



## Research & Development Team

# DevOps

[www.pnpsw.com](http://www.pnpsw.com)

[sommai.k@gmail.com](mailto:sommai.k@gmail.com)

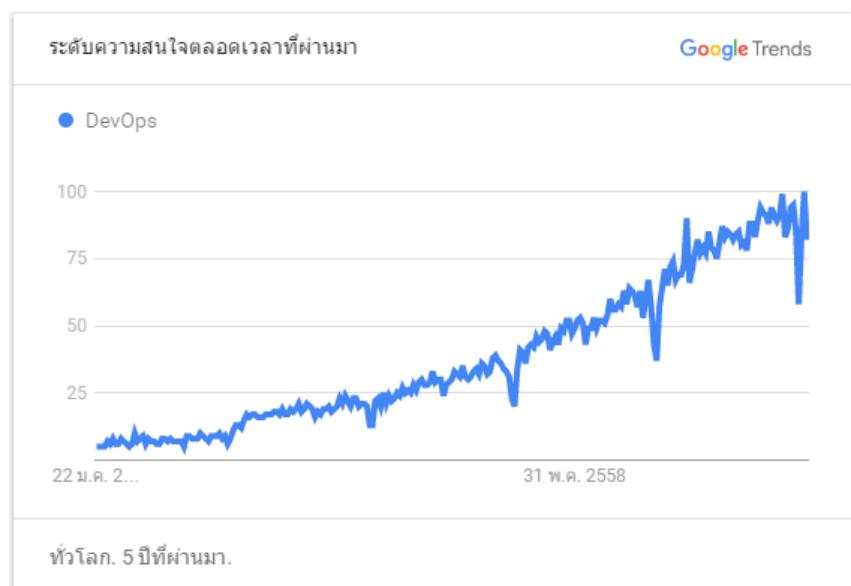
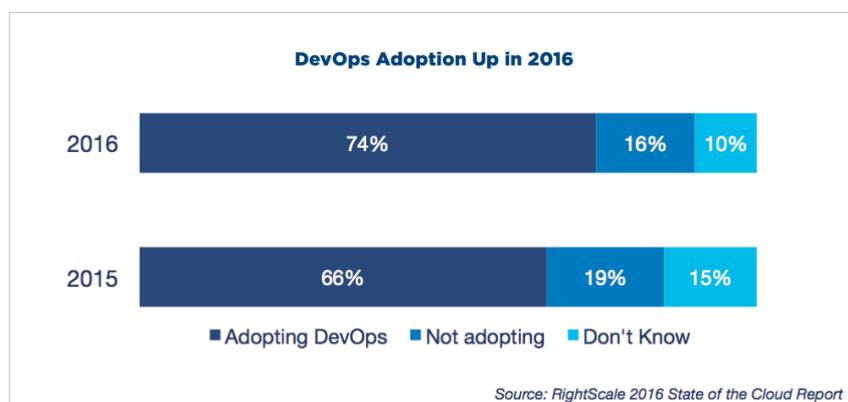
**081-754-4663**

**Line Id : sommai.k**

# 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 เป็นเรื่องทั่วไปที่กำกับในชีวิตประจำวัน 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 พ布ว่าในปี 2016 มีองค์กรนำ DevOps ไปปรับใช้แล้วถึง 74% ซึ่งเพิ่มขึ้นจากปีที่แล้วถึง 8% และจำนวนการ Search คำว่า DevOps ใน google ก็ยังเพิ่มขึ้นเรื่อยๆ ด้วย

# ระดับความสนใจ

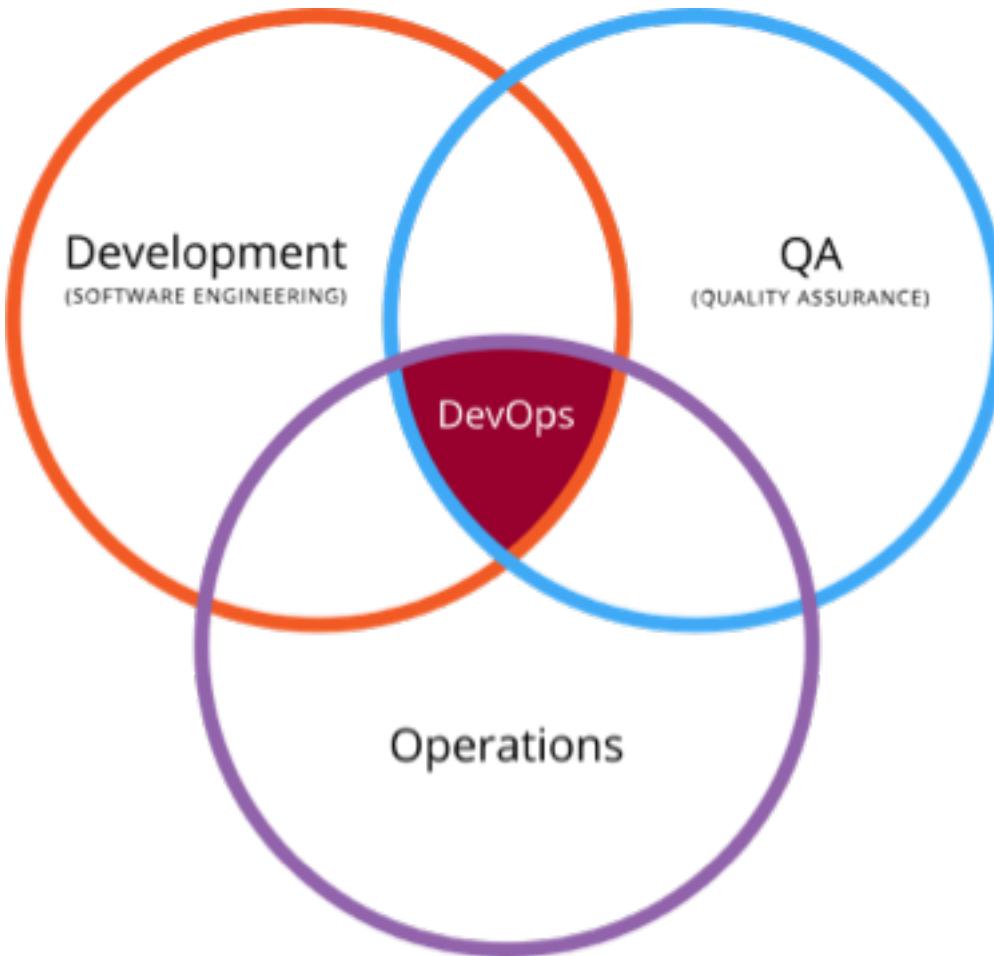


# Cultural change

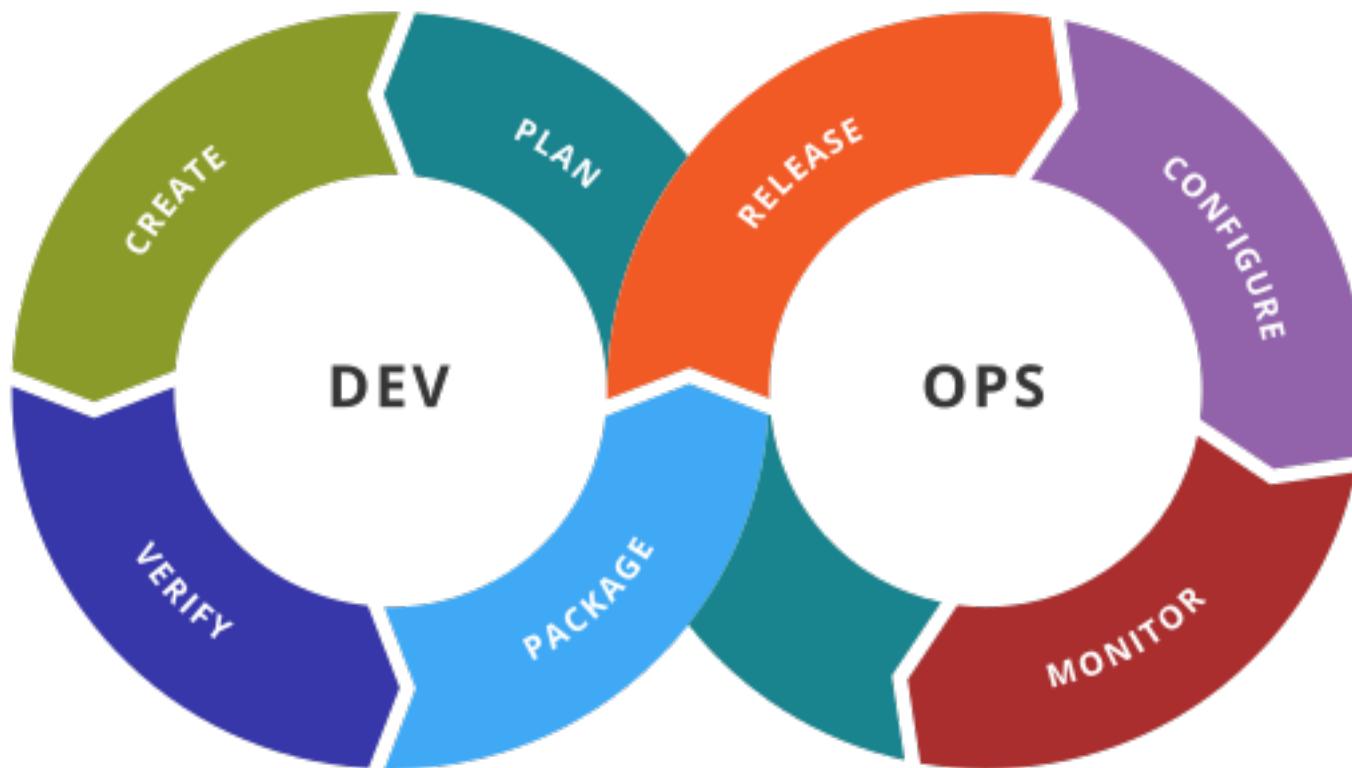
DevOps is more than just a tool or a process change.

