



Research & Development Team

DevOps CI/CD with Jenkins & Docker

www.pnpsw.com

sommai.k@gmail.com

081-754-4663

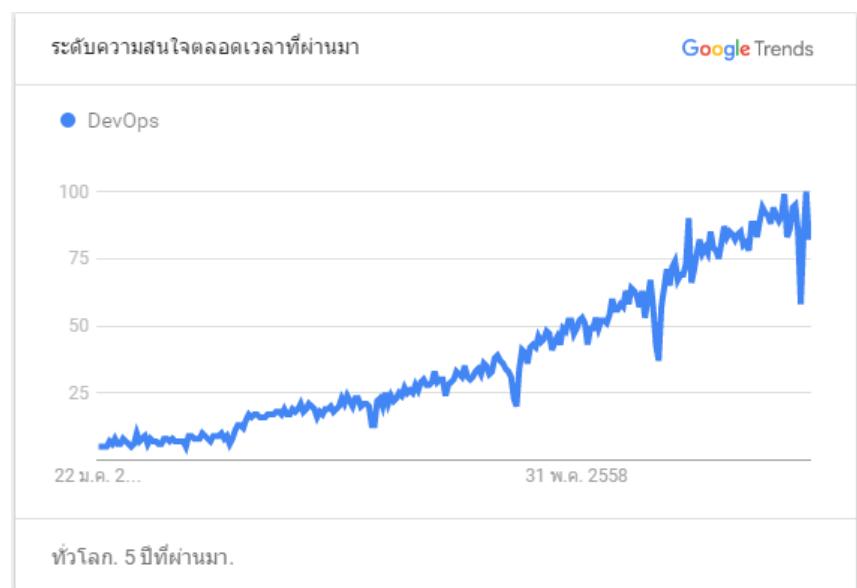
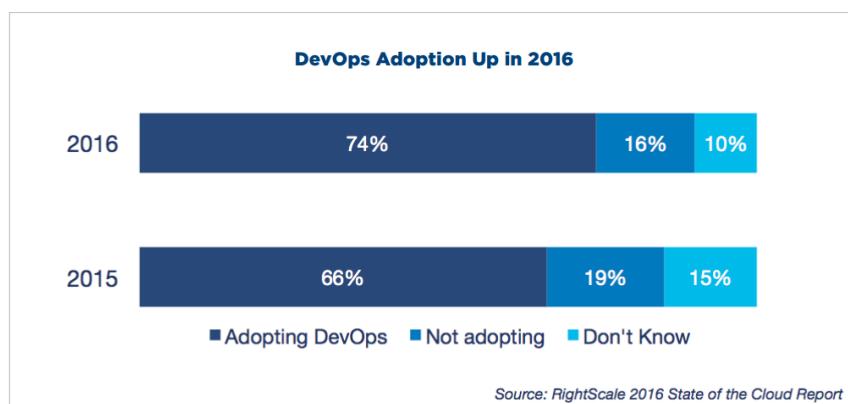
Lineid : sommaik

Medium: sommaikrangpanich

Definitions and History

- ในปี ค.ศ. 2009 ที่ Velocity Conference John Allspaw และ Paul Hammond ได้นำเสนอเรื่อง 10 Deploys per Day: Dev and Ops Cooperation at Flickr ซึ่งกล่าวถึงวิธีการสร้างเป้าหมายร่วมกันระหว่างแผนก Development และแผนก Operation และวิธีการที่ทำให้การ Deployment เป็นเรื่องทั่วไปที่กำกันในเซิร์ฟตประจวบัน การนำเสนอเรื่องนี้เป็นแรงบันดาลใจให้ Patrick Debois จัดงาน DevOpsDay ขึ้นมาในปีเดียวกัน คำว่า DevOps ที่ย่อมาจาก Development และ Operations จึงถูกสร้างขึ้นตั้งแต่นั้นเป็นต้นมา
- ปัจจุบัน DevOps เป็นแนวคิดที่มีประสิทธิภาพและแพร่หลายออกไปทั่วโลก จากผลสำรวจองค์กรกว่า 1,000 แห่งจากรายงาน RightScale 2016 State of the Cloud Report: DevOps Trends พบว่าในปี 2016 มีองค์กรนำ DevOps ไปปรับใช้แล้วถึง 74% ซึ่งเพิ่มขึ้นจากปีที่แล้วถึง 8% และจำนวนการ Search คำว่า DevOps ใน google ก็ยังเพิ่มขึ้นเรื่อยๆ ด้วย

ระดับความสนใจ

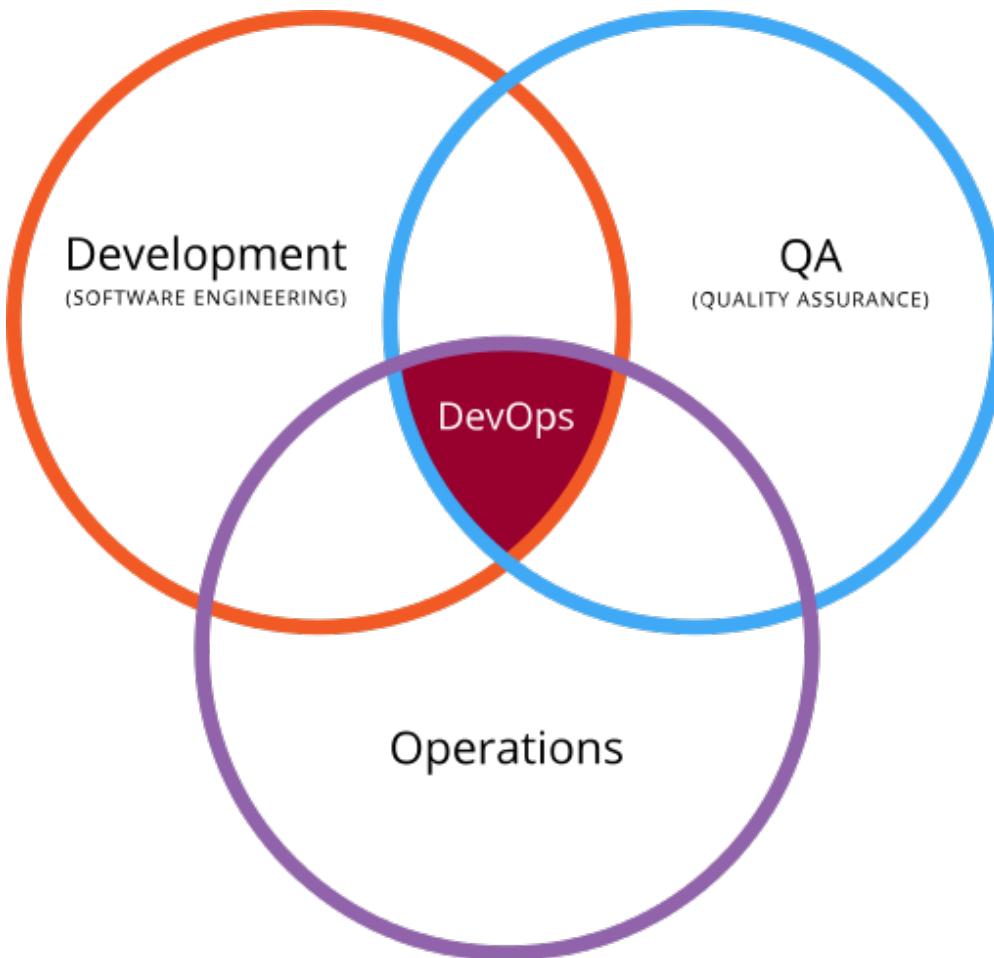


Cultural change

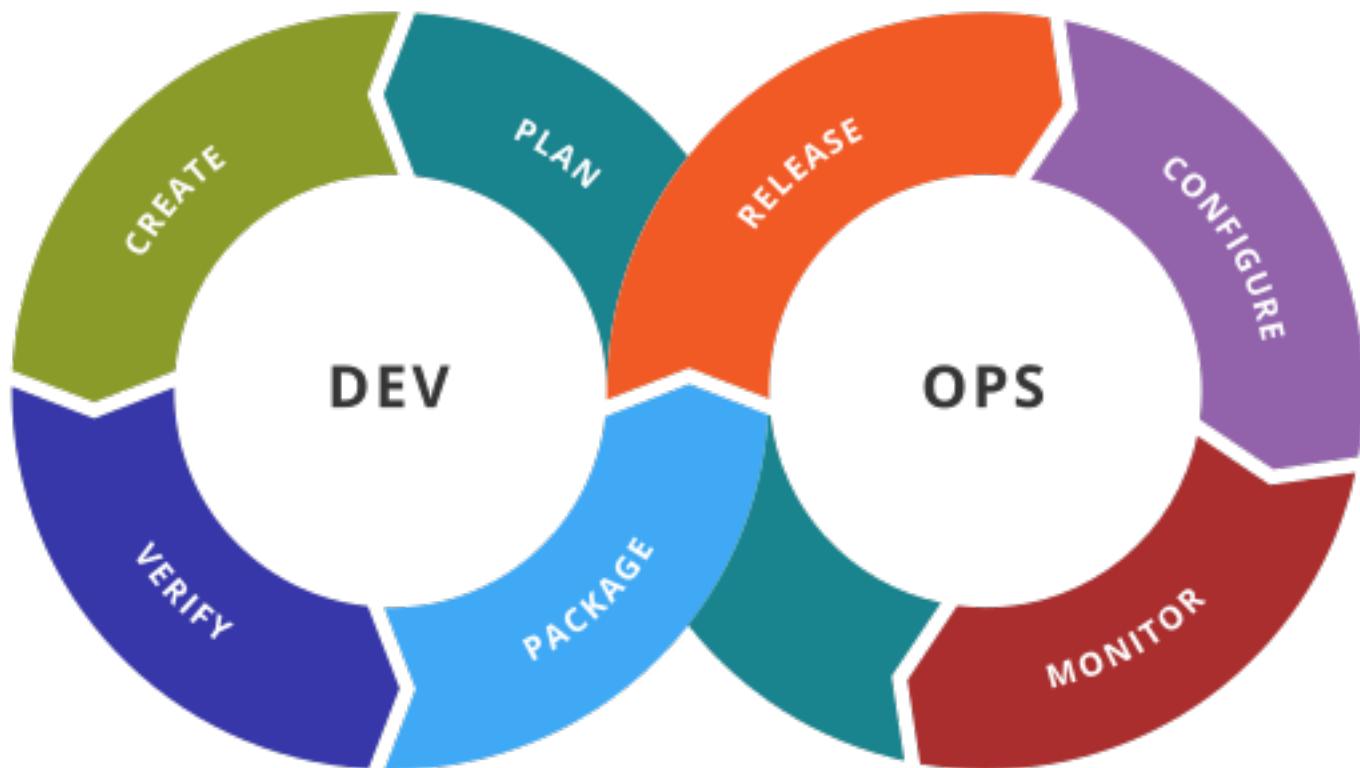
DevOps is more than just a tool or a process change.

- Operations— seeks organisational stability
- Developers— seek change
- Testers— seek risk reduction

Overview



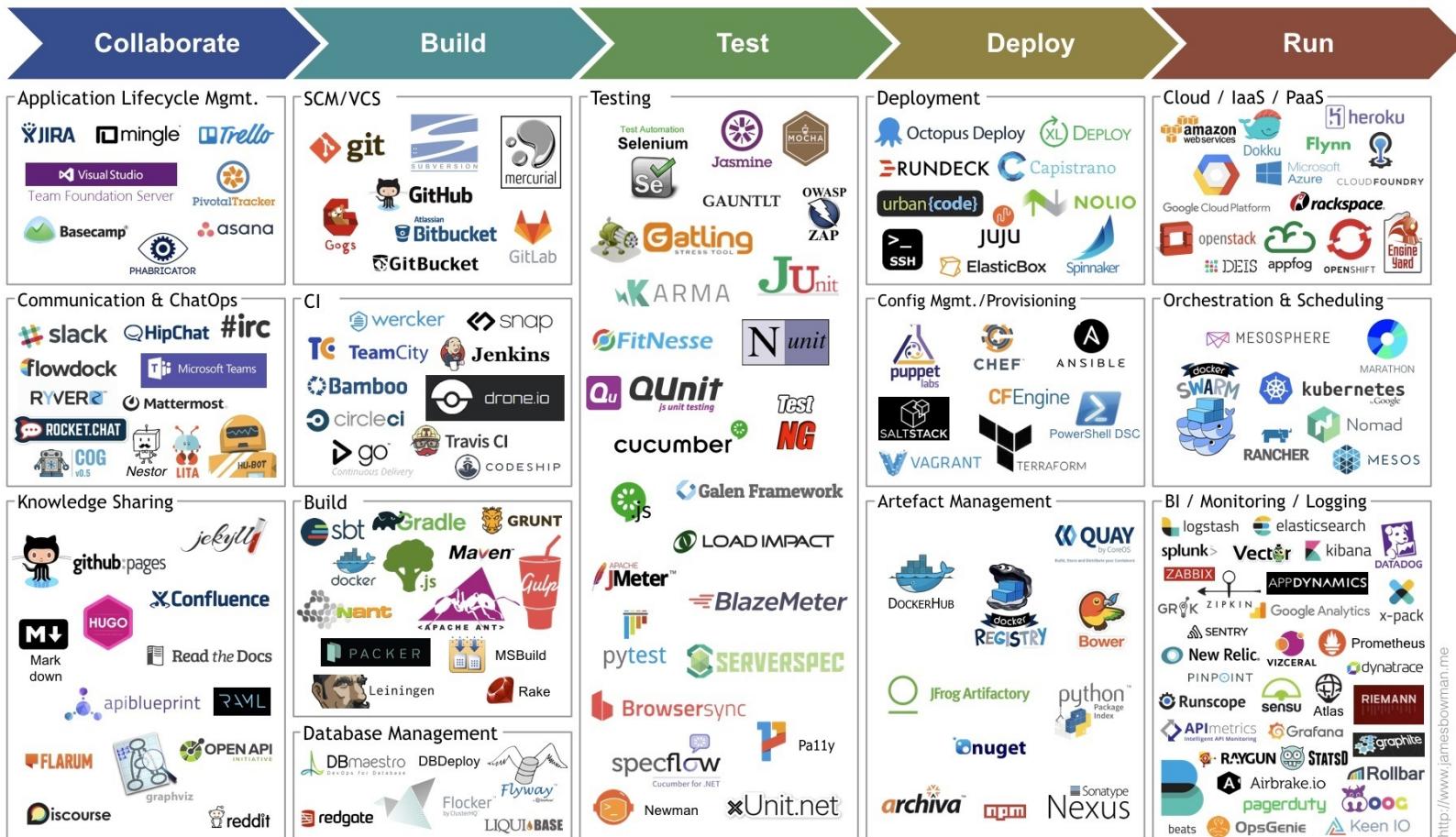
Workflow



7 Ways to Get Started with DevOps

- Invite Your Operations Team Into Your Development Process
- Visualise the Work Together
- Automate Your Test/Build Process
- Create a Deployment Plan
- Identify Fragile Systems
- Smooth Out Wait States
- Link Your Work to Your Value

DevOps Ecosystem



<http://www.jamesbowman.me>

Collaborate

Application Lifecycle Mgmt.

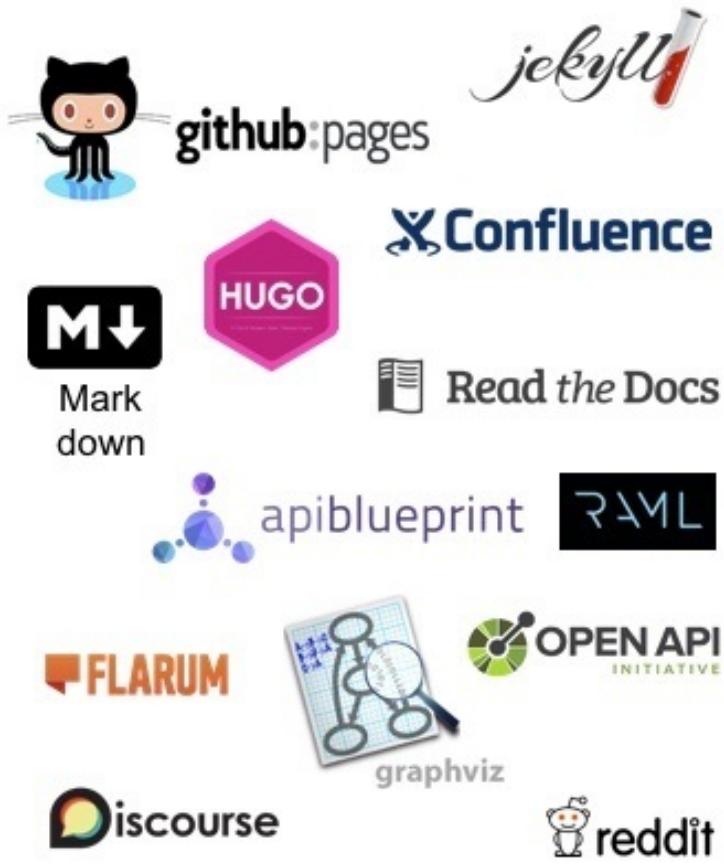


Communication & ChatOps



Collaborate

Knowledge Sharing



Build

SCM/VCS



CI



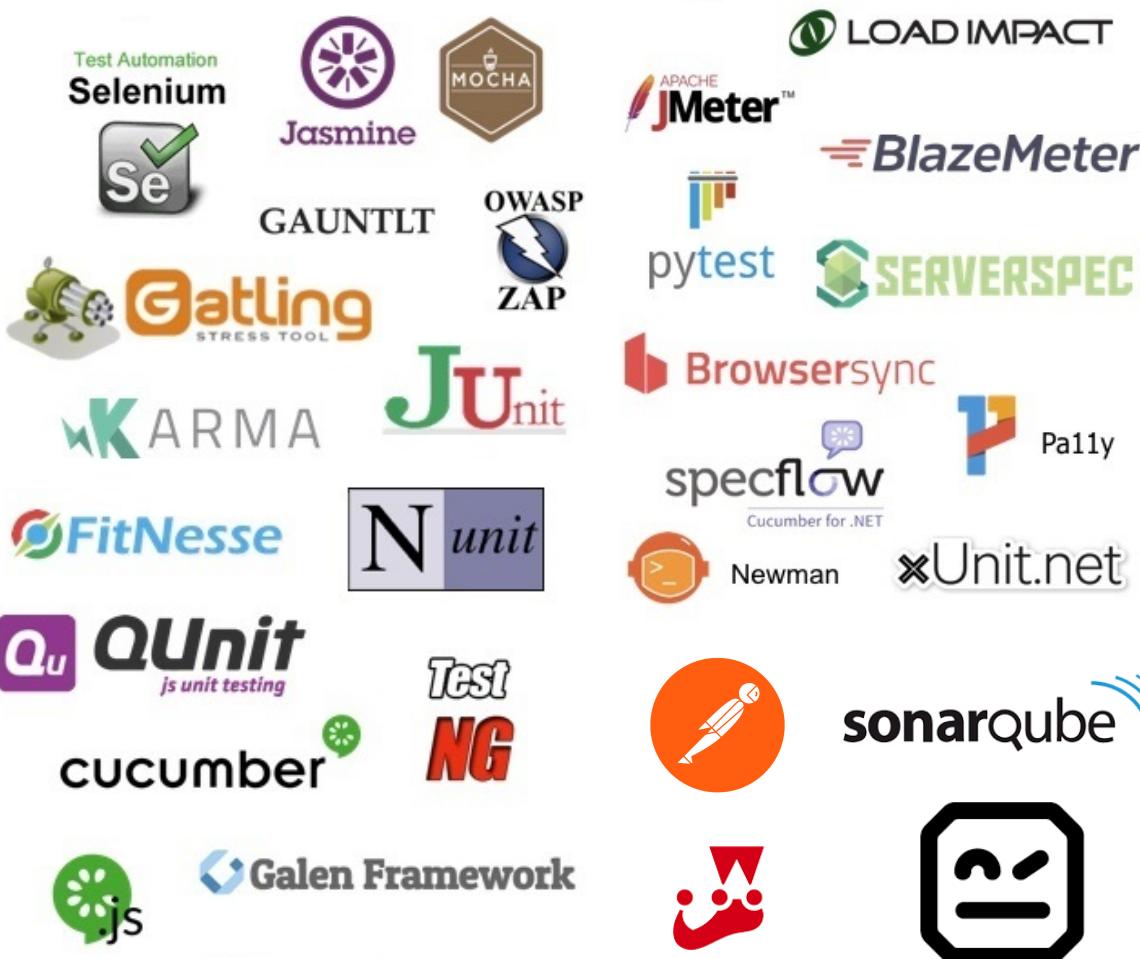
Build



Database Management



Test



Deploy

Deployment



Artifact Management



Config Mgmt./Provisioning



Run

Cloud / IaaS / PaaS



Orchestration & Scheduling



BI / Monitoring / Logging



CI/CD

- Continuous Integration (CI)
- Continuous Deployment (CD)
- Continuous Delivery (CD)

Continuous Integration

- Continuous Integration uses automation tools that empower development teams to build and test code after each merge as seamlessly as possible

Continuous Delivery

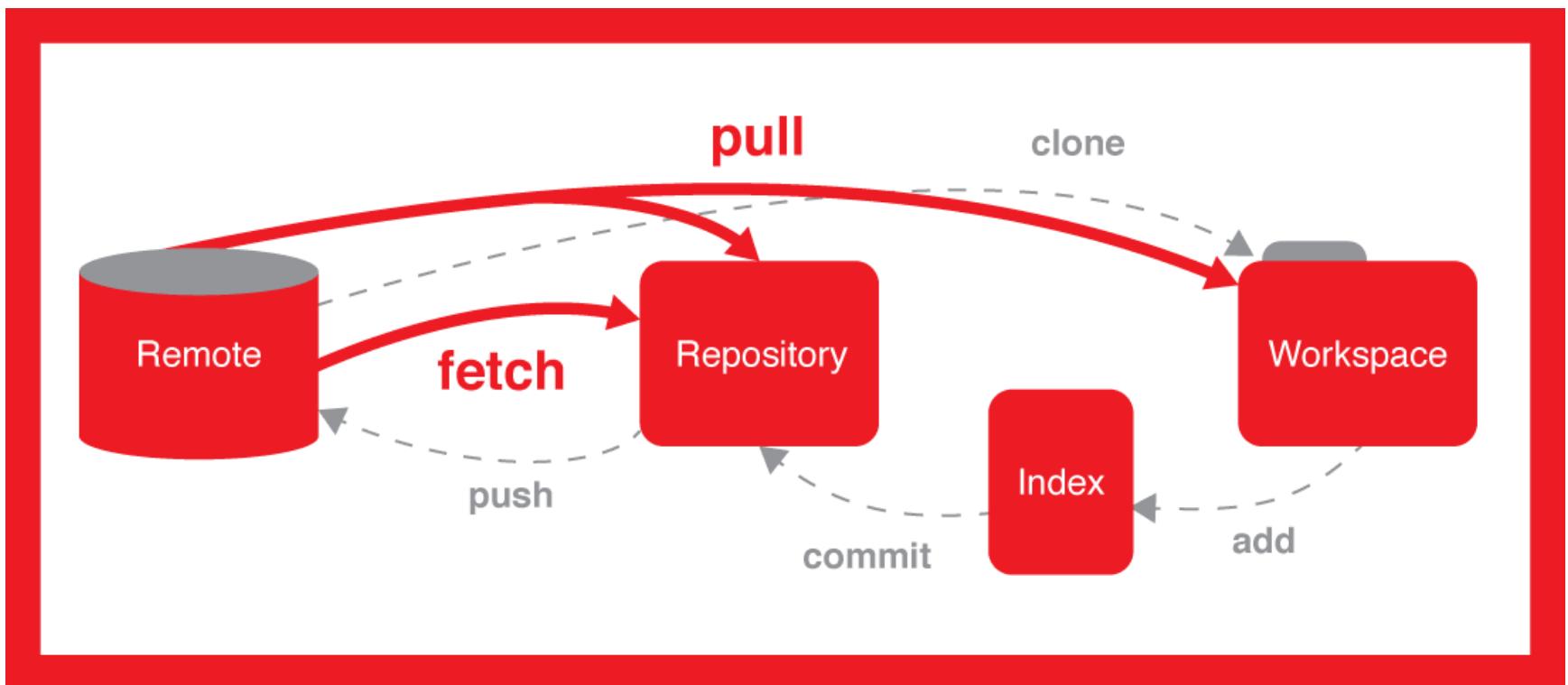
- Focused on keeping the code ready to deploy at any given time.
- It's not about making bugged-code available for the production environment.
- Rather, all the sets of features are ready to go, and the latest build is ready to be delivered at any time given.

Continuous Deployment

- Production deployment of every change done to the code.
- The changes are ideally automated, without human intervention.
- The approach works well in a corporate environment, where the user and the tester are ultimately one person.

Software version control with
GIT

Overview



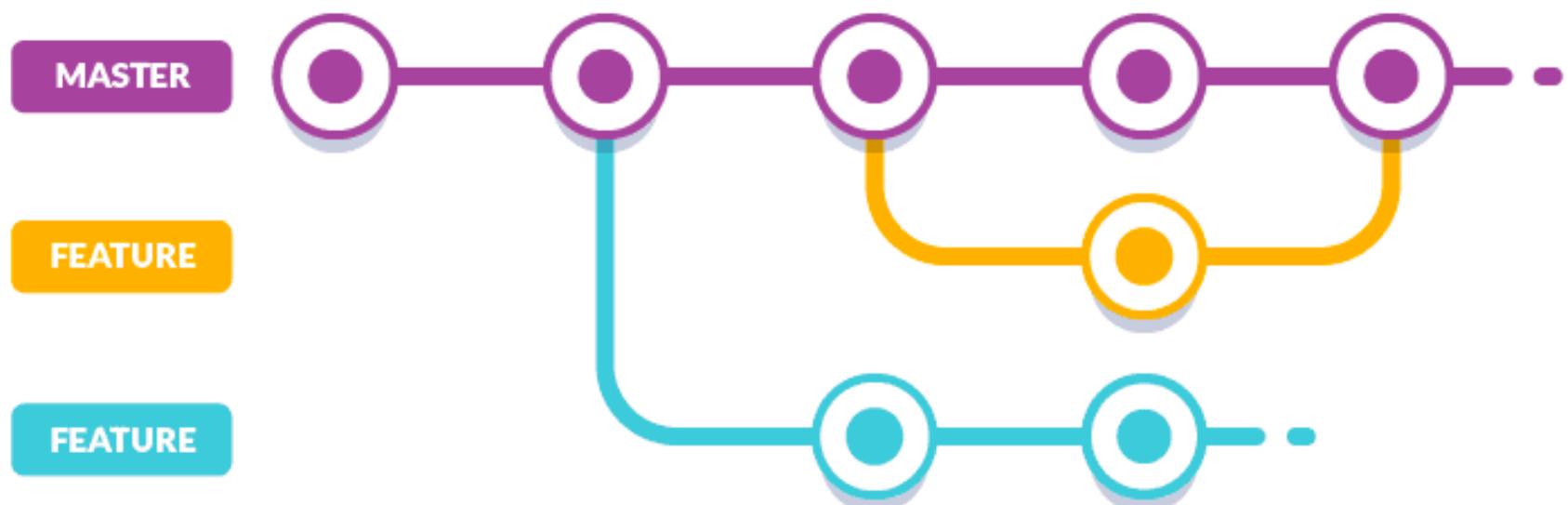
Git workflow type

- Basic ໃໝ່ກັບ Project ຂາດເລືກ
- Feature Branch ໃໝ່ກັບ Project ຂາດກລາງ
- Gitflow ໃໝ່ກັບ Project ຂາດໃຫຍ່

