source
2. Git pull "config" to /config
3. Git pull "docker" to /dockers



```
function run() {  
  
    local CMD=$1  
  
    checkEnv  
  
    gitUpdate $SOURCE_DIR $SOURCE_REPO $RELEASE_BRANCH $RELEASE_VERSION  
    gitUpdate $CONFIG_DIR $CONFIG_REPO $CONFIG_ENV  
    gitUpdate $DOCKER_DIR $DOCKER_REPO $SERVER_ENV  
  
    #sync file to build dir  
    rsyncHard $SOURCE_DIR/$SOURCE_SUBDIR $BUILD_DIR  
    #override config in build dir  
    rsyncSoft $CONFIG_DIR/config $BUILD_DIR  
  
    local SOURCE_ZIP="$FINISH_DIR/$PROJECT_NAME-code.tar.gz"  
    copyToMaster $BUILD_DIR $SOURCE_ZIP  
  
    local DOCKER_ZIP="$FINISH_DIR/$PROJECT_NAME-docker.tar.gz"  
    copyToMaster $DOCKER_DIR/dockers $DOCKER_ZIP  
  
    if [ $CMD = "test" ]  
    then  
        | runtest $(basename $SOURCE_ZIP) $(basename $DOCKER_ZIP)  
    else  
        | redeploy $(basename $SOURCE_ZIP) $(basename $DOCKER_ZIP)  
    fi  
}
```

Deployment Flow deploy-base.sh



4. Sync file from /source to /build
5. Sync file from /config to /build
6. Zip /build and save in /finish
7. Zip /dockers and save in /finish
8. Copy /finish/* to "sourcemaster"

9. Ask App servers to run re-deploy.sh

```
function run() {  
  
    local CMD=$1  
  
    checkEnv  
  
    gitUpdate $SOURCE_DIR $SOURCE_REPO $RELEASE_BRANCH $RELEASE_VERSION  
    gitUpdate $CONFIG_DIR $CONFIG_REPO $CONFIG_ENV  
    gitUpdate $DOCKER_DIR $DOCKER_REPO $SERVER_ENV  
  
    #sync file to build dir  
    rsyncHard $SOURCE_DIR/$SOURCE_SUBDIR $BUILD_DIR  
    #override config in build dir  
    rsyncSoft $CONFIG_DIR/config $BUILD_DIR  
  
    local SOURCE_ZIP="$FINISH_DIR/$PROJECT_NAME-code.tar.gz"  
    copyToMaster $BUILD_DIR $SOURCE_ZIP  
  
    local DOCKER_ZIP="$FINISH_DIR/$PROJECT_NAME-docker.tar.gz"  
    copyToMaster $DOCKER_DIR/dockers $DOCKER_ZIP  
  
    if [ $CMD = "test" ]  
    then  
        runtest $(basename $SOURCE_ZIP) $(basename $DOCKER_ZIP)  
    else  
        redeploy $(basename $SOURCE_ZIP) $(basename $DOCKER_ZIP)  
    fi  
}
```

Deployment Flow (re-deploy.sh)



10. Download dockers to /var/dockers

```
rm -f $DOCKER_FILE
wget $SOURCE_MASTER_URL/$DOCKER_FILE
sudo rm -Rf $DOCKER_PATH
sudo mkdir -p $DOCKER_PATH
sudo tar -xzf $DOCKER_FILE -C $DOCKER_PATH
sudo chown ubuntu:ubuntu $DOCKER_PATH
sudo mkdir -p $DOCKER_PATH/logs
sudo chown -R 1001:1001 $DOCKER_PATH/logs
sudo chown -R 1001:1001 $DOCKER_PATH/nginx
sudo chown -R 1001:1001 $DOCKER_PATH/php-fpm
```

11. Run docker-compose pull

```
sudo docker-compose -f $DOCKER_PATH/docker-compose.yml pull
```

12. Download source to /var/www

```
rm -f $SOURCE_FILE
wget $SOURCE_MASTER_URL/$SOURCE_FILE
sudo rm -Rf $SOURCE_PATH
sudo mkdir -p $SOURCE_PATH
sudo tar -xzf $SOURCE_FILE -C $SOURCE_PATH
sudo chown -R 1001 $SOURCE_PATH
```

```
#create file for elb to make health check
echo 'pong' | sudo tee $SOURCE_PATH/ping.html
```

13. Run docker-compose down

```
sudo docker-compose -f $DOCKER_PATH/docker-compose.yml down
sleep 5
sudo docker-compose -f $DOCKER_PATH/docker-compose.yml up -d
```