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

# Overview



# Workflow

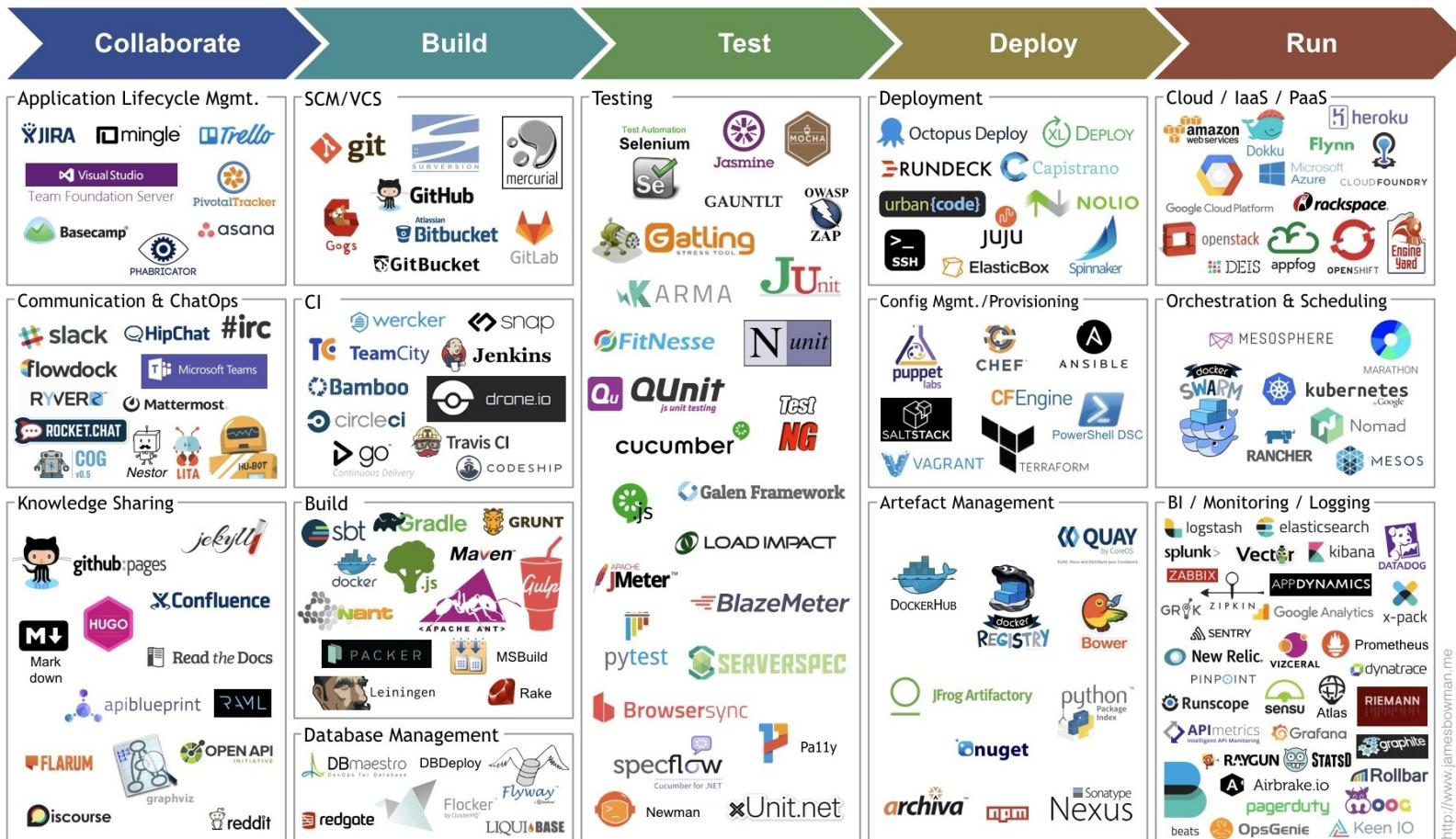


# 7 Ways to Get Started with DevOps

- Invite Your Operations Team Into Your Development Process
- Visualize 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



Software

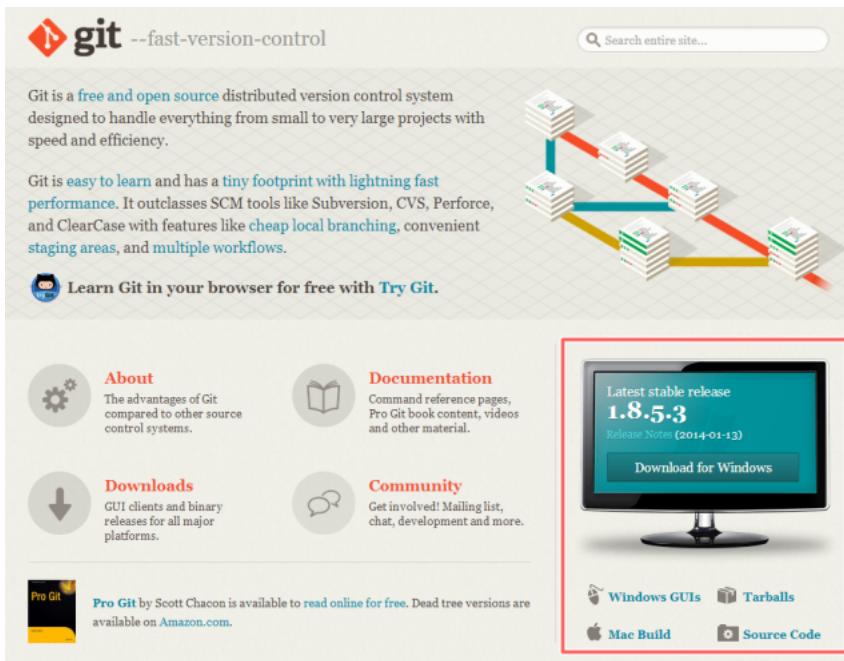
# INSTALLATION

การติดตั้ง

# **GIT FOR WINDOWS**

# การติดตั้ง GIT สำหรับ Windows

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



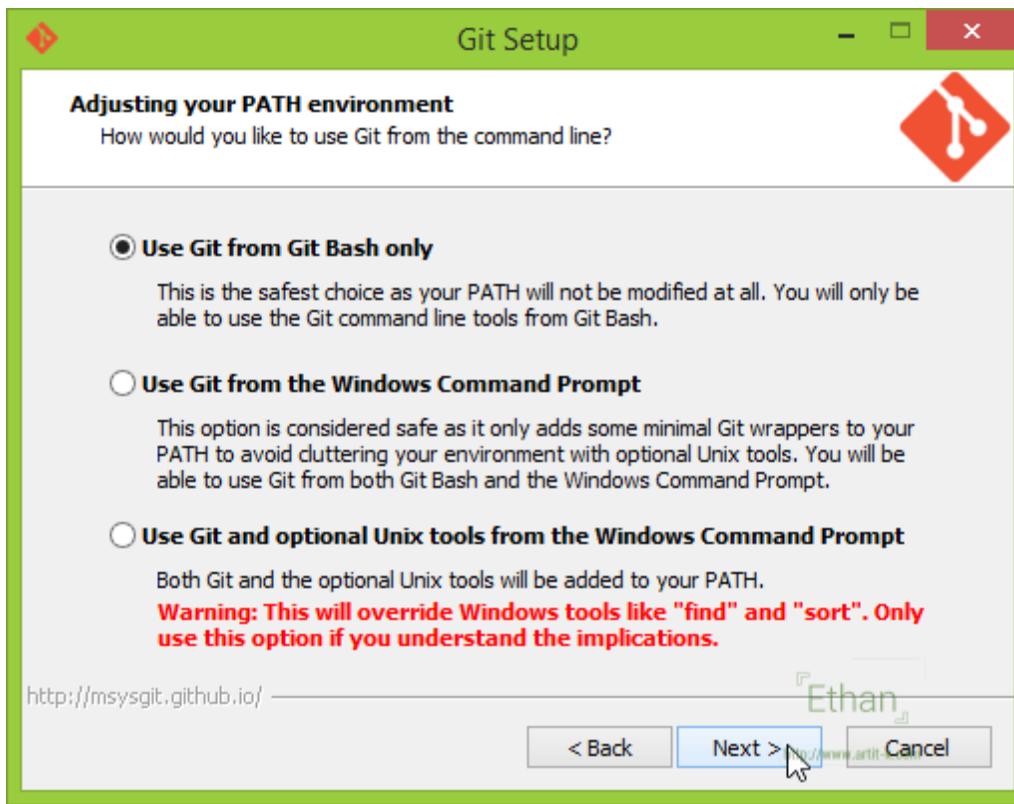
# การติดตั้ง GIT สำหรับ Windows #2

- ติดตั้งโดยใช้สิทธิ Administrator ในการติดตั้ง



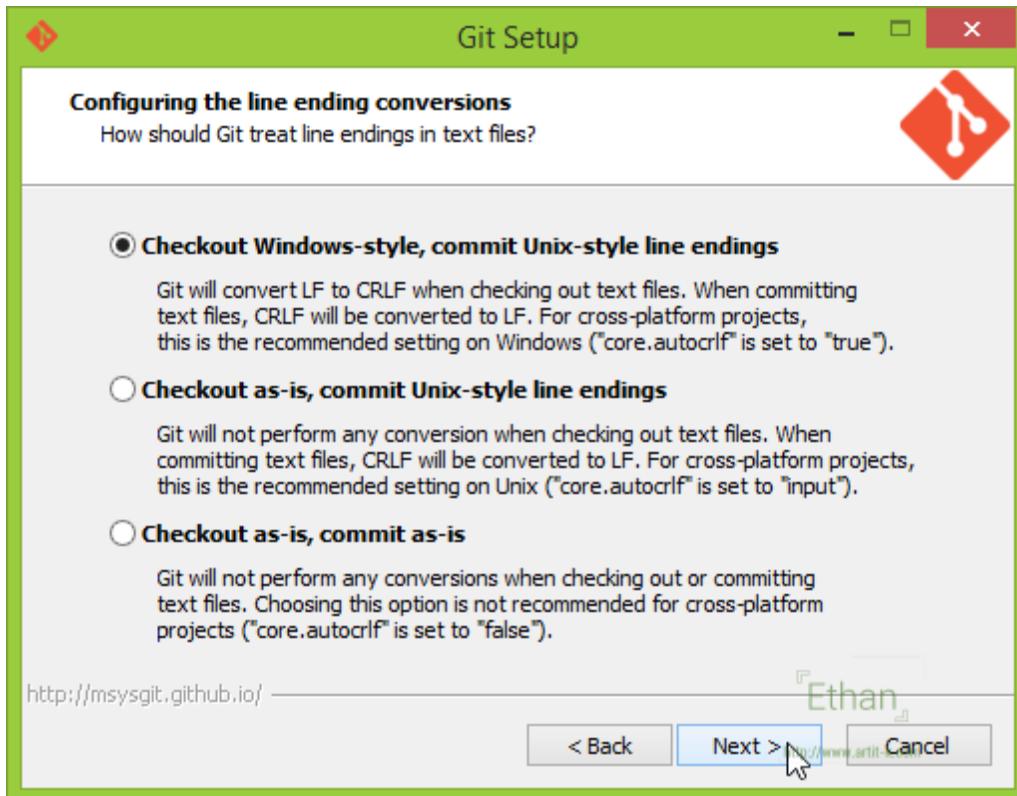
# การติดตั้ง GIT สำหรับ Windows #3

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



# การติดตั้ง GIT สำหรับ Windows #4

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



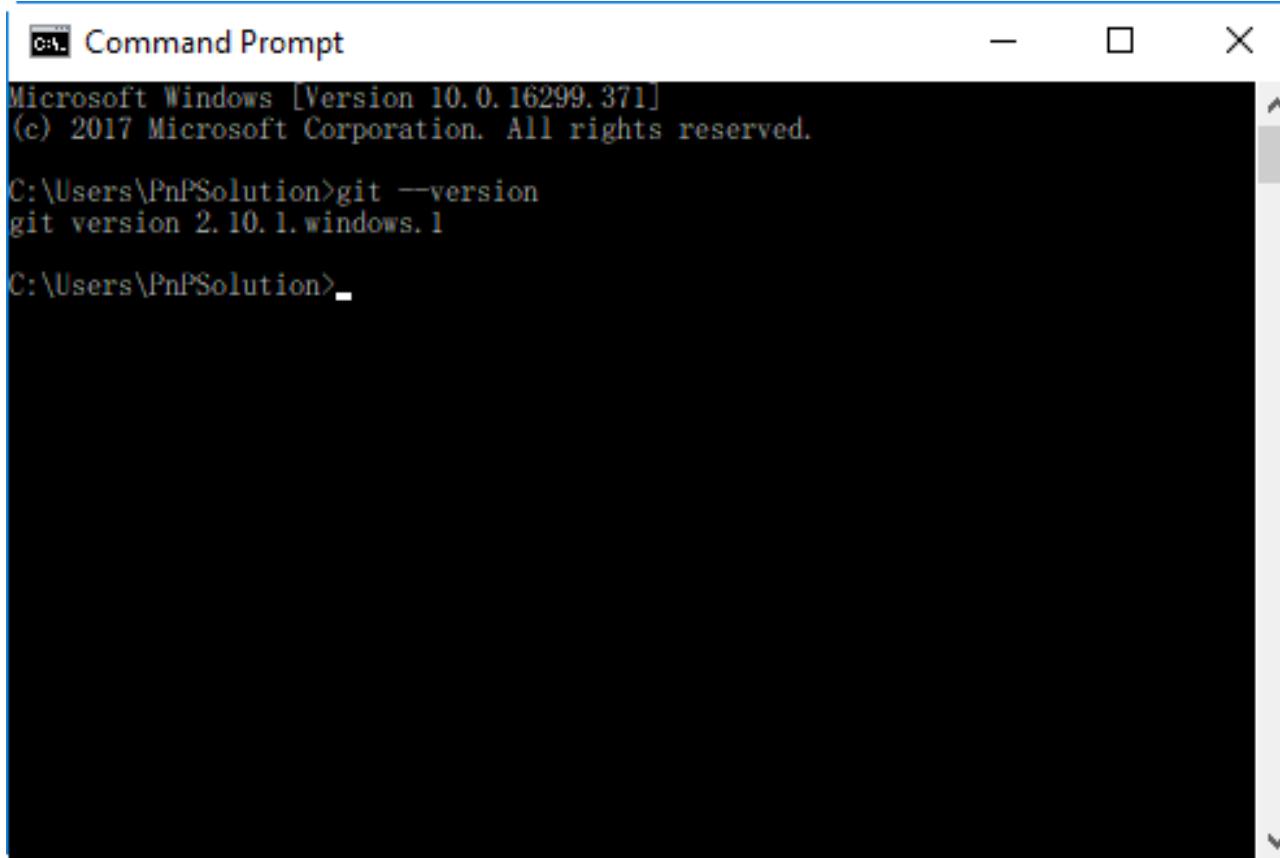
# การติดตั้ง GIT สำหรับ Windows #5

- 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\PnPSSolution>git --version
git version 2.10.1.windows.1