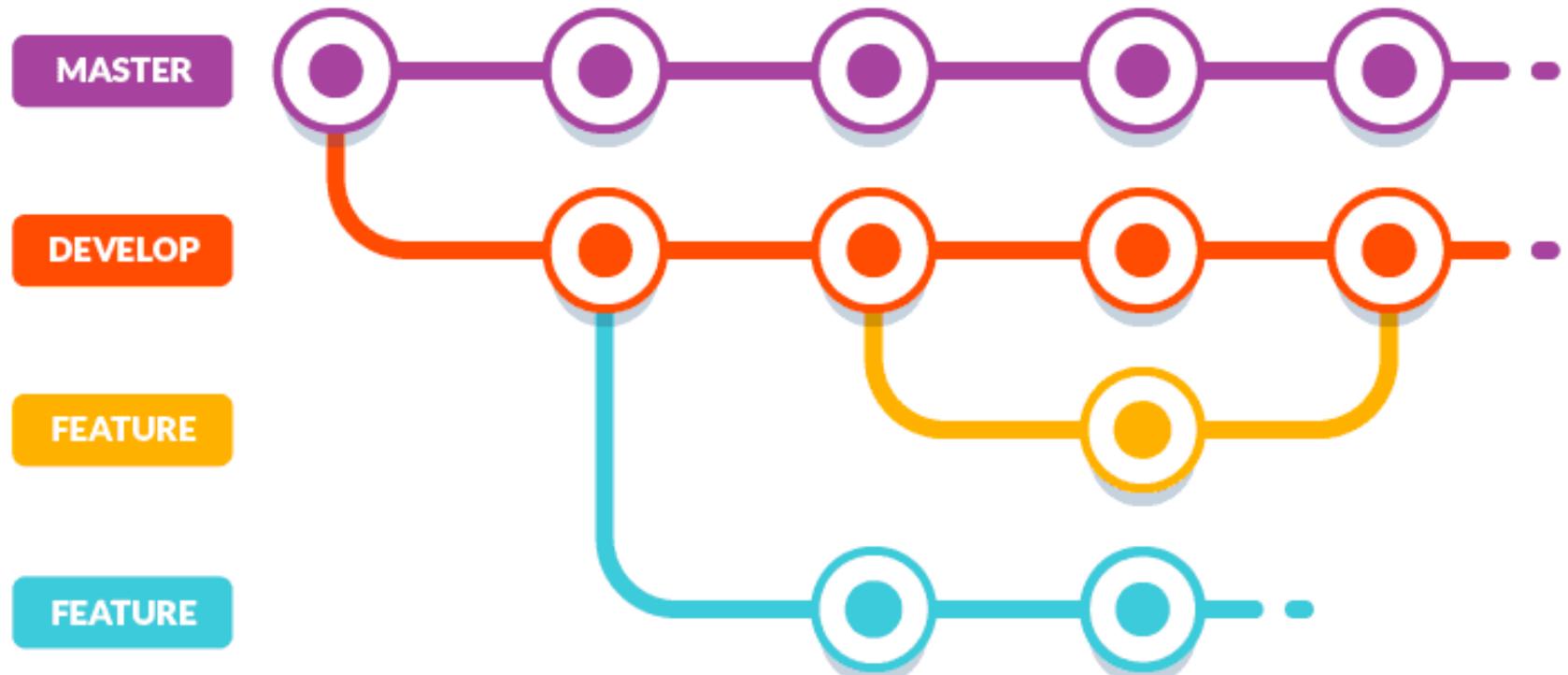
Basic



Feature Branch



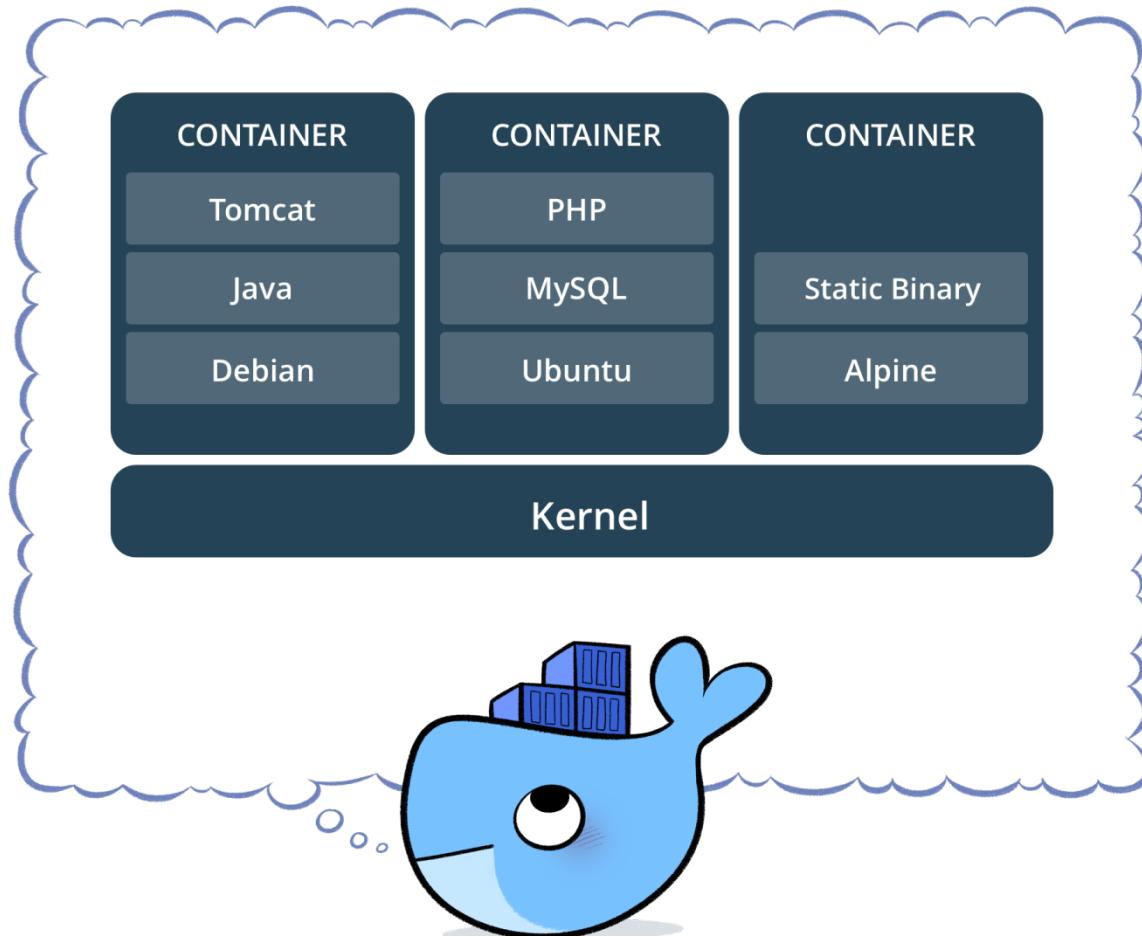
Gitflow



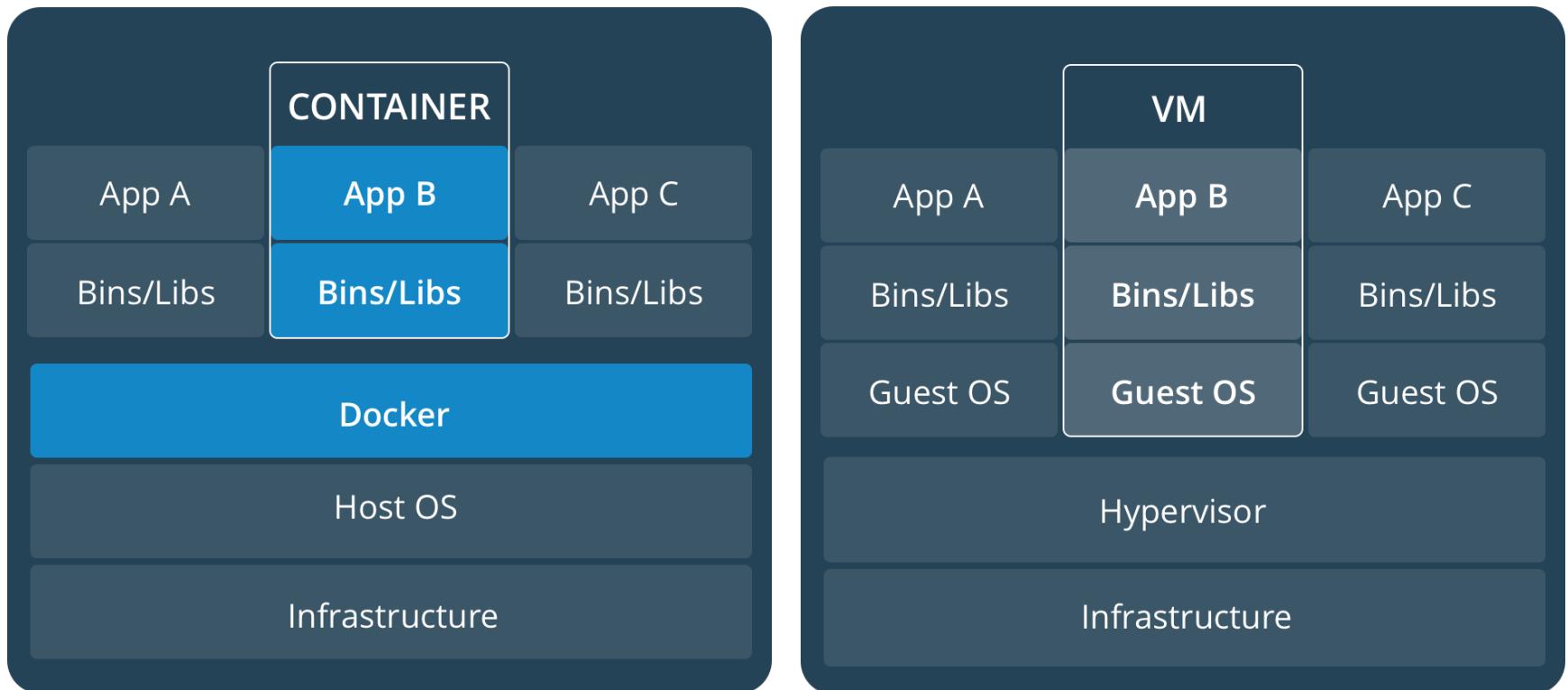
Containers virtualise with

DOCKER

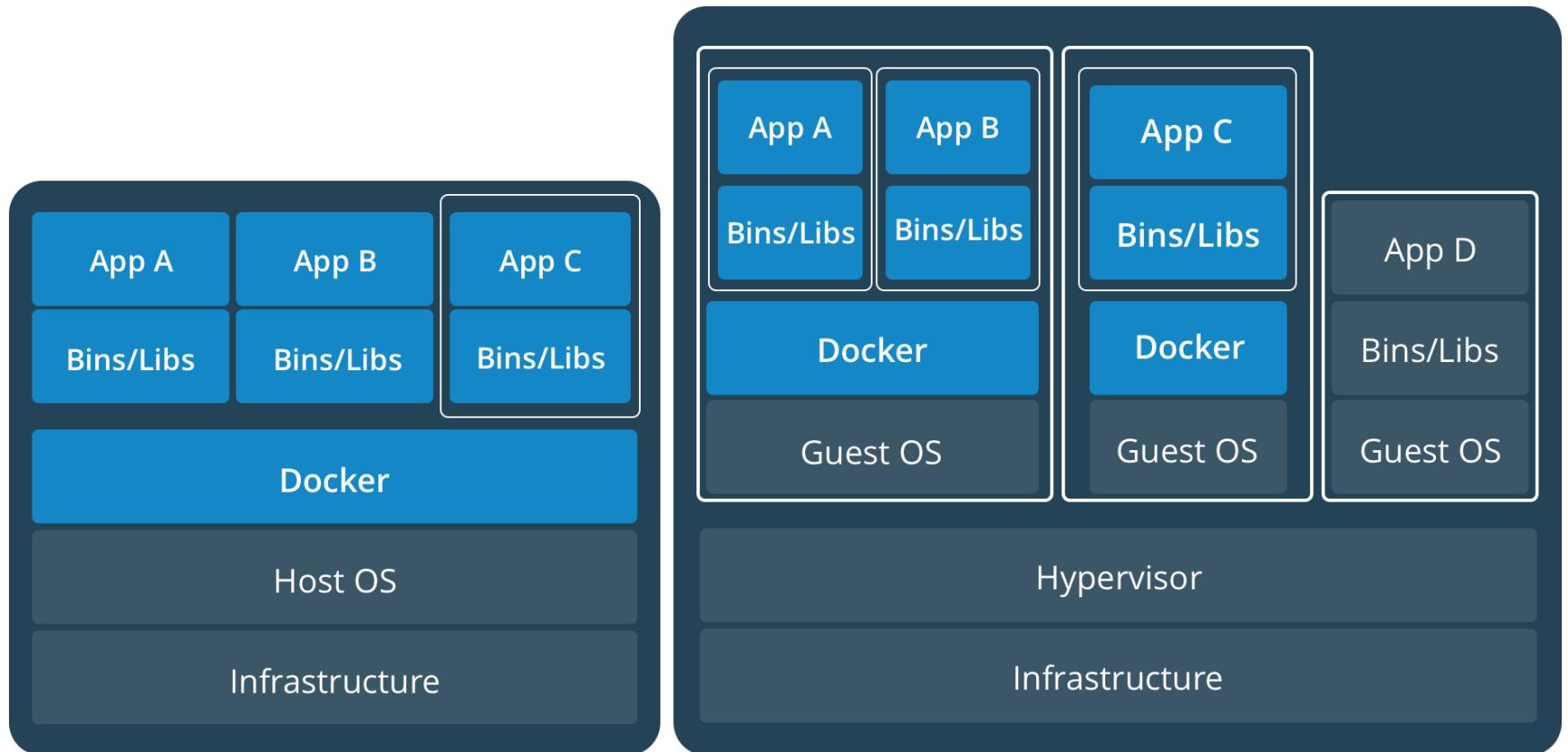
About Container



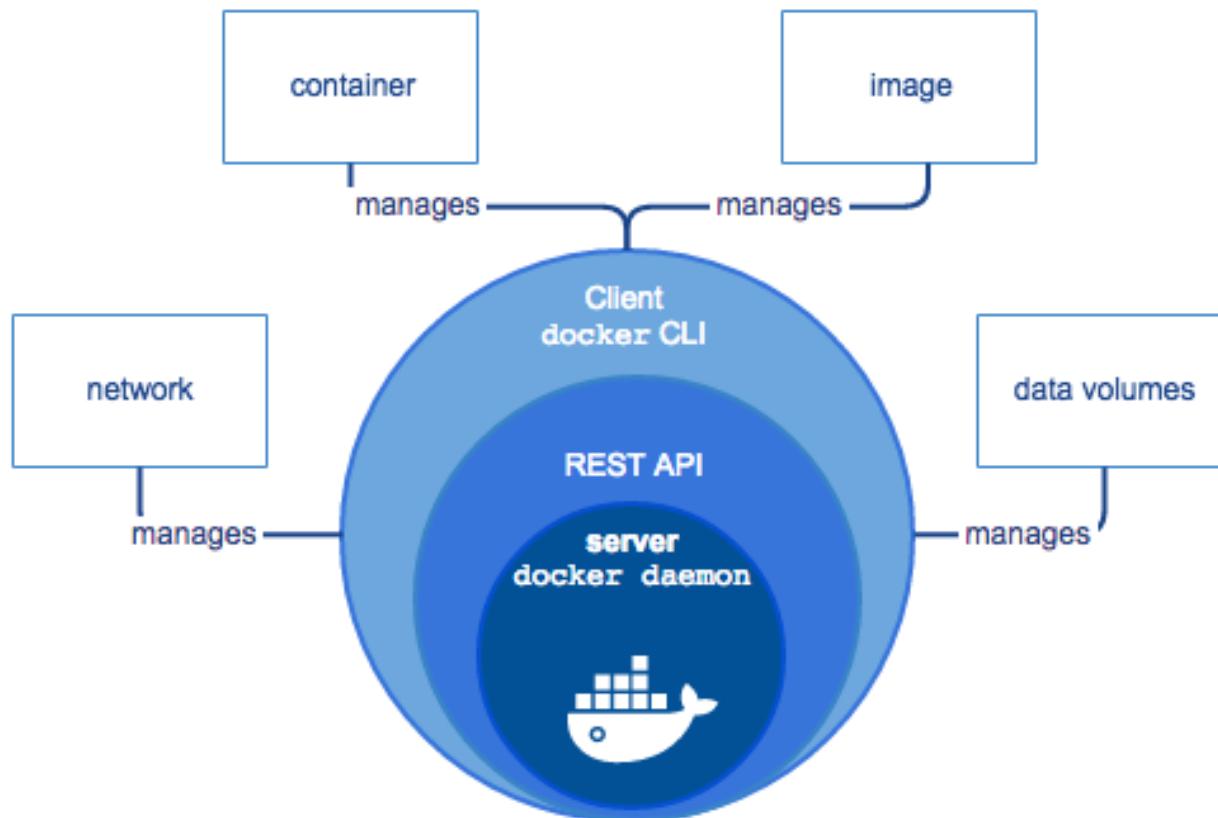
Comparing Containers and Virtual Machines



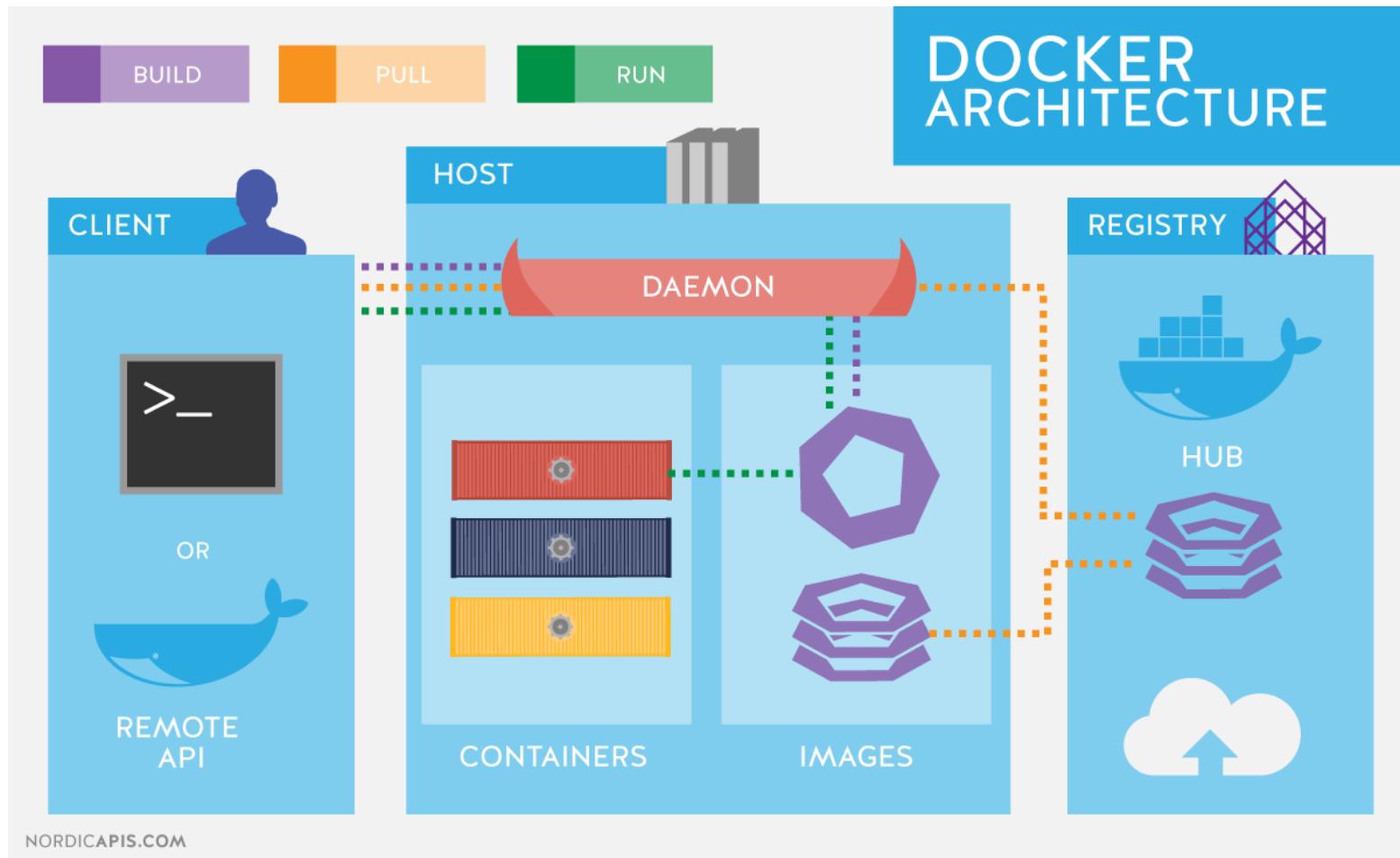
Containers and Virtual Machines Together



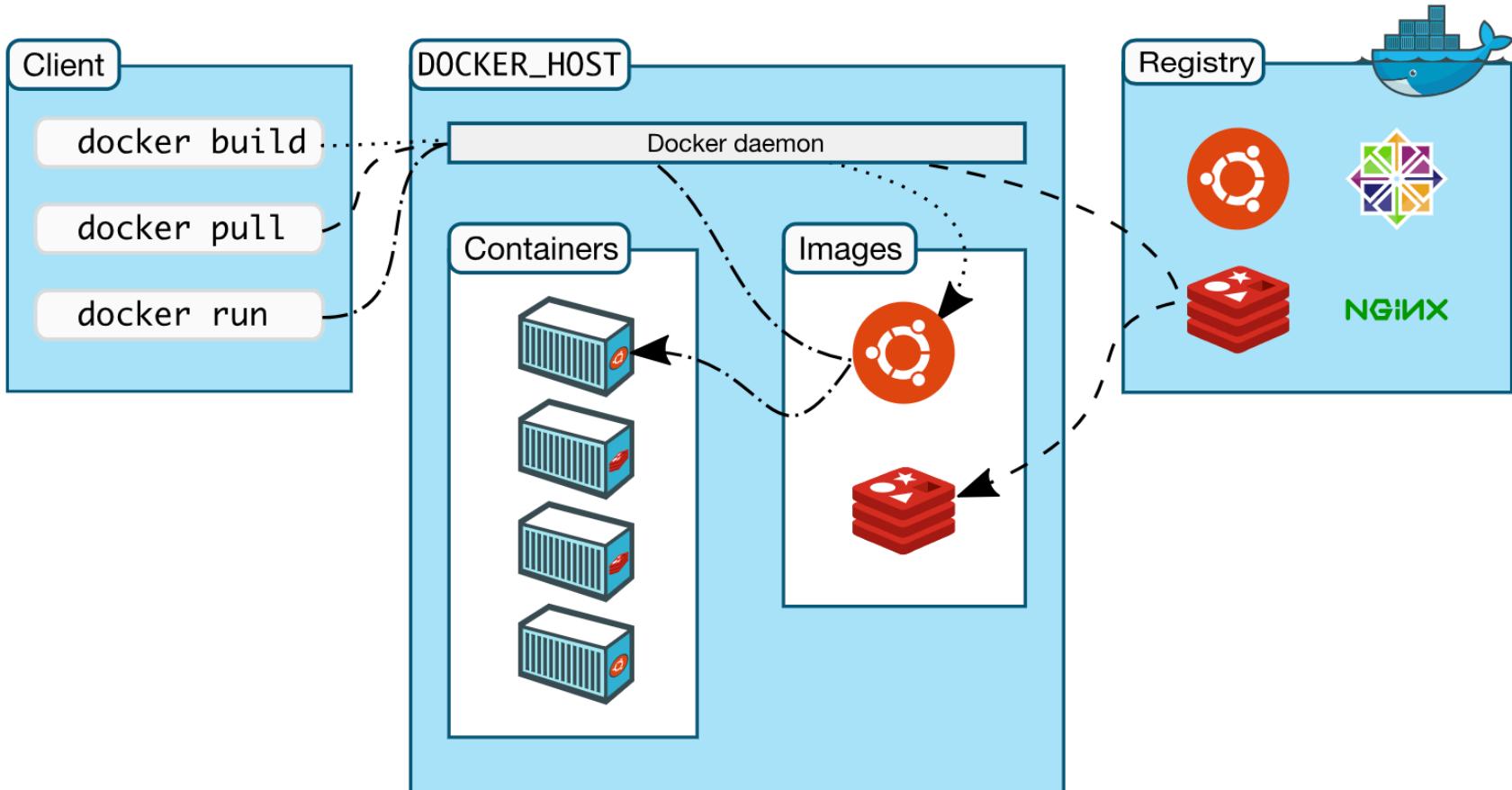
Docker engine



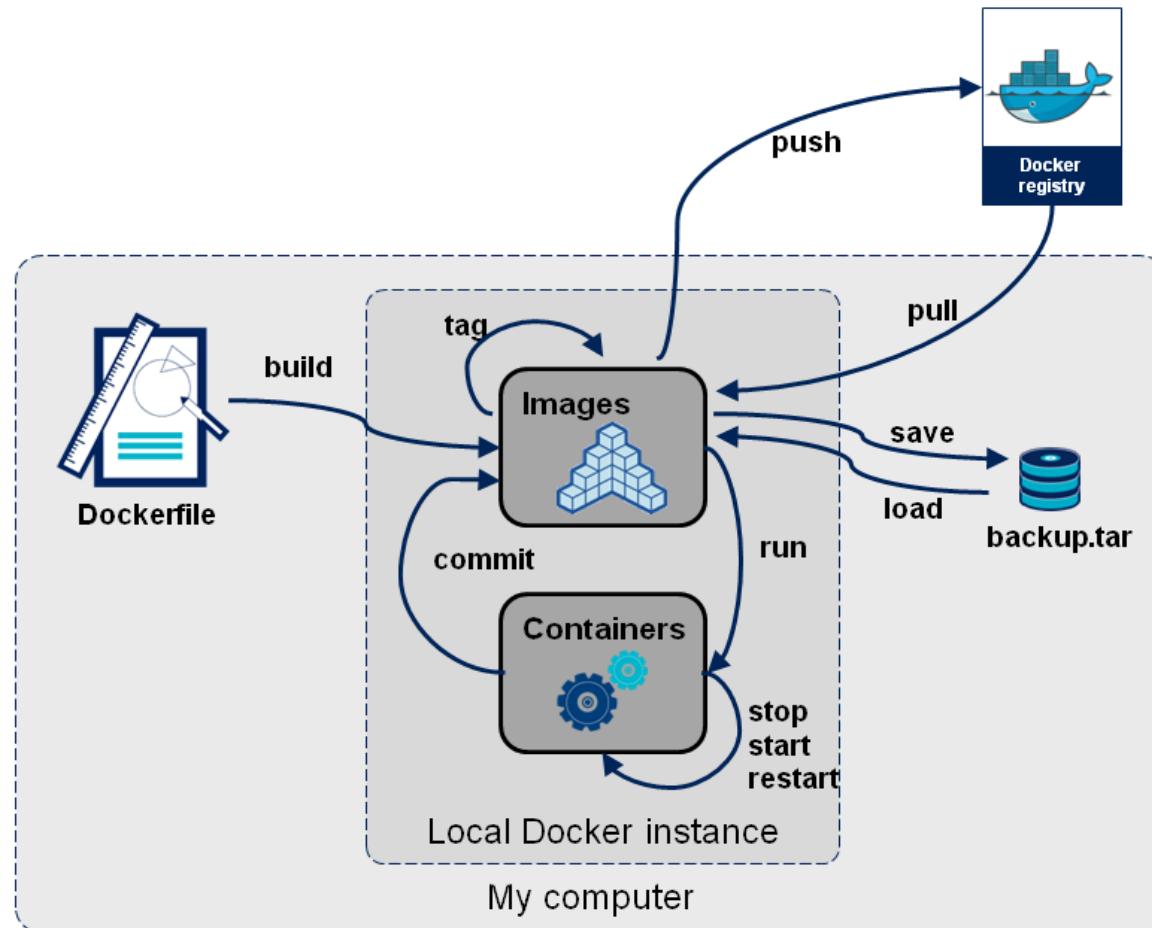
Docker Architecture



Docker Architecture#2



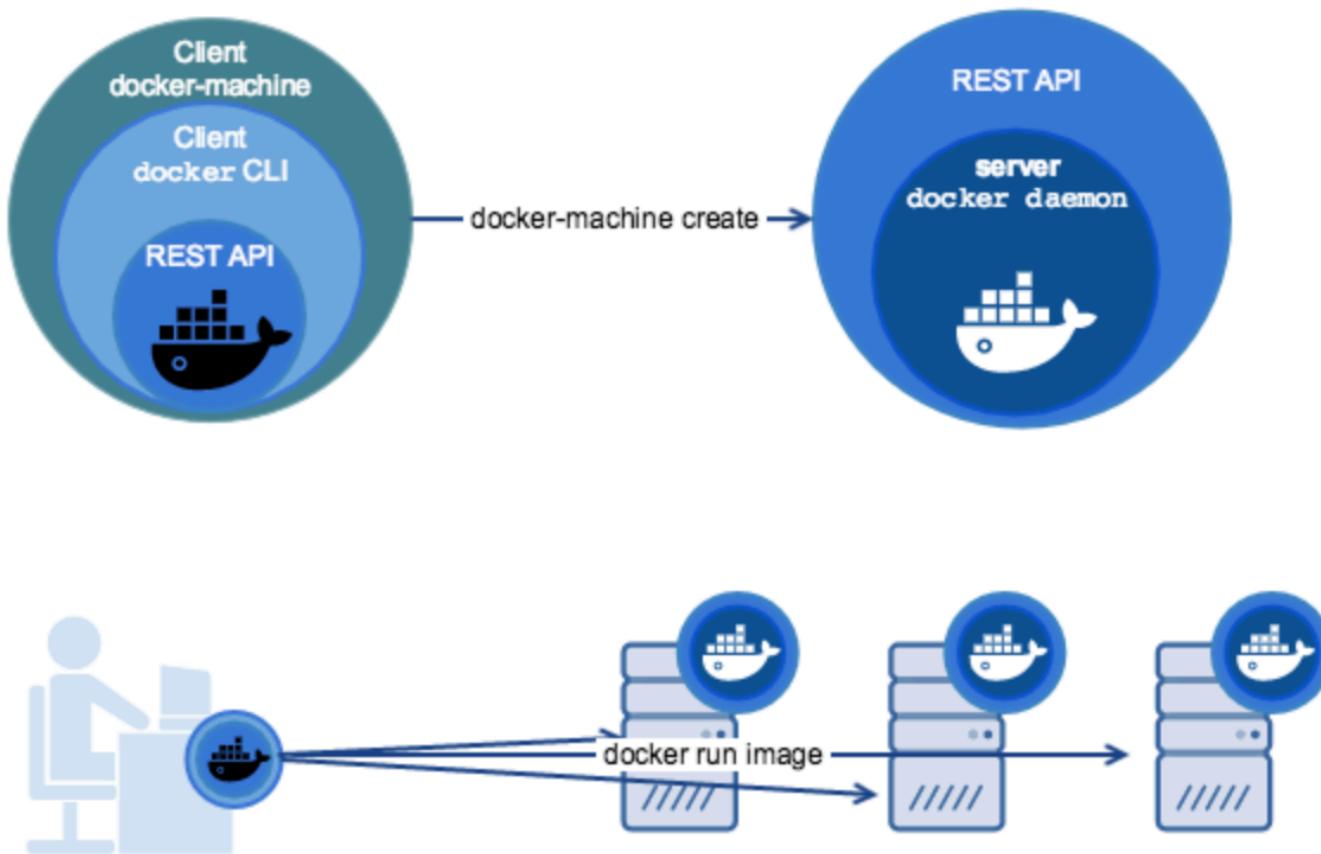
Docker Architecture#3



remote manage docker with

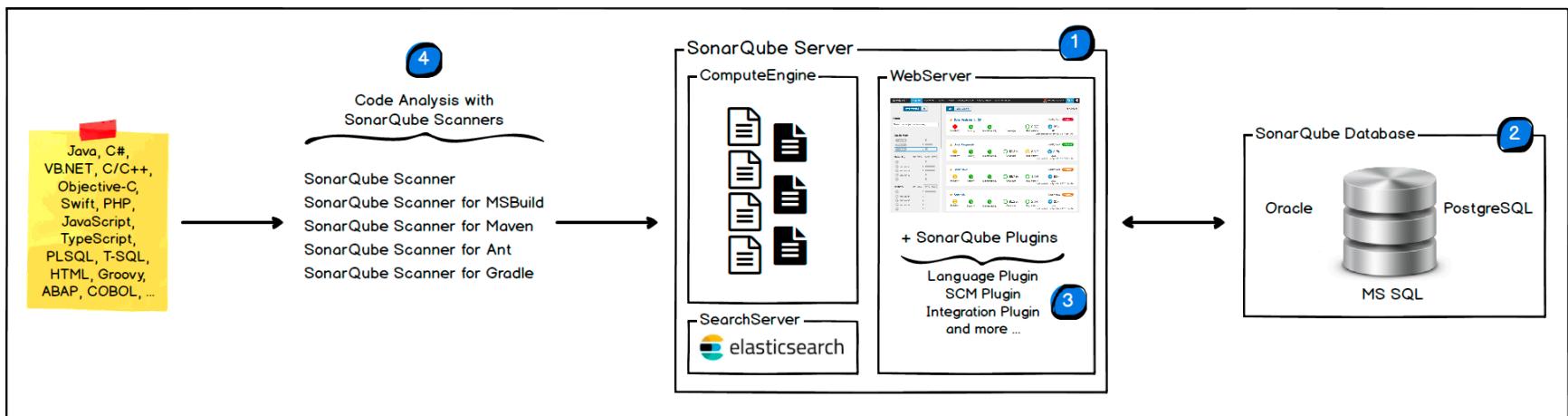
DOCKER MACHINE

Overview



Code Quality with
SONARQUBE

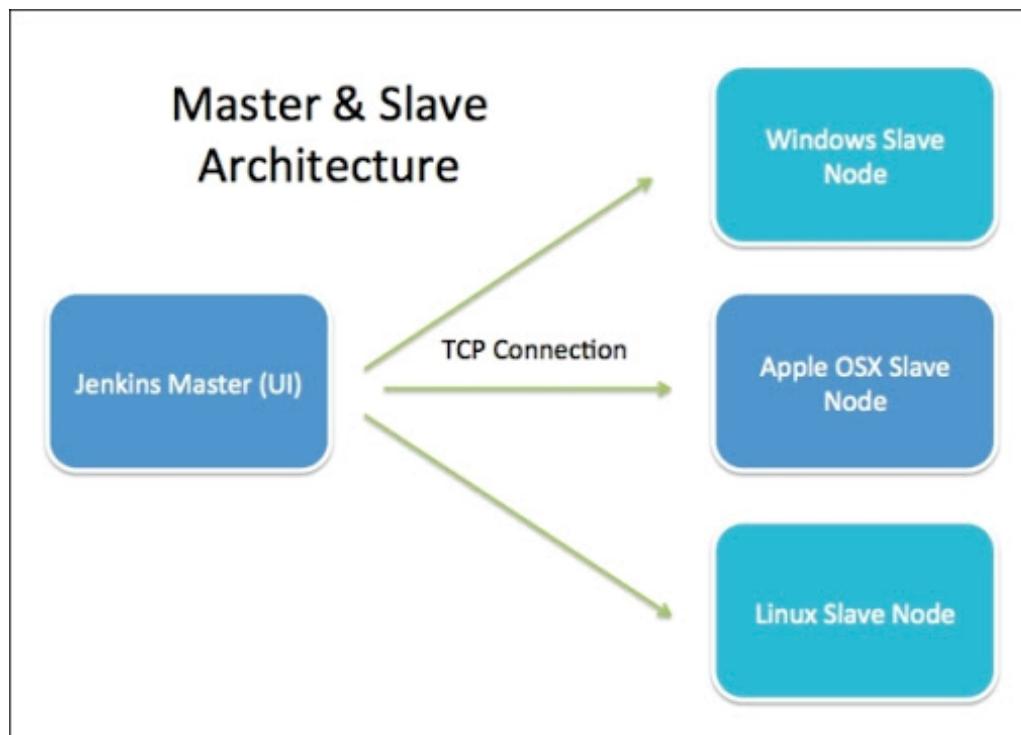
Architecture



Automate Build and Deploy with

JENKINS

Master & Slave Architecture



Choose Pipeline

Enter an item name

» Required field



Freestyle project

This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.



Pipeline

Orchestrates long-running activities that can span multiple build slaves. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.



External Job

This type of job allows you to record the execution of a process run outside Jenkins, even on a remote machine. This is designed so that you can use Jenkins as a dashboard of your existing automation system.



Multi-configuration project

Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

Pipeline script

General Build Triggers Advanced Project Options Pipeline

Advanced Project Options

Advanced...

Pipeline

Definition Pipeline script

Script

```
1 ~ node //{
2 ~   stage("stage 1") {
3 |     echo "Hello"
4 }
5 ~   stage("stage 2") {
6 |     echo "World!"
7 }
8 }
```

try sample Pipeline... ?

Use Groovy Sandbox ?