14. Run docker-compose up -d

/internal/deploy.sh (Deploy settings)

```
PROJECT_NAME="laravel-quickstart-internal.com"
SITE_NAME="laravel-quickstart-internal.com"

BASE="/bitnami/jenkins/1001/deploy-scripts/base"
DEPLOY_BASEDIR="/bitnami/jenkins/1001/deploys"

#SOURCE MASTER
SRC_MASTER_DIR="/var/www/html"
SRC_MASTER_URL="http://<your-jenkins-server-ip>:8080"

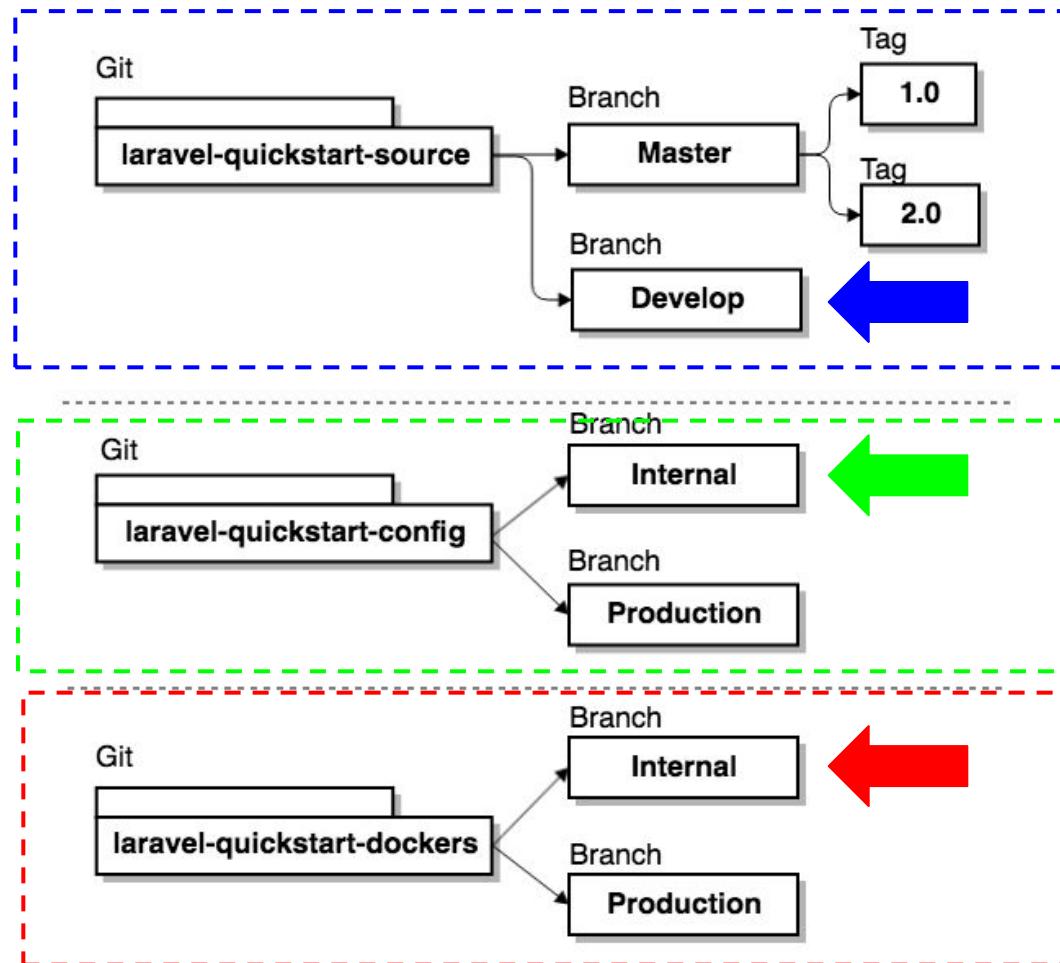
#SERVER
SERVERS="ubuntu@<your-app-server-ip>"
SERVER_DOCKER_PATH="/var/dockers/$SITE_NAME"
SERVER_SOURCE_PATH="/var/www/$SITE_NAME"

#GITS
GITREPO_URL="https://$GIT_USER:$GIT_PASS@github.com/<your-github-username>"
SOURCE_REPO="$GITREPO_URL/laravel-quickstart-source.git"
CONFIG_REPO="$GITREPO_URL/laravel-quickstart-config.git"
DOCKER_REPO="$GITREPO_URL/laravel-quickstart-dockers.git"
SOURCE_SUBDIR="web"
```

/production/deploy.sh (Deploy settings)

```
PROJECT_NAME="laravel-quickstart-production.com"  
SITE_NAME="laravel-quickstart-production.com"  
  
BASE="/bitnami/jenkins/1001/deploy-scripts/base"  
DEPLOY_BASEDIR="/bitnami/jenkins/1001/deploys"  
  
#SOURCE MASTER  
SRC_MASTER_DIR="/var/www/html"  
SRC_MASTER_URL="http://<your-jenkins-server-ip>:8080"  
  
#SERVER  
SERVERS="ubuntu@<your-app-server-ip>"  
SERVER_DOCKER_PATH="/var/dockers/$SITE_NAME"  
SERVER_SOURCE_PATH="/var/www/$SITE_NAME"  
  
#GITS  
GITREPO_URL="https://$GIT_USER:$GIT_PASS@gitHub.com/<your-github-username>"  
SOURCE_REPO="$GITREPO_URL/laravel-quickstart-source.git"  
CONFIG_REPO="$GITREPO_URL/laravel-quickstart-config.git"  
DOCKER_REPO="$GITREPO_URL/laravel-quickstart-dockers.git"  
SOURCE_SUBDIR="web"
```