C:\Users\PnPSSolution>
```

การติดตั้ง

# **DOCKER FOR WINDOWS**

# การติดตั้ง Docker สำหรับ Windows

- ความต้องการขั้นพื้นฐานสำหรับการติดตั้ง Docker for windows
  - Windows ต้องเป็น version 64bit เท่านั้น
  - Windows 7 or higher
  - Cpu ต้องมีความสามารถ Hyper-v สามารถเปิดได้ที่ BIOS

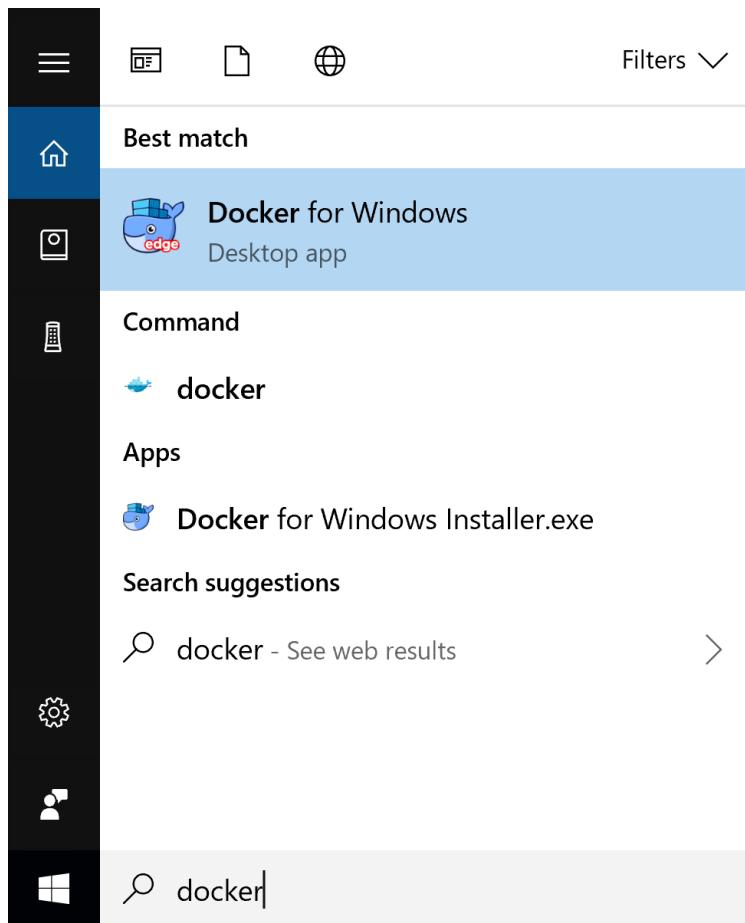


# การติดตั้ง Docker สำหรับ Windows #2

- เข้า link  
<https://download.docker.com/win/stable/Docker%20for%20Windows%20Installer.exe>
- กด next ไปเรื่อยๆ จนสิ้นสุดการติดตั้ง

# การติดตั้ง Docker สำหรับ Windows #3

- หลังจากติดตั้งเสร็จแล้วให้ทำการเปิด docker ดังนี้



# การติดตั้ง Docker สำหรับ Windows #4

- จะมี docker run อยู่ที่ taskbar ดังภาพ



- ทดสอบการติดตั้งโดยการเปิด cmd และพิมพ์คำสั่งดังนี้
- docker ps

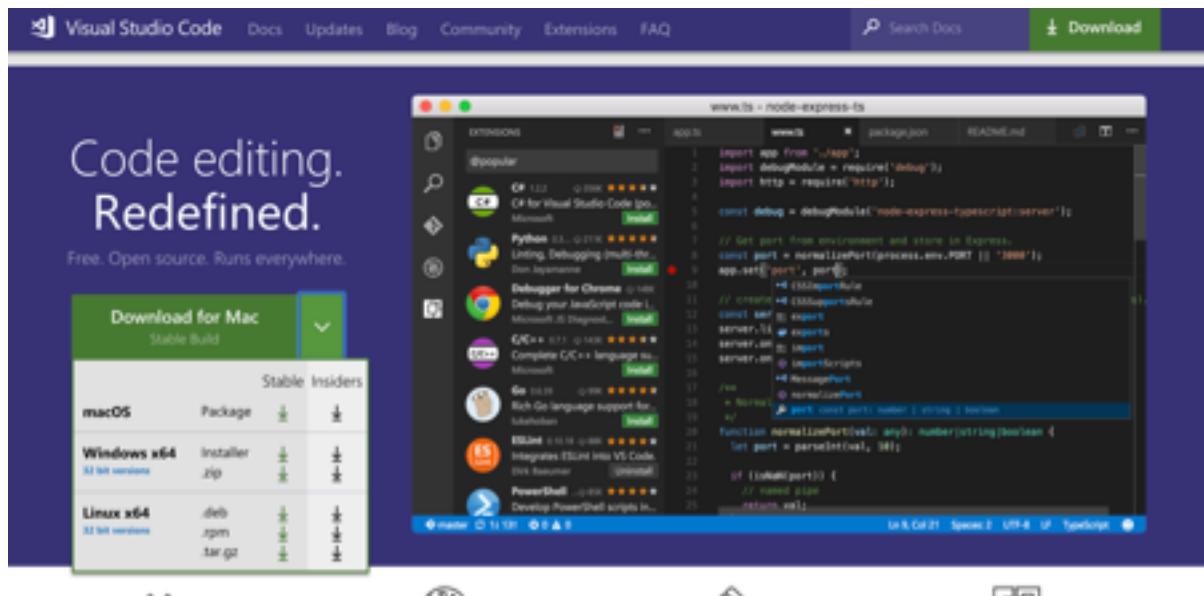
Sommais-MacBook-Pro:~ sommaik\$ docker ps	CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES

การติดตั้ง

# VISUAL STUDIO CODE

# การติดตั้ง VSC

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



การติดตั้ง

# VIRTUAL BOX

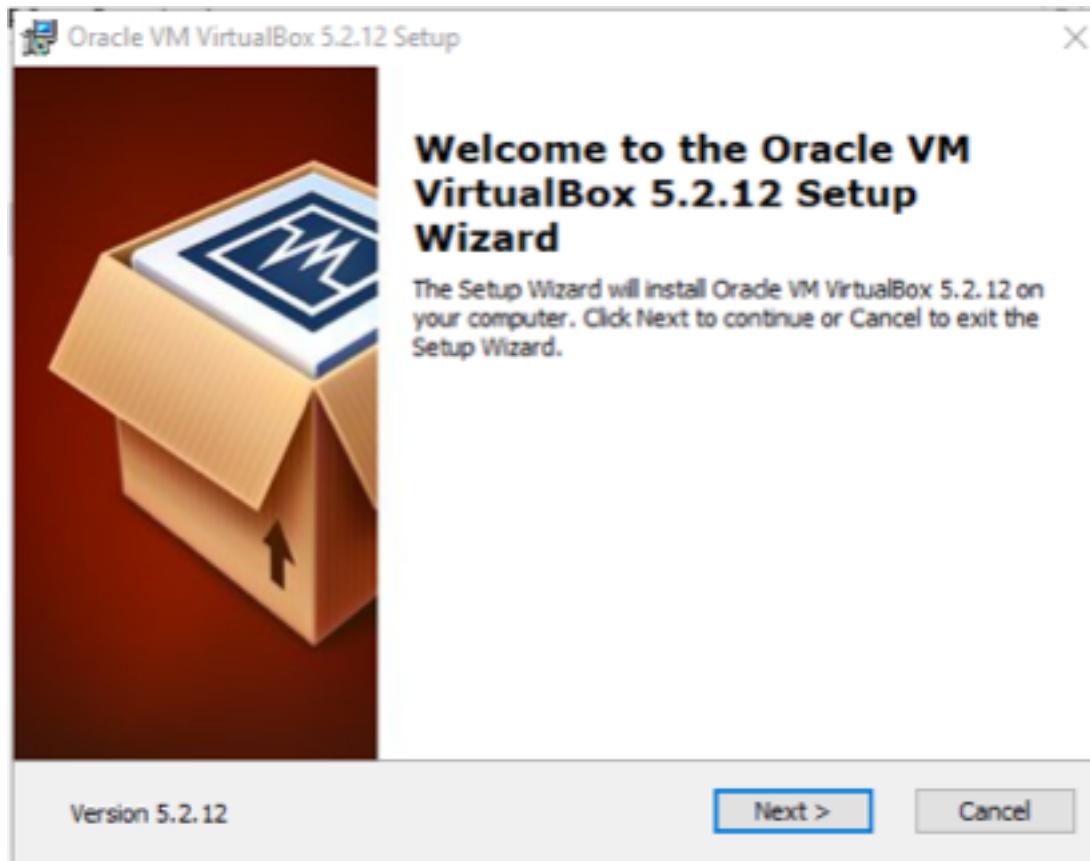
# การติดตั้ง Virtual Box

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

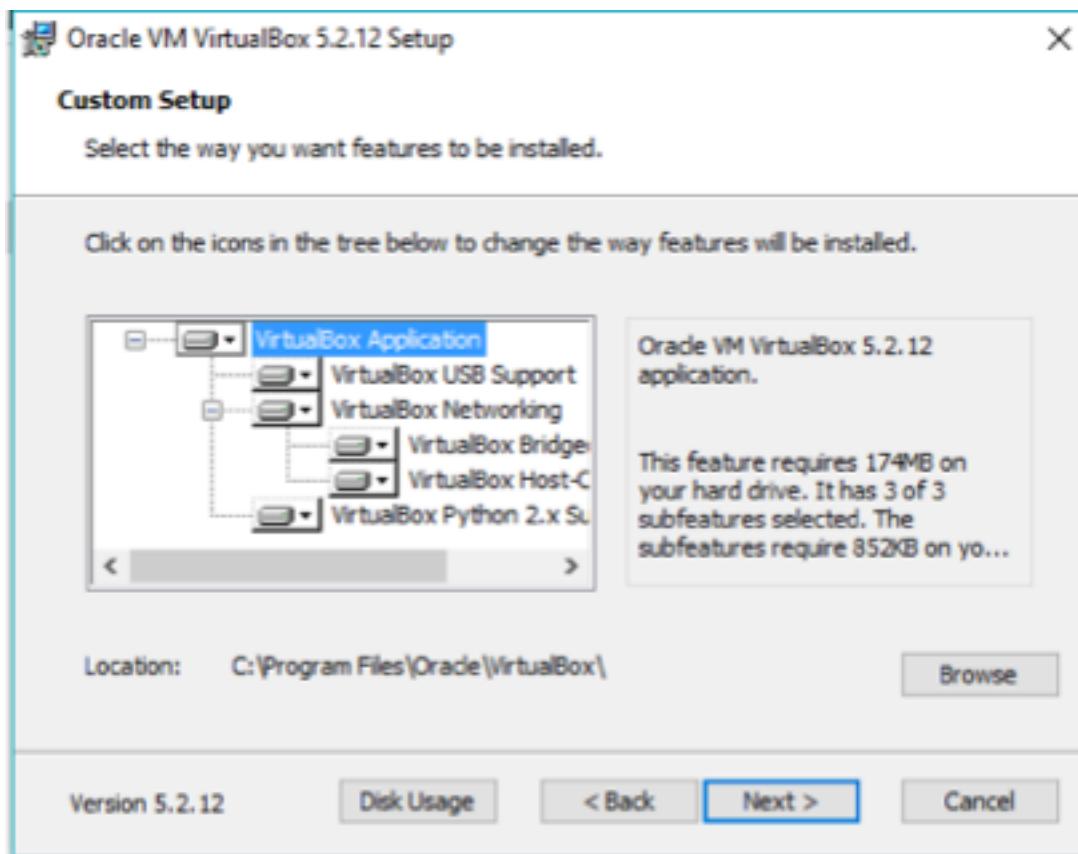