[Pipeline Syntax](#)

Save Apply

Pipeline template

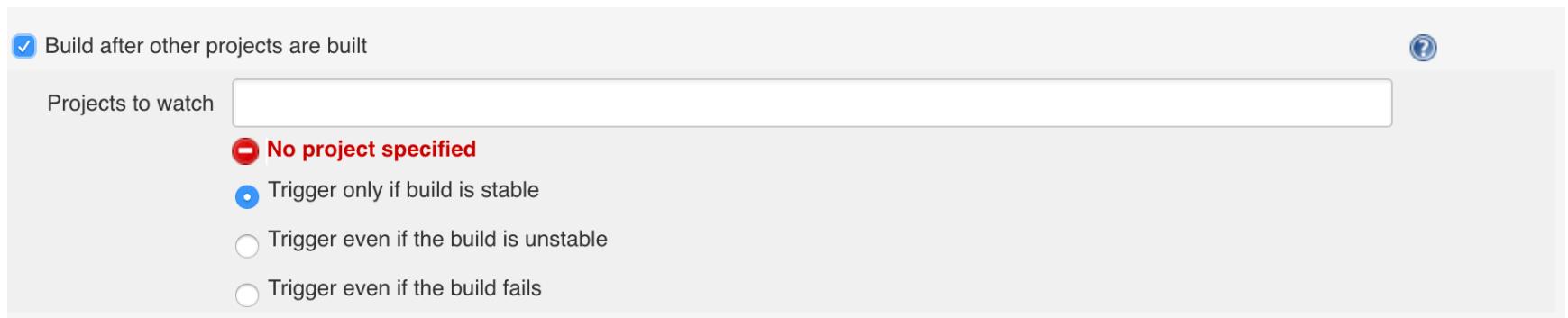
```
pipeline {
    agent any
    environment {
        APP_NAME = "test app name"
    }
    stages {
        stage('Build Image'){
            steps {
                sh "echo ${env.APP_NAME}"
            }
        }
    }
}
```

Jenkins Config Build Trigger

Build Triggers

- Build after other projects are built 
- Build periodically 
- GitHub hook trigger for GITScm polling 
- Poll SCM 
- Disable this project 
- Quiet period 
- Trigger builds remotely (e.g., from scripts) 

Build after other projects are built



The screenshot shows a configuration panel for a build step. At the top left is a checked checkbox labeled "Build after other projects are built". To its right is a blue circular icon with a question mark. Below this is a section titled "Projects to watch" containing an empty input field and a red error message "No project specified". Underneath are three radio button options: "Trigger only if build is stable" (selected), "Trigger even if the build is unstable", and "Trigger even if the build fails".

สั่งให้ทำงานอัตโนมัติ หลังจากที่ project กำหนดทำงานเสร็จ โดยมี 3 option คือ
ทำงานเมื่อ build สำเร็จ ไม่มี warning
ทำงานเมื่อ build สำเร็จ แบบมี warning
ทำงานไม่สำเร็จ

Build periodically

Build periodically ?

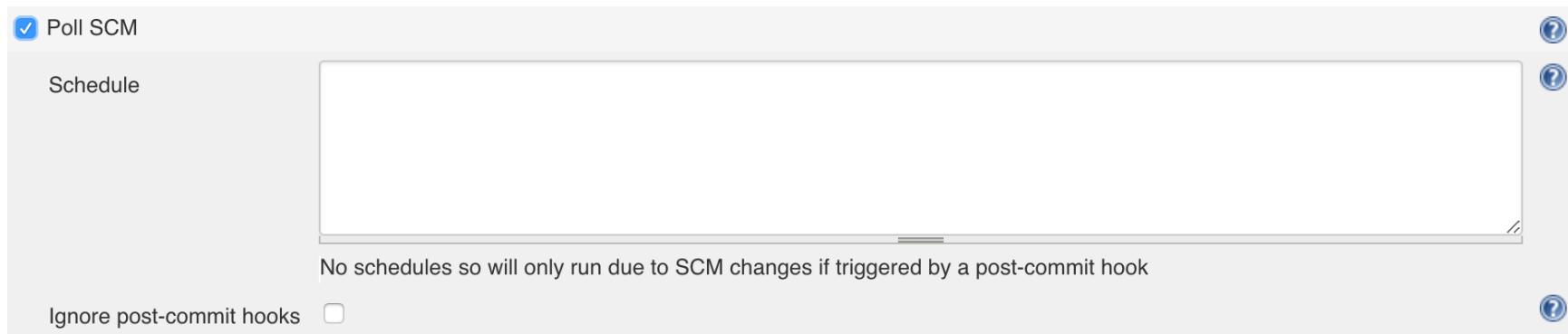
Schedule ?

Would last have run at Wednesday, June 27, 2018 9:43:05 PM ICT; would next run at Wednesday, June 27, 2018 9:55:05 PM ICT.

This field follows the syntax of cron (with minor differences). Specifically, each line consists of 5 fields separated by TAB or whitespace:
MINUTE HOUR DOM MONTH DOW
MINUTE Minutes within the hour (0–59)
HOUR The hour of the day (0–23)
DOM The day of the month (1–31)
MONTH The month (1–12)
DOW The day of the week (0–7) where 0 and 7 are Sunday.

ສ້າງໃຫ້ກໍາງານວັດໂນມັຕິຕາມໜ່ວຍເວລາທີ່ກໍານົດ ໂດຍສາມາດກໍາບັນດາເວລາໄດ້ວູ້
ໃນຮູບແບບ crontab format

Poll SCM



สั่งให้ทำงานอัตโนมัติตามช่วงเวลาที่กำหนด โดยสามารถกำหนดเวลาได้อยู่ในรูปแบบ crontab format

โดยมีลักษณะการทำงานคือจะไปทำการตรวจสอบดูก่อนว่า source code มีการเปลี่ยนแปลงหรือไม่ ถ้าไม่มีเปลี่ยนแปลงก็จะไม่ทำการ build job

Trigger builds remotely

Build Triggers

- Build after other projects are built (?)
- Build periodically (?)
- GitHub hook trigger for GITScm polling (?)
- Poll SCM (?)
- Disable this project (?)
- Quiet period (?)
- Trigger builds remotely (e.g., from scripts) (?)

Authentication Token 0817544663

Use the following URL to trigger build remotely: `JENKINS_URL/job/first/build?token=TOKEN_NAME` or `/buildWithParameters?token=TOKEN_NAME`
Optionally append `&cause=Cause+Text` to provide text that will be included in the recorded build cause.

เป็นการตั้งค่าให้สามารถสั่งให้ job ทำงานโดยการเรียกผ่าน url เช่น ใส่ค่า Authentication Token เป็น **123** Jenkins run ที่เครื่อง 127.0.0.1 ก็จะเรียกผ่าน url ดังนี้

`http://127.0.0.1:8080/job/{job_name}/build?token=123`

Jenkins Job List

Jenkins [ENABLE AUTO REFRESH](#)

[New Item](#) [add description](#)

[People](#)

[Build History](#)

[Project Relationship](#)

[Check File Fingerprint](#)

[Manage Jenkins](#)

[My Views](#)

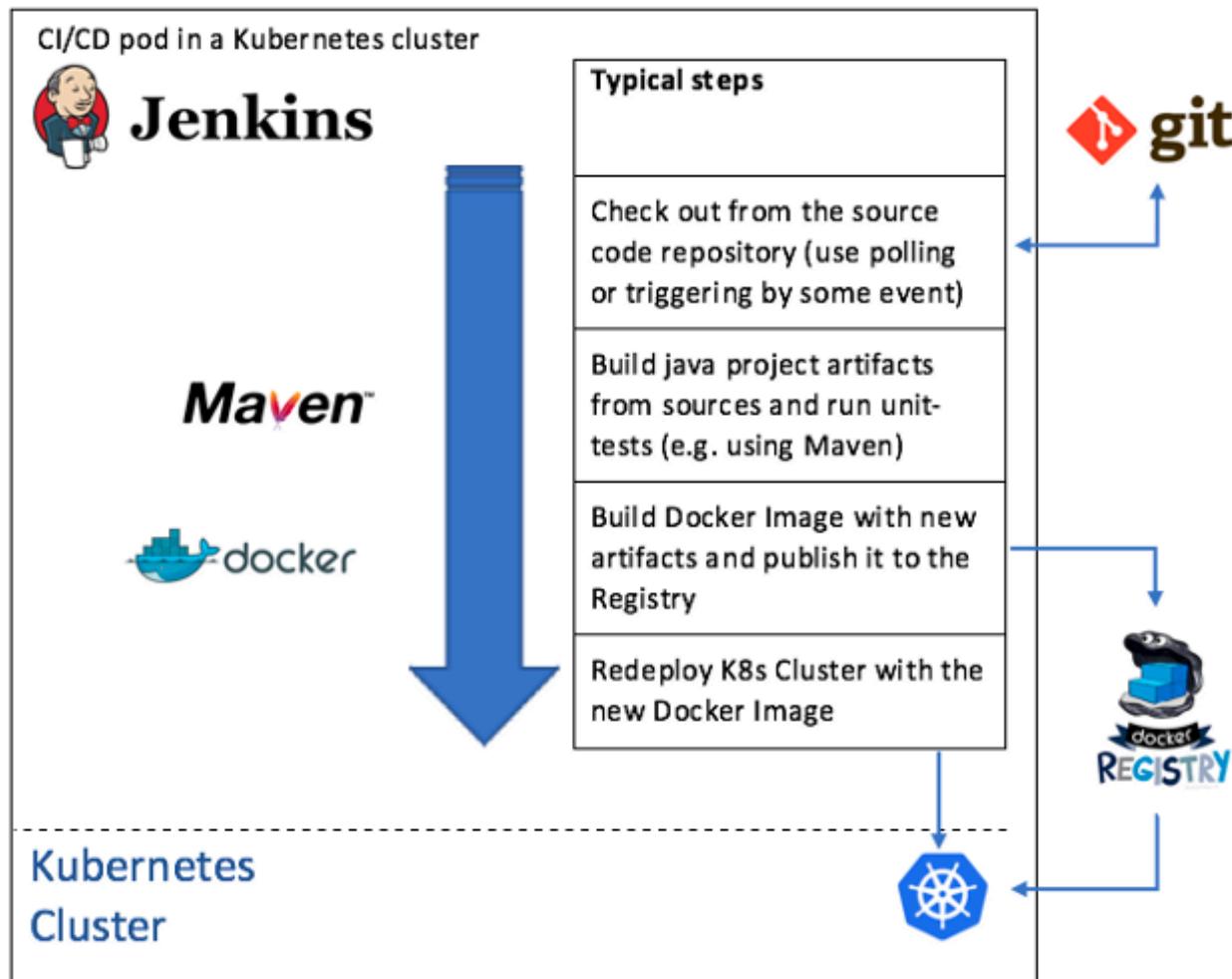
[Open Blue Ocean](#)

[Credentials](#)

[New View](#)

S	W	Name ↓	Last Success	Last Failure	Last Duration	Fav
		api.pnpsw.com	1 mo 0 days - #22	4 mo 25 days - #13	23 sec	
		api.wisdomairways.com	5 days 10 hr - #25	8 days 11 hr - #20	2 min 8 sec	
		api.zone9sport.com	4 mo 14 days - #11	5 mo 4 days - #8	45 sec	
		birt.pnpsw.com	2 mo 17 days - #8	N/A	57 sec	
		booking.wisdomairways.com	5 days 10 hr - #5	N/A	1 min 46 sec	
		checker.pnpsw.com	5 mo 10 days - #9	5 mo 10 days - #8	50 sec	

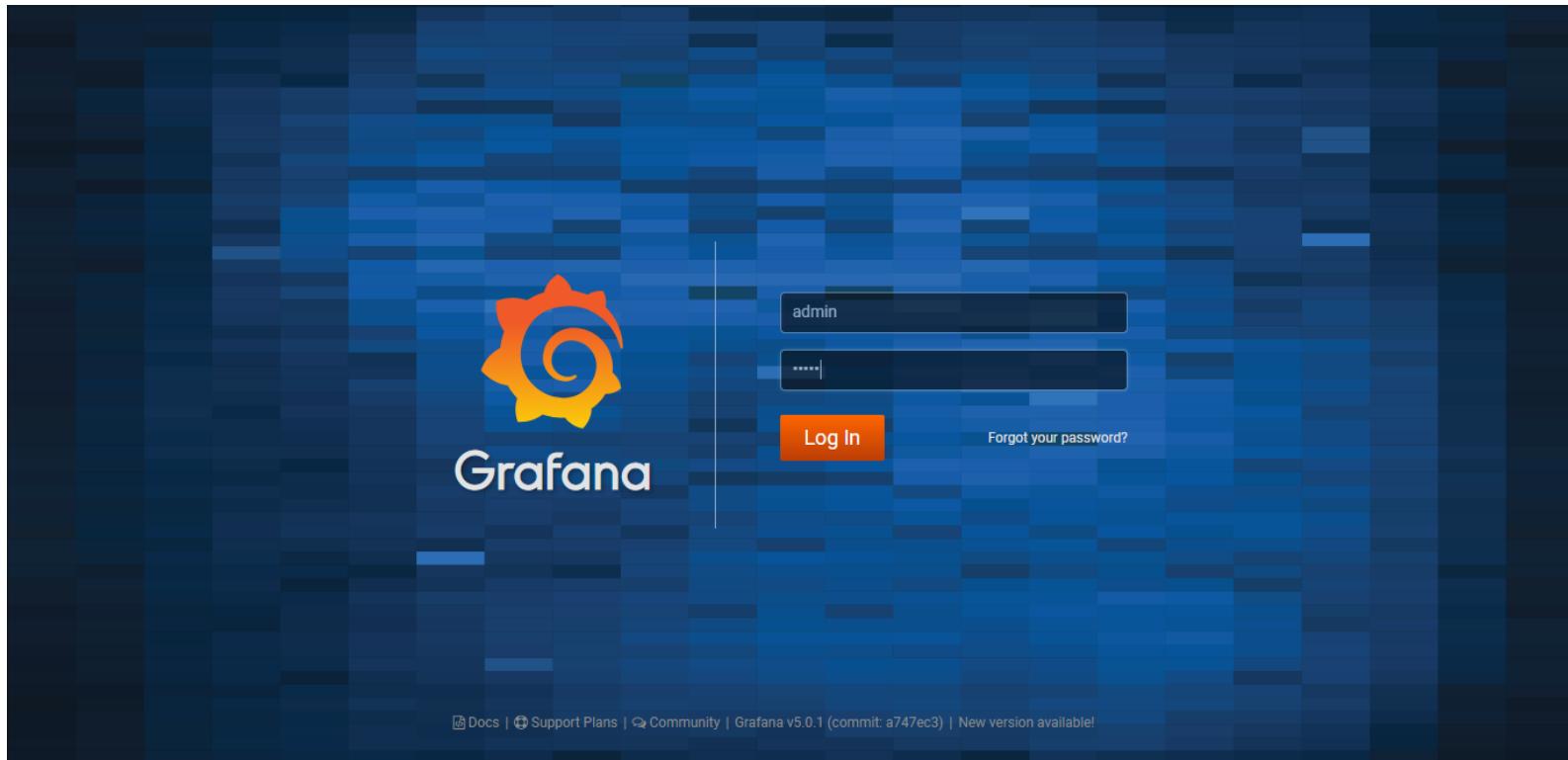
Jenkins with k8s



Monitoring with

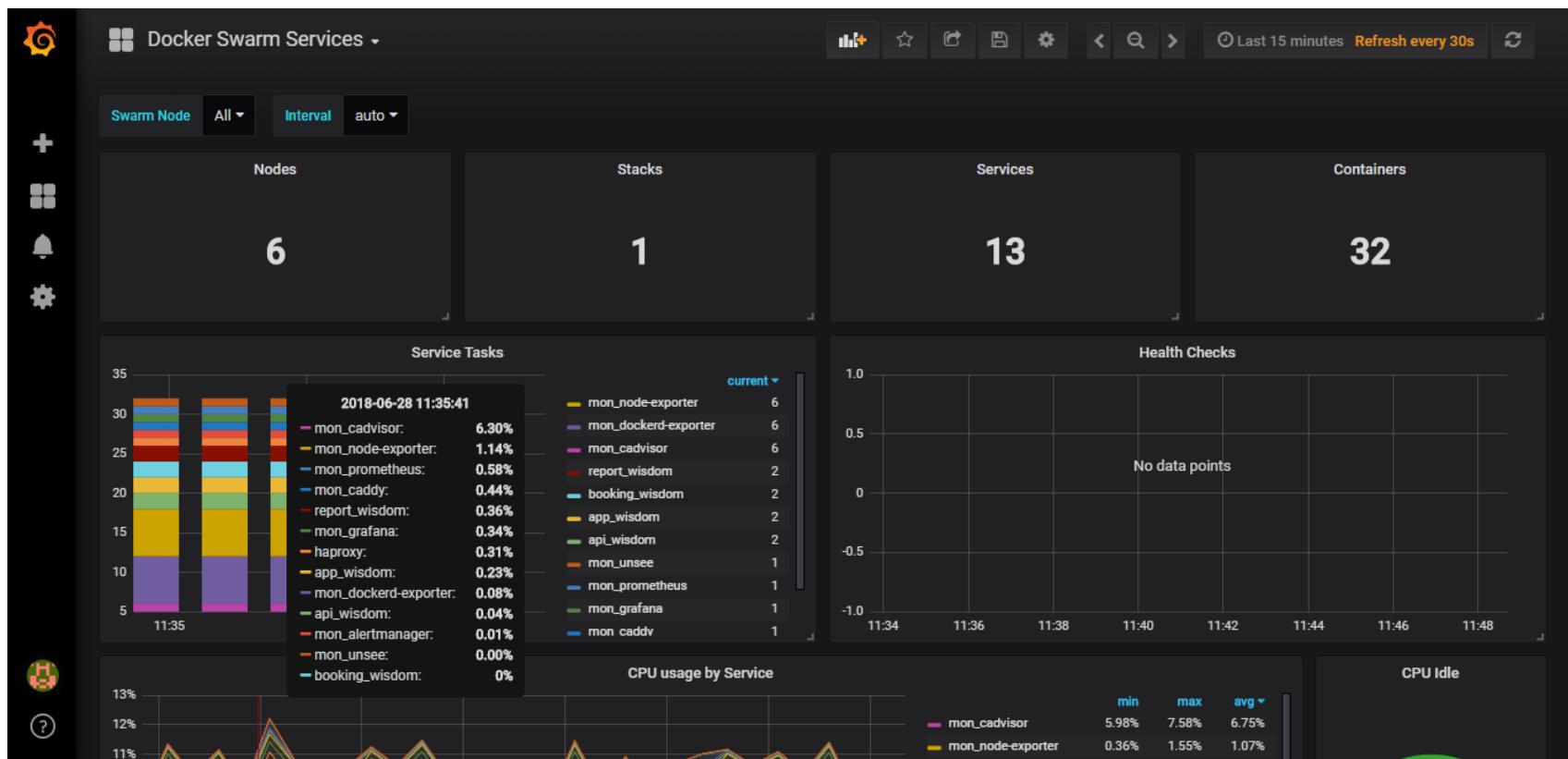
GRAFANA

การใช้งาน Grafana



การใช้งาน Grafana

- หน้าจอแสดงรายละเอียดแต่ละ Service



DevOps

WORKSHOP

Software

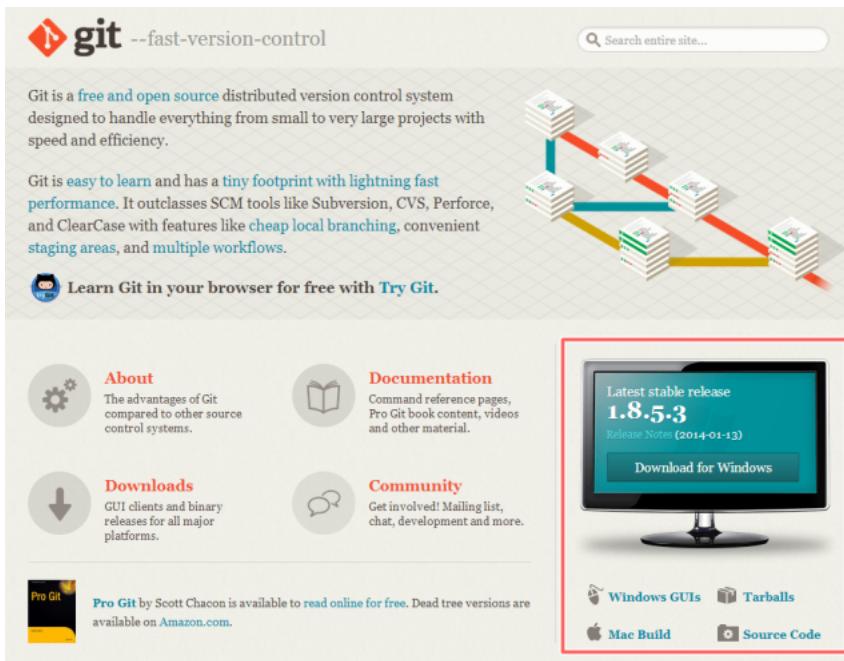
INSTALLATION

การติดตั้ง

GIT FOR WINDOWS

การติดตั้ง GIT สำหรับ Windows

- เข้า website <https://git-scm.com/downloads>
- เลือก Download Git เพื่อติดตั้งบน Windows



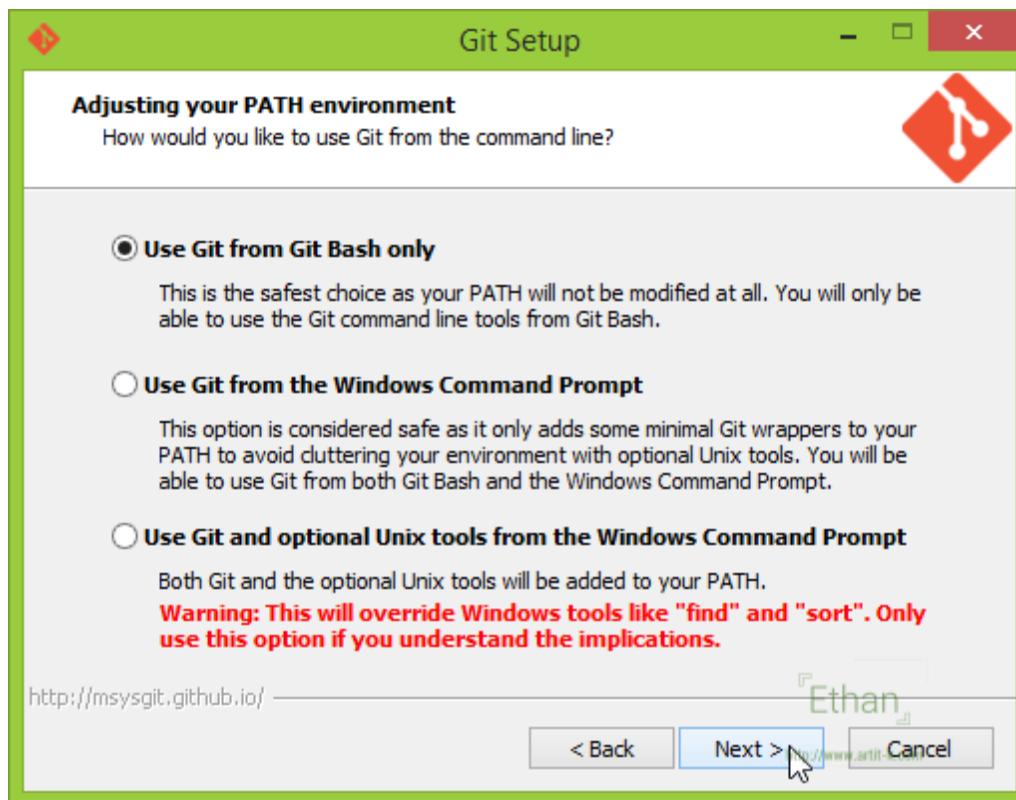
การติดตั้ง GIT สำหรับ Windows #2

- ติดตั้งโดยใช้สิทธิ Administrator ในการติดตั้ง



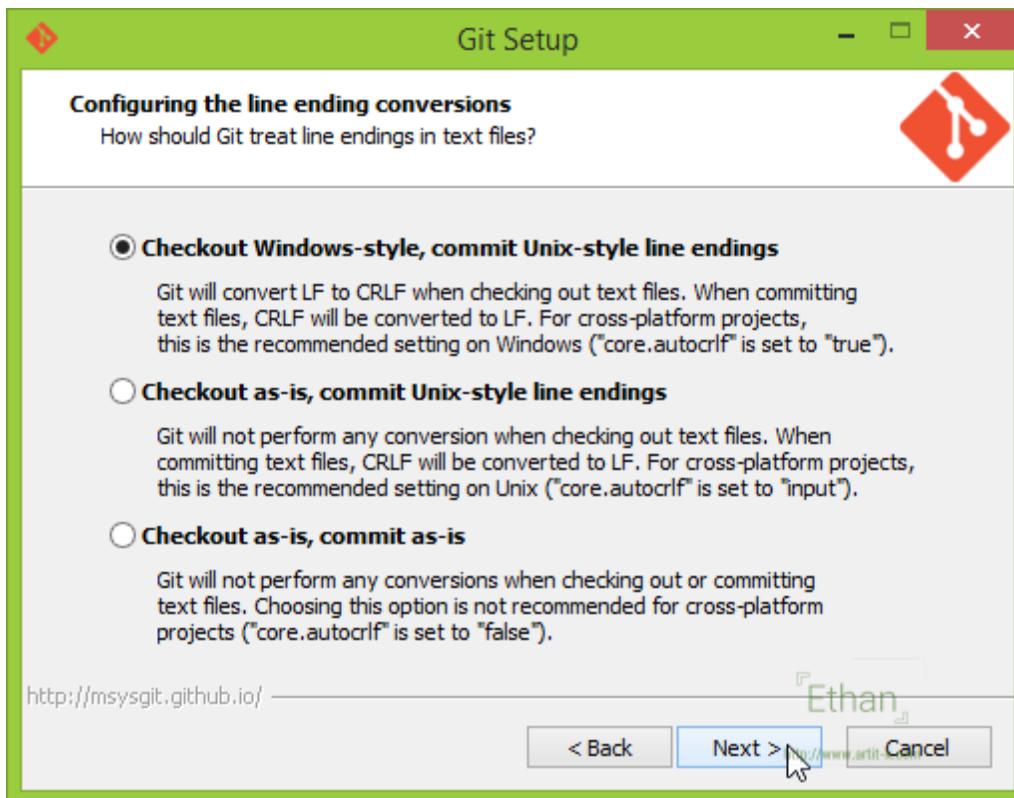
การติดตั้ง GIT สำหรับ Windows #3

- เลือกติดตั้งแบบ **Use Git from the Windows Command Prompt**



การติดตั้ง GIT สำหรับ Windows #4

- เลือกเป็น **Checkout Windows-style, commit Unix-style line endings**



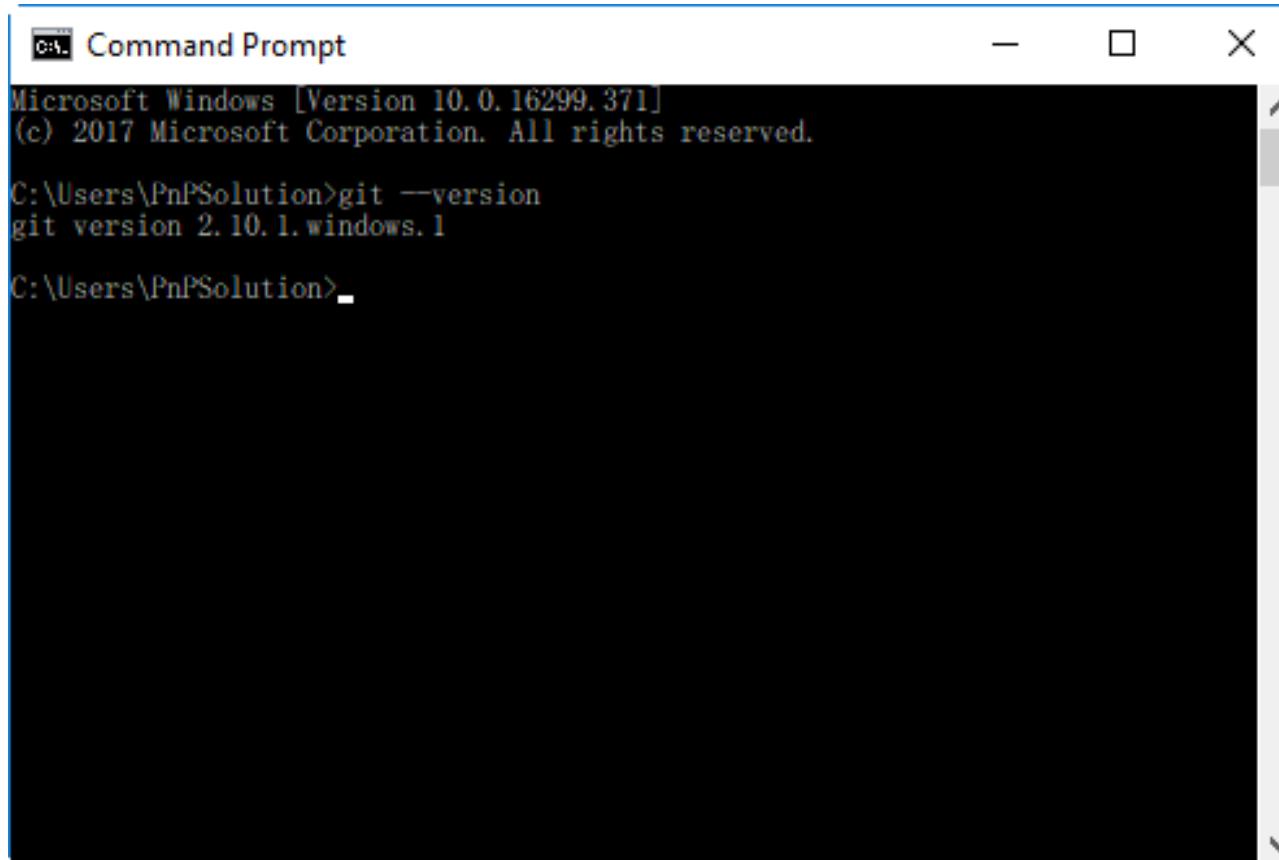
การติดตั้ง GIT สำหรับ Windows #5

- กดปุ่ม next ไปจนถึงหน้าสุดท้าย



ทดสอบหลังการติดตั้ง Git

- เปิดโปรแกรม cmd และพิมพ์คำสั่ง git --version



```
Command Prompt
Microsoft Windows [Version 10.0.16299.371]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\PnP\Documents>git --version
git version 2.10.1.windows.1

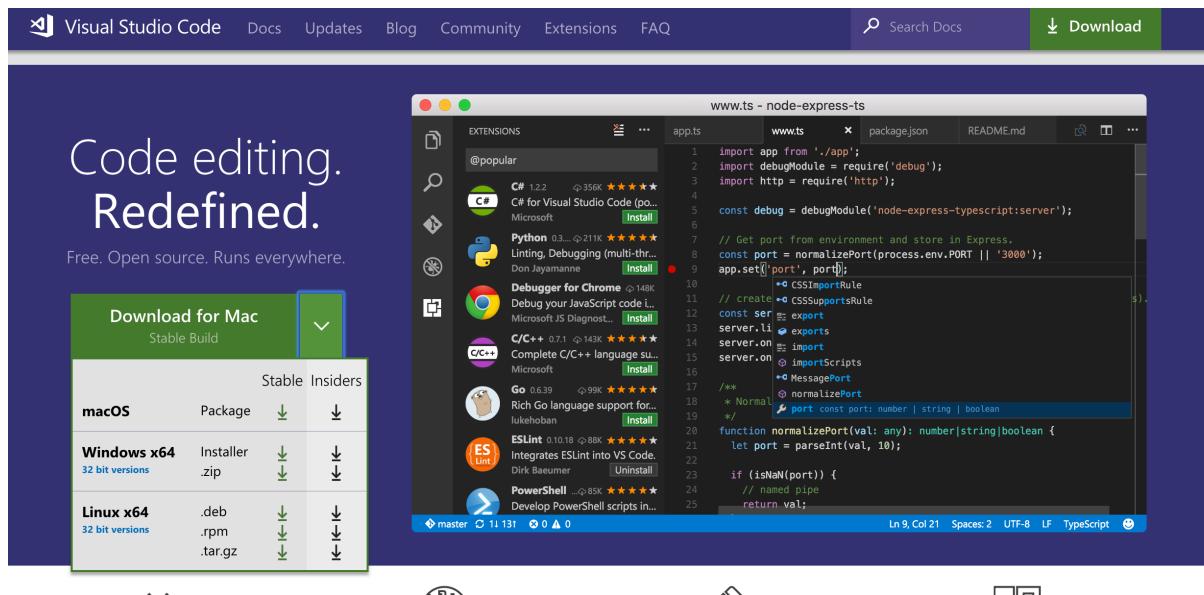
C:\Users\PnP\Documents>
```

การติดตั้ง

VISUAL STUDIO CODE

การติดตั้ง VSC

- เข้าไปที่ website <https://code.visualstudio.com/>
- เลือก download สำหรับ windows (stable)