Deploy Command (Internal)

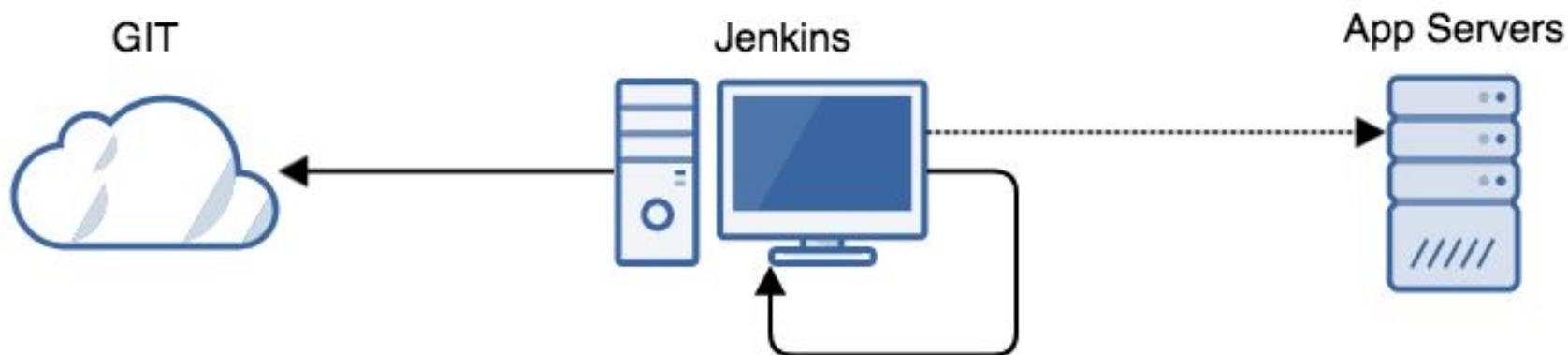


```
./.../internal/deploy.sh internal internal develop
```

Git Deploy Scripts Workshop & Deploy Internal Workshop

Automate Test Flow

10. Download dockers to /var/dockers
11. Download source to /var/source
12. Run Docker-compose up -d
13. Run docker exec migrate database
14. Run docker exec phpunit
15. Return exit code to Jenkins



1. Git pull "source" to /source
2. Git pull "config" to /config
3. Git pull "docker" to /dockers
4. Sync file from /source to /build
5. Sync file from /config to /build
6. Zip /build and save in /finish
7. Zip /dockers and save in /finish
8. Copy /finish/* to "sourcemaster"
9. Ask App Servers to run **runtest.sh**

Automate Test Flow (runtest.sh)



10. Download dockers to /var/dockers

```
rm -f $DOCKER_FILE
wget $SOURCE_MASTER_URL/$DOCKER_FILE
sudo rm -Rf $DOCKER_PATH
sudo mkdir -p $DOCKER_PATH
sudo tar -xzf $DOCKER_FILE -C $DOCKER_PATH
sudo chown ubuntu:ubuntu $DOCKER_PATH
sudo mkdir -p $DOCKER_PATH/logs
sudo chown -R 1001:1001 $DOCKER_PATH/logs
sudo chown -R 1001:1001 $DOCKER_PATH/nginx
sudo chown -R 1001:1001 $DOCKER_PATH/php-fpm
```

```
sudo docker-compose -f $DOCKER_PATH/docker-compose.yml pull
```

11. Download source to /var/source

```
rm -f $SOURCE_FILE
wget $SOURCE_MASTER_URL/$SOURCE_FILE
sudo rm -Rf $SOURCE_PATH
sudo mkdir -p $SOURCE_PATH
sudo tar -xzf $SOURCE_FILE -C $SOURCE_PATH
sudo chown -R 1001 $SOURCE_PATH
```