The screenshot shows the official Oracle VM VirtualBox download page. At the top left is the VirtualBox logo, a blue cube with 'VirtualBox' and 'Oracle' on it. The main title 'VirtualBox' is in large blue letters. Below it is a sub-section title 'Download VirtualBox'. A sidebar on the left contains links to 'About', 'Screenshots', 'Downloads', 'Documentation', 'End-user docs', 'Technical docs', 'Contribute', and 'Community'. The main content area starts with a note about finding binary and source code links. It then has a section for 'VirtualBox binaries' with a note about accepting license terms. It mentions 'VirtualBox 5.2.12 platform packages' and lists links for Windows hosts, OS X hosts, Linux distributions, and Solaris hosts. It notes that binaries are released under GPL v2. There's a 'changelog' link, a note about comparing checksums (favoring SHA256 over MD5), and a note about upgrading guest additions. Finally, it lists the 'VirtualBox 5.2.12 Oracle VM VirtualBox Extension Pack' and a link to all supported platforms.

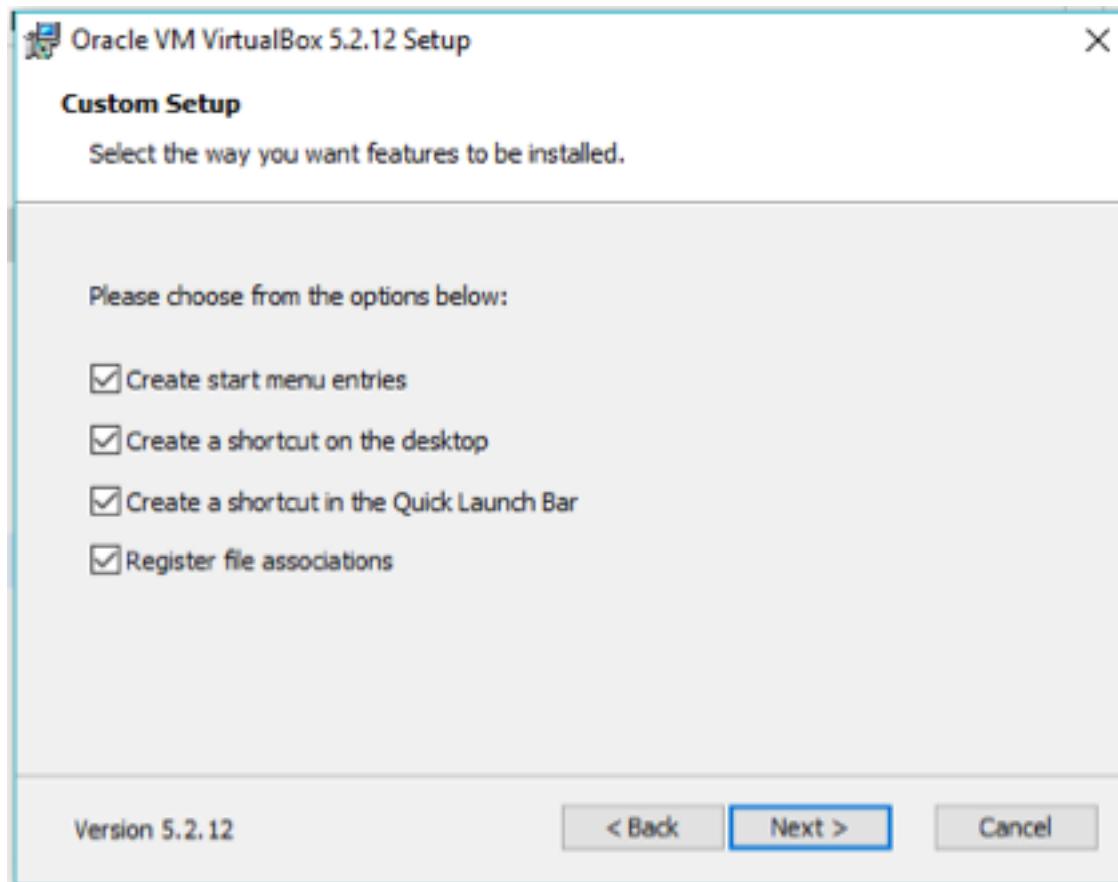
# การติดตั้ง Virtual Box



# การติดตั้ง Virtual Box



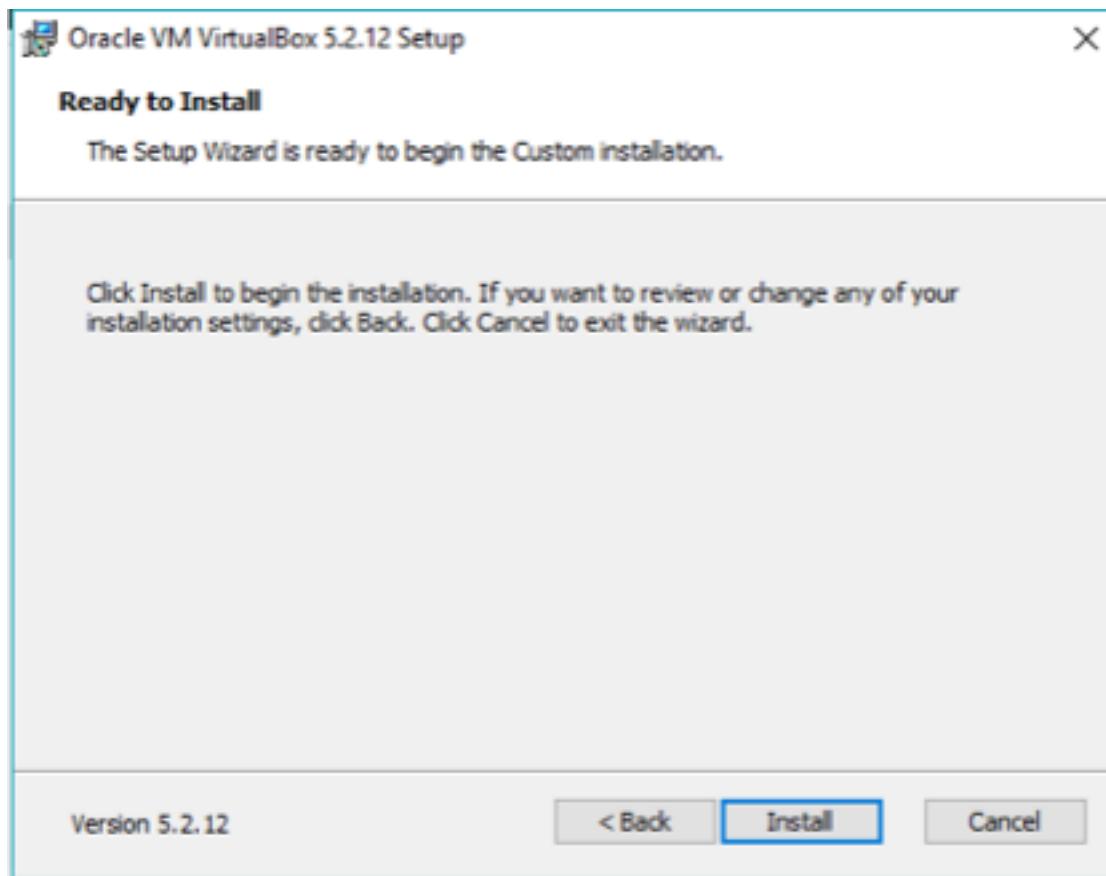
# การติดตั้ง Virtual Box



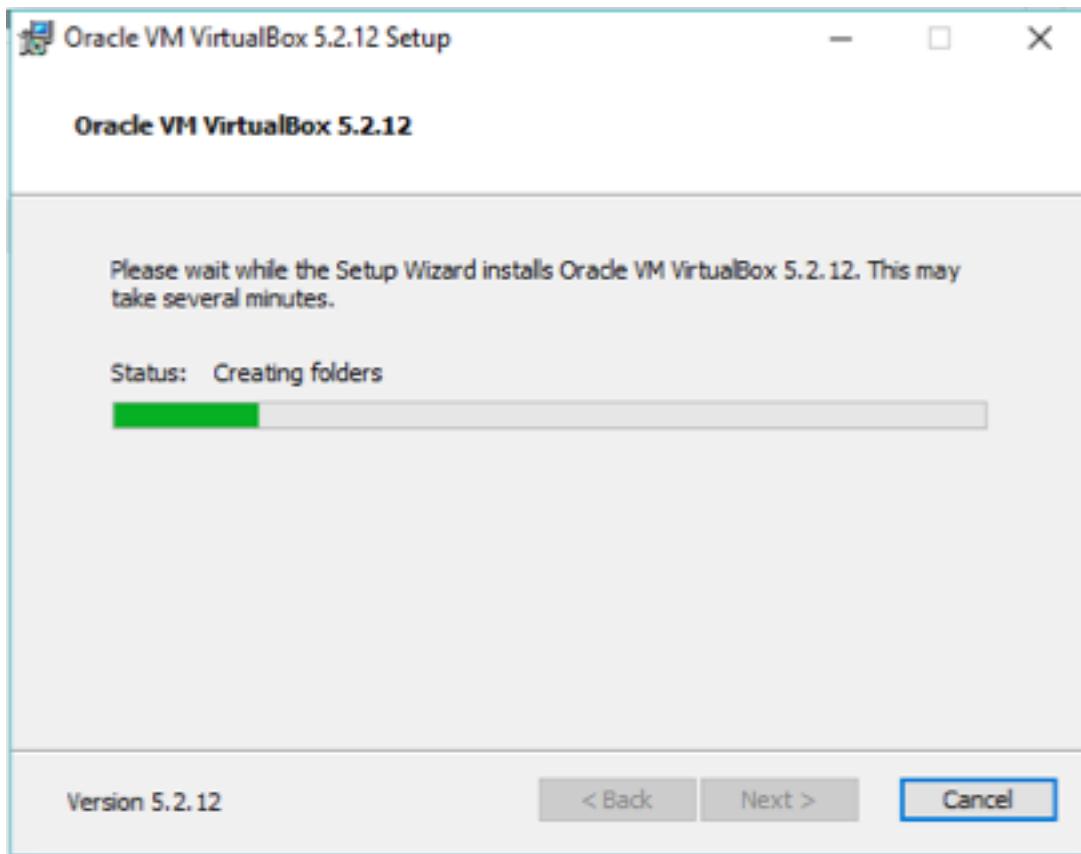
# การติดตั้ง Virtual Box



# การติดตั้ง Virtual Box



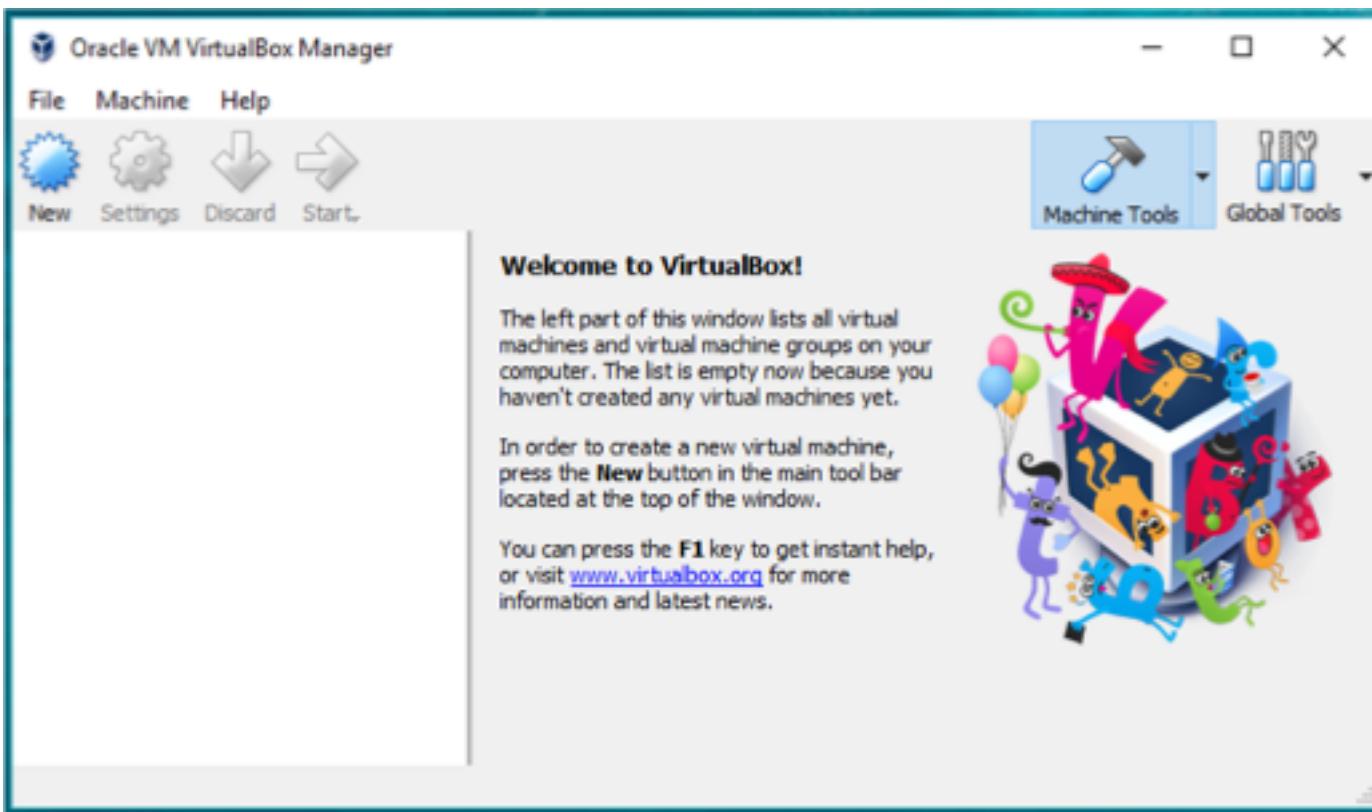