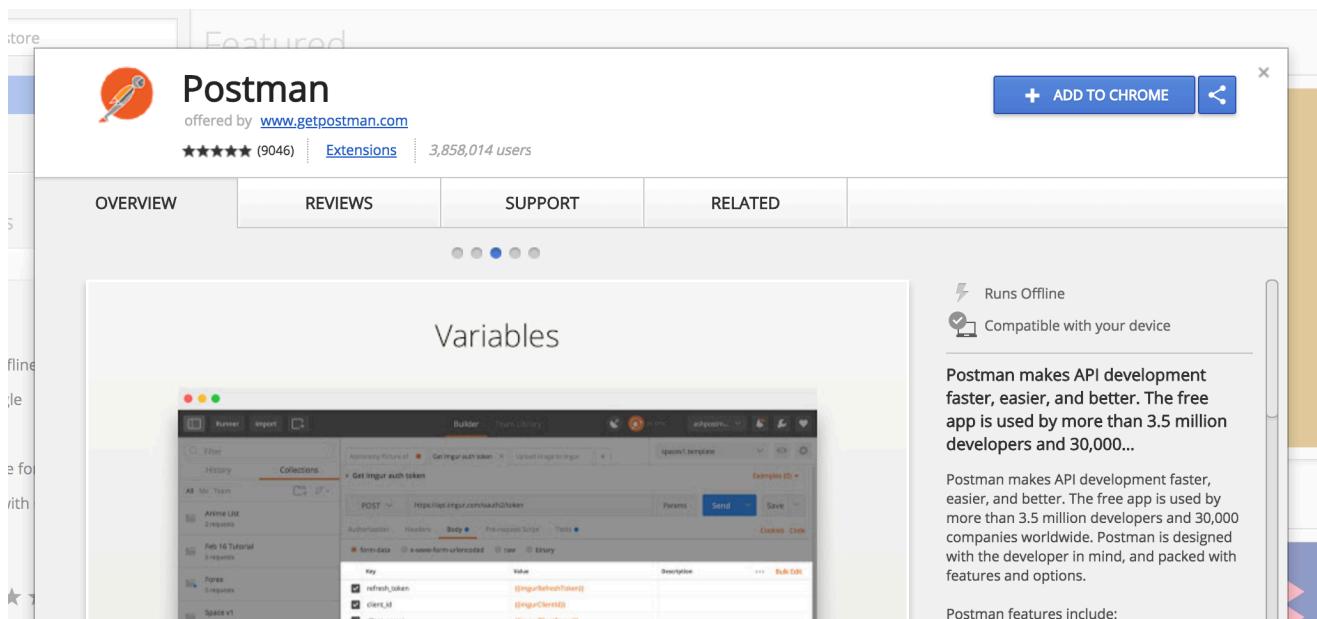


การติดตั้ง

POSTMAN

การติดตั้ง postman

- เข้า url <https://chrome.google.com/webstore/detail/postman/fhbjgbiflinjb dggeh cddcbn cddd domop?hl=en>



การติดตั้ง

VIRTUAL BOX

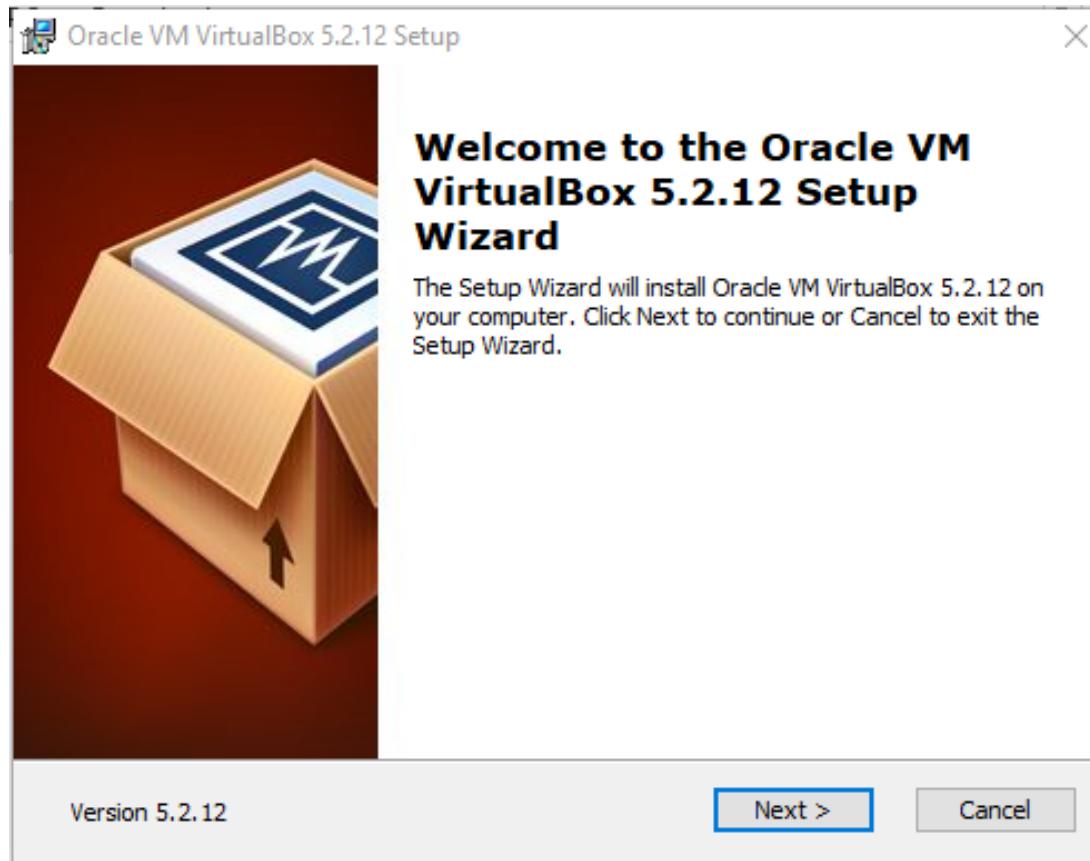
การติดตั้ง Virtual Box

- <https://www.virtualbox.org/wiki/Downloads>

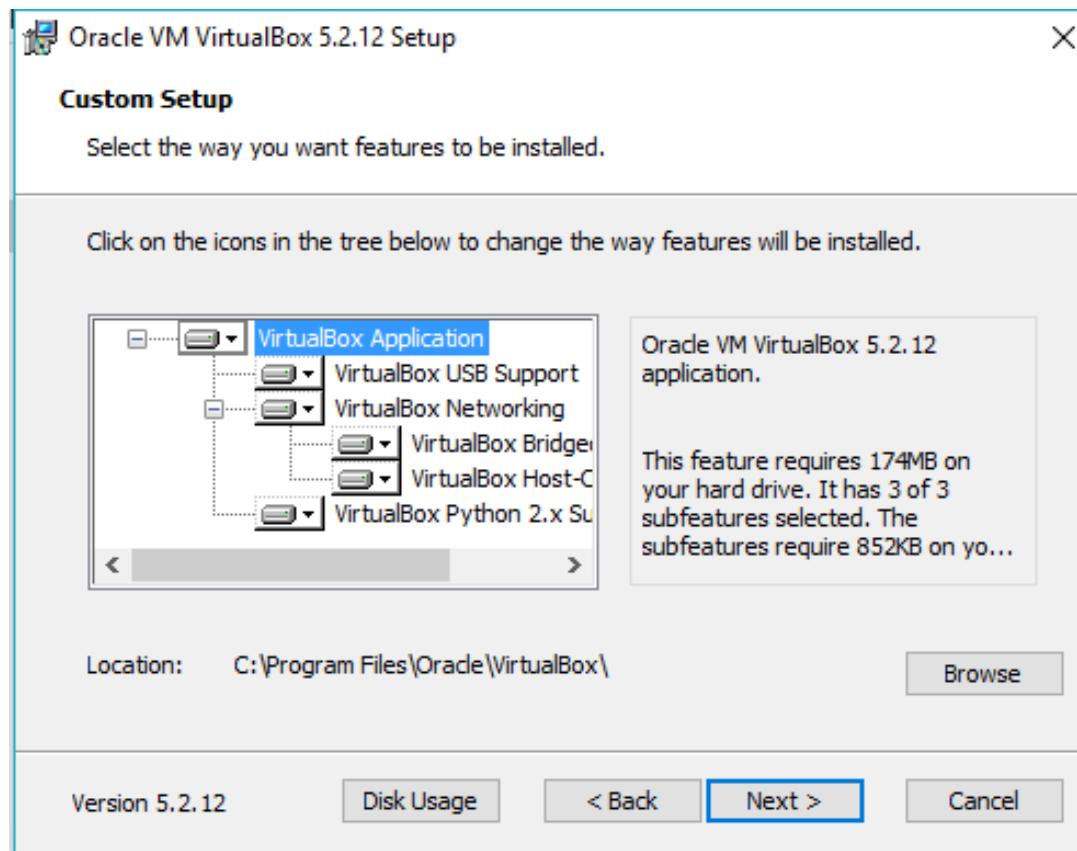


The screenshot shows the official VirtualBox download page. At the top left is the Oracle VM VirtualBox logo, which is a blue cube with the letters 'V' and 'M' on it, and 'ORACLE' and 'VirtualBox' written on its sides. To the right of the logo is the large title 'VirtualBox'. Below the title is a sub-header 'Download VirtualBox'. A horizontal dotted line separates this from the main content area. On the left side of the content area is a sidebar with links: 'About', 'Screenshots', 'Downloads', 'Documentation', 'End-user docs', 'Technical docs', 'Contribute', and 'Community'. The 'Downloads' link is highlighted in blue. The main content area starts with a paragraph: 'Here you will find links to VirtualBox binaries and its source code.' Below this is a section titled 'VirtualBox binaries' with the subtext: 'By downloading, you agree to the terms and conditions of the respective license.' It also mentions: 'If you're looking for the latest VirtualBox 5.1 packages, see [VirtualBox 5.1 builds](#). Consider upgrading.' Underneath is a section titled 'VirtualBox 5.2.12 platform packages' with a bulleted list: '⇒ Windows hosts', '⇒ OS X hosts', 'Linux distributions', and '⇒ Solaris hosts'. Below this is a note: 'The binaries are released under the terms of the GPL version 2.' Another note follows: 'See the [changelog](#) for what has changed.' A final note at the bottom states: 'You might want to compare the checksums to verify the integrity of downloaded packages. *The SHA256 checksums should be favored as the MD5 algorithm must be treated as insecure!*' with a bullet point: 'SHA256 checksums, MD5 checksums'. A note below that says: 'Note: After upgrading VirtualBox it is recommended to upgrade the guest additions as well.' At the very bottom of the content area is another section titled 'VirtualBox 5.2.12 Oracle VM VirtualBox Extension Pack' with a single bullet point: '⇒ All supported platforms'.

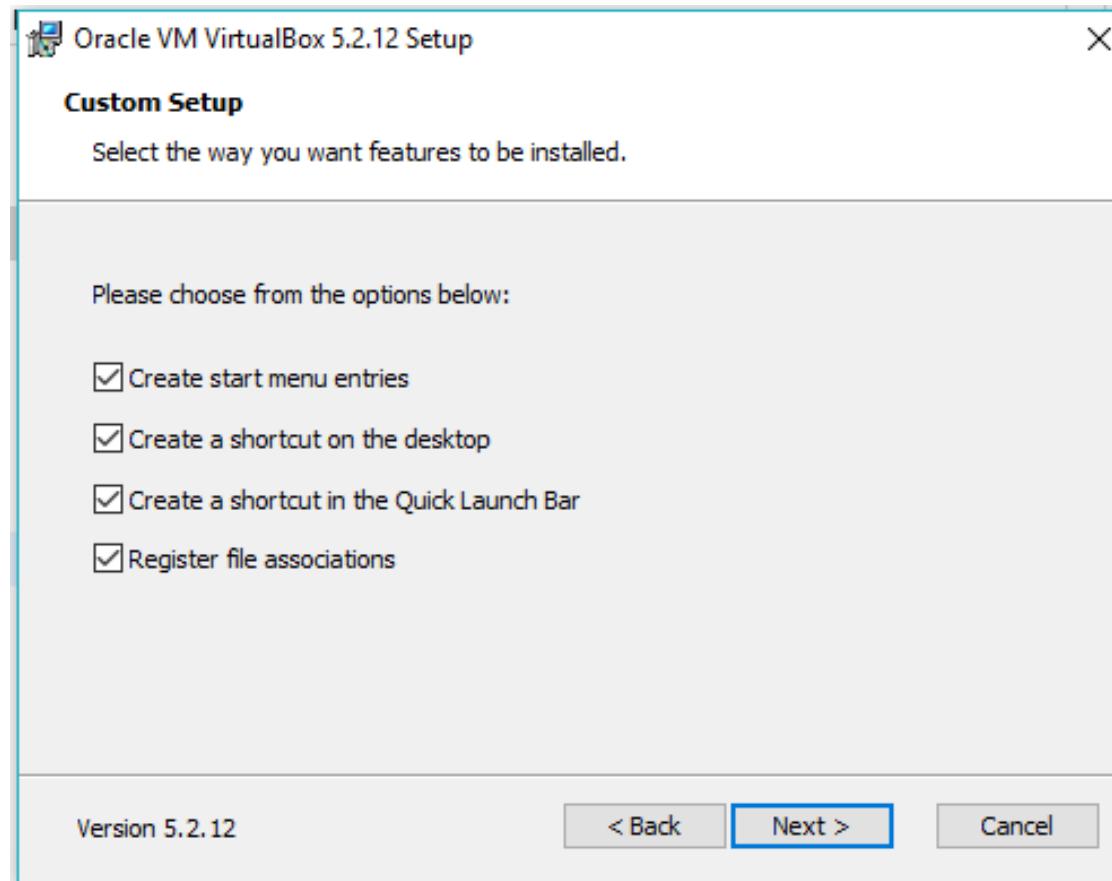
การติดตั้ง Virtual Box



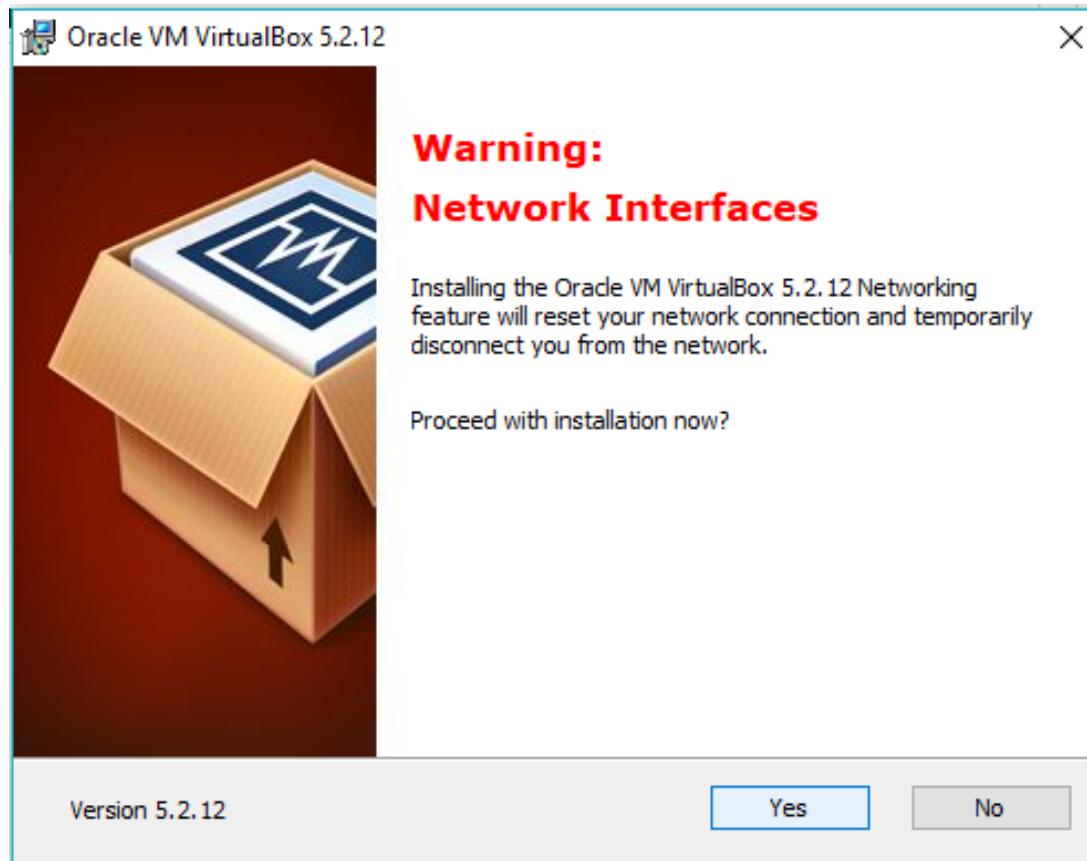
การติดตั้ง Virtual Box



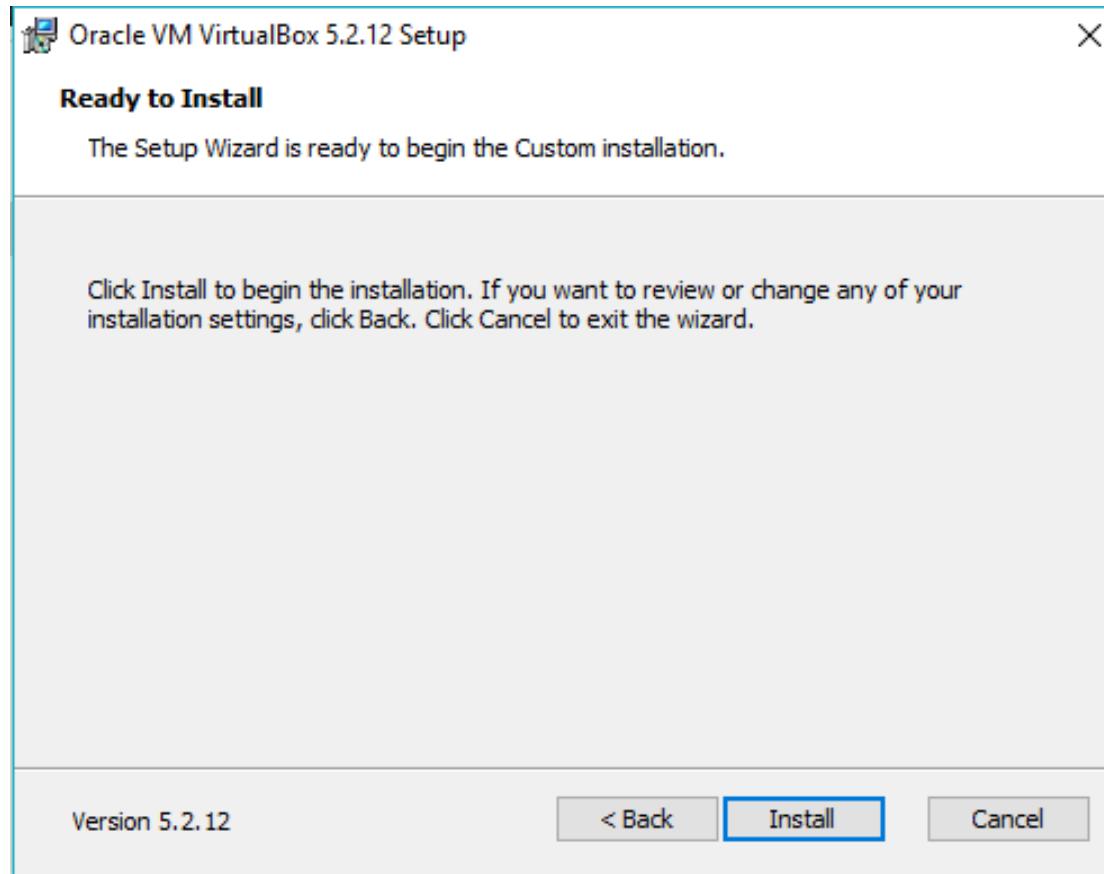
การติดตั้ง Virtual Box



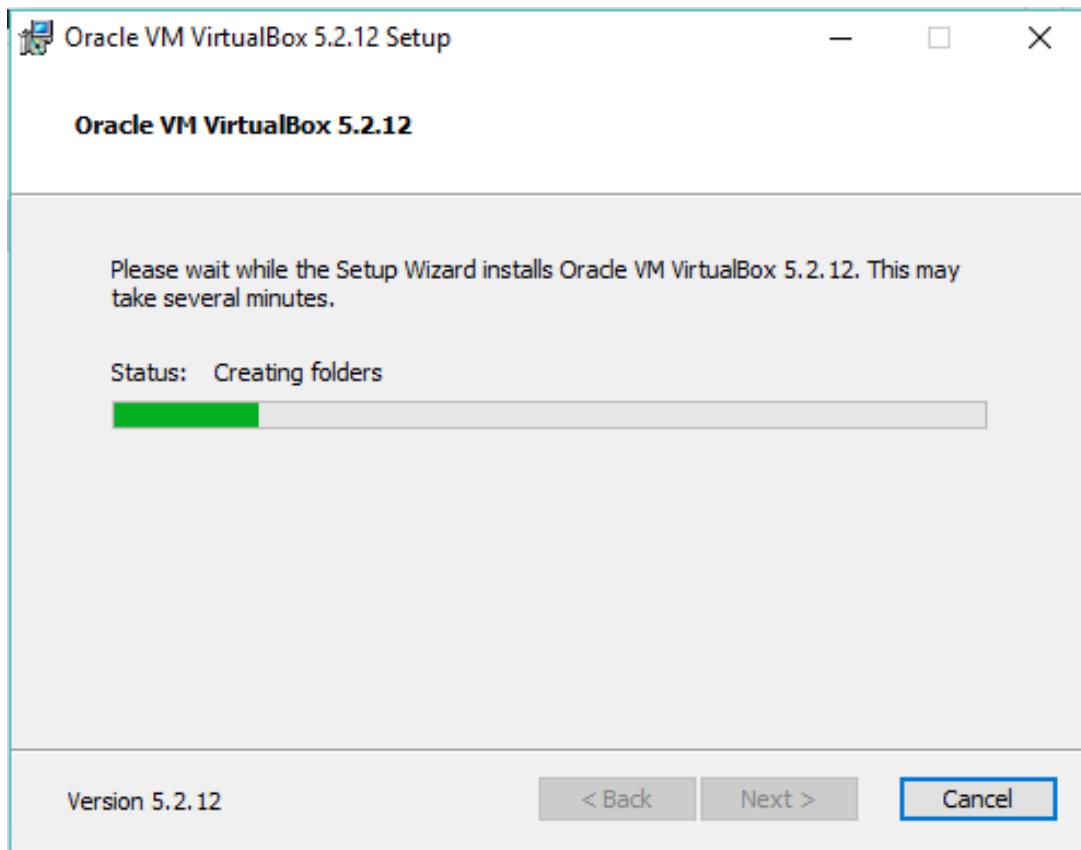
การติดตั้ง Virtual Box



การติดตั้ง Virtual Box



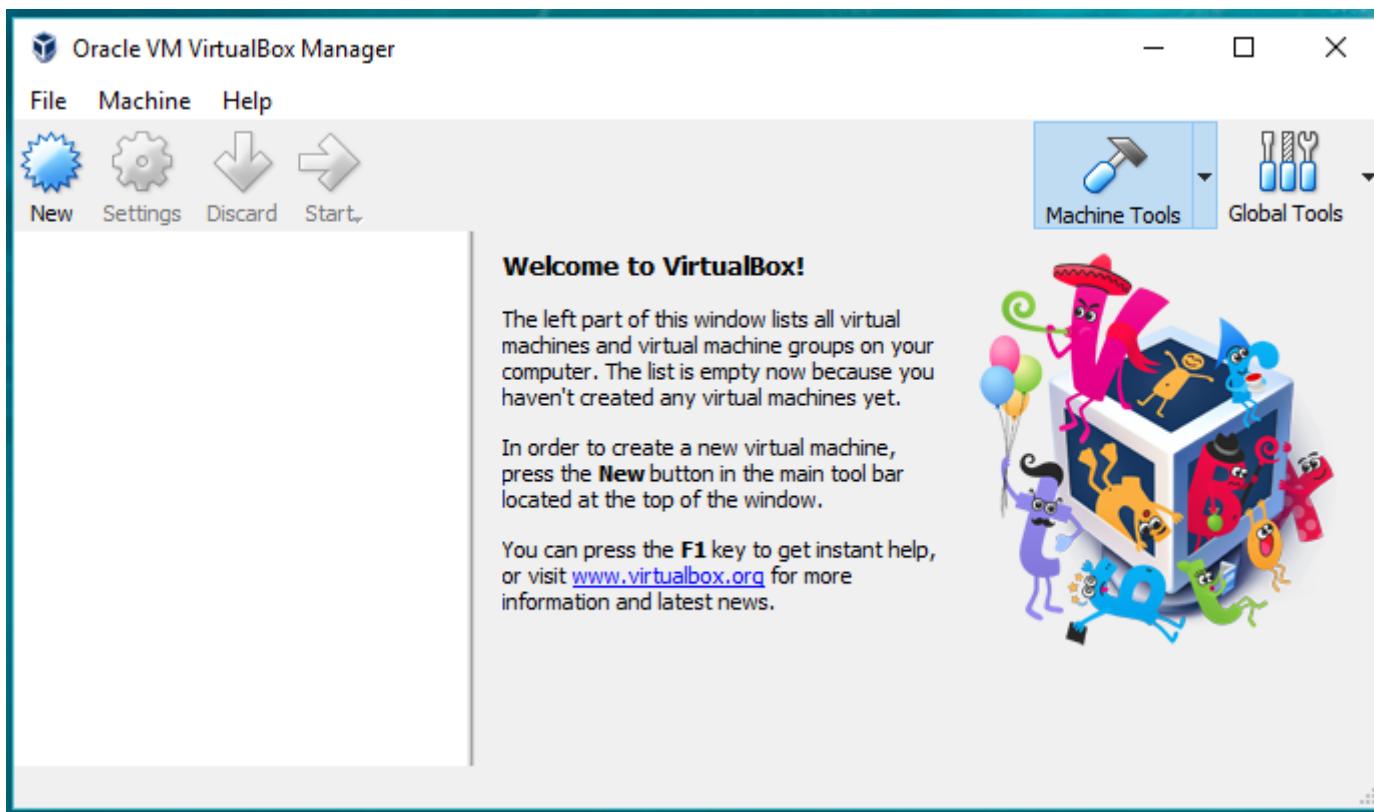
การติดตั้ง Virtual Box



การติดตั้ง Virtual Box



การติดตั้ง Virtual Box



การติดตั้ง

DOCKER TOOLBOX

การติดตั้ง Docker Toolbox

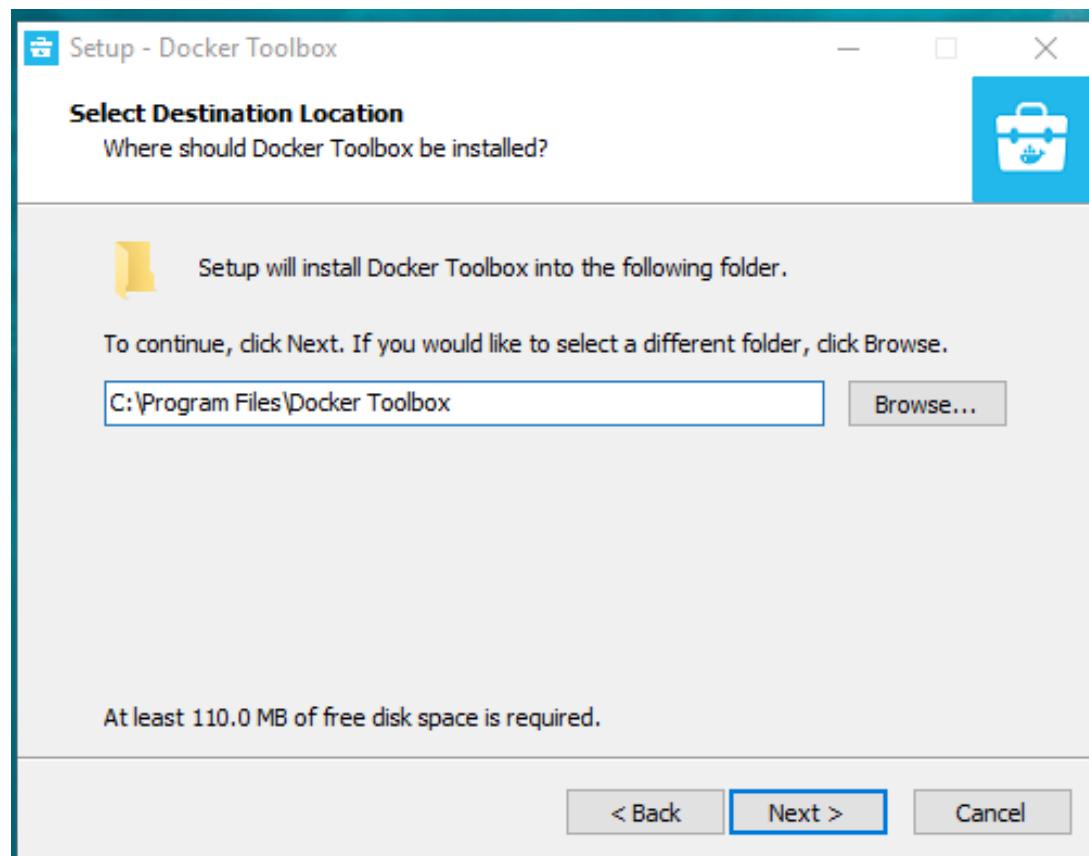
- <https://github.com/docker/toolbox/releases>

The screenshot shows the Docker documentation homepage. On the left, there's a sidebar with navigation links for Docker editions (CE, Cloud, Linux, MacOS, Microsoft Windows), Docker CE Edge releases, Docker Toolbox (legacy), and a Toolbox overview section which is currently selected. The main content area has a heading 'Ready to get started?'. It contains two numbered steps: 1. Get the latest Toolbox installer for your platform, with buttons for 'Toolbox for Mac' and 'Toolbox for Windows'. Step 2. Choose the install instructions for your platform, and follow the steps: 'Install Docker Toolbox on macOS' and 'Install Docker Toolbox for Windows'. Below this, there's a 'Next steps' section with a bulleted list: 'Try the [Get started](#) tutorial.', 'Dig in deeper with [more tutorials and examples](#) on building images, running containers, networking, managing data, and storing images on Docker Hub.', 'Learn about [Kitematic](#)', and 'Learn about [Docker Machine](#)'. On the right side of the page, there are several sidebar links: 'Edit this page', 'Request docs changes', 'Get support' (with a toggle switch), 'On this page: What's in the box', 'Ready to get started?', and 'Next steps'.