12. Run Docker-compose up -d

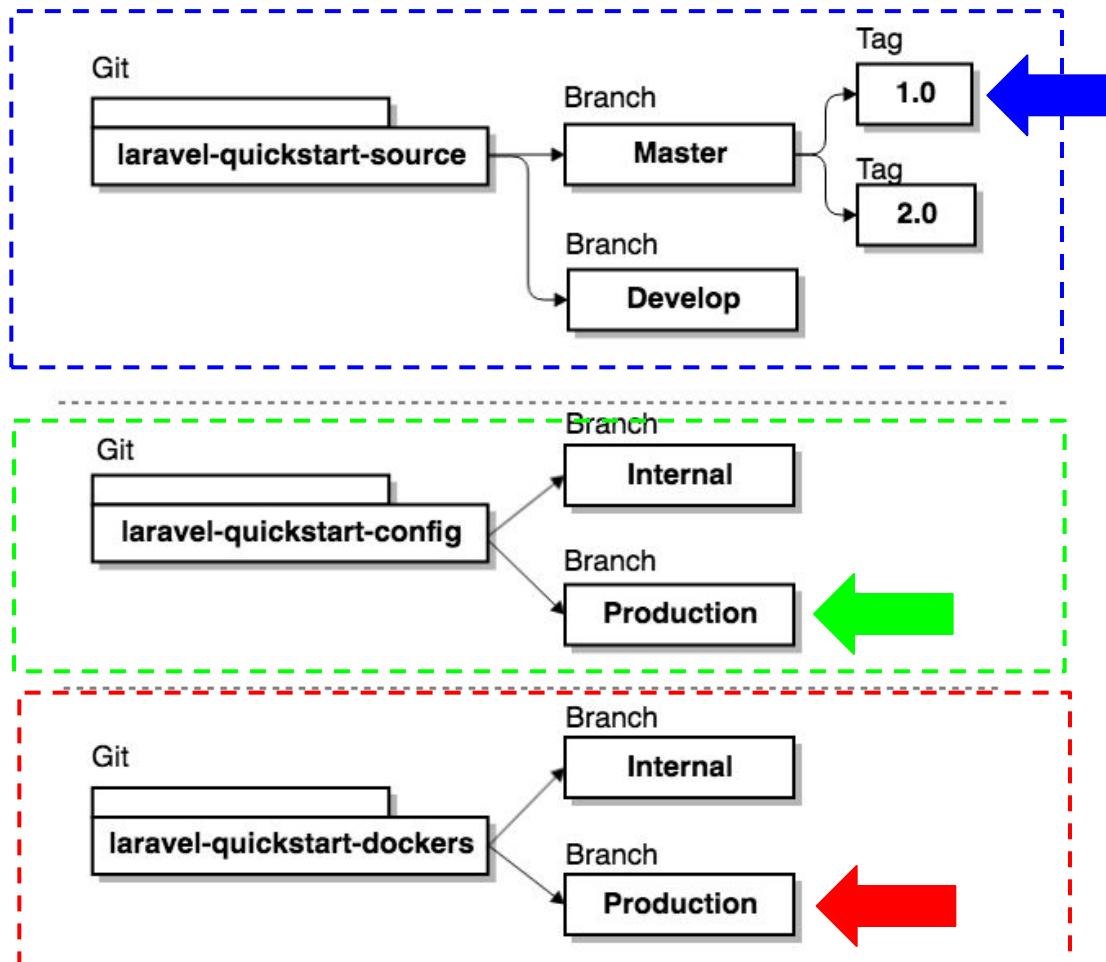
```
sudo docker-compose -f $DOCKER_PATH/docker-compose.yml down
sleep 5
sudo docker-compose -f $DOCKER_PATH/docker-compose.yml up -d
sleep 20
```

13. Run docker exec migrate database
14. Run docker exec phpunit
15. Return exit code to Jenkins

```
PHPFPM_NAME=$(docker-compose -f $DOCKER_PATH/docker-compose.yml ps -q phpfpm)
sudo docker exec -i $PHPFPM_NAME php artisan migrate:refresh --database=mysqltest
sudo docker exec -i $PHPFPM_NAME vendor/bin/phpunit
exit $? 
```

Deploy Automate Test Workshop

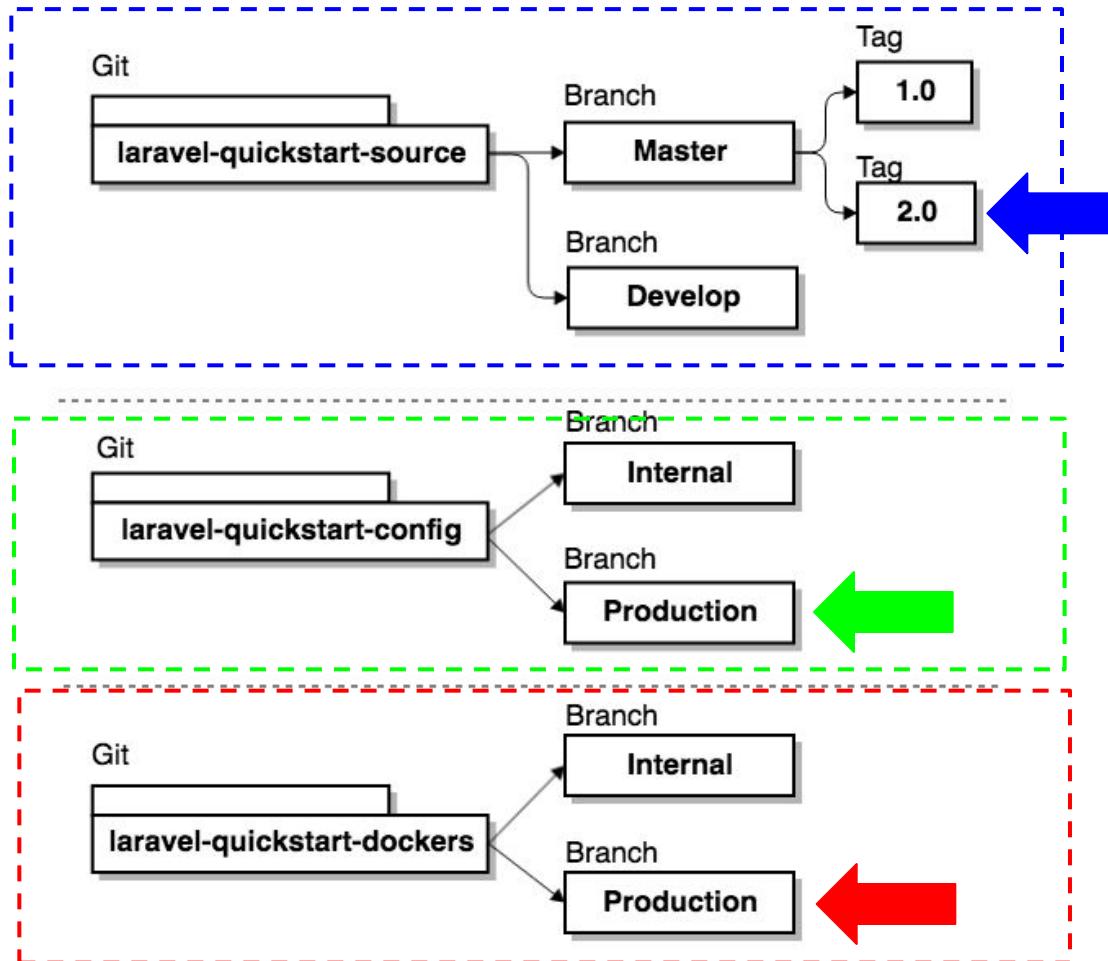
Deploy Command (Production)



```
./.../production/deploy.sh production production master 1.0
```