# การติดตั้ง Virtual Box



# การติดตั้ง Virtual Box



# การติดตั้ง Virtual Box

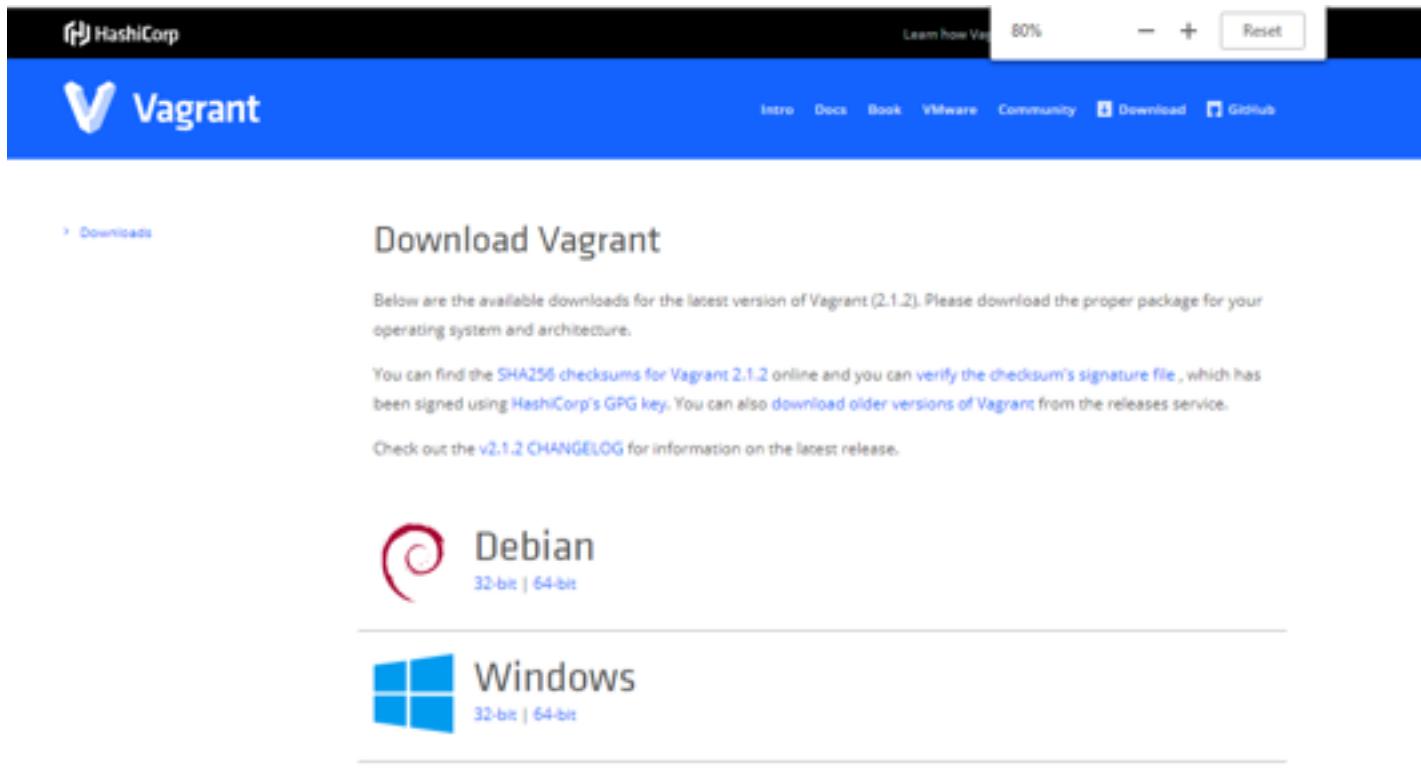


การติดตั้ง

# VAGRANT

# การติดตั้ง Vagrant

- <https://www.vagrantup.com/downloads.html>



The screenshot shows the 'Downloads' section of the Vagrant website. At the top, there's a navigation bar with the HashiCorp logo, a search bar, and links for 'Learn how Vagrant works', '80%', and 'Reset'. Below the navigation is a blue header with the Vagrant logo and links for 'Intro', 'Docs', 'Book', 'VMware', 'Community', 'Download', and 'GitHub'. The main content area has a breadcrumb trail 'Downloads' and a heading 'Download Vagrant'. It states: 'Below are the available downloads for the latest version of Vagrant (2.1.2). Please download the proper package for your operating system and architecture.' It provides links for 'SHA256 checksums for Vagrant 2.1.2' and 'verify the checksum's signature file'. It also mentions 'older versions of Vagrant' and the 'CHANGELOG'. Below this, there are two sections: one for 'Debian' (with 32-bit and 64-bit options) and one for 'Windows' (with 32-bit and 64-bit options).

Downloads