การติดตั้ง Docker Toolbox

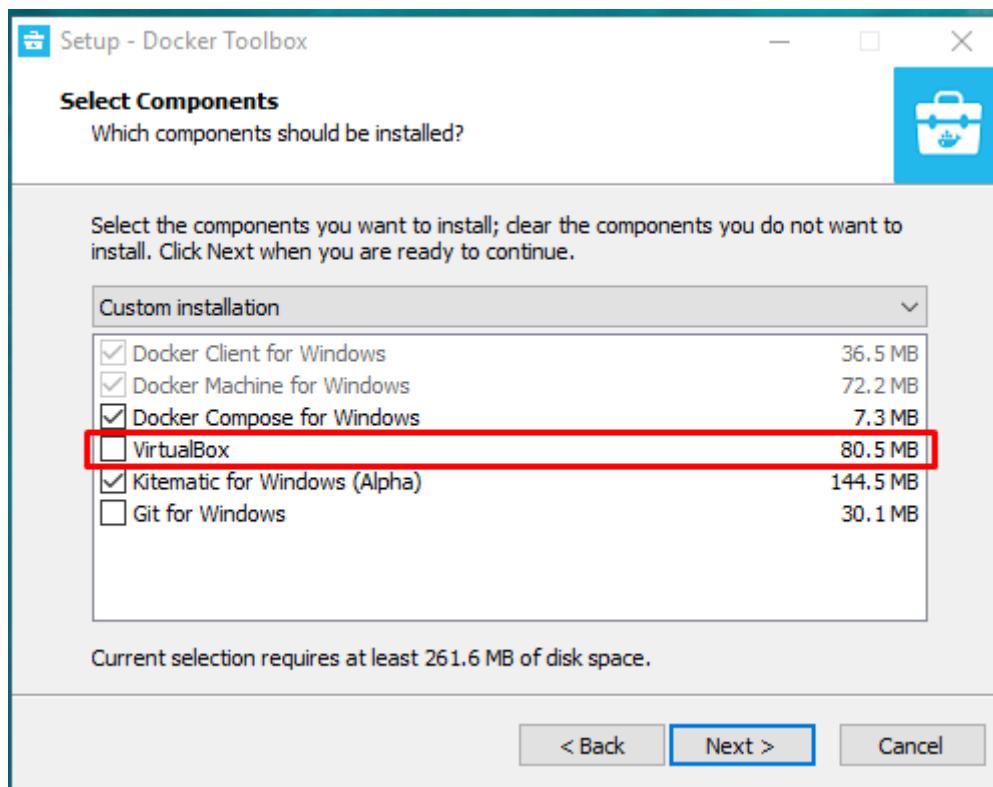


การติดตั้ง Docker Toolbox

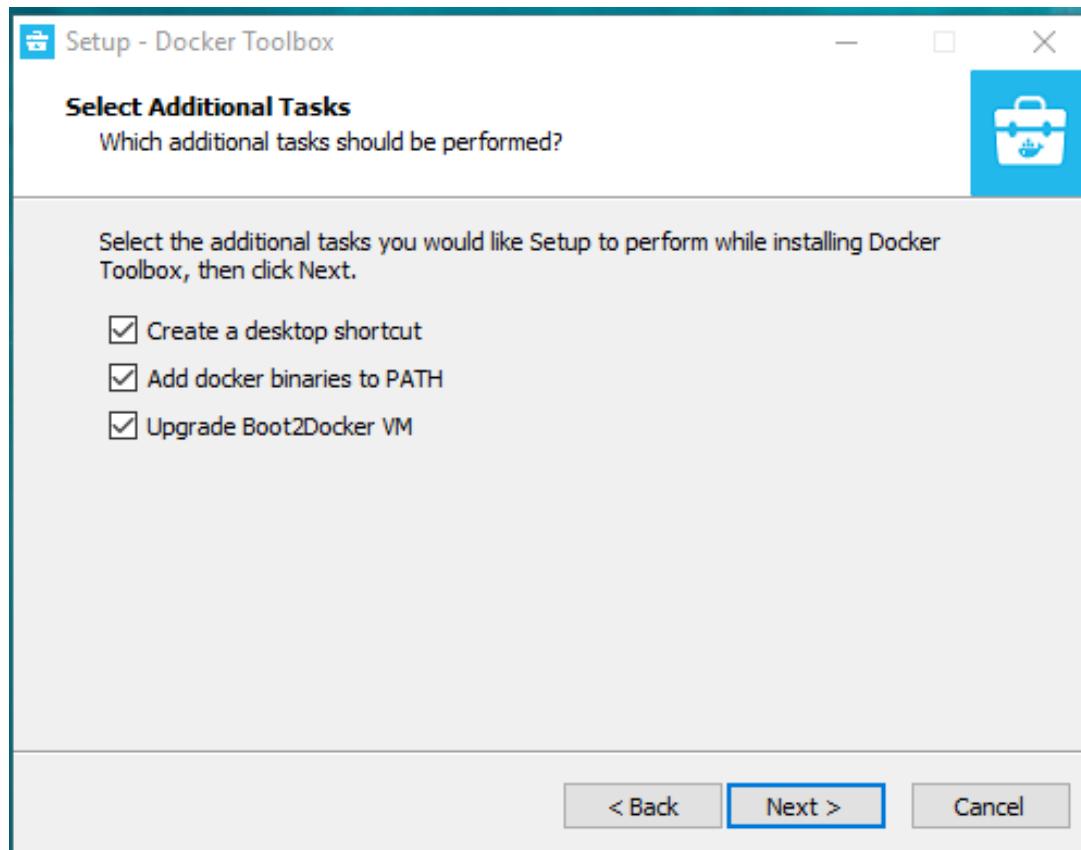


การติดตั้ง Docker Toolbox

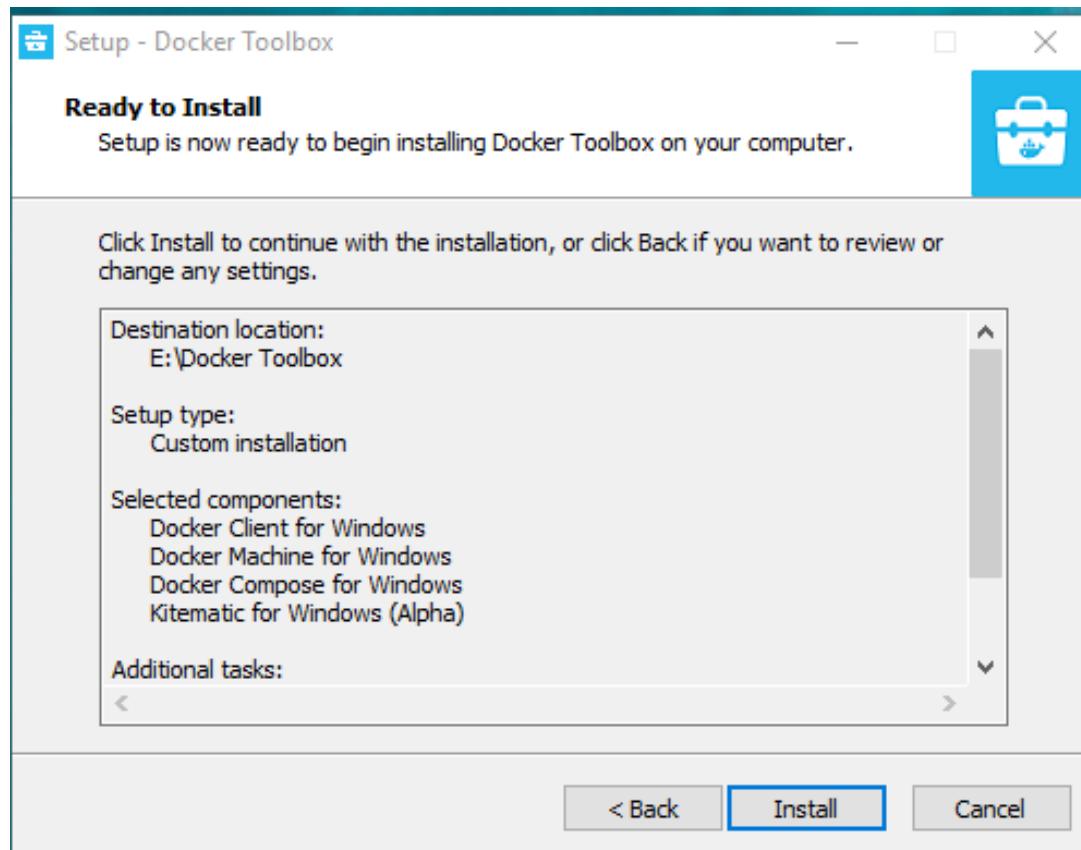
- เอาเครื่องหมายถูกหน้า VirtualBox ออก



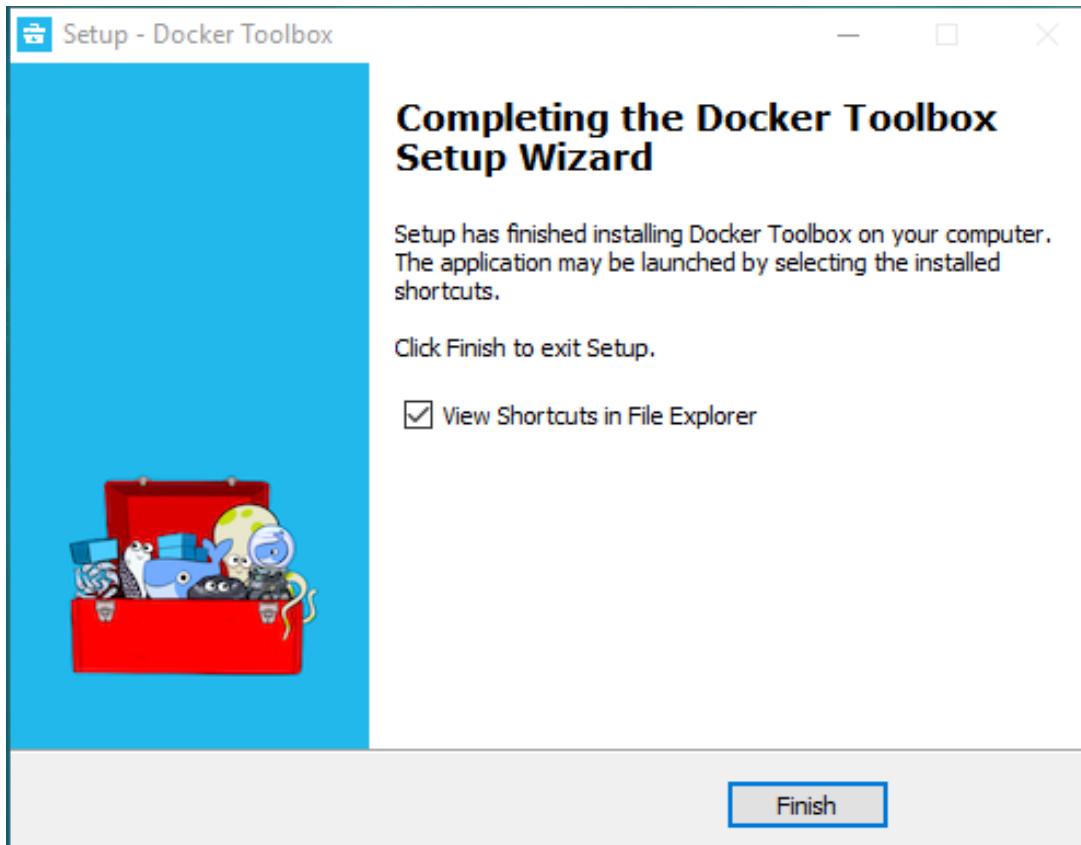
การติดตั้ง Docker Toolbox



การติดตั้ง Docker Toolbox

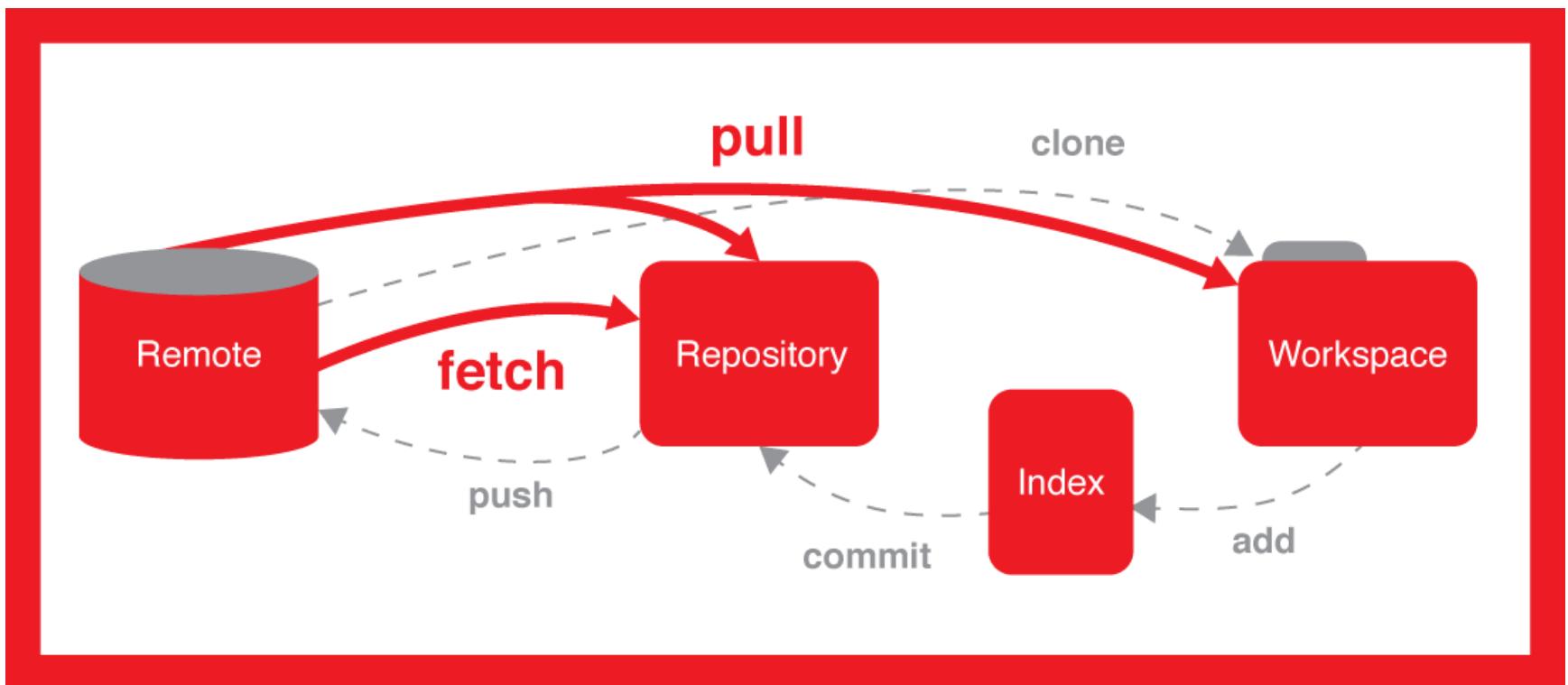


การติดตั้ง Docker Toolbox



Software version control with
GIT

Overview



CONFIGURE TOOLING

Sets the name you want attached to your commit transactions

- git config --global user.name "[name]"
- git config --global user.name "XXX"

Sets the email you want attached to your commit transactions

- git config --global user.email "[email address]"
- git config --global user.email "XXXX@hotmail.com"

Enables helpful colorization of command line output

- git config --global color.ui auto

CREATE REPOSITORIES

Creates a new local repository with the specified name

- git init [project-name]
- git init XXX

Downloads a project and its entire version history

- git clone [url]
- git clone <https://github.com/Sommaik/XXX.git>

MAKE CHANGES

Lists all new or modified files to be committed

- git status

Shows file differences not yet staged

- git diff

Snapshots the file in preparation for versioning

- git add [file]
- git add readme.txt
- git add .

Shows file differences between staging and the last file version

- git diff --staged

MAKE CHANGES

Unstages the file, but preserve its contents

- git reset [file]
- git reset readme.txt

Records file snapshots permanently in version history

- git commit -m "[descriptive message]"
- git commit -m "Initial Project"

GROUP CHANGES

Lists all local branches in the current repository

- git branch

Creates a new branch

- git branch [branch-name]
- git branch XXX

Switches to the specified branch and updates the working directory

- git checkout [branch-name]
- git checkout XXX

GROUP CHANGES

Combines the specified branch's history into the current branch

- git merge [branch]
- git merge XXX

Deletes the specified branch

- git branch -d [branch-name]
- git branch -d XXX

REFACTOR FILENAMES

Deletes the file from the working directory and stages the deletion

- `git rm [file]`
- `git rm XXX.txt`

Removes the file from version control but preserves the file locally

- `git rm --cached [file]`
- `git rm --cached XXX.txt`

Changes the file name and prepares it for commit

- `git mv [file-original] [file-renamed]`
- `git mv XXX.txt YYY.txt`

SUPPRESS TRACKING

A text file named .gitignore suppresses accidental versioning of files and paths matching the specified patterns

- *.log
- build/
- temp-*

Lists all ignored files in this project

- git ls-files --other --ignored --exclude-standard

REVIEW HISTORY

Lists version history for the current branch

- git log

Lists version history for a file, including renames

- git log --follow [file]
- git log --follow XXX.txt

Shows content differences between two branches

- git diff [first-branch]...[second-branch]

Outputs metadata and content changes of the specified commit

- git show [commit]
- git show XXX

REDO COMMITS

Undoes all commits after [commit], preserving changes locally

- git reset [commit]
- git reset XXX

Discards all history and changes back to the specified commit

- git reset --hard [commit]
- git reset --hard XXX

SYNCHRONIZE CHANGES

Uploads all local branch commits to Git

- git push [alias] [branch]
- git push origin master

Downloads bookmark history and incorporates changes

- git pull

Containers virtualize with

DOCKER

Hello World Docker

- เปิด cmd แล้วรันคำสั่งดังนี้

```
docker run --name some-nginx \
-p 80:80 \
-d nginx
```

*** link สำหรับหา image <https://hub.docker.com>

Docker parameter

Run in the background

- -d

Create name to container is running

- --name

Port mapping

- -p (local_port:container_port)

Container host name

- -h

Docker parameter

Environment

- -e

Map volume paths

- -v

Keep STDIN open even if not attached

- -i

Allocate a pseudo-TTY

- -t

Docker Command

Login

- docker login
- docker login -u <user_name>
- docker login -u <user_name> -p <password>

Logout

- docker logout

List all image

- docker images
- docker image ls

Docker Command

Search image

- docker search <image name>

Pull image

- docker pull <image name>

Create container from image

- docker create <options> <image name>
 - --name
 - -v
 - -p
- docker create --name ubuntu14 -v /user/sommaik:/home ubuntu:14.04

Docker Command

Start Container

- docker start <container_id> or <container_name>

Stop Container

- docker stop <container_id> or <container_name>

Stop all container

- docker stop \$(docker ps -a -q)

List all container

- docker ps <options>

Docker Command

Pause Container

- docker pause <container_id> or <container_name>

Unpause Container

- docker unpause <container_id> or <container_name>

Exec Container

- docker exec -it <container_id> bash

Inspect Container

- docker inspect <container_id>

Docker Command

Logs container

- docker logs

Commit Container

- docker commit <container_id> <new_image_name>
- docker commit 2x5t aloha

Push Image

- docker push <account>/<image name>

Tag

- docker tag ubuntu ubuntu-x

Docker Command

Export container

- docker export <container_id> > <to_path>

Import container

- docker import - <from_path>

Save Image

- docker save <image name> > <to path>
- docker save <image name>:<tag> > <to path>

Load Image

- docker load < <from path>

Docker Command

Remove container

- docker rm <container_id>

Remove all stop container

- docker rm \$(docker ps -a -q)

Remove Image

- docker rmi <image_id>

Docker Network

- docker network ls
- docker network create <network_name> default bridge
- docker network create --subnet 10.0.0.1/24 <network_name>
- docker network inspect <network_name> or <container_id>
- docker network create my-net (create images networks)
- docker run --network <network_name> <image_name>
- docker run -it --name <container_name> --net--alias alias2 --network <network_name> <image_name>

Dockerfile

FROM

- FROM <image>[:<tag>]
- FROM ubuntu:14.04

RUN

- RUN <command>
- RUN echo “Hello World”

EXPOSE

- EXPOSE <port>
- EXPOSE 8080

COPY

- COPY <local_path> <container_path>
- COPY ./tomcat/context.xml /usr/local/tomcat/conf

Dockerfile

ENV

- ENV <key> <value>

CMD

- CMD command param1 param2

WORKDIR

- WORKDIR /path/to/workdir

VOLUME

- VOLUME /path

Dockerfile

Build

- docker build <option> <path>
 - -t tag name

Example

- docker build -t first .

Compose file

file name

- docker-compose.yaml

Example

```
version: '3'
services:
  jenkins:
    container_name: jenkins
    image: jenkins
    volumes:
      - ./jenkins:/var/jenkins_home
    ports:
      - 8080:8080
      - 5000:5000
  ubuntu:
    container_name: ubuntu14
    image: "ubuntu:14.04"
```

Docker Compose

- docker-compose up -d
- docker-compose up --force-recreate
- docker-compose ps <service>
- docker-compose stop <service>
- docker-compose rm <service>

remote manage docker with

DOCKER MACHINE

Create new machine

- เปิด cmd และรันคำสั่งดังนี้

```
docker-machine create --driver hyperv vm1
```

*** link สำหรับหา driver <https://docs.docker.com/machine/drivers/>

Docker Machine Command

list machine

- docker-machine ls

remote

- docker-machine ssh <<your machine>>
- docker-machine ssh vm1

get environment

- docker-machine env <<your machine>>
- docker-machine env vm1

Docker machine command

set docker env

- eval \$(docker-machine env <>your machine>>)
- eval \$(docker-machine env vm1)

check active

- docker-machine active

reset env

- docker-machine env --unset
- eval \$(docker-machine env --unset)

Docker machine command

view config

- docker-machine config <<your machine>>
- docker-machine config vm1

inspect

- docker-machine inspect <<your machine>>
- docker-machine inspect vm1

get ip address

- docker-machine ip <<your machine>>
- docker-machine ip vm1

Docker machine command

force stop machine

- docker-machine kill <<your machine>>
- docker-machine kill vm1

restart machine

- docker-machine restart <<your machine>>
- docker-machine restart vm1

remove machine

- docker-machine rm <<your machine>>
- docker-machine rm vm1

Docker machine command

copy file

- docker-machine scp <<source>> <<target>>
- docker-machine scp index.html vm1:/home/docker/
- ****format** machinename:/path**

start server

- docker-machine start <<your machine>>

check status

- docker-machine status <<your machine>>

stop server

- docker-machine stop <<your machine>>

Docker machine command

upgrade docker version

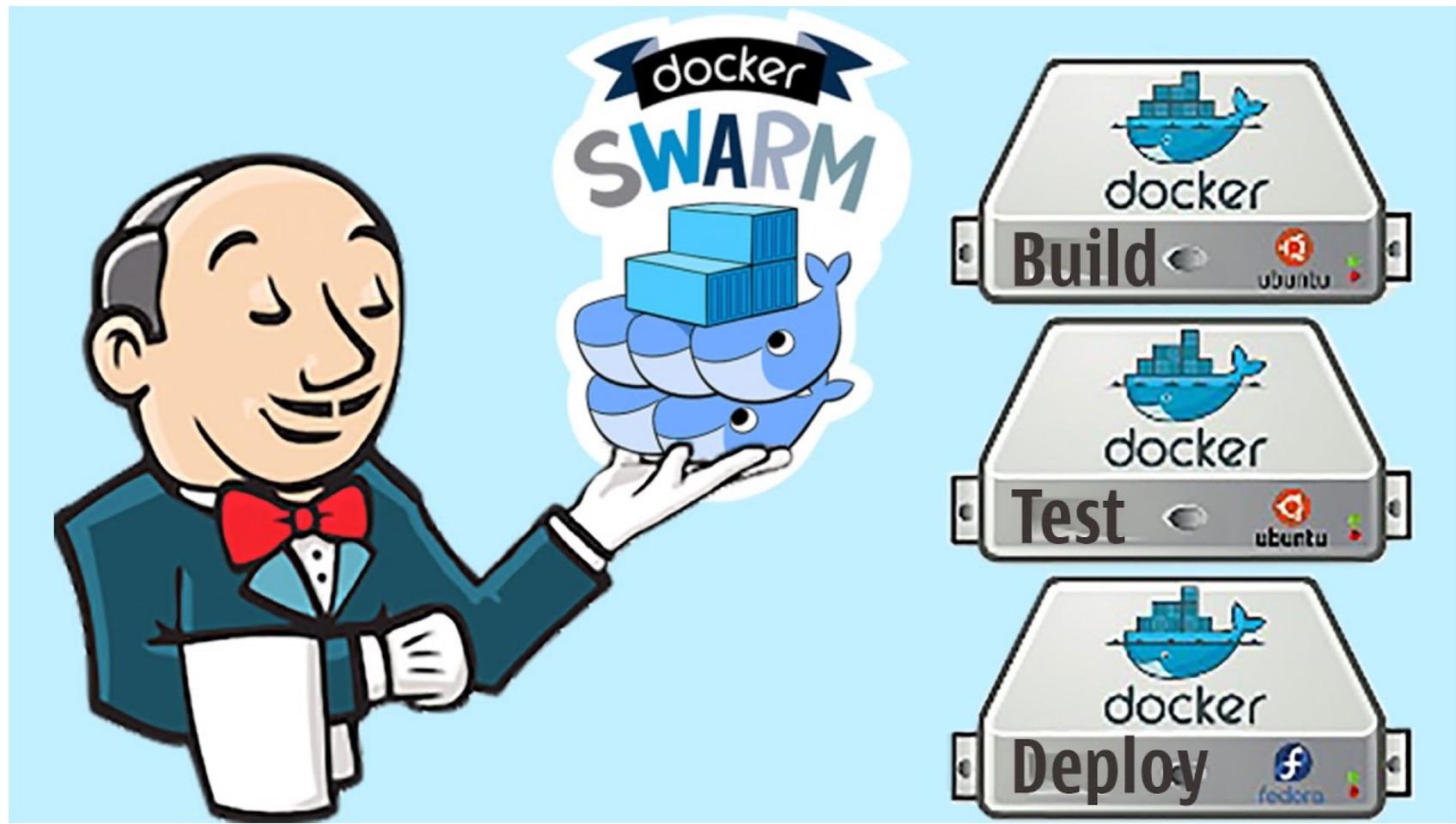
- docker-machine upgrade <<your machine>>

get machine url

- docker-machine url <<your machine>>

ORCHESTRATION WITH
DOCKER SWARM

Jenkins With docker swarm



Docker Swarm command

- docker swarm init : สำหรับเริ่มต้น docker swarm mode (manager)
- docker swarm join : สำหรับให้เครื่อง worker join เข้า manager
- docker swarm join-token : สำหรับสร้าง key เพื่อให้เข้ามา join
- docker service create : สร้าง service เพื่อให้บริการ
- docker service inspect : ตรวจสอบ service
- docker service ls : ดู service ทั้งหมด
- docker service rm : ลบ service
- docker service scale : เพิ่ม / ลด จำนวน node ที่ รัน service
- docker service ps : ดูสถานะ service
- docker service update : ปรับปรุง service
- docker node ls : ดู node ทั้งหมด

เริ่มสร้าง docker swarm (manager)

- run คำสั่ง ใน terminal ดังนี้

```
docker swarm init --advertise-addr=192.168.99.100
```

```
[vagrant@mgr1:~]$ docker swarm init --advertise-addr=192.168.33.12
Swarm initialized: current node (izupnb1ra7cv2a9ur4k95ve) is now a manager.
```

To add a worker to this swarm, run the following command:

```
docker swarm join --token SWMTKN-1-3dip03ya9iixhhsk1aegfs3ptd30pwuzpk7413i0okg6nca99-egmkmkx8hvhh8fok2d8zfppmx 192.168.33.12:2377
```

To add a manager to this swarm, run 'docker swarm join-token manager' and follow the instructions.



หลังจากที่รันคำสั่งเสร็จจะได้ผลลัพธ์เป็น เลข token เพื่อเอาไป run ที่เครื่อง worker ต่อไป

Join เข้ากลุ่ม swarm

- หลังจากที่ได้สร้างเครื่อง manager ไปแล้ว ก็ต้องนำคำสั่งที่ได้จาก console ของเครื่อง manager มารันในเครื่อง worker ดังตัวอย่าง

```
vagrant@node1:~$ docker swarm join --token SWMTKN-1-3dip03ya9iixhhsk1aegfs3ptd30pwvuzpk7413iookg6nca99-egmkmkx8hvhh8fok2d8zfppmx 192.168.33.12:2377
This node joined a swarm as a worker.
```

ตรวจสอบจำนวน node ทั้งหมด

- รุกคำสั่ง ใน terminal ดังนี้

docker node list

ID	HOSTNAME	STATUS	AVAILABILITY	MANAGER STATUS	ENGINE VERSION
izupnbin1ra7cv2a9ur4k95ve *	mgr1	Ready	Active	Leader	18.03.1-ce
mdi3i3cjpknx8d58ujrfc9ujz	node1	Ready	Active		18.03.1-ce

สร้าง service บนเครื่อง manager

- run คำสั่ง ใน terminal ดังนี้

docker service create <>option>> nginx

```
[root@centos ~]# docker service create nginx
99nj4lwci1spqifd7a2iwn7qm
overall progress: 1 out of 1 tasks
1/1: running [=====>]
verify: Service converged
```

** option ที่ควรมี

--replicas 2

--name web_server

--constraint "node.role != manager"

--publish 8080:80

replicas คือ จำนวน node ที่ต้องการสร้าง

name ชื่อที่เอาไว้อ้างอิง

constraint เงื่อนไขในการเลือกเครื่อง ©2003 PnP Solution Co., Ltd.

ดู service กันหมด

- run คำสั่ง ใน terminal ดังนี้

docker service ls

```
verity: Service converged
[vagrant@mgr1:~$ docker service ls
ID                  NAME                MODE            REPLICAS        IMAGE
99nj4lwci1sp       competent_chatterjee   replicated      1/1             nginx:latest
PORTS
```

เพิ่มลดจำนวน node

- run คำสั่ง ใน terminal ดังนี้

docker service scale <service_name>=2

```
vagrant@engr1:~$ docker service scale competent_chatterjee=2
competent_chatterjee scaled to 2
overall progress: 2 out of 2 tasks
1/2: running  [=====>]
2/2: running  [=====>]
verify: Service converged
```

ตรวจสอบ service

- รุกคำสั่ง ใน terminal ดังนี้

docker service ps <service_name>

No such service: competent_chatterjee.1							
ID	NAME	IMAGE	NODE	DESIRED STATE	CURRENT STATE	ERROR	PORTS
txebpk419vzb	competent_chatterjee.1	nginx:latest	mgr1	Running	Running 6 minutes ago		
q0u7gd9qkvo	competent_chatterjee.2	nginx:latest	node1	Running	Running 37 seconds ago		

จะแสดงข้อมูลของ service ว่าทำงานอยู่บน node ไหนตามจำนวน replicas ที่ได้ตั้งเอาไว้

update service

- runคำสั่ง ใน terminal ดังนี้

docker service update <service_name> --image nginx:alpine

```
vagrant@mgr1:~$ docker service update competent_chatterjee --image nginx:alpine
competent_chatterjee
overall progress: 2 out of 2 tasks
1/2: running  [=====>]
2/2: running  [=====>]
verify: Service converged
```

ลบ service

- รุกคำสั่ง ใน terminal ดังนี้

docker service rm <service_name>

```
vagrant@engr1:~$ docker service rm competent_chatterjee
competent_chatterjee
```

Automate Build and Deploy with

JENKINS

Unlock Jenkins

Getting Started

Unlock Jenkins

To ensure Jenkins is securely set up by the administrator, a password has been written to the log ([not sure where to find it?](#)) and this file on the server:

`/var/jenkins_home/secrets/initialAdminPassword`

Please copy the password from either location and paste it below.

Administrator password

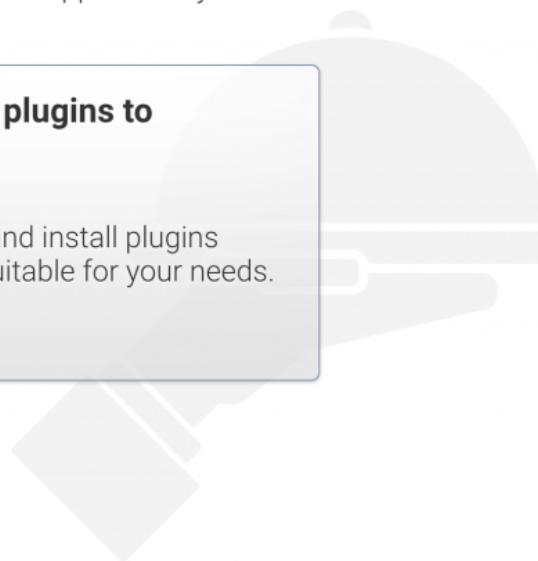
Continue

Install Plugins

Getting Started X

Customize Jenkins

Plugins extend Jenkins with additional features to support many different needs.



Install suggested plugins
Install plugins the Jenkins community finds most useful.

Select plugins to install
Select and install plugins most suitable for your needs.

Jenkins 2.46.1

Loading Plugins

Getting Started

Getting Started

✓ Folders Plugin	✓ OWASP Markup Formatter Plugin	✓ build timeout plugin	✓ Credentials Binding Plugin	** Pipeline: Shared Groovy Libraries ** Branch API Plugin ** Pipeline: Multibranch ** Authentication Tokens API Plugin ** Docker Commons Plugin ** Durable Task Plugin ** Pipeline: Nodes and Processes ** Docker Pipeline ** Pipeline: Stage Tags Metadata ** Pipeline: Declarative Agent API ** Pipeline: Model Definition Pipeline ** GitHub API Plugin Jenkins Git plugin ** GitHub plugin ** GitHub Branch Source Plugin ** Pipeline: GitHub Groovy Libraries ** - required dependency
✓ Timestamper	✓ Workspace Cleanup Plugin	✓ Ant Plugin	✓ Gradle Plugin	
✓ Pipeline	⌚ GitHub Organization Folder Plugin	✓ Pipeline: Stage View Plugin	✓ Git plugin	
⌚ Subversion Plug-in	⌚ SSH Slaves plugin	✓ Matrix Authorization Strategy Plugin	✓ PAM Authentication plugin	
✓ LDAP Plugin	⌚ Email Extension Plugin	✓ Mailer Plugin		

Jenkins 2.46.1

©2003 PnP Solution Co., Ltd.