Deploy Production Workshop

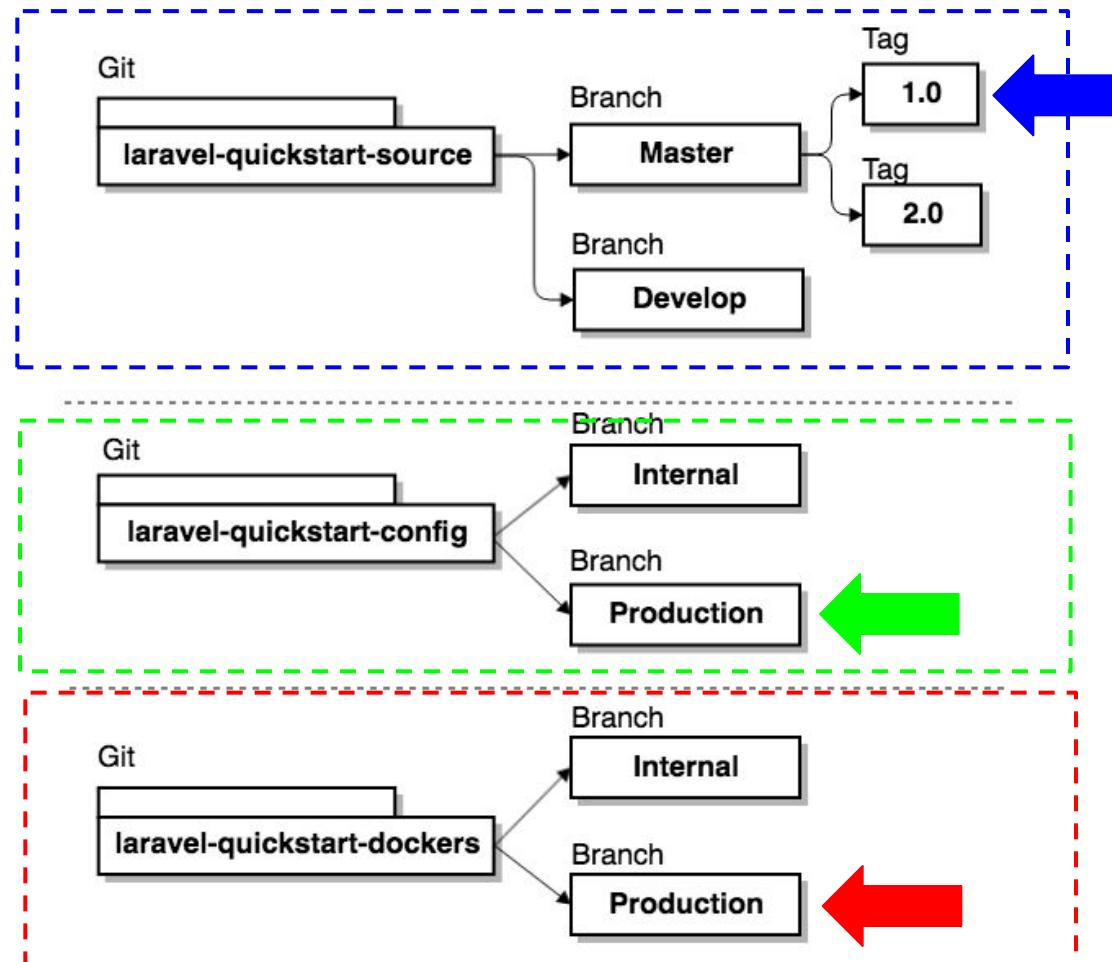
Deploy Command (Version 2.0)



```
./.../production/deploy.sh production production master 2.0
```