## Download Vagrant

Below are the available downloads for the latest version of Vagrant (2.1.2). Please download the proper package for your operating system and architecture.

You can find the [SHA256 checksums for Vagrant 2.1.2](#) online and you can [verify the checksum's signature file](#), which has been signed using [HashiCorp's GPG key](#). You can also [download older versions of Vagrant](#) from the releases service.

Check out the [v2.1.2 CHANGELOG](#) for information on the latest release.

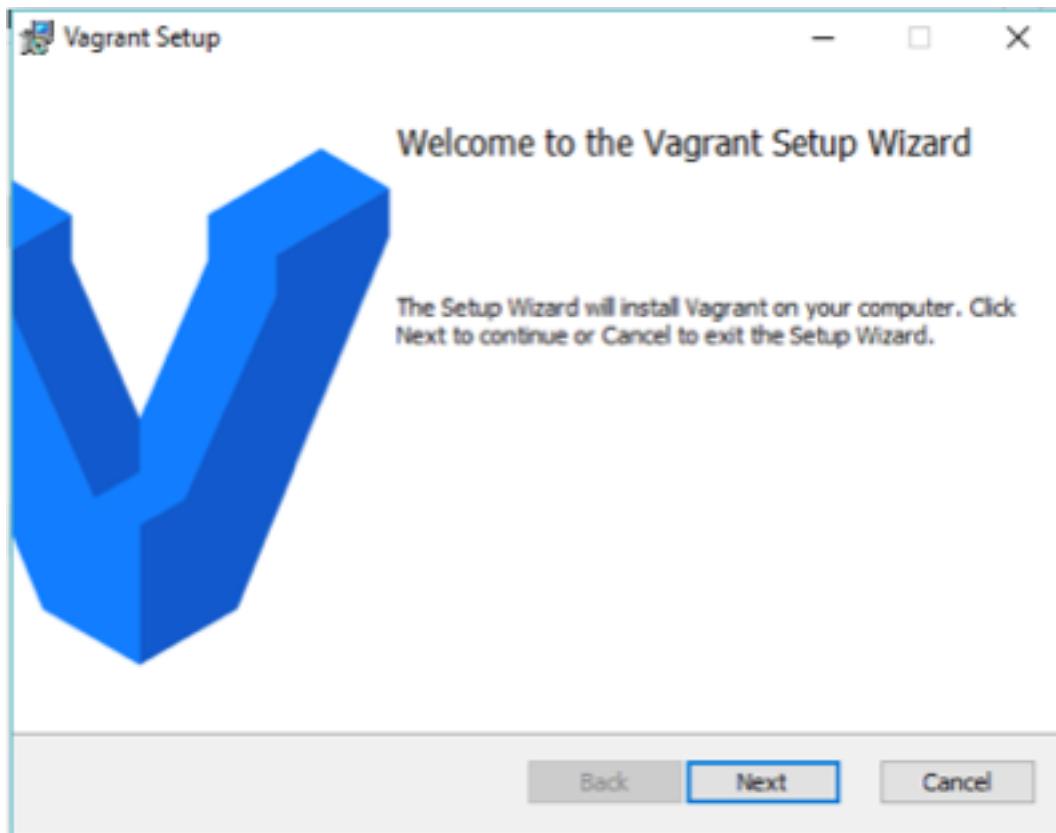
---

 **Debian**  
32-bit | 64-bit

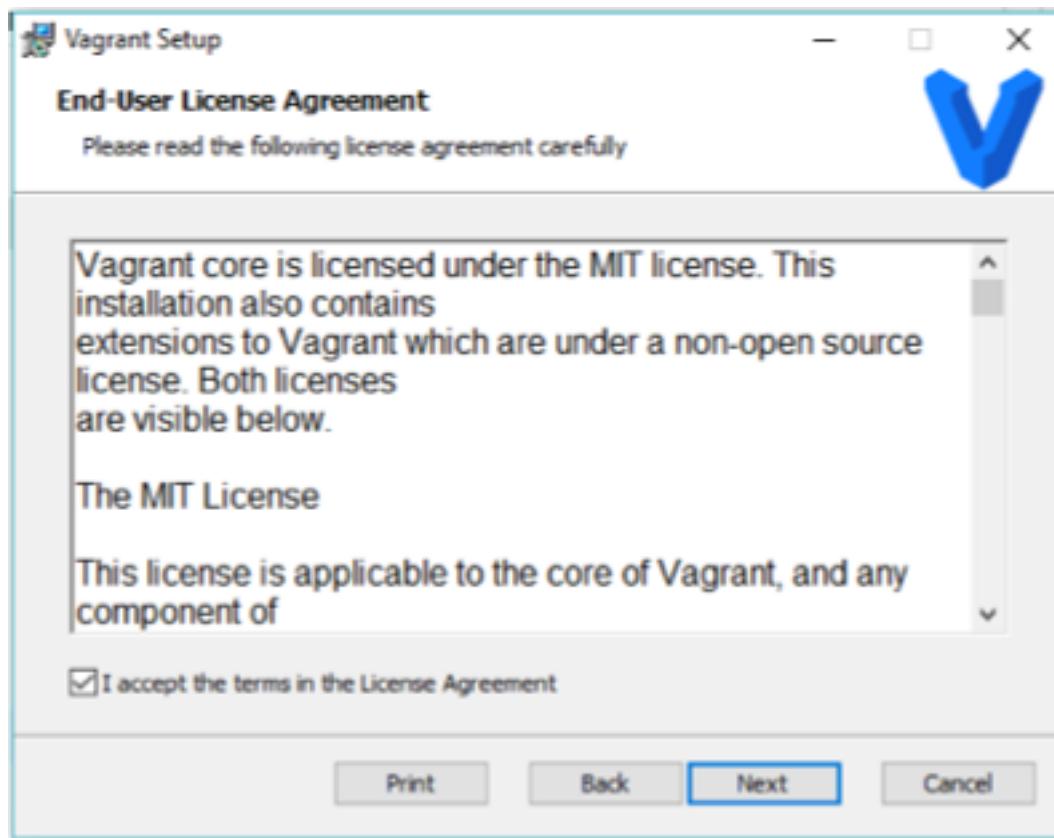
---

 **Windows**  
32-bit | 64-bit

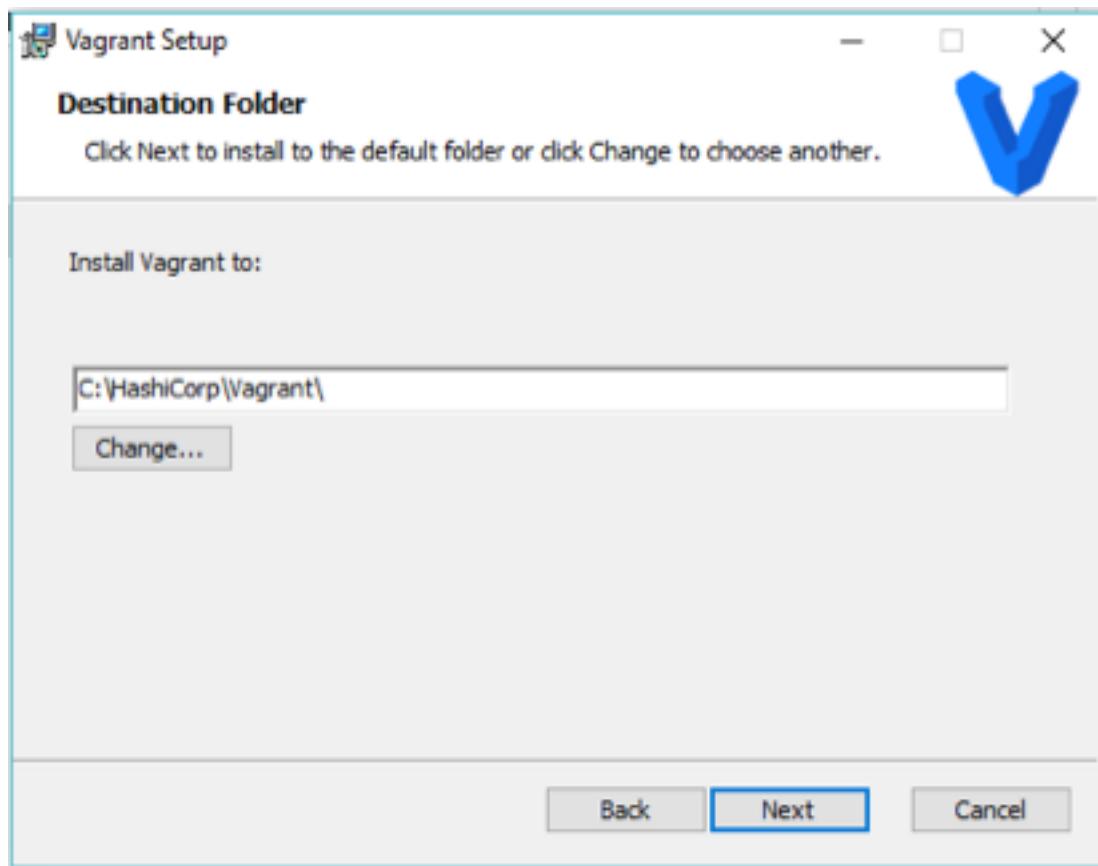
# การติดตั้ง Vagrant



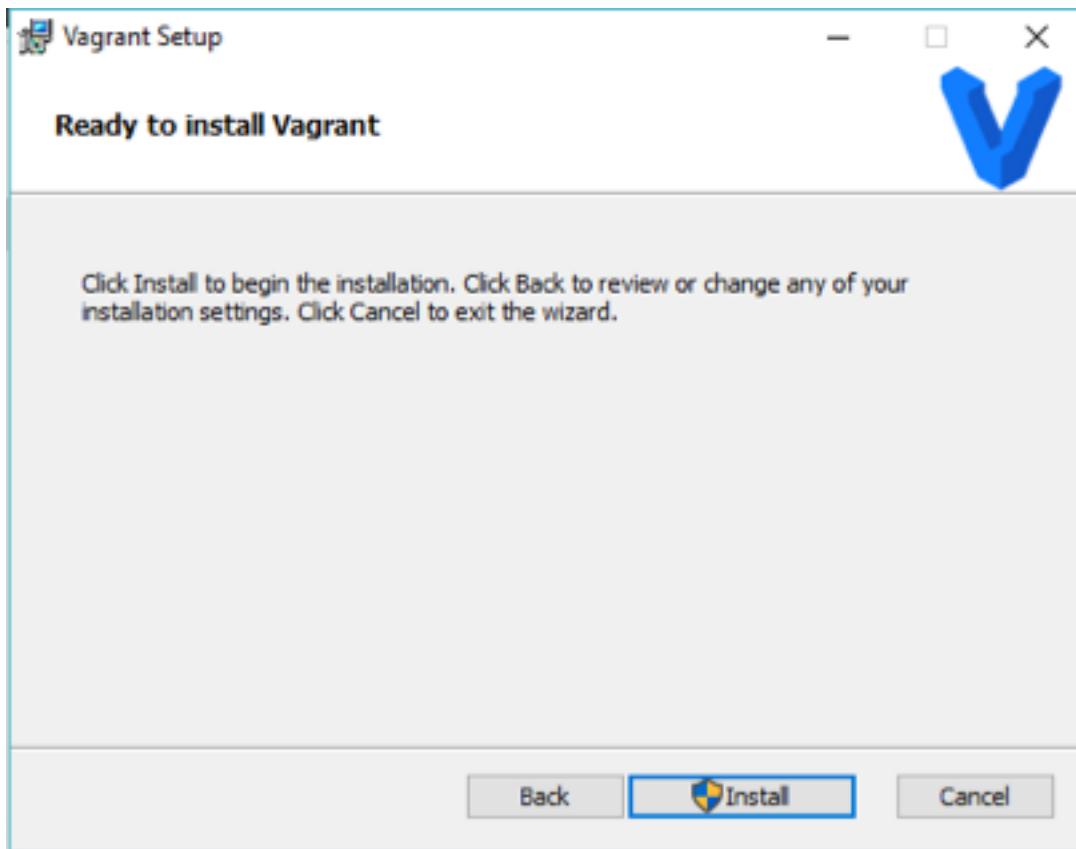
# การติดตั้ง Vagrant



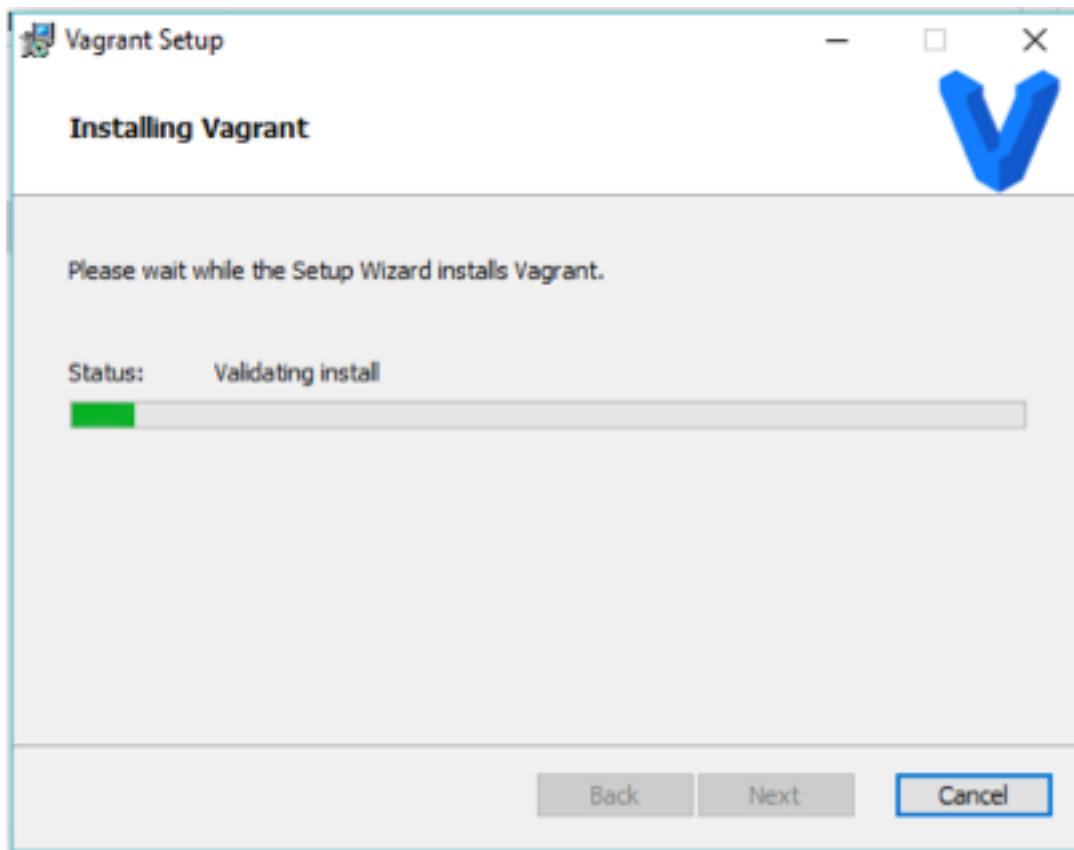
# การติดตั้ง Vagrant



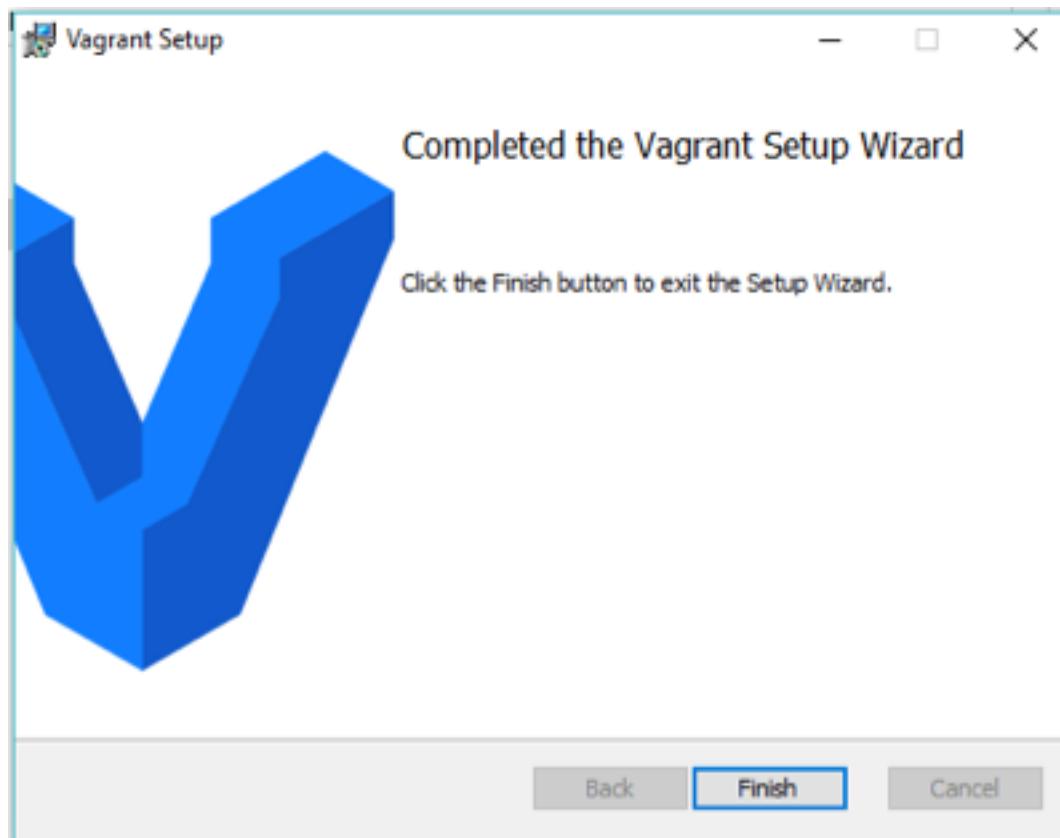
# การติดตั้ง Vagrant



# การติดตั้ง Vagrant



# การติดตั้ง Vagrant



การติดตั้ง

# DOCKER TOOLBOX

# การติดตั้ง Docker Toolbox

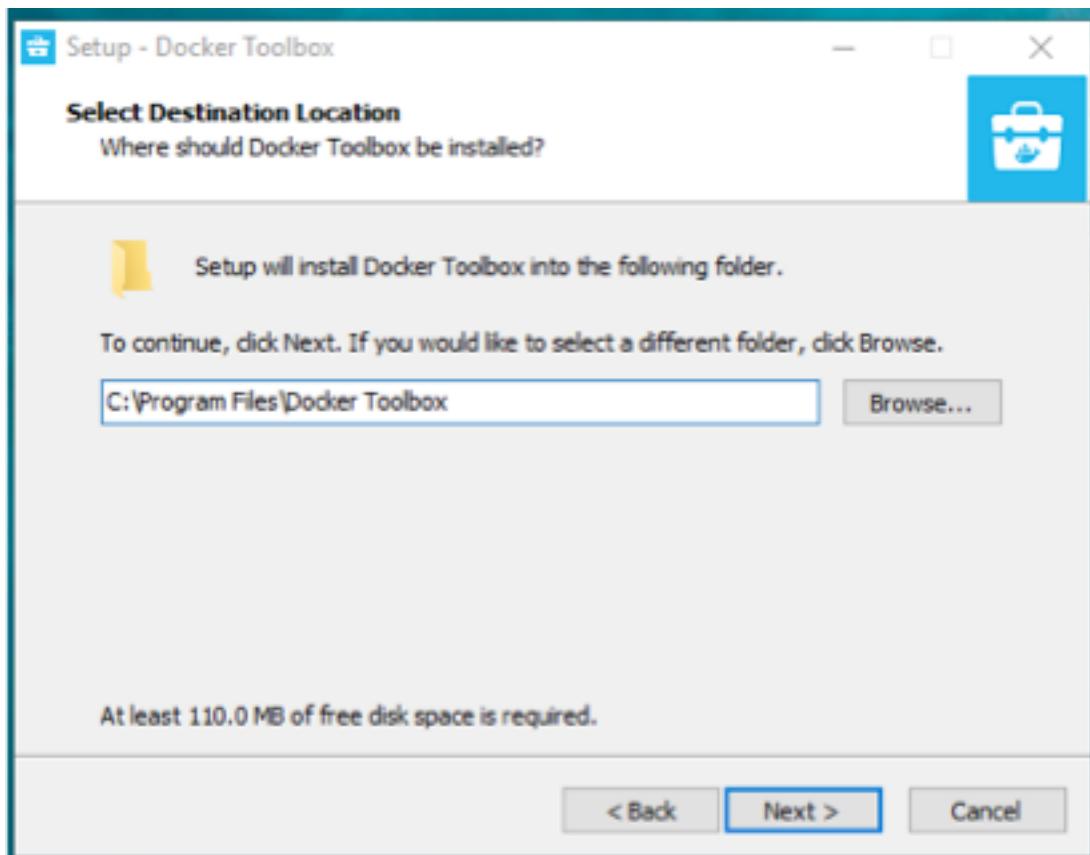
- <https://docs.docker.com/toolbox/overview/#ready-to-get-started>

The screenshot shows the Docker Documentation homepage. The left sidebar has a 'Toolbox overview' section selected. The main content area is titled 'Ready to get started?'