136

Create Admin User

Getting Started

Create First Admin User

Username:

Password:

Confirm password:

Full name:

E-mail address:

Jenkins 2.46.1

Continue as admin Save and Finish

Setup Complete

The screenshot shows a web-based Jenkins setup completion interface. At the top left, there is a "Getting Started" link. The main content area features a large, bold heading "Jenkins is ready!" followed by the subtext "Your Jenkins setup is complete." Below this, there is a blue button labeled "Start using Jenkins". At the bottom left of the main content area, the Jenkins version "Jenkins 2.46.1" is displayed. The entire interface is contained within a light gray box with a dark gray border.

Getting Started

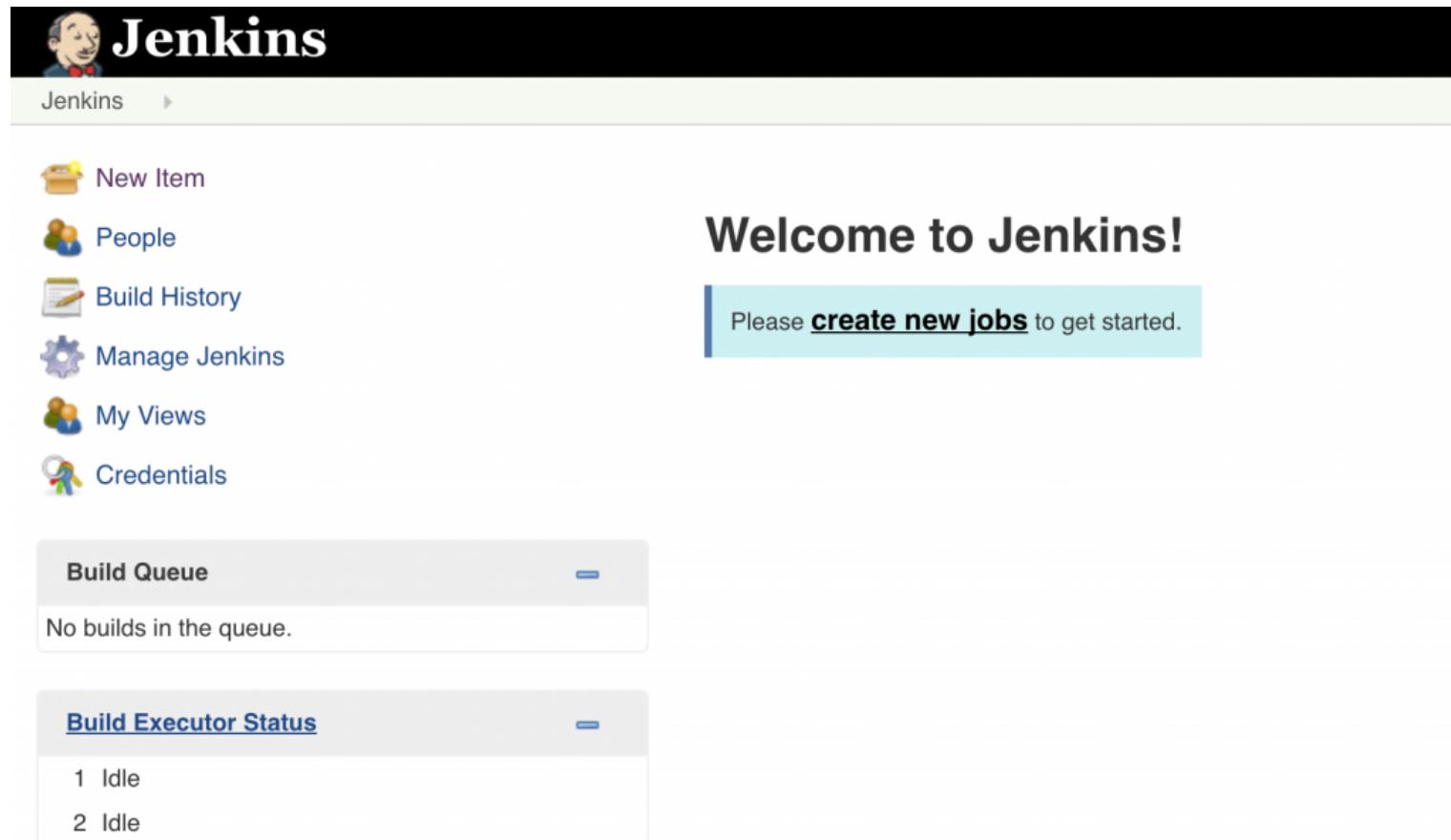
Jenkins is ready!

Your Jenkins setup is complete.

[Start using Jenkins](#)

Jenkins 2.46.1

Create New Jobs



The screenshot shows the Jenkins dashboard. At the top left is the Jenkins logo with a cartoon character. The main title "Jenkins" is displayed prominently. Below the title is a navigation bar with the word "Jenkins" and a right-pointing arrow. On the left side, there is a sidebar with the following links:

- New Item (with icon)
- People (with icon)
- Build History (with icon)
- Manage Jenkins (with icon)
- My Views (with icon)
- Credentials (with icon)

In the center, a large "Welcome to Jenkins!" message is displayed in bold. Below it, a call-to-action message says "Please [create new jobs](#) to get started." At the bottom of the dashboard, there are two sections: "Build Queue" (showing "No builds in the queue.") and "Build Executor Status" (showing "1 Idle" and "2 Idle").

Choose Pipeline

Enter an item name

» Required field



Freestyle project

This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.



Pipeline

Orchestrates long-running activities that can span multiple build slaves. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.



External Job

This type of job allows you to record the execution of a process run outside Jenkins, even on a remote machine. This is designed so that you can use Jenkins as a dashboard of your existing automation system.



Multi-configuration project

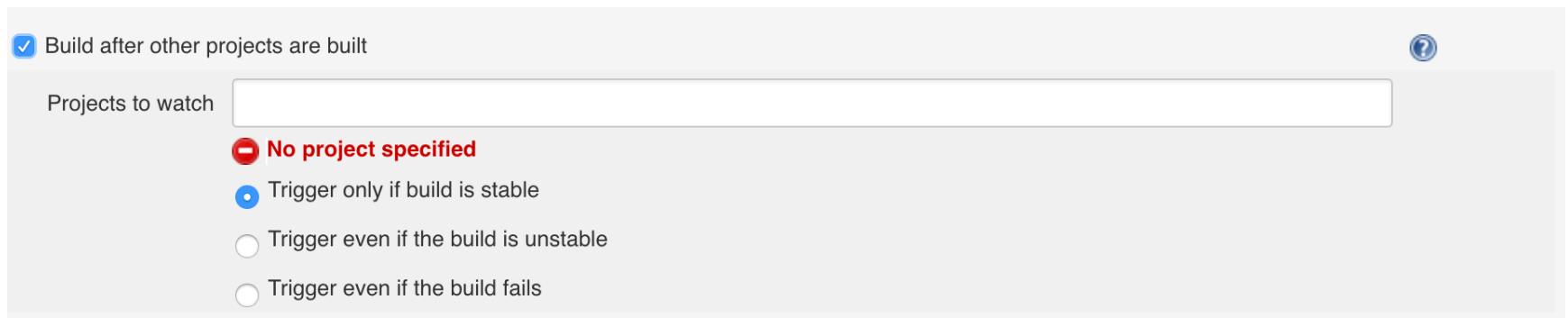
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

Jenkins Config Build Trigger

Build Triggers

- Build after other projects are built 
- Build periodically 
- GitHub hook trigger for GITScm polling 
- Poll SCM 
- Disable this project 
- Quiet period 
- Trigger builds remotely (e.g., from scripts) 

Build after other projects are built



The screenshot shows a configuration panel for a build step. At the top left is a checked checkbox labeled "Build after other projects are built". To its right is a blue circular icon with a question mark. Below this is a section titled "Projects to watch" containing an empty input field and a red error message "No project specified". Underneath are three radio button options: "Trigger only if build is stable" (selected), "Trigger even if the build is unstable", and "Trigger even if the build fails".

สั่งให้ทำงานอัตโนมัติ หลังจากที่ project กำหนดทำงานเสร็จ โดยมี 3 option คือ
ทำงานเมื่อ build สำเร็จ ไม่มี warning
ทำงานเมื่อ build สำเร็จ แบบมี warning
ทำงานไม่สำเร็จ

Build periodically

Build periodically ?

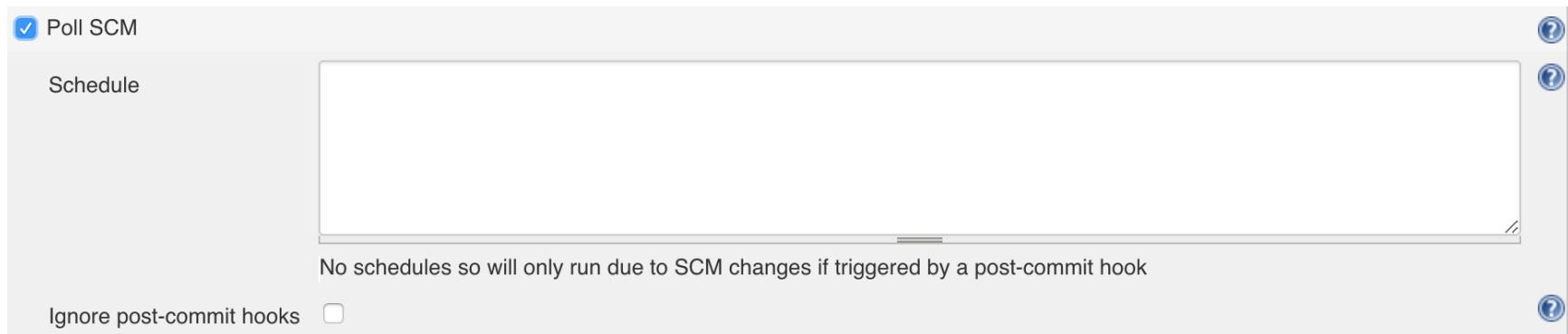
Schedule ?

Would last have run at Wednesday, June 27, 2018 9:43:05 PM ICT; would next run at Wednesday, June 27, 2018 9:55:05 PM ICT.

This field follows the syntax of cron (with minor differences). Specifically, each line consists of 5 fields separated by TAB or whitespace:
MINUTE HOUR DOM MONTH DOW
MINUTE Minutes within the hour (0–59)
HOUR The hour of the day (0–23)
DOM The day of the month (1–31)
MONTH The month (1–12)
DOW The day of the week (0–7) where 0 and 7 are Sunday.

ສ້າງໃຫ້ກໍາງານວັດໂນມັຕິຕາມໜ່ວຍເວລາທີ່ກໍານົດ ໂດຍສາມາດກໍາບັນດາເວລາໄດ້ວູ້
ໃນຮູບແບບ crontab format

Poll SCM



สั่งให้ทำงานอัตโนมัติตามช่วงเวลาที่กำหนด โดยสามารถกำหนดเวลาได้อยู่ในรูปแบบ crontab format

โดยมีลักษณะการทำงานคือจะไปทำการตรวจสอบดูก่อนว่า source code มีการเปลี่ยนแปลงหรือไม่ ถ้าไม่มีเปลี่ยนแปลงก็จะไม่ทำการ build job

Trigger builds remotely

Build Triggers

 Build after other projects are built Build periodically GitHub hook trigger for GITScm polling Poll SCM Disable this project Quiet period Trigger builds remotely (e.g., from scripts)

Authentication Token

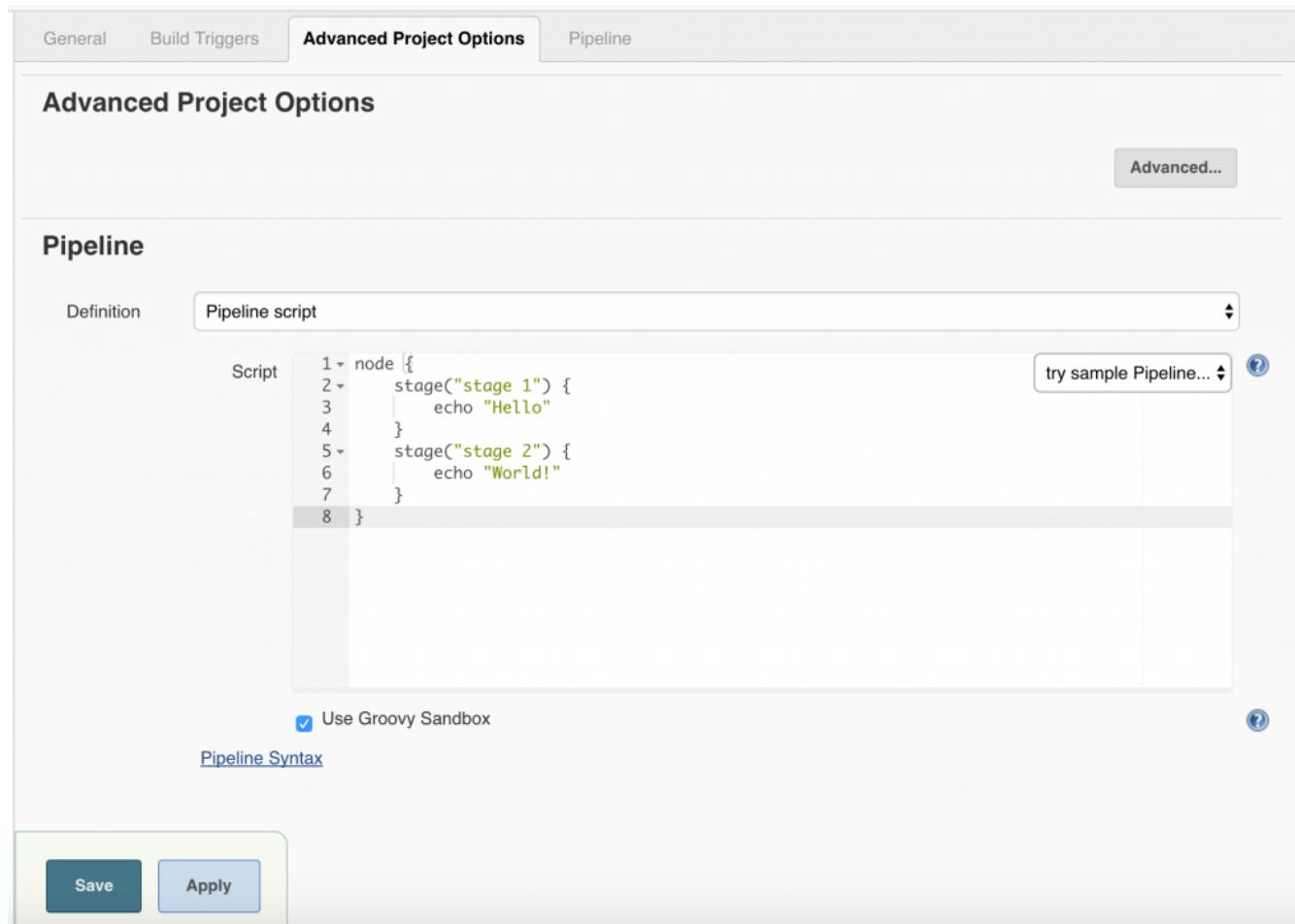
Use the following URL to trigger build remotely: JENKINS_URL/job/first/build?token=TOKEN_NAME or /buildWithParameters?token=TOKEN_NAME

Optionally append &cause=Cause+Text to provide text that will be included in the recorded build cause.

เป็นการตั้งค่าให้สามารถสั่งให้ job ทำงานโดยการเรียกผ่าน url เช่น ใส่ค่า Authentication Token เป็น **123** Jenkins run ที่เครื่อง 127.0.0.1 ก็จะเรียกผ่าน url ดังนี้

http://127.0.0.1:8080/job/{job_name}/build?token=123

Pipeline script



The screenshot shows the Jenkins interface for configuring a Pipeline script. The top navigation bar includes tabs for General, Build Triggers, Advanced Project Options (which is selected), and Pipeline. The main content area is titled "Advanced Project Options" and contains a "Pipeline" section. In the Pipeline section, the "Definition" dropdown is set to "Pipeline script". Below it, a code editor displays the following Groovy script:

```
1 ~ node //{
2 ~   stage("stage 1") {
3 ~     echo "Hello"
4 ~   }
5 ~   stage("stage 2") {
6 ~     echo "World!"
7 ~   }
8 }
```

Next to the code editor is a "try sample Pipeline..." button with a question mark icon. At the bottom of the code editor, there is a checked checkbox labeled "Use Groovy Sandbox" and a "Pipeline Syntax" link with a question mark icon. At the very bottom of the window are "Save" and "Apply" buttons.

Pipeline template

```
pipeline {
    agent any
    environment {
        APP_NAME = "test app name"
    }
    stages {
        stage('Build Image'){
            steps {
                sh "echo ${env.APP_NAME}"
            }
        }
    }
}
```

Jenkins Job List

Jenkins [ENABLE AUTO REFRESH](#)

New Item 

People 

Build History 

Project Relationship 

Check File Fingerprint 

Manage Jenkins 

My Views 

Open Blue Ocean 

Credentials 

New View 

All Checker PnP Training Zone9 wcom wisdom + 

[add description](#)

S	W	Name ↓	Last Success	Last Failure	Last Duration	Fav
		api.pnpsw.com	1 mo 0 days - #22	4 mo 25 days - #13	23 sec	 
		api.wisdomairways.com	5 days 10 hr - #25	8 days 11 hr - #20	2 min 8 sec	 
		api.zone9sport.com	4 mo 14 days - #11	5 mo 4 days - #8	45 sec	 
		birt.pnpsw.com	2 mo 17 days - #8	N/A	57 sec	 
		booking.wisdomairways.com	5 days 10 hr - #5	N/A	1 min 46 sec	 
		checker.pnpsw.com	5 mo 10 days - #9	5 mo 10 days - #8	50 sec	 

กดเครื่องหมาย + เพื่อกำการเพิ่ม tab เพื่อการกรองข้อมูลของ job

Set List View

View name

List View 
Shows the simple list format. You can choose which jobs are to be displayed in which view.

My View
This view automatically displays all the jobs that the current user has an access to.

OK

Name

Description

[Plain text] [Preview](#)

Filter build queue

Filter build executors

Job Filters

Status Filter

Recurse in subfolders

Jobs

- api.pnpsw.com
- api.wisdomairways.com
- api.zone9sport.com
- birt.pnpsw.com
- booking.wisdomairways.com
- checker.pnpsw.com
- checkerapi.pnpsw.com
- demo.pnpsw.com
- issue.pnpsw.com
- issueapi.pnpsw.com
- train-api.pnpsw.com
- www.pnpsw.com

Logs

Jenkins Kan Ouivirach | log out

ENABLE AUTO REFRESH

Back to Dashboard Status Changes Build Now Delete Pipeline Configure Full Stage View Pipeline Syntax

Recent Changes

add description

Pipeline FizzBuzz

Stage View

Average stage times:

stage 1	stage 2
132ms	145ms

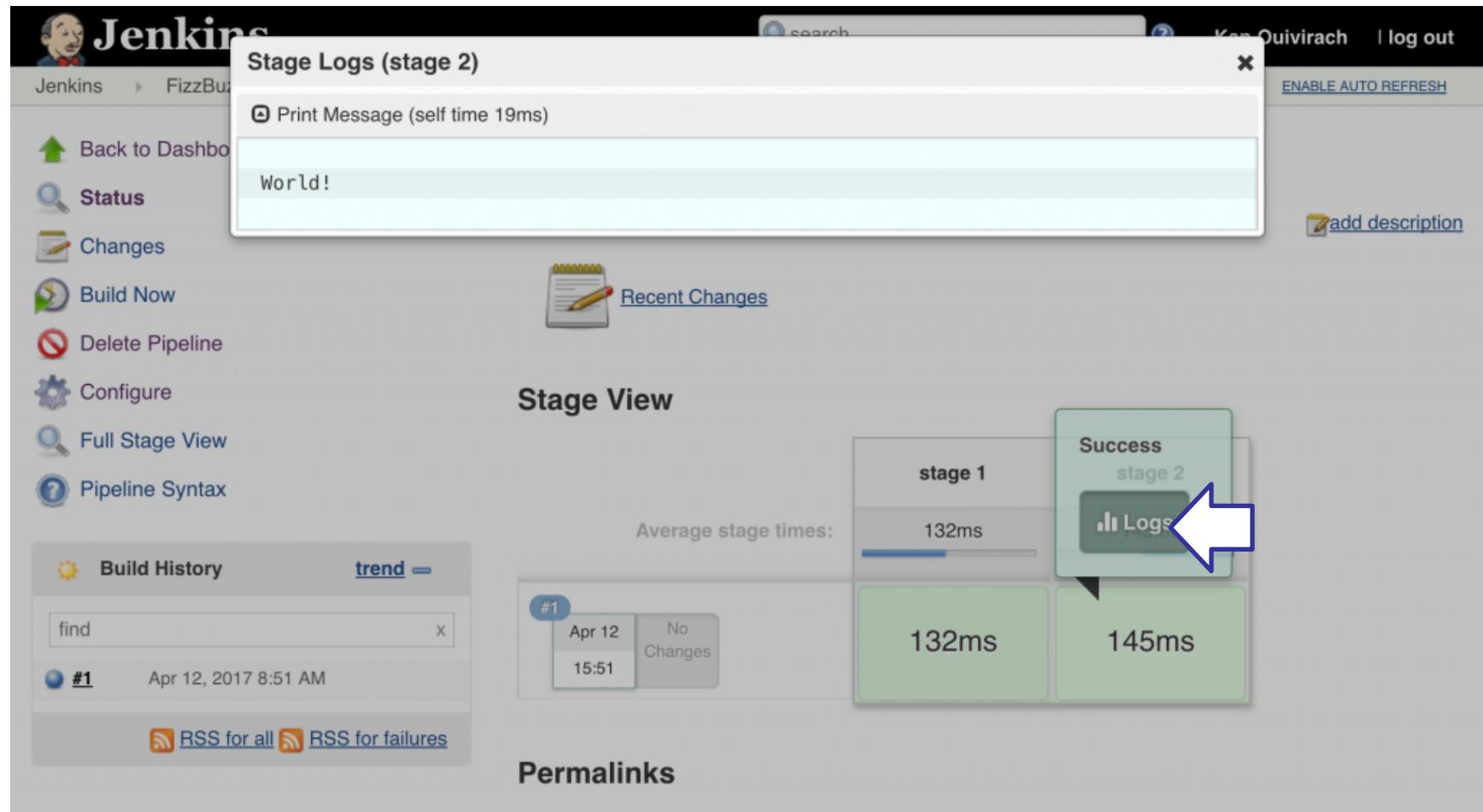
#1 Apr 12, 2017 8:51 AM No Changes 15:51 132ms 145ms

RSS for all RSS for failures

Permalinks



Logs



The screenshot shows the Jenkins interface for a pipeline named "FizzBuzz".

Stage Logs (stage 2)

- Print Message (self time 19ms)
World!

Recent Changes

Stage View

Average stage times:

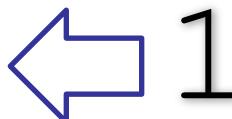
stage 1	stage 2
132ms	Success Logs
132ms	145ms

Permalinks

A blue arrow points to the "Logs" button in the Stage View section of the pipeline summary.

Add Credentials #1

-  New Item
-  People
-  Build History
-  Project Relationship
-  Check File Fingerprint
-  Manage Jenkins
-  My Views
-  Open Blue Ocean
-  **Credentials**
-  System
-  New View

Stores scoped to Jenkins

P	Store ↓	Domains
	Jenkins	 (global)



2

1

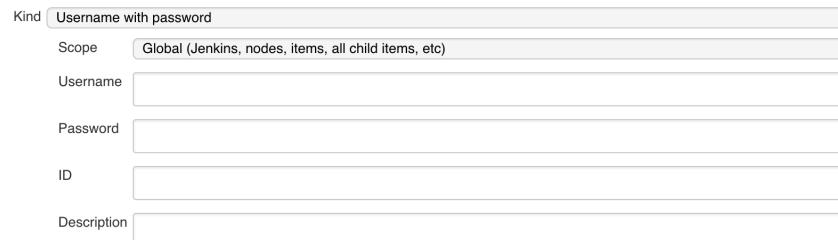
Add Credentials #2



The screenshot shows the Jenkins Global credentials configuration page. The URL in the browser is `Jenkins > Credentials > System > Global credentials (unrestricted)`. On the left, there are navigation links: "Back to credential domains" (with a blue arrow pointing to it) and "Add Credentials". The main content area is titled "Global" and shows a list of credentials. The first three entries are:

Icon	Name
	tiewy
	somi
	core

Below the table, there is a link "Icon: S M L".



A modal dialog box for adding a new credential. The "Kind" dropdown is set to "Username with password". The "Scope" field contains "Global (Jenkins, nodes, items, all child items, etc)". The form fields are as follows:

Username	<input type="text"/>
Password	<input type="password"/>
ID	<input type="text"/>
Description	<input type="text"/>

An "OK" button is at the bottom right of the dialog.

Install Plugins

The screenshot shows the Jenkins dashboard with the 'Manage Jenkins' menu open. The 'Manage Plugins' option is highlighted. On the right, the 'Updates' tab is selected in the 'Install' section, displaying a list of available Jenkins plugins:

Name	Version	Installed
Apache HttpClient 4.x API	4.5.5-2.0	4.5.3-2.0
Autofavorite for Blue Ocean	1.2.2	1.2.1
Bitbucket Branch Source	2.2.11	2.2.8
Bitbucket Pipeline for Blue Ocean	1.5.0	1.4.0
Blue Ocean	1.5.0	1.4.0
Blue Ocean Core JS	1.5.0	1.4.0
Blue Ocean Pipeline Editor	1.5.0	1.4.0
Branch API	2.0.20	2.0.18
Common API for Blue Ocean	1.5.0	1.4.0
Config API for Blue Ocean	1.5.0	1.4.0
Credentials Binding		

Add new node

The screenshot shows the Jenkins dashboard with the following interface elements:

- Left Sidebar:** Contains links like New Item, People, Build History, Project Relationship, Check File Fingerprint, Manage Jenkins (selected), My Views, Open Blue Ocean, Credentials, and New View.
- Central Area:** Shows a grid of build status indicators (S, W) for various projects. Below the grid are links: Back to Dashboard, Manage Jenkins, New Node (highlighted with a blue arrow), and Configure.
- Build Queue:** Displays "No builds in the queue."
- Build Executor Status:** Displays "1 Idle" and "2 Idle" for the master node.
- Bottom Sidebar:** Contains links for master (1 Idle, 2 Idle) and wisdom (offline). The "Manage Nodes" link is highlighted with a blue box.
- Table on the right:** Shows a list of nodes with columns for S, Name, and Architecture. One row is highlighted with a blue box and labeled "Data obtained".

Add new node

Node name

Permanent Agent
Adds a plain, permanent agent to Jenkins. This is called "permanent" provisioning. Select this type if no other agent types apply — for example Jenkins, etc.

Name	<input type="text" value="mynode"/>
Description	<input type="text"/>
# of executors	<input type="text" value="1"/>
Remote root directory	<input type="text" value="/home/user"/>
Labels	<input type="text"/>
Usage	<input type="text" value="Use this node as much as possible"/>
Launch method	<input type="text" value="Launch slave agents via SSH"/>
Host	<input type="text"/>
Credentials	<input type="text" value="root"/> <input type="button" value="Add"/>
Host Key Verification Strategy <input type="text" value="Known hosts file Verification Strategy"/>	

Availability

Node Properties

Environment variables
 Tool Locations

Add new node

- Remote root directory : ใช้ชื่อ path ของเครื่องปลายทาง
- Labels : ชื่อจะถูกเรียกใช้ใน pipeline
- Usage : Only Build job with label expression matching this node
- Launch Method : Launch slave agents via SSH
 - Host : ชื่อเครื่องหรือ ip
 - Credentials : ค่า Credentials ที่ตั้งไว้
 - Host Key Verification Strategy : None verifying Verification Strategy

Use node in pipeline

Declarative pipeline

```
pipeline {  
    agent {label 'slave'}  
    stages {  
        ...  
    }  
}
```