Deploy New Feature Workshop

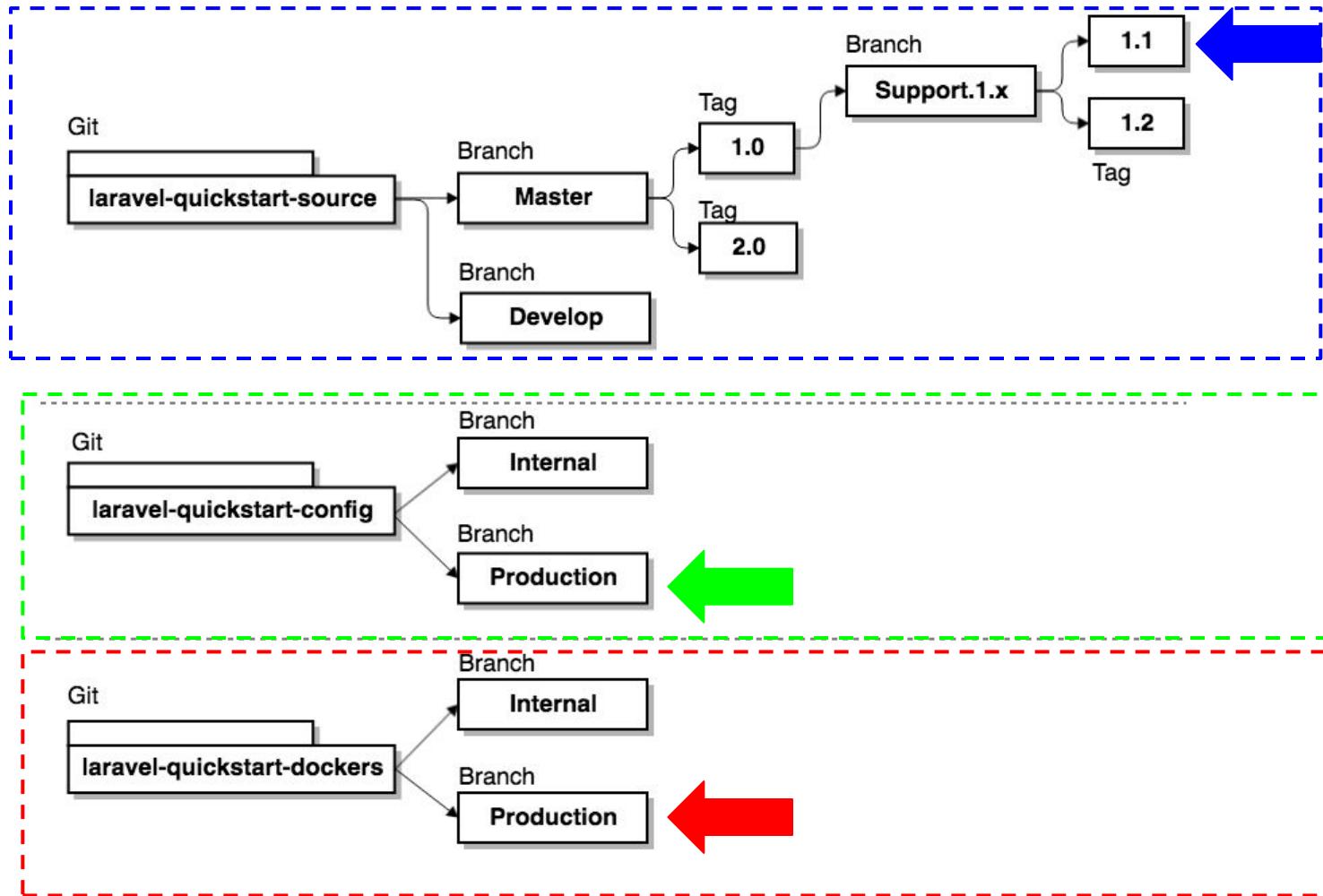
Deploy Command (Rollback to 1.0)



```
./.../production/deploy.sh production production master 1.0
```

Rollback Workshop

Deploy Hotfixes (Support Branch)



`./.../production/deploy.sh production production support.1.x
1.1`

Hotfixes Workshop

Tuning

Tuning

- Linux
- Nginx
- PHP-FPM

Tuning Linux

Linux ที่ติดตั้งมา แบบ Out of the box
ค่าคอนฟิกถูกตั้งให้ใช้ได้กับเครื่องทั่วๆไป

Tuning sysctl.conf

sysctl หรือ System Config

คือ การปรับตั้งค่า Kernel Parameter

Tuning sysctl.conf

Config file

ອយ່ທີ່

/etc/sysctl.conf

Tuning sysctl.conf

แสดงค่า config ทั้งหมด

```
# sysctl -a
```

```
vm.overcommit_memory = 0
vm.overcommit_ratio = 50
vm.page-cluster = 3
vm.panic_on_oom = 0
vm.percpu_pagelist_fraction = 0
vm.stat_interval = 1
vm.swappiness = 0
vm.user_reserve_kbytes = 63225
vm.vfs_cache_pressure = 50
vm.zone_reclaim_mode = 0
```