. It contains two main steps: 1. Get the latest Toolbox installer for your platform, with 'Toolbox for Mac' and 'Toolbox for Windows' buttons. The 'Toolbox for Mac' button is highlighted with a red box. 2. Choose the install instructions for your platform, and follow the steps: 'Install Docker Toolbox on macOS' and 'Install Docker Toolbox for Windows'. To the right, there's a sidebar with options like 'Edit this page', 'Request docs changes', 'Get support', and 'On this page: What's in the box'. Below the sidebar, there's a 'Ready to get started?' section with a 'Next steps' link.

# การติดตั้ง Docker Toolbox

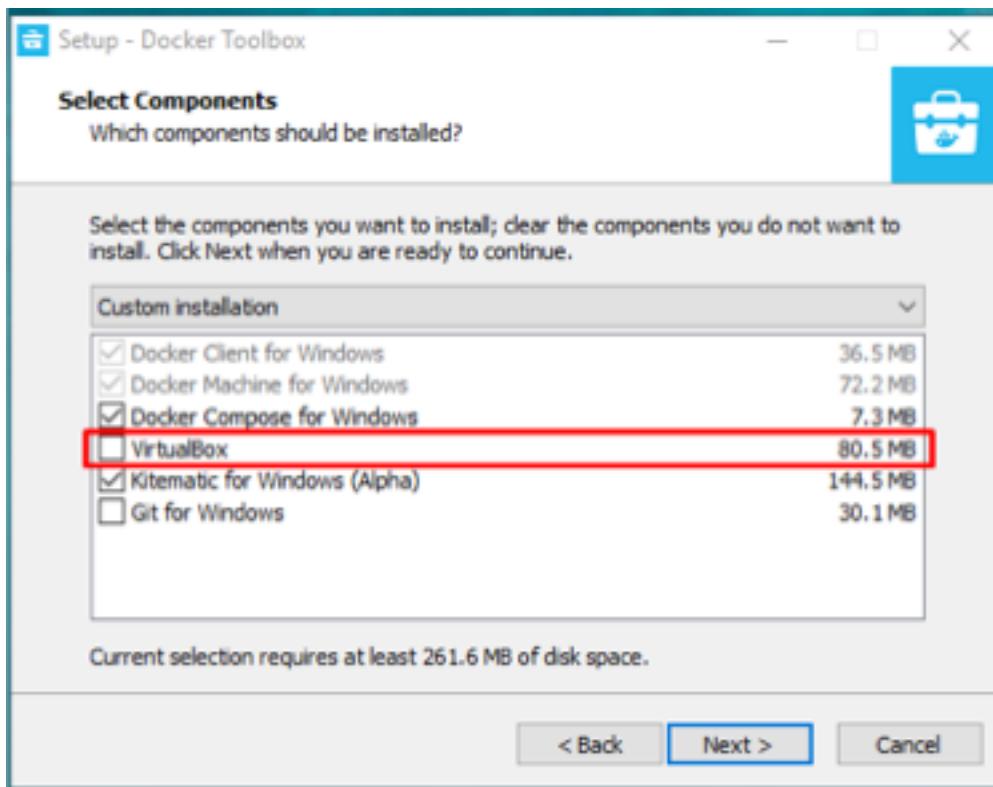


# การติดตั้ง Docker Toolbox

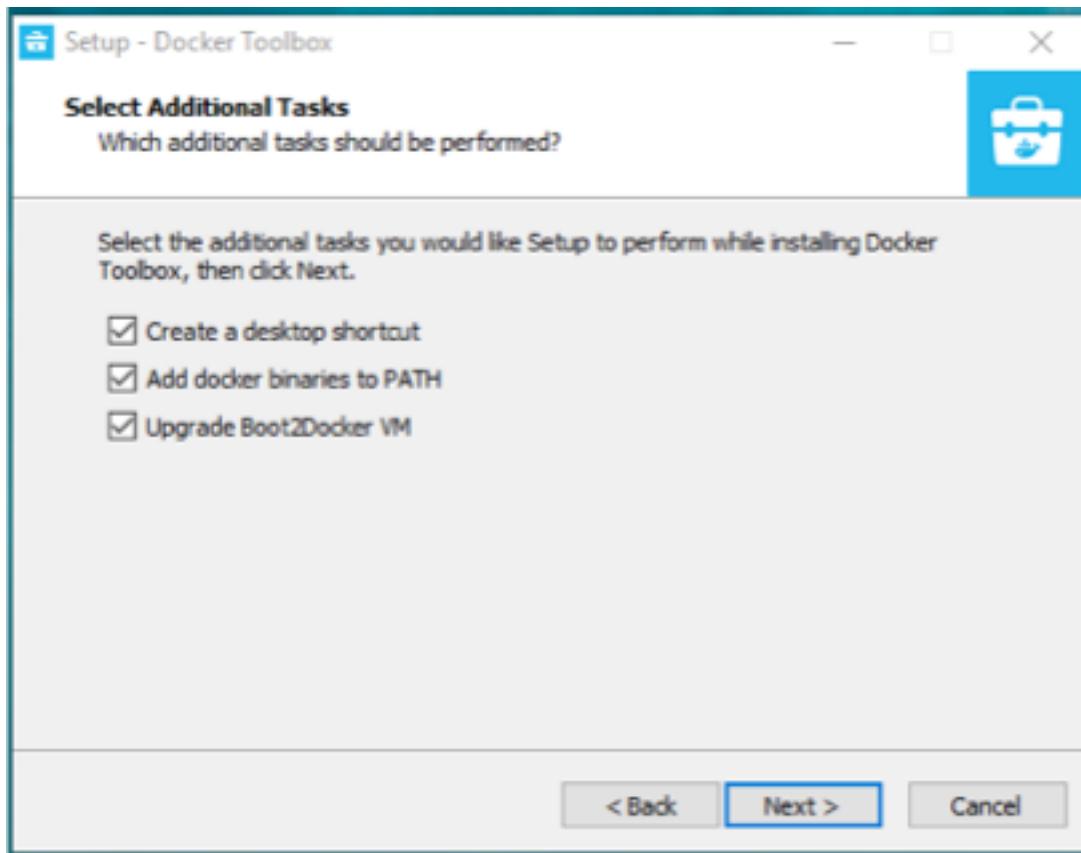


# การติดตั้ง Docker Toolbox

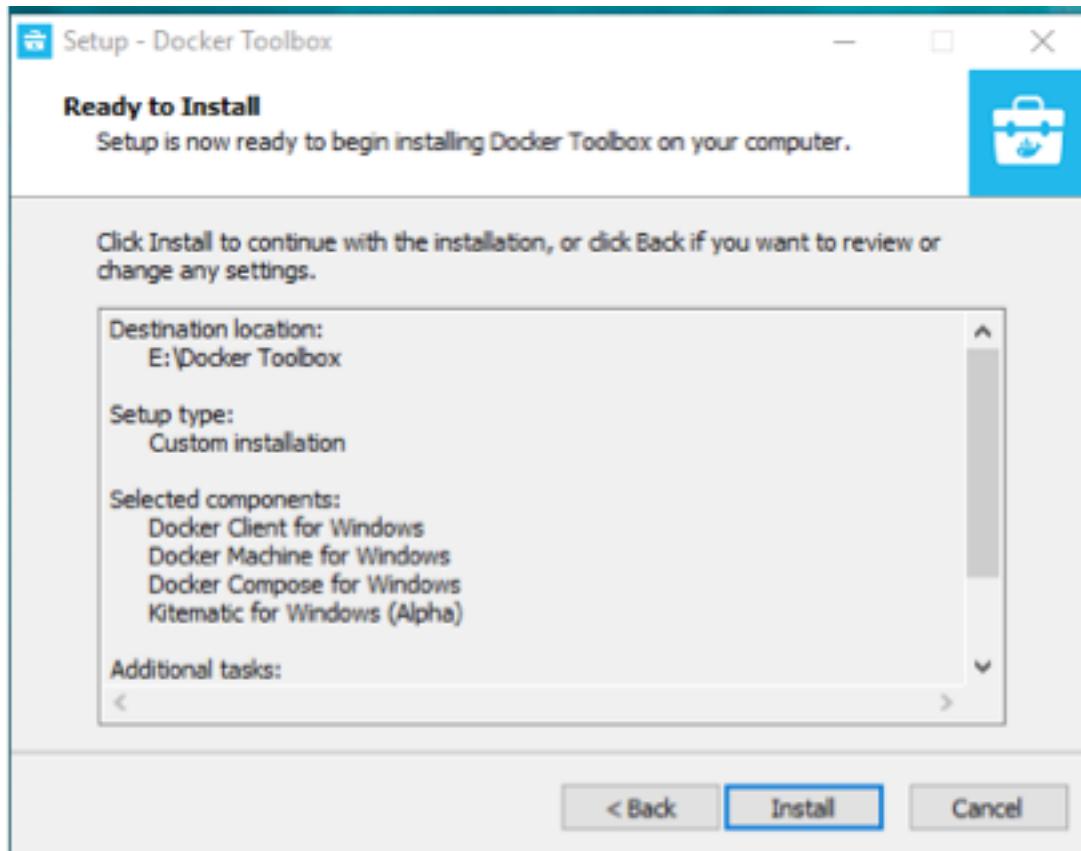
- เลือกเครื่องหมายถูกหน้า VirtualBox ออก



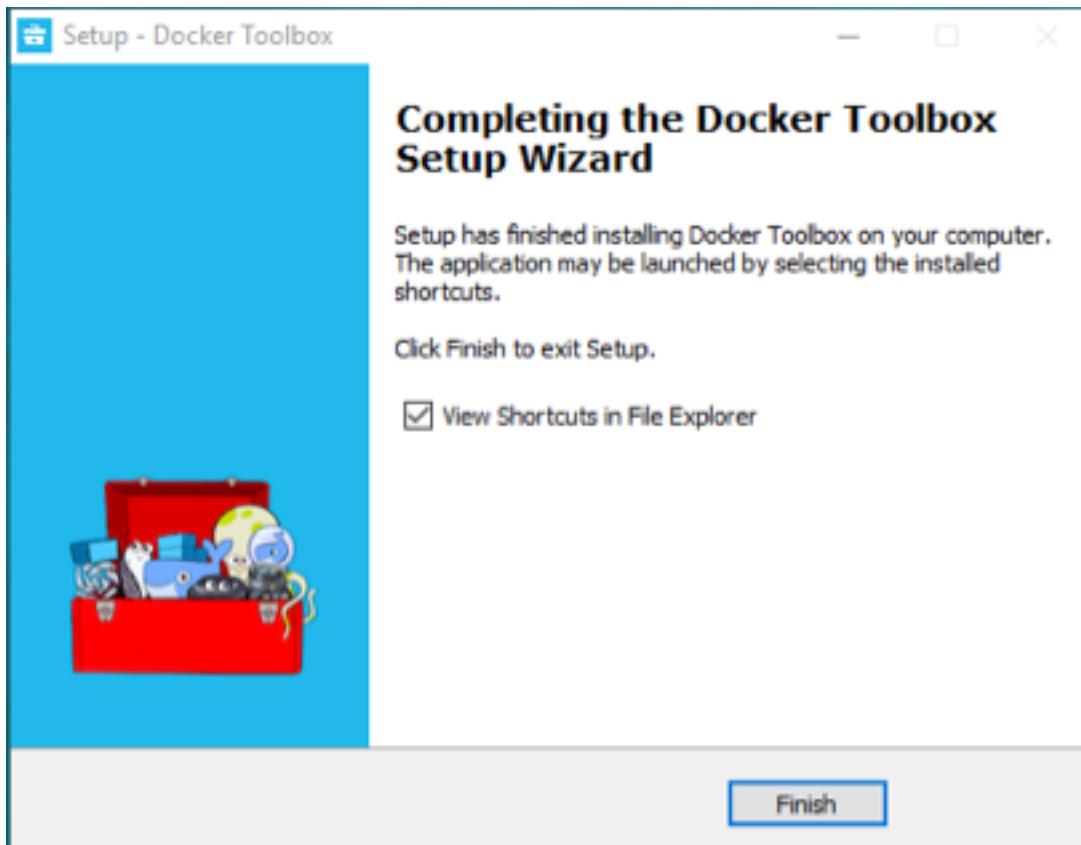
# การติดตั้ง Docker Toolbox



# การติดตั้ง Docker Toolbox



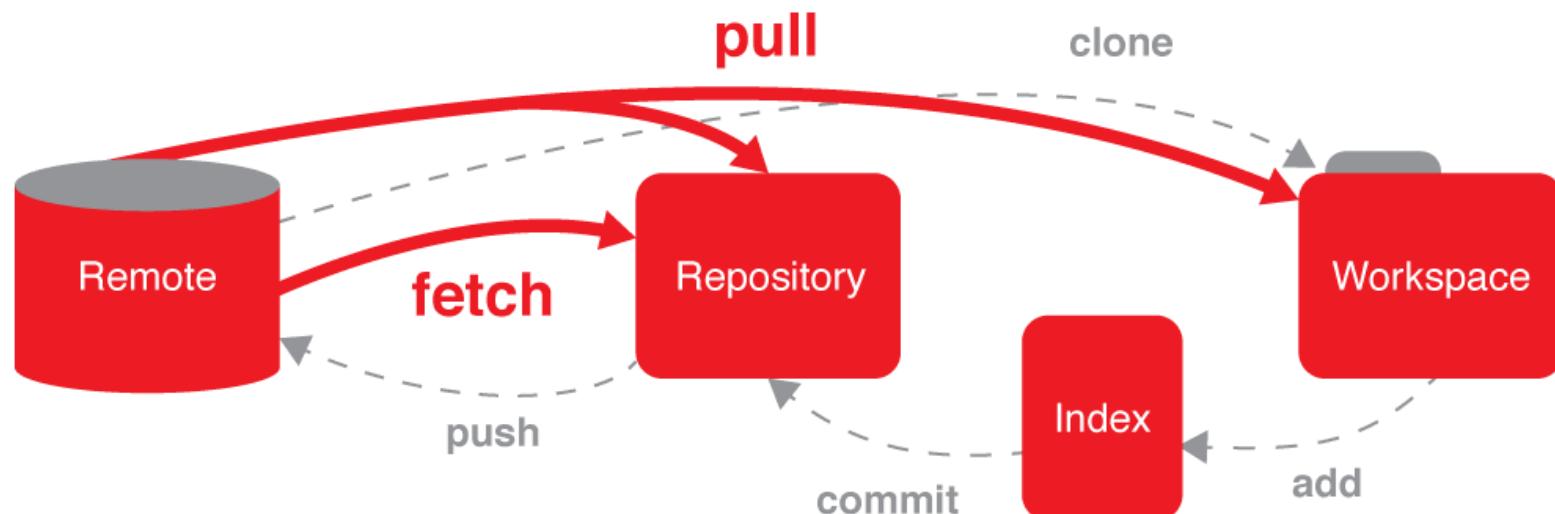
# การติดตั้ง Docker Toolbox



Software version control with

**GIT**

# Overview



# Create a new Git Repository (Remote)

## Create a new repository

A repository contains all the files for your project, including the revision history.

Owner      Repository name

 Sommailk / XXX 

Great repository names are short and memorable. Need inspiration? How about [improved-adventure](#).

Description (optional)

 Public  
Anyone can see this repository. You choose who can commit.

 Private  
You choose who can see and commit to this repository.

Initialize this repository with a README  
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: None  Add a license: None 

**Create repository**

# 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`

# SAVE FRAGMENTS

**Temporarily stores all modified tracked files**

- git stash

**Restores the most recently stashed files**

- git stash pop

**Lists all stashed changesets**

- git stash list

**Discards the most recently stashed changeset**

- git stash drop

# 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

**Downloads all history from the repository bookmark**

- git fetch [bookmark]
- git fetch origin

**Combines bookmark's branch into current local branch**

- git merge [bookmark]/[branch]
- git merge origin/master2

# SYNCHRONIZE CHANGES

**Uploads all local branch commits to Git**

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

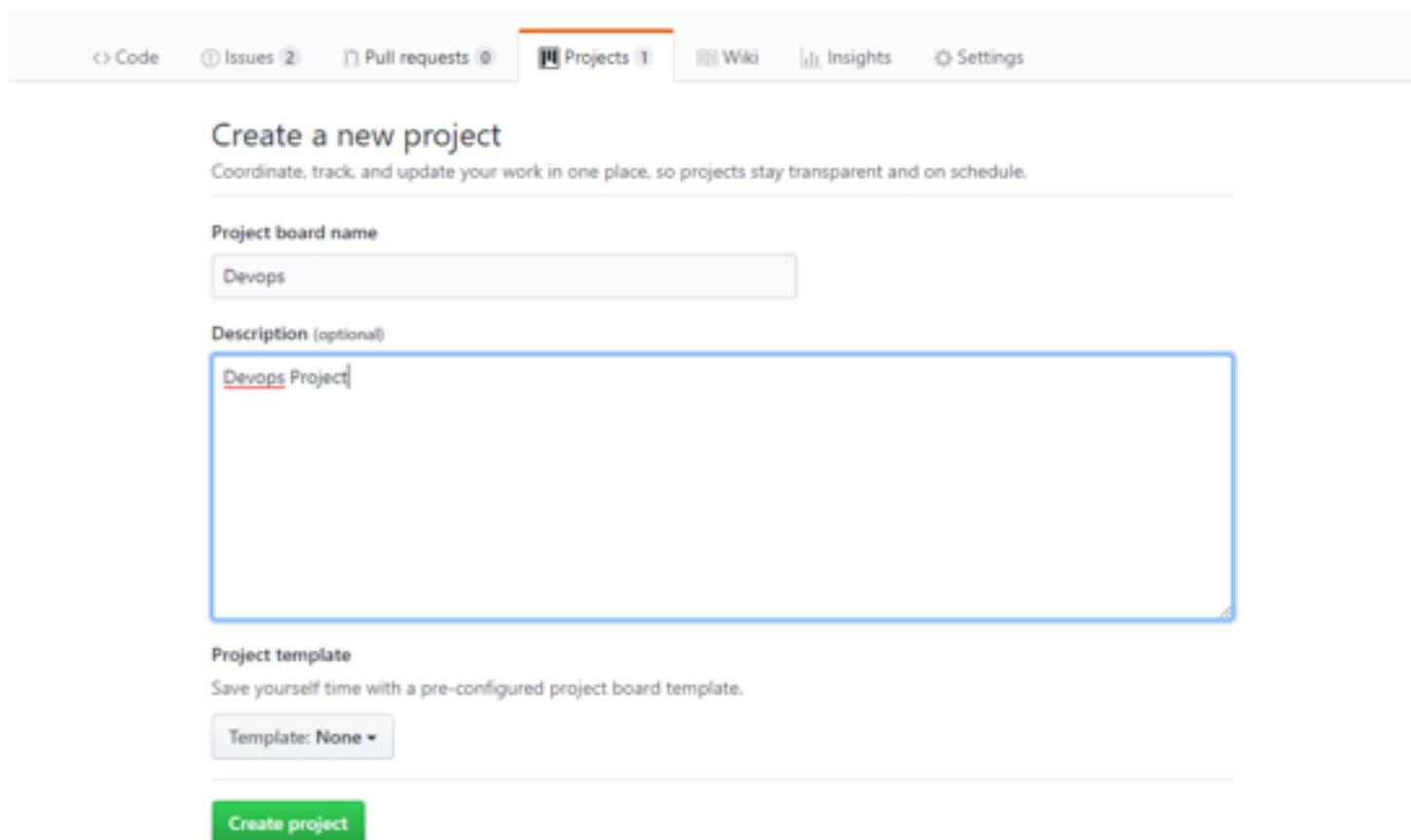
**Downloads bookmark history and incorporates changes**

- git pull

การใช้งาน Issues ใน