Scripted pipeline

```
node (label: 'slave') {  
    ...  
}
```

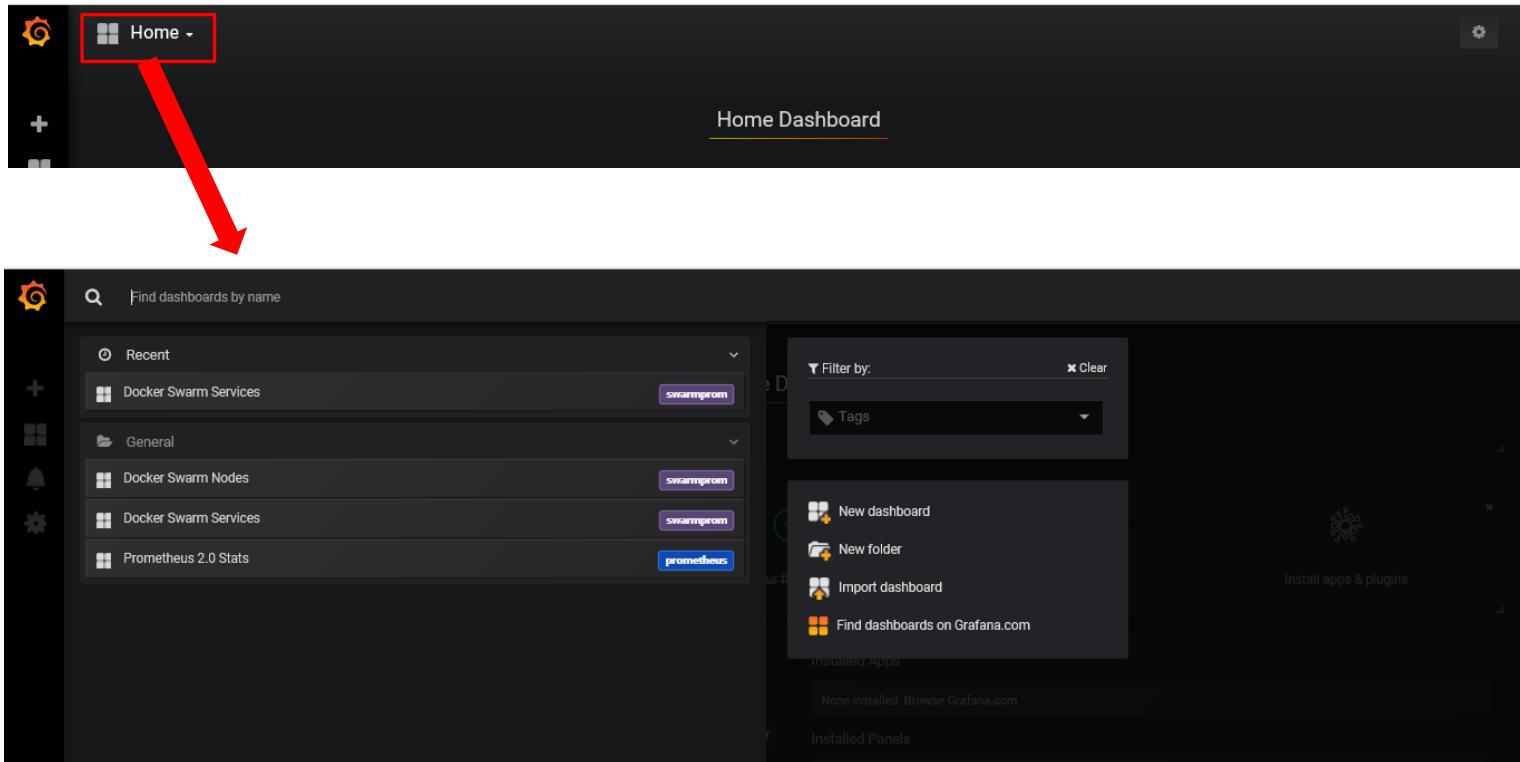
Monitoring with

GRAFANA

การใช้งาน Grafana

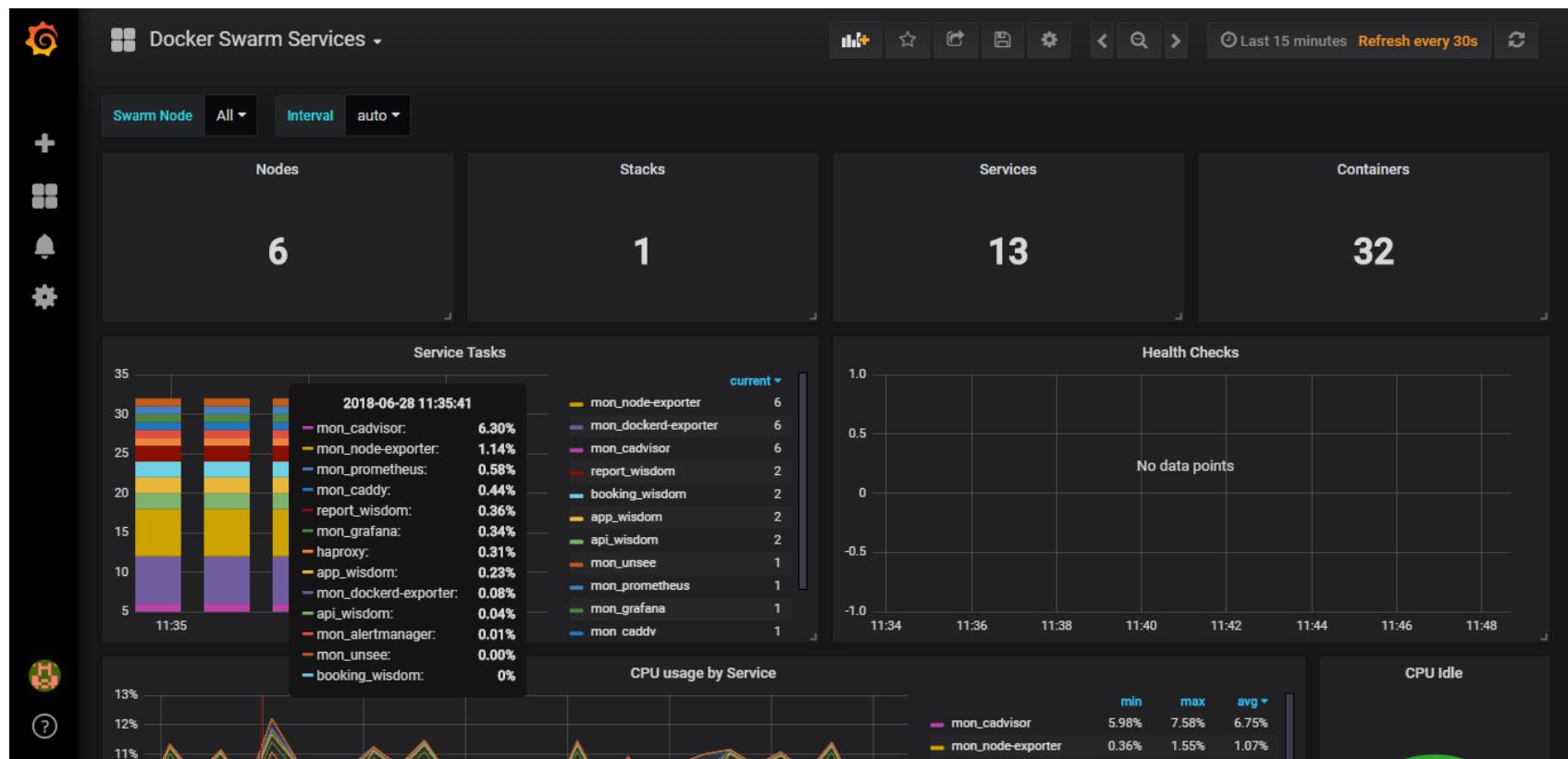


การใช้งาน Grafana

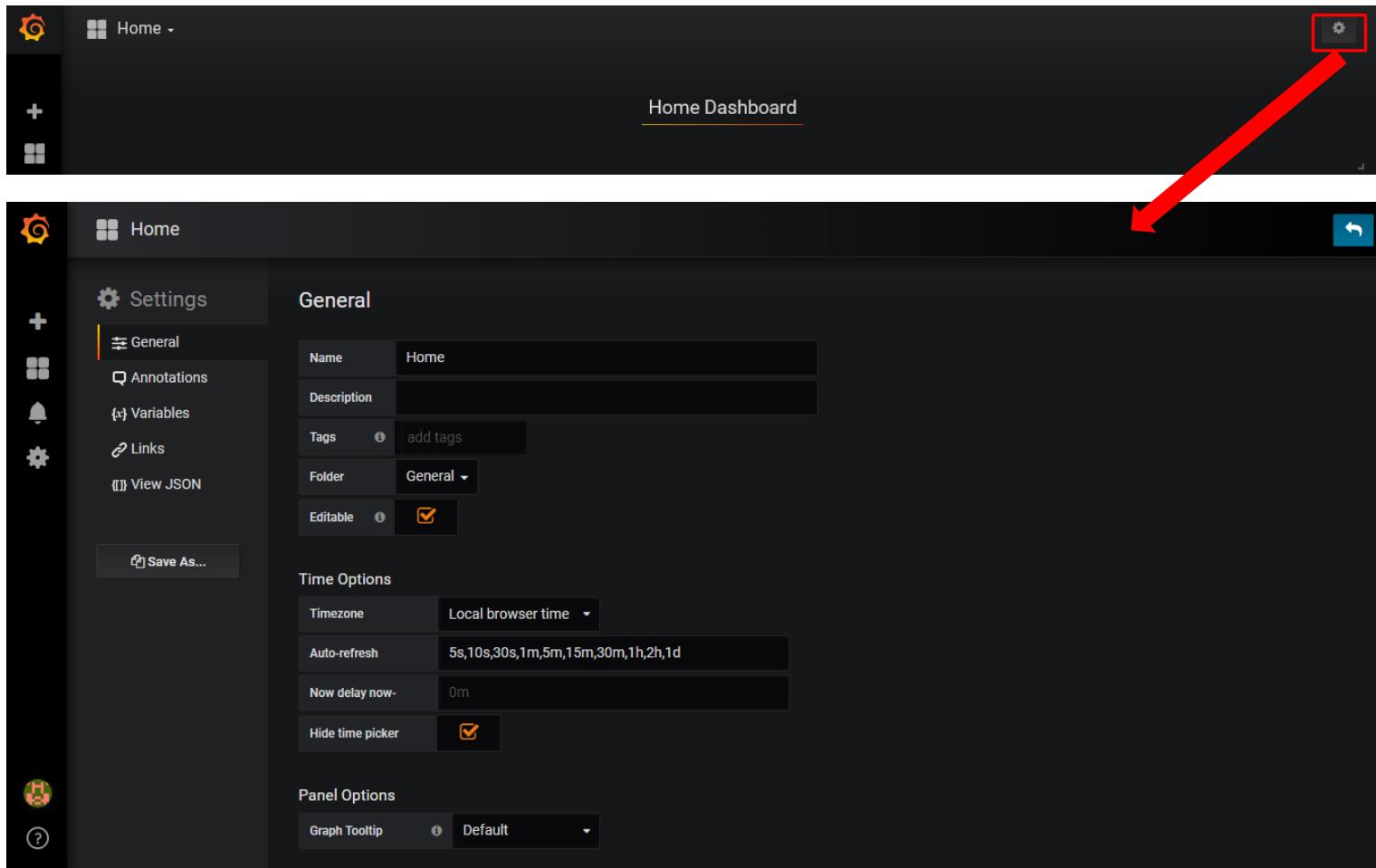


การใช้งาน Grafana

- หน้าจอแสดงรายละเอียดแต่ละ Service



การใช้งาน Grafana



The screenshot shows the Grafana interface. At the top, there's a navigation bar with icons for Home, Settings, and a gear icon. Below the navigation bar is the title "Home Dashboard". On the left side, there's a sidebar with various icons and a "Settings" tab which is currently selected. The main content area is titled "General" and contains the following fields:

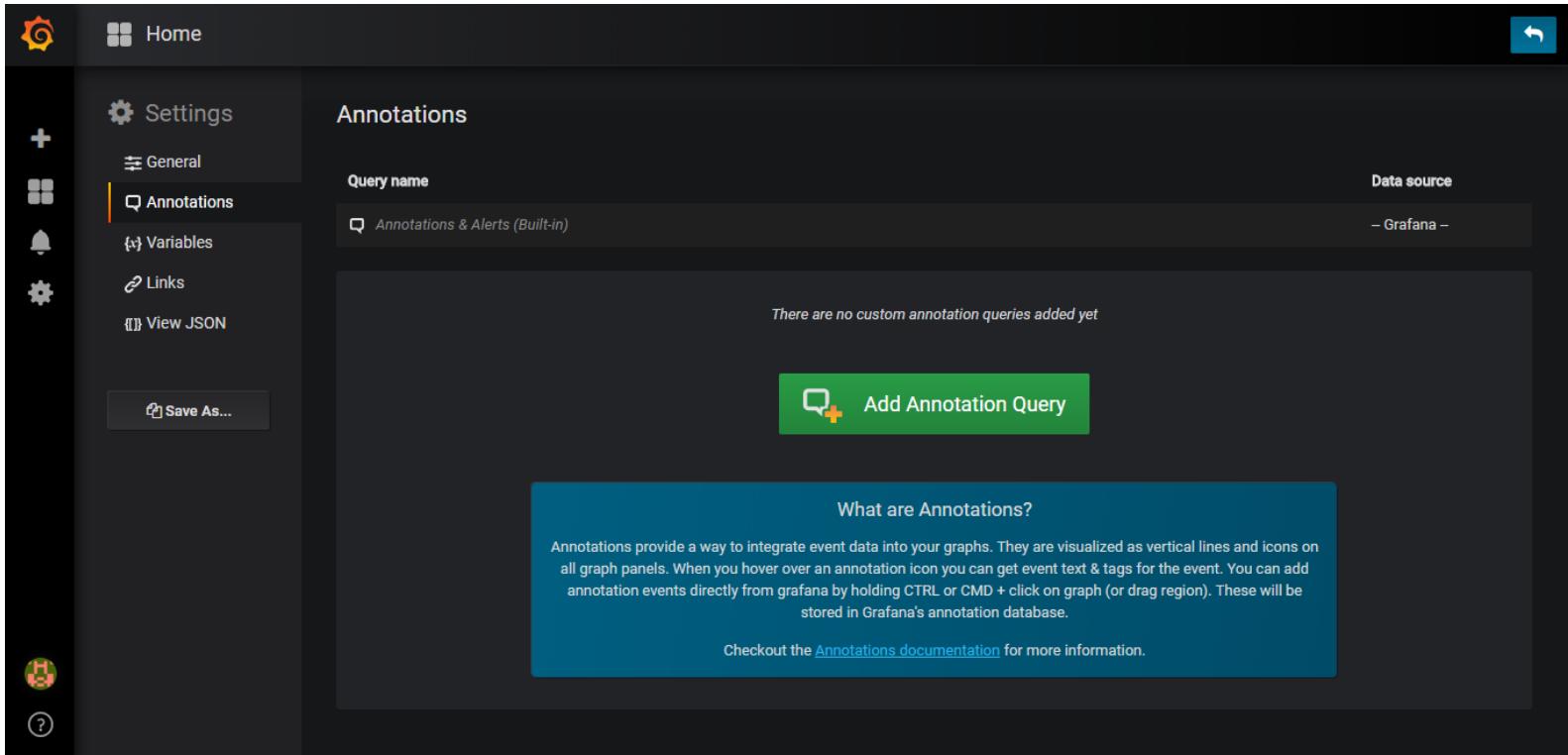
Name	Home
Description	[empty]
Tags	add tags
Folder	General
Editable	<input checked="" type="checkbox"/>

Below these fields is a section titled "Time Options" with the following settings:

Timezone	Local browser time
Auto-refresh	5s,10s,30s,1m,5m,15m,30m,1h,2h,1d
Now delay now-	0m
Hide time picker	<input checked="" type="checkbox"/>

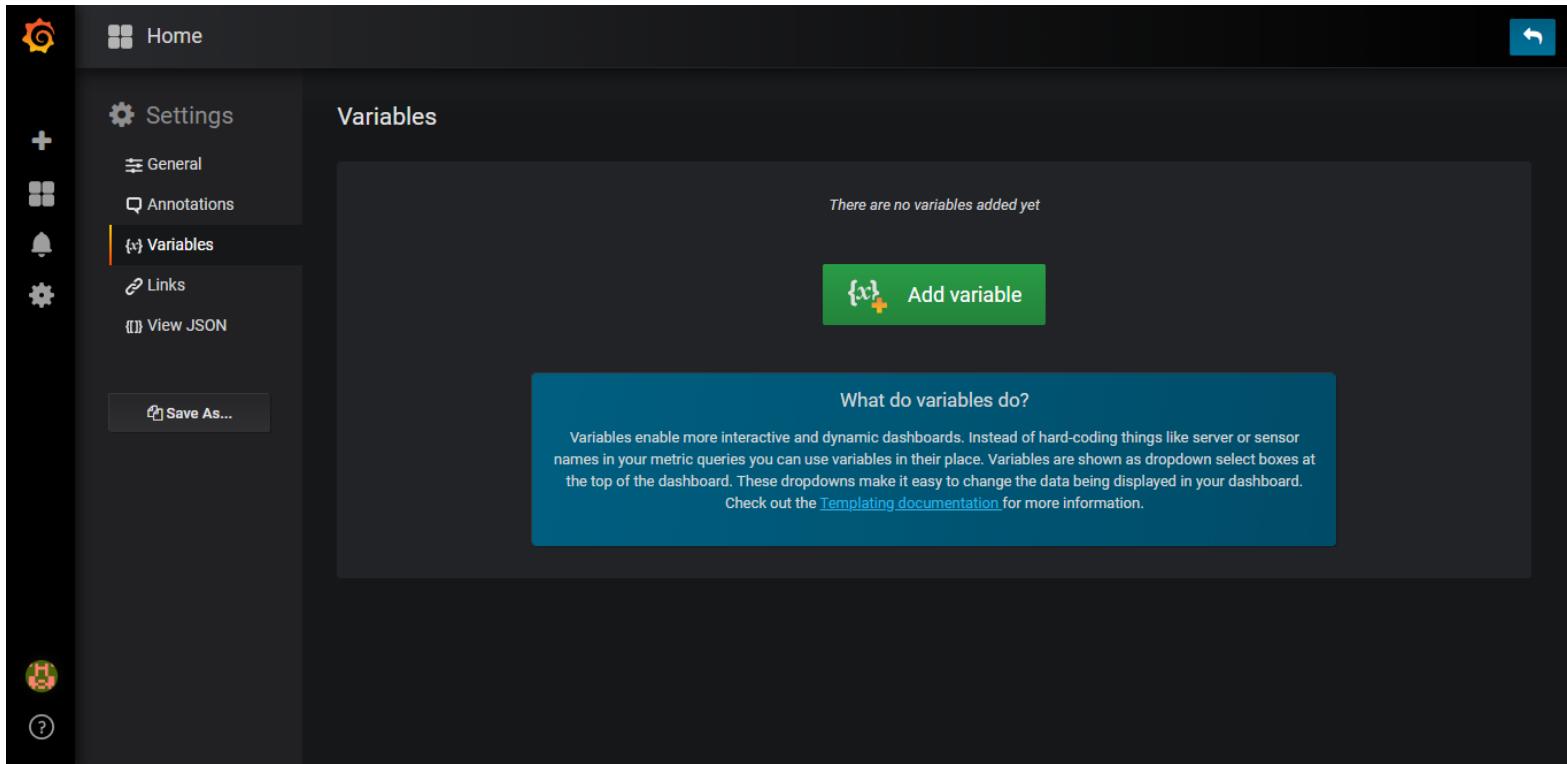
At the bottom of the main content area is a "Panel Options" section with a "Graph Tooltip" dropdown set to "Default".

การใช้งาน Grafana



The screenshot shows the Grafana interface for managing annotations. The left sidebar has a 'Annotations' section selected. The main area is titled 'Annotations' and contains a 'Query name' dropdown set to 'Annotations & Alerts (Built-in)' and a 'Data source' dropdown set to 'Grafana'. A message states 'There are no custom annotation queries added yet'. A green button labeled 'Add Annotation Query' with a plus icon is visible. A callout box titled 'What are Annotations?' provides a brief explanation: 'Annotations provide a way to integrate event data into your graphs. They are visualized as vertical lines and icons on all graph panels. When you hover over an annotation icon you can get event text & tags for the event. You can add annotation events directly from grafana by holding CTRL or CMD + click on graph (or drag region). These will be stored in Grafana's annotation database.' It also links to 'Annotations documentation'.

การใช้งาน Grafana



The screenshot shows the 'Variables' section of the Grafana configuration interface. On the left, a sidebar menu includes 'Home', 'Settings' (selected), 'General', 'Annotations', 'Variables' (highlighted with an orange border), 'Links', and 'View JSON'. Below the sidebar is a 'Save As...' button. The main content area is titled 'Variables' and displays the message 'There are no variables added yet'. A prominent green button at the bottom right says '{x} + Add variable'. A blue callout box contains the text 'What do variables do?'. It explains that variables enable more interactive and dynamic dashboards by allowing users to use variables in metric queries instead of hard-coding server or sensor names. It also mentions that variables are shown as dropdown select boxes at the top of the dashboard. A link to 'Templating documentation' is provided for more information.

Variables

There are no variables added yet

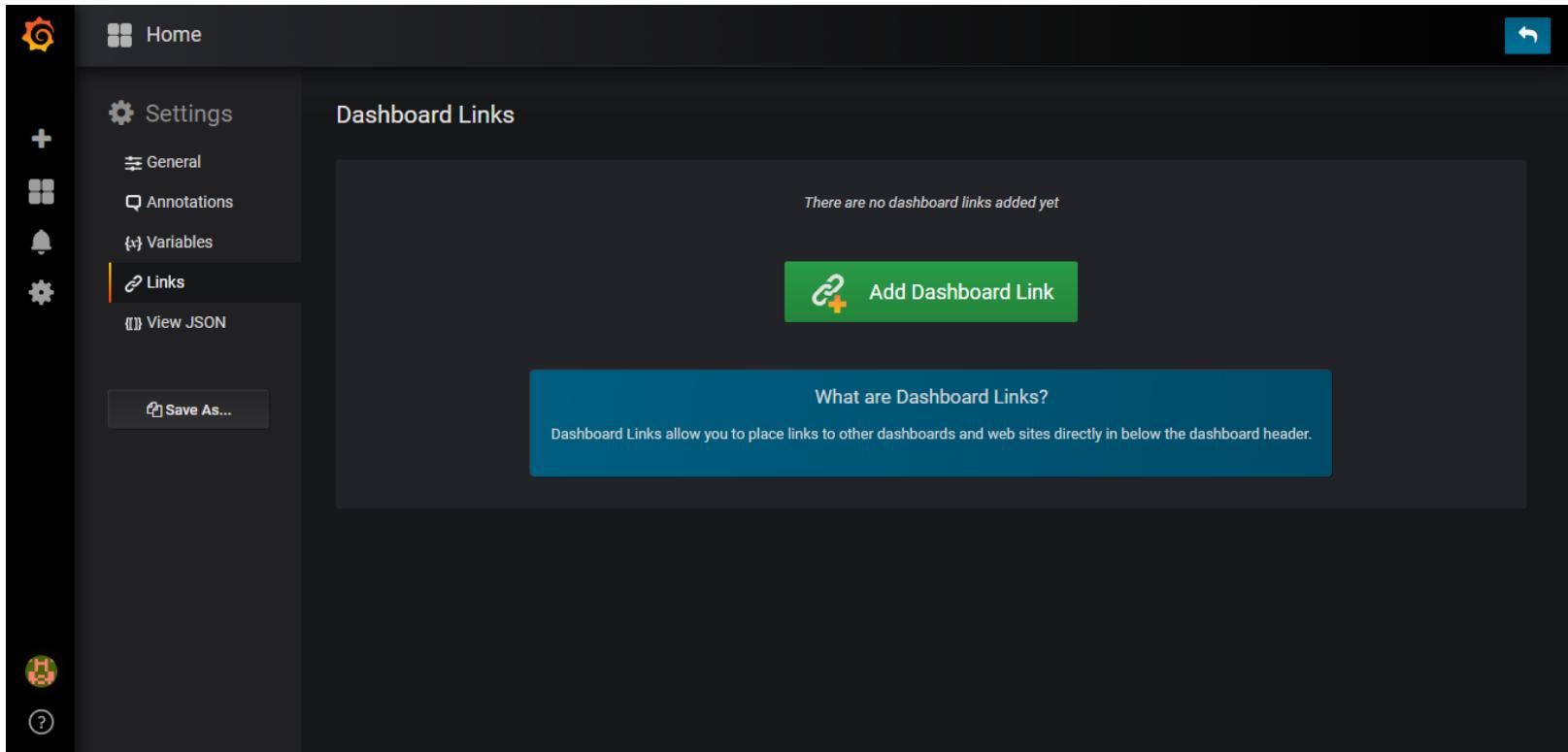
{x} + Add variable

What do variables do?

Variables enable more interactive and dynamic dashboards. Instead of hard-coding things like server or sensor names in your metric queries you can use variables in their place. Variables are shown as dropdown select boxes at the top of the dashboard. These dropdowns make it easy to change the data being displayed in your dashboard.

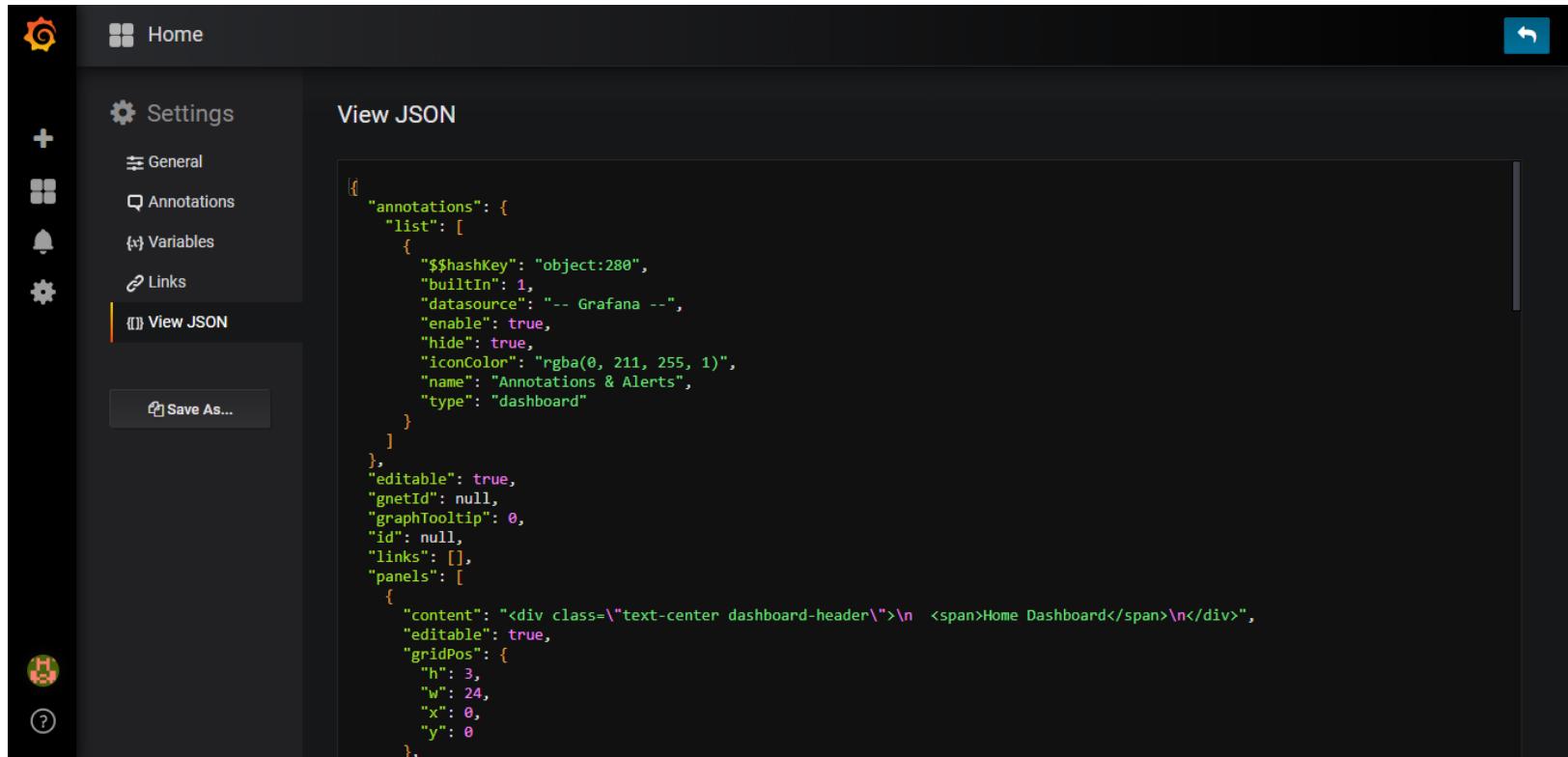
Check out the [Templating documentation](#) for more information.

การใช้งาน Grafana



The screenshot shows the 'Dashboard Links' configuration page in Grafana. The left sidebar has a 'Links' item selected, indicated by an orange border. The main area displays a message: 'There are no dashboard links added yet'. A prominent green button at the bottom right says 'Add Dashboard Link' with a plus sign icon. Below this button, a blue callout box contains the text: 'What are Dashboard Links? Dashboard Links allow you to place links to other dashboards and web sites directly in below the dashboard header.' On the far left of the sidebar, there are icons for Home, Settings, General, Annotations, Variables, and View JSON, along with a 'Save As...' button. At the bottom of the sidebar are icons for a gear and a question mark.

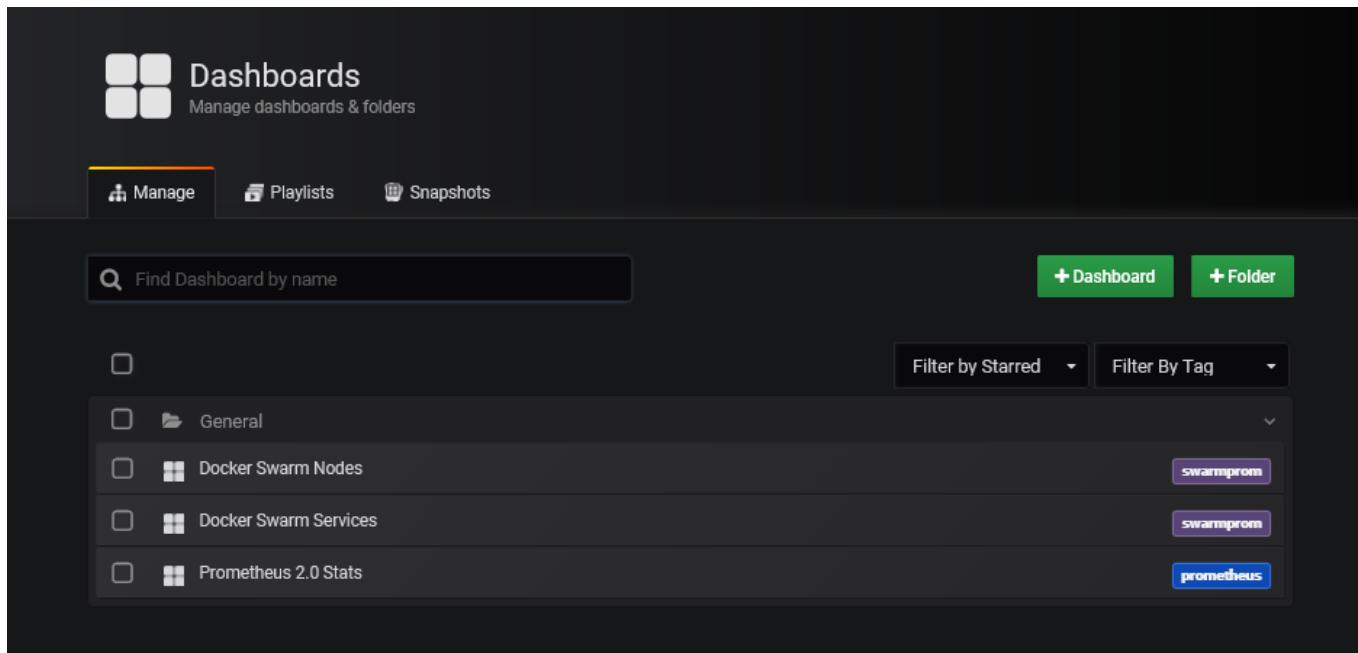
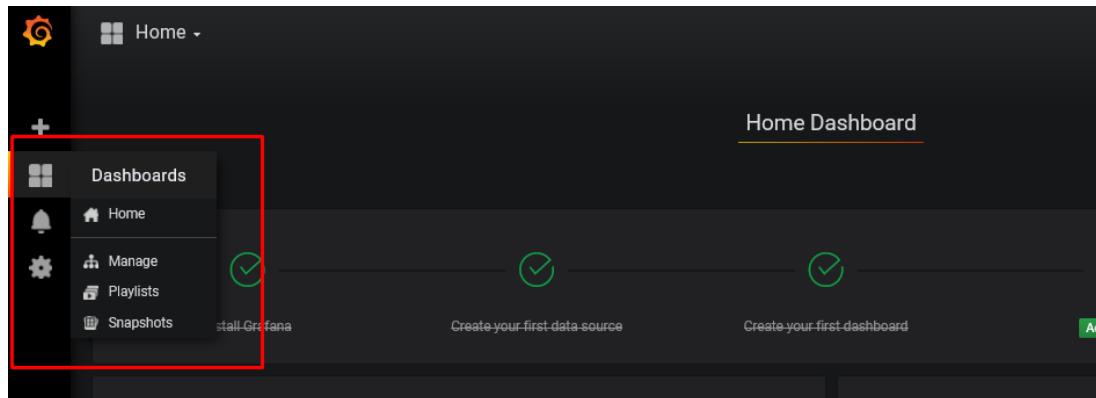
การใช้งาน Grafana



The screenshot shows the Grafana interface with a dark theme. On the left, there is a sidebar with various icons: a gear for settings, a plus sign for creating new dashboards, a grid for annotations, a bell for variables, a link icon for links, and a 'View JSON' icon which is highlighted with an orange border. Below these are 'Save As...' and a question mark icon. The main area is titled 'View JSON' and displays a JSON configuration file for a dashboard. The JSON code includes sections for annotations, editable fields, and panels, with some content being rendered as HTML code.