Tuning sysctl.conf

แสดงค่า config ที่สนใจ

```
# sysctl vm.swappiness
```

or

```
# sysctl -a | grep vm.swappiness
```

or

```
# cat /proc/sys/vm/swappiness
```

```
vm.swappiness = 60
```

Tuning sysctl.conf

ปรับค่า config

```
# sysctl -w vm.swappiness=0  
# echo 0 > /proc/sys/vm/swappiness  
# sysctl vm.swappiness
```

vm.swappiness = 0

Tuning sysctl.conf

ปรับค่า config โดยการแก้ /etc/sysctl.conf

Example.

```
# Increase size of file handles and inode cache
fs.file-max = 2000000

# Do less swapping
vm.swappiness = 10
vm.dirty_ratio = 60
vm.dirty_background_ratio = 2
...
```

Tuning sysctl.conf

เรียกใช้ค่า config ในมด้วย sysctl -p

**** NOTE ****

```
# sysctl -p # <= จะโหลดค่า config จาก  
/etc/sysctl.conf ค่าจาก sysctl -w จะถูกเรียกแทนที่  
จากไฟล์ /etc/sysctl.conf
```

Tuning Nginx

Nginx ที่ถูกติดตั้งมาจาก Package Manager หรือ
Compile จาก Source Code ไม่ได้ถูกคอนฟิกมาเพื่อรับ
โหลดมากๆ

Tuning Nginx

Default config file location

Package Manager or compile source [w/o --prefix]

> **/etc/nginx/nginx.conf**

Docker Images : **3dsinteractive/nginx:1.12**

> **/opt/bitnami/nginx/conf/nginx.conf**

Tuning Nginx

Basic Command Line

- Start
- Stop
- Reload

Tuning Nginx

Basic Command Line.

Start Nginx.

```
nginx
```

Tuning Nginx

Basic Command Line.

Stop Nginx.

```
nginx -s stop
```

Tuning Nginx

Basic Command Line.

Reload Nginx or reload new config.

```
nginx -s reload
```

Tuning Nginx

Nginx Configuration Structure

```
main
  └── events
      └── ...
  └── http
      └── server
          └── location ***
  └── mail
      └── ...
  └── stream
      └── ...
  ...
```

Tuning Nginx

Nginx Configuration Structure

```
events { ... }

http {
    upstream { ... }

    server {
        listen 80; # <-- directive
        location / {
            # location context, block
            if( ... ) {
                # if block, context
            }
        }
    }
}
```

Tuning Nginx | main

- **worker_processes**: Defines the number of worker processes., **auto** = autodetected
- **worker_rlimit_nofile** : The limit on the maximum number of open files

```
worker_processes 4;  
worker_rlimit_nofile 500000;
```

Tuning Nginx | events

- **worker_connections**: Maximum number of simultaneous connections that can be opened by a worker process
- **use** : Specifies the connection processing method
- **worker_rlimit_nofile** : The limit on the maximum number of open files

Tuning Nginx | events

- **multi_accept** : A worker process will accept all new connections at a time.

```
events {  
    worker_connections 200240;  
    use epoll;  
    multi_accept on;  
}
```

FYI : ulimit

Ulimit | User limit

- Show all limit
- Set limit

```
ulimit -a  
ulimit -n  
ulimit -n 200240
```

Tuning Nginx | http

```
# Most client buffers are coming from POST method form submissions
client_body_buffer_size      128k;

# Sets the max body buffer size, Error 413
client_max_body_size        10m;

# Sets buffer size for reading client request header, default 1k
client_header_buffer_size   1k;

# The maximum number and size of buffers used for reading large client request header.
large_client_header_buffers 4 4k;
```

Tuning Nginx | http

```
# cont.

sendfile    on;
tcp_nopush  on;
tcp_nodelay on;

keepalive_timeout  65;
keepalive_requests 100000;

gzip on;
gzip_http_version 1.0;
gzip_comp_level 2;
gzip_proxied any;
gzip_types text/plain text/css application/x-javascript
text/xml application/xml application/xml+rss text/javascript;
```

Tuning Nginx | http

```
# cont., extras.

proxy_cache_path /data/nginx/cache levels=1:2
keys_zone=web:10m;

upstream web {
    ...
    server web1:8080;
    server web2:8080;
}
...
```

Tuning Nginx | server

```
server {  
    listen 80;  
    server_name example.com;  
    resolver 127.0.0.11; # docker embedded DNS  
  
    # CORS = Cross Origin Resource Sharing, AJAX API Call  
    add_header Access-Control-Allow-Origin $http_origin  
    always;  
    add_header Access-Control-Allow-Credentials true always;  
    add_header Access-Control-Allow-Methods 'GET, POST, PUT,  
    DELETE, OPTIONS, HEAD' always;  
    add_header Access-Control-Allow-Headers 'Content-Type,  
    Cache-Control, X-Requested-With, X-Prototype-Version' always;  
    ...  
}
```