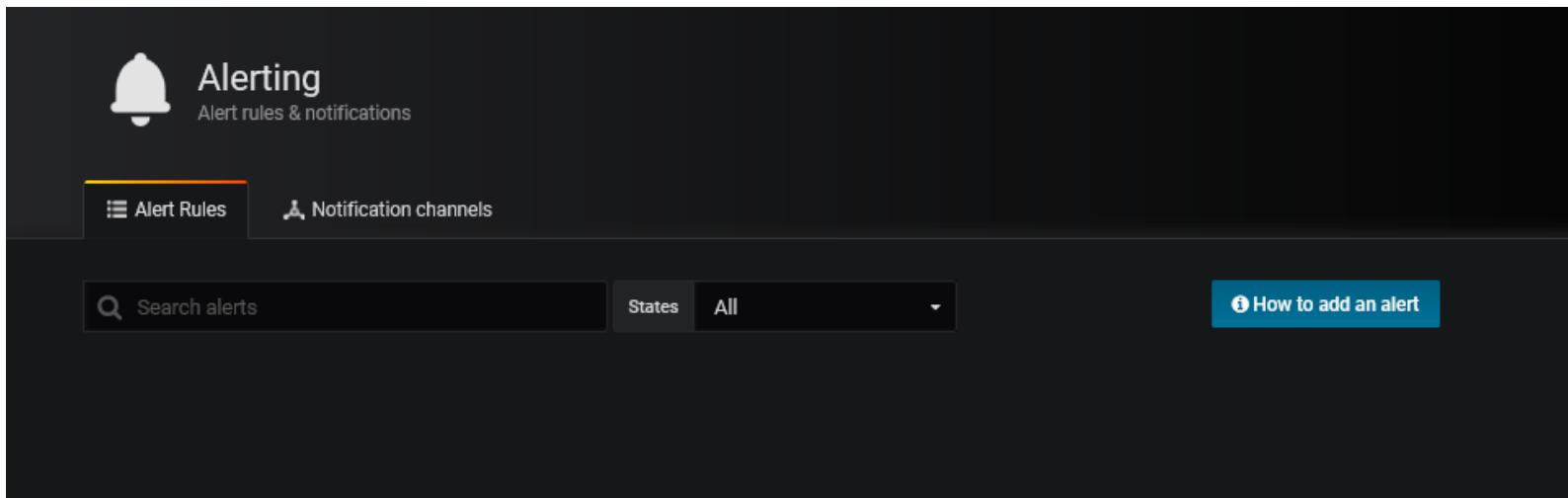
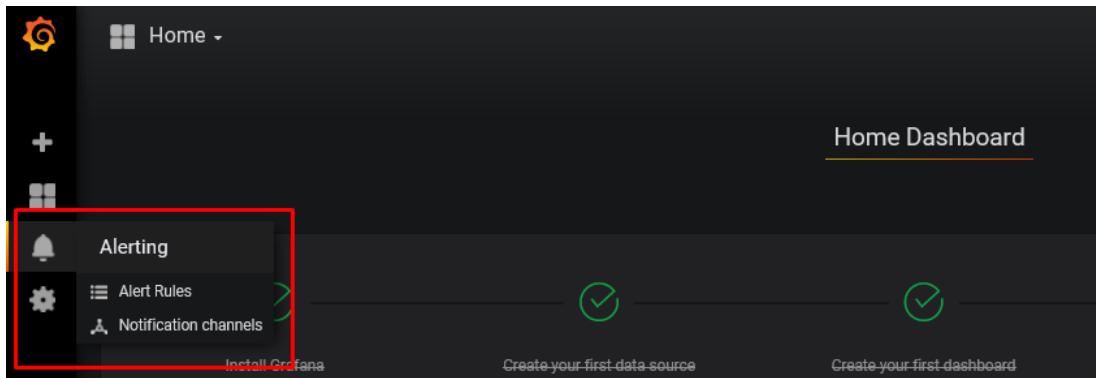
```
{
  "annotations": {
    "list": [
      {
        "$$hashKey": "object:280",
        "builtin": 1,
        "datasource": "-- Grafana --",
        "enable": true,
        "hide": true,
        "iconColor": "rgba(0, 211, 255, 1)",
        "name": "Annotations & Alerts",
        "type": "dashboard"
      }
    ],
    "editable": true,
    "gnetId": null,
    "graphTooltip": 0,
    "id": null,
    "links": [],
    "panels": [
      {
        "content": "<div class=\"text-center dashboard-header\">\n  <span>Home Dashboard</span>\n</div>",
        "editable": true,
        "gridPos": {
          "h": 3,
          "w": 24,
          "x": 0,
          "y": 0
        },
        "type": "text"
      }
    ]
  }
}
```

การใช้งาน Grafana



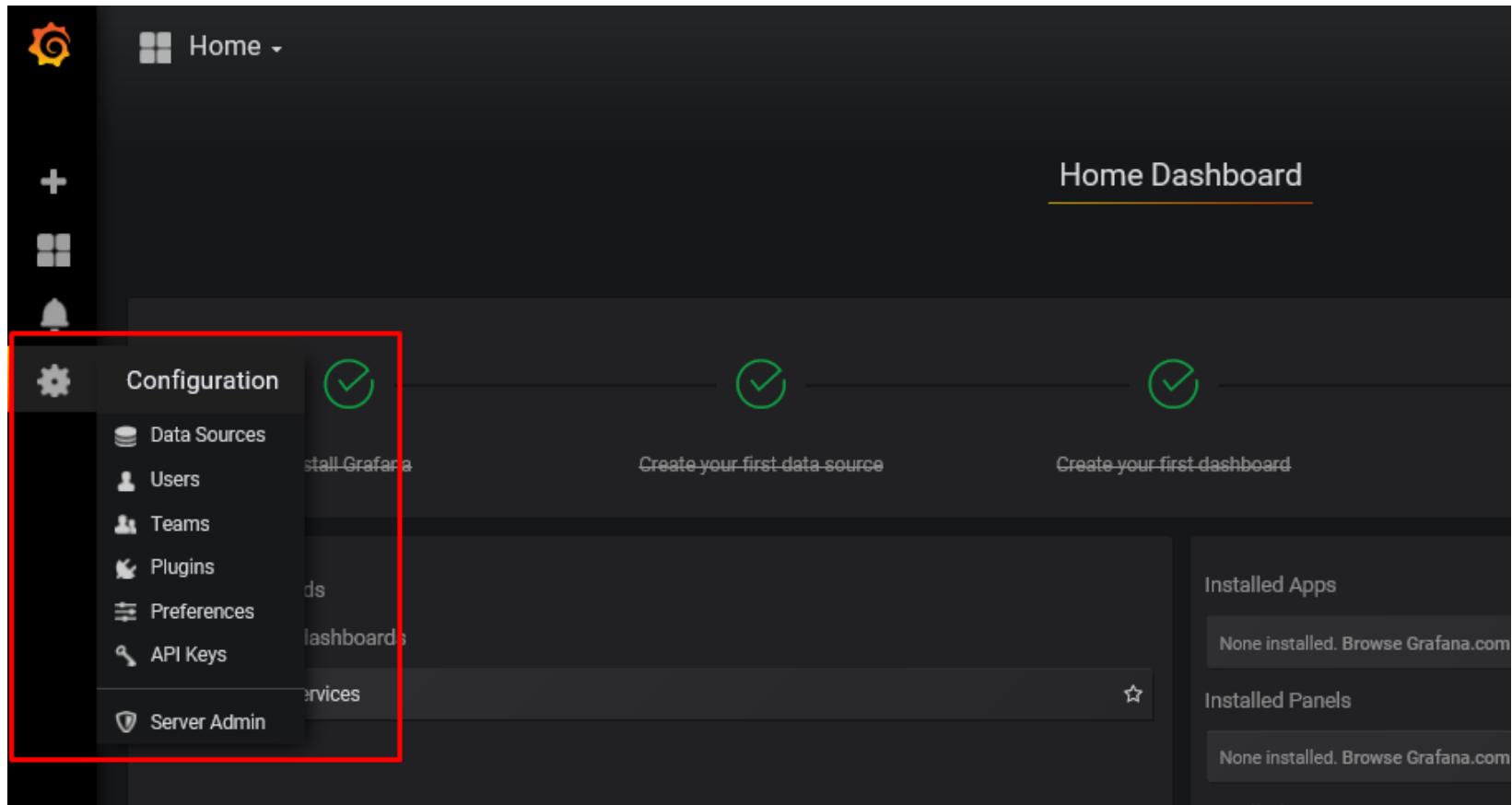
The screenshot shows the 'Dashboards' management interface. It features a search bar at the top with the placeholder 'Find Dashboard by name' and two green buttons for '+ Dashboard' and '+ Folder'. Below the search bar are filters for 'Filter by Starred' and 'Filter By Tag'. The main list contains four dashboard entries: 'General' (under 'General'), 'Docker Swarm Nodes' (tagged 'swarmprom'), 'Docker Swarm Services' (tagged 'swarmprom'), and 'Prometheus 2.0 Stats' (tagged 'prometheus').

การใช้งาน Grafana



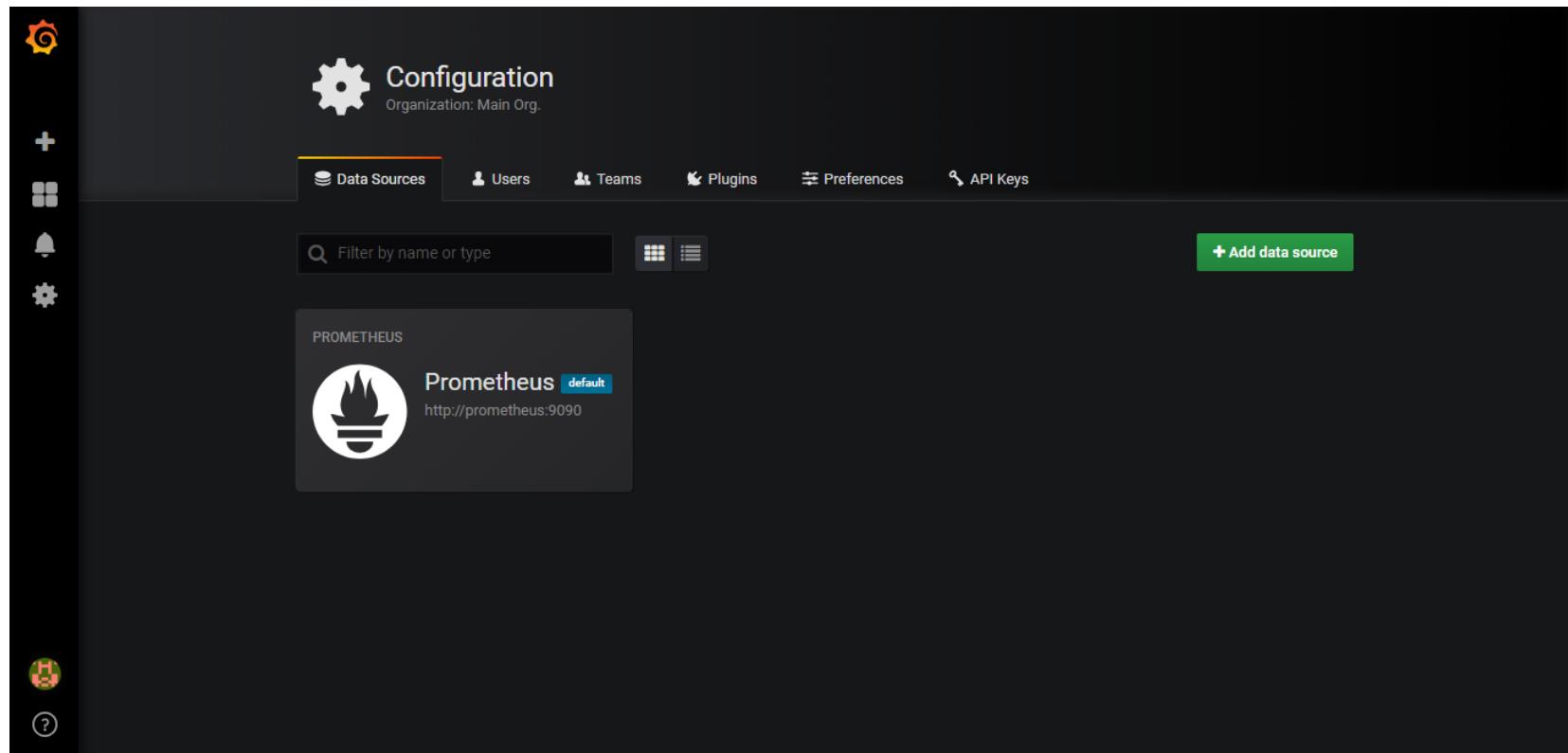
The screenshot shows the Grafana Alerting interface. It features a header with a bell icon and the title "Alerting". Below the header are two tabs: "Alert Rules" (selected) and "Notification channels". A search bar with the placeholder "Search alerts" is followed by a "States" dropdown set to "All". In the top right corner, there's a blue button labeled "How to add an alert".

การใช้งาน Grafana

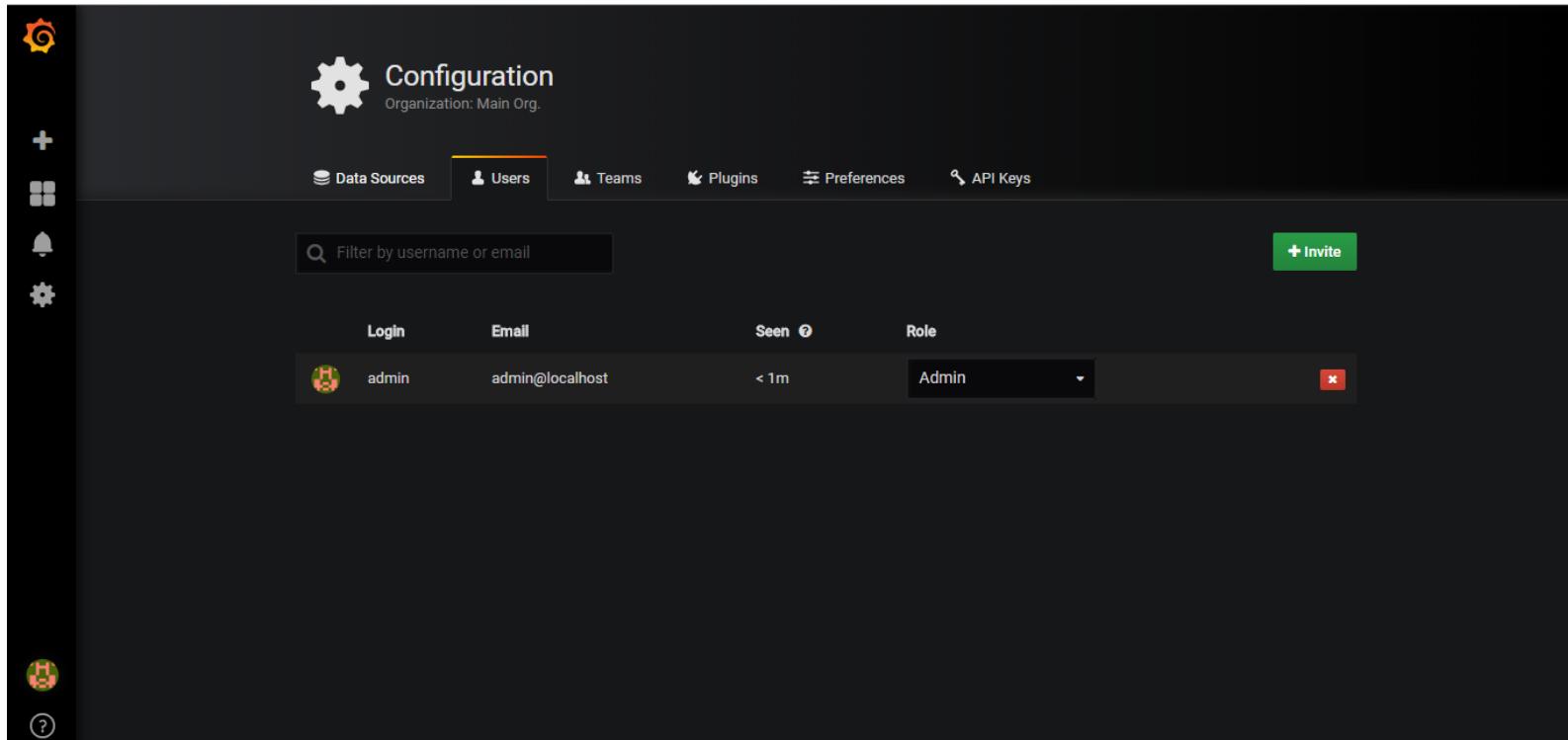


The screenshot shows the Grafana Home Dashboard. On the left, there is a sidebar with various icons and links. A red box highlights the "Configuration" section, which includes links for Data Sources, Users, Teams, Plugins, Preferences, API Keys, and Server Admin. To the right of the sidebar, there are three main sections: "Install Grafana" (with a checkmark), "Create your first data-source" (with a checkmark), and "Create your first dashboard" (with a checkmark). Below these sections, there are two panels: "Installed Apps" (which says "None installed. Browse Grafana.com") and "Installed Panels" (which says "None installed. Browse Grafana.com").

การใช้งาน Grafana

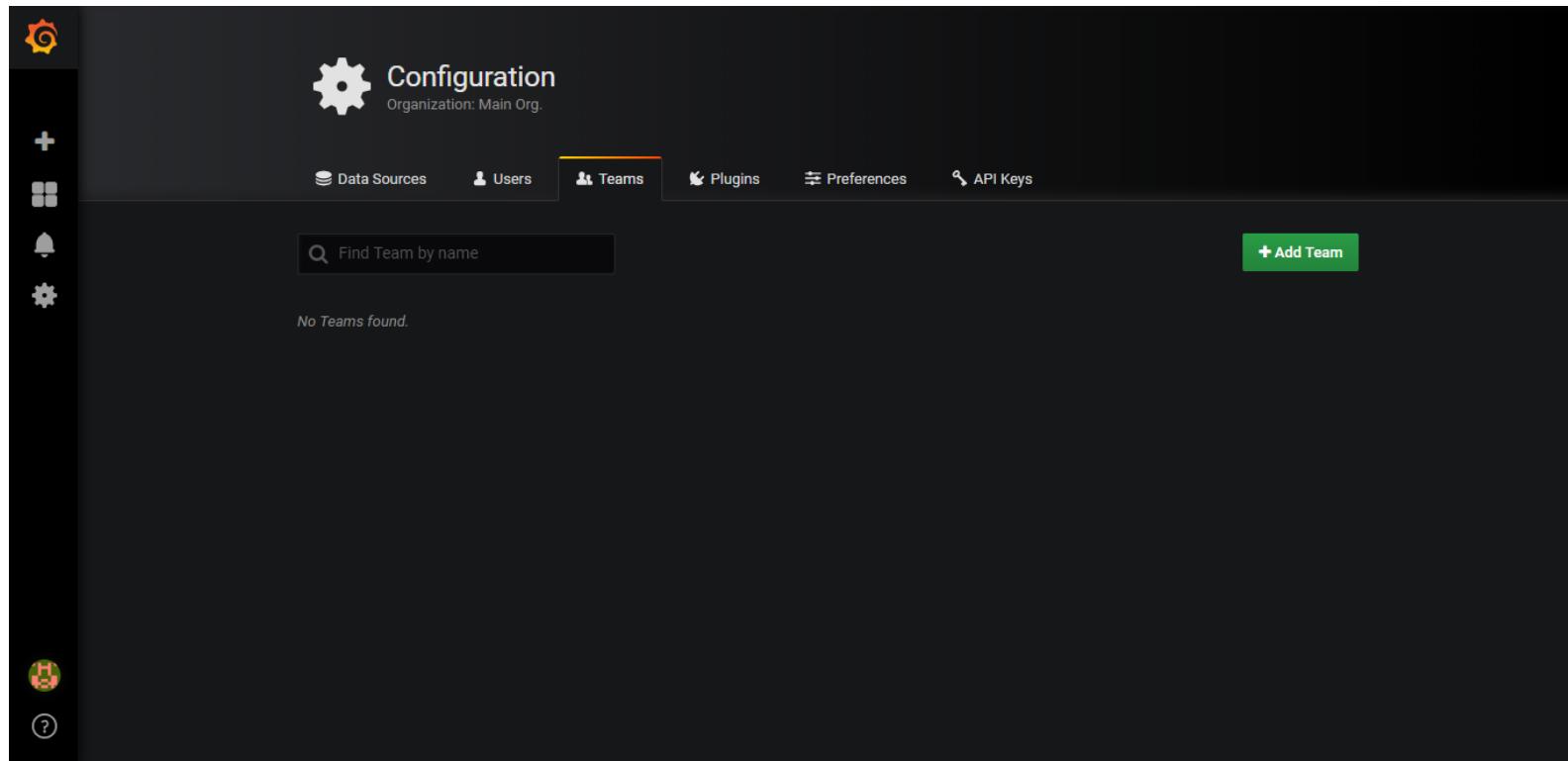


การใช้งาน Grafana

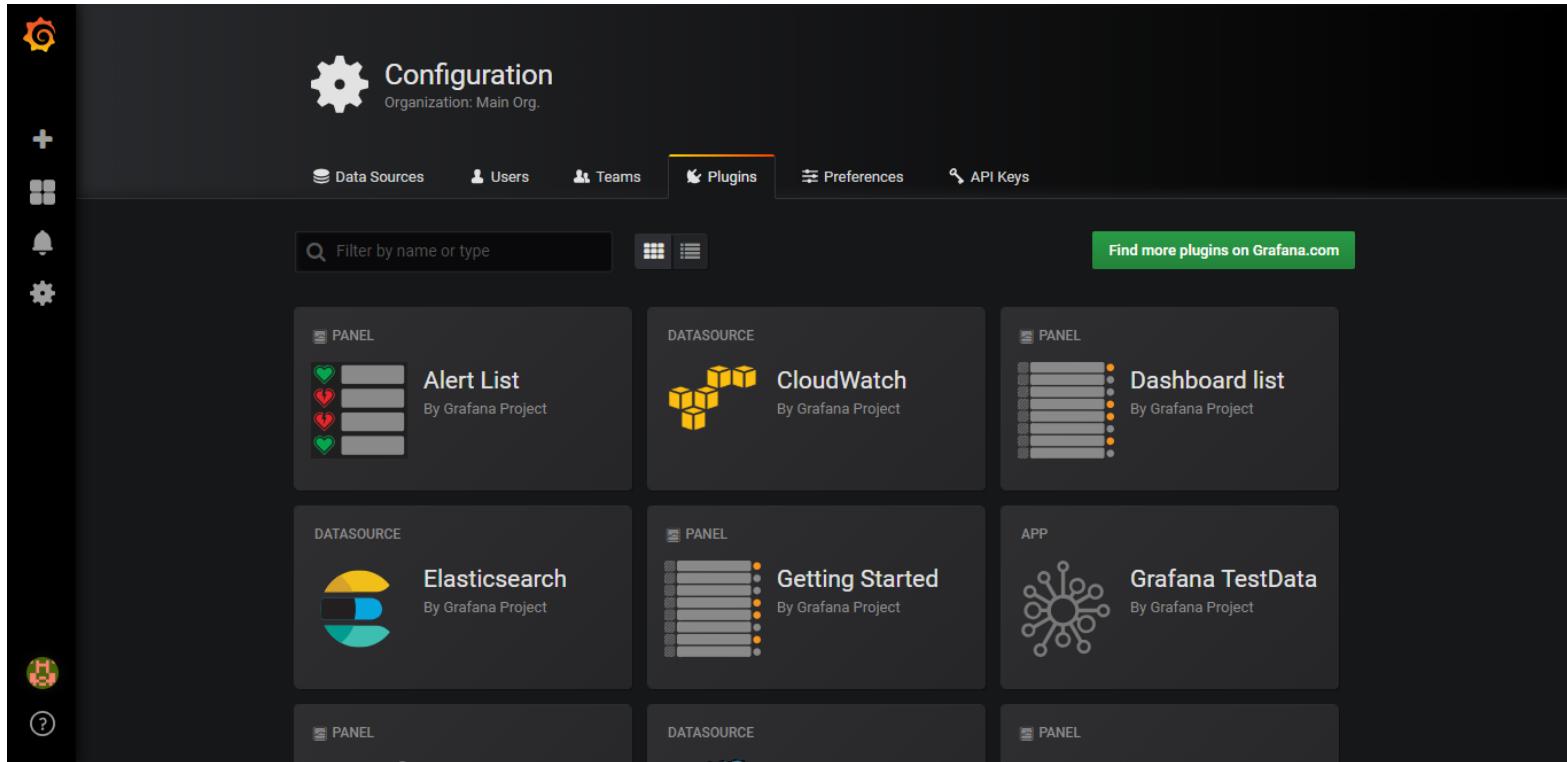


The screenshot shows the Grafana Configuration interface. On the left is a sidebar with icons for Data Sources, + Add, Plugins, Preferences, API Keys, and Help. The main area has a dark header with a gear icon and the text "Configuration". Below it says "Organization: Main Org.". The top navigation bar includes tabs for Data Sources, Users (which is selected), Teams, Plugins, Preferences, and API Keys. A search bar with the placeholder "Filter by username or email" is followed by a green "Invite" button. The main content area displays a table with columns: Login, Email, Seen, and Role. One row is shown for the user "admin" (email admin@localhost, last seen < 1m ago, role Admin). There is a red "x" icon to the right of the Role column for this user.

การใช้งาน Grafana



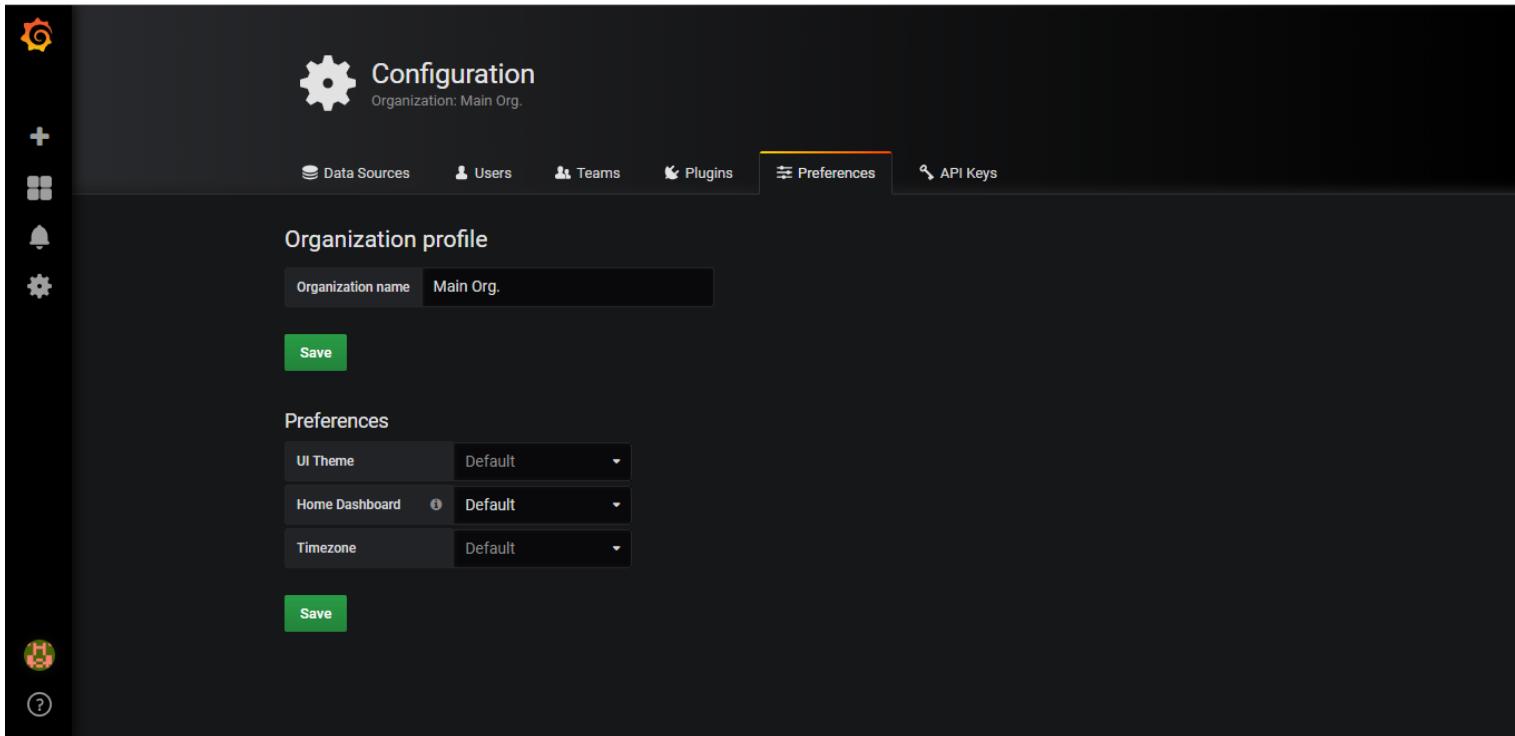
การใช้งาน Grafana



The screenshot shows the Grafana Configuration interface with the following elements:

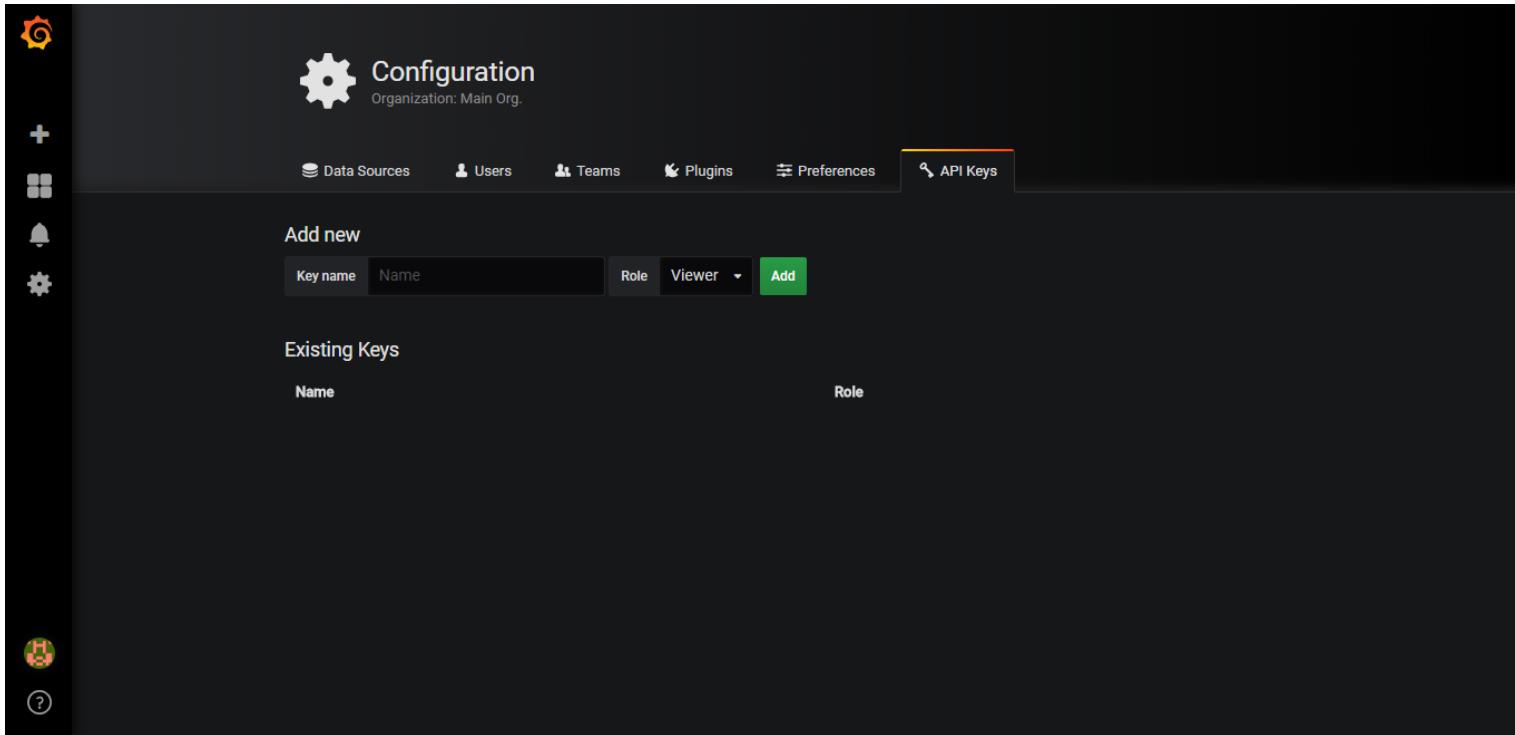
- Left Sidebar:** A vertical sidebar with icons for Home, Plugins, Data Sources, Users, Teams, and Help.
- Header:** "Configuration" and "Organization: Main Org."
- Top Navigation:** Tabs for Data Sources, Users, Teams, Plugins (selected), Preferences, and API Keys. A search bar "Filter by name or type" and a button "Find more plugins on Grafana.com" are also present.
- Plugin Cards:** A grid of cards representing different components:
 - Alert List:** PANEL, By Grafana Project, featuring a red and green heart icon.
 - CloudWatch:** Datasource, By Grafana Project, featuring a yellow cube icon.
 - Dashboard list:** PANEL, By Grafana Project, featuring a grey bar chart icon.
 - Elasticsearch:** Datasource, By Grafana Project, featuring a blue and yellow circle icon.
 - Getting Started:** PANEL, By Grafana Project, featuring a grey bar chart icon.
 - Grafana TestData:** APP, By Grafana Project, featuring a network node icon.

การใช้งาน Grafana



The screenshot shows the Grafana Configuration interface. On the left is a sidebar with icons for Data Sources, Users, Teams, Plugins, Preferences (which is selected), and API Keys. The main content area has a dark header with a gear icon and the word "Configuration". Below it says "Organization: Main Org.". The "Organization profile" section contains a text input for "Organization name" with "Main Org." typed in. A green "Save" button is below this. The "Preferences" section contains three dropdown menus: "UI Theme" set to "Default", "Home Dashboard" set to "Default", and "Timezone" set to "Default". Another green "Save" button is at the bottom of this section. The background is dark, and the overall theme is professional.

การใช้งาน Grafana



The screenshot shows the Grafana Configuration interface. On the left is a sidebar with icons for Home, Data Sources, Users, Teams, Plugins, Preferences, and Help. The main area has a dark header with a gear icon and the title "Configuration". Below it says "Organization: Main Org.". A navigation bar includes "Data Sources", "Users", "Teams", "Plugins", "Preferences", and "API Keys" (which is highlighted). The "API Keys" section contains a sub-header "Add new" with fields for "Key name" (set to "Name") and "Role" (set to "Viewer"). A green "Add" button is next to the role dropdown. Below this is a table titled "Existing Keys" with columns "Name" and "Role". There are no entries in the table.

สิ่งที่ควรศึกษาเพิ่มเติม

- Methodology
 - Agile
- Container
 - Docker Swarm. <https://docs.docker.com/engine/swarm/>
 - kubernetes <https://kubernetes.io/>
- Programming
 - Linux shell command <https://docs.cs.cf.ac.uk/notes/linux-shell-commands/>
 - Html, CSS, Javascript <https://www.w3schools.com/>
 - Node.js <https://nodejs.org/en/>