Tuning Nginx | location

```
# Reverse Proxy
location / {
    proxy_pass http://web;
    proxy_redirect off;
    proxy_set_header Host $host;
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-For
    $proxy_add_x_forwarded_for;           proxy_connect_timeout
90;
    proxy_send_timeout      90;
    proxy_read_timeout       90;
    proxy_buffer_size        4k;
    proxy_buffers            4 32k;
    proxy_busy_buffers_size   64k;
    proxy_temp_file_write_size 64k;
}
```

Tuning Nginx | location

```
# Caching
proxy_cache web;
proxy_cache_valid 200 302 10m;
proxy_cache_valid 404 1m;
add_header X-Cache-Status $upstream_cache_status;
expires      10m;

proxy_ignore_headers Set-Cookie;
proxy_ignore_headers expires;
proxy_hide_header pragma;
proxy_ignore_headers Cache-Control;
proxy_hide_header Cache-Control;
```

Tuning PHP-FPM

Configuration file

>

/opt/bitnami/php/etc/php-fpm.d/www.conf

```
pm = dynamic
pm.max_children = 200
pm.start_servers = 100
pm.min_spare_servers = 50
pm.max_spare_servers = 150
```

Tuning PHP-FPM

pm.max_children

The number of child processes to be created when pm is set to static and the maximum number of child processes to be created when pm is set to dynamic. This option is mandatory.

Calculate:

```
pm.max_children = RAM / Memory per child process size
```

Tuning PHP-FPM

pm.max_children (cont.)

Show php-fpm Memory use.

```
ps -ylC php-fpm --sort/rss
```

Tuning PHP-FPM

pm.max_children (cont.)

Show average memory use per process

```
ps ax --no-headers -o "rss,cmd" -C php-fpm | awk '{ sum+=$1 }  
END { printf ("%d%s\n", sum/NR/1024,"M") }'
```

Tuning PHP-FPM

`pm.max_children` (cont.)

Example:

- RAM 16GB
- Memory per process = 64MB

$$\begin{aligned}\text{MAX_CHILDREN} &= (16 * 1024) / 64\text{MB} \\ &= 256\end{aligned}$$

Tuning PHP-FPM

`pm.max_children (cont.)`

Example:

Or

$$\begin{aligned}\text{MAX_CHILDREN} &= ((16 - 2) * 1024) / 64\text{MB} \\ &= 224\end{aligned}$$

p.s 2GB for other process

Tuning PHP-FPM

`pm.start_servers`

The number of child processes created on startup.
Used only when pm is set to dynamic.

Default Value: **`min_spare_servers +
(max_spare_servers - min_spare_servers) / 2`**

Tuning PHP-FPM

`pm.start_servers` (cont.)

Example:

- MIN_SPARE = 100
- MAX_SPARE = 150

$$\begin{aligned}\text{START_SERVER} &= 100 + (150 - 100) / 2 \\ &= \mathbf{125}\end{aligned}$$

Tuning PHP-FPM

pm.min_spare_servers

The desired minimum number of idle server processes. Used only when pm is set to dynamic. Also mandatory in this case.

```
pm.min_spare_servers = 50
```

Tuning PHP-FPM

`pm.max_spare_servers`

The desired maximum number of idle server processes. Used only when pm is set to dynamic. Also mandatory in this case.

```
pm.max_spare_servers = 150
```

Load Testing

wrk : <https://github.com/wg/wrk>

Or

Docker image : williamyeh/wrk

Load Testing

Basic Usage

```
wrk -t 4 -c 400 -d 30s http://127.0.0.1:8080/index.html
```

```
Running 30s test @ http://127.0.0.1:8080/index.html
  12 threads and 400 connections
```

Thread Stats	Avg	Stdev	Max	+/- Stdev
Latency	635.91us	0.89ms	12.92ms	93.69%
Req/Sec	56.20k	8.07k	62.00k	86.54%

```
22464657 requests in 30.00s, 17.76GB read
```

```
Requests/sec: 748868.53
```

```
Transfer/sec: 606.33MB
```

Load Testing

Or via Docker

```
docker run --rm --net=host williamyeh/wrk -t 4 -c 400 -d 30s
http://127.0.0.1:8080/index.html
```

```
Running 30s test @ http://127.0.0.1:8080/index.html
 12 threads and 400 connections
 Thread Stats      Avg      Stdev      Max      +/- Stdev
   Latency    635.91us    0.89ms    12.92ms    93.69%
   Req/Sec    56.20k     8.07k    62.00k    86.54%
 22464657 requests in 30.00s, 17.76GB read
 Requests/sec: 748868.53
 Transfer/sec: 606.33MB
```

Load Testing

wrk

```
Usage: wrk <options> <url>
```

Options:

- c, --connections <N> Connections to keep open
- d, --duration <T> Duration of test
- t, --threads <N> Number of threads to use

- s, --script <S> Load Lua script file
- H, --header <H> Add header to request
- latency Print latency statistics
- timeout <T> Socket/request timeout
- v, --version Print version details

Numeric arguments may include a SI unit (1k, 1M, 1G)
Time arguments may include a time unit (2s, 2m, 2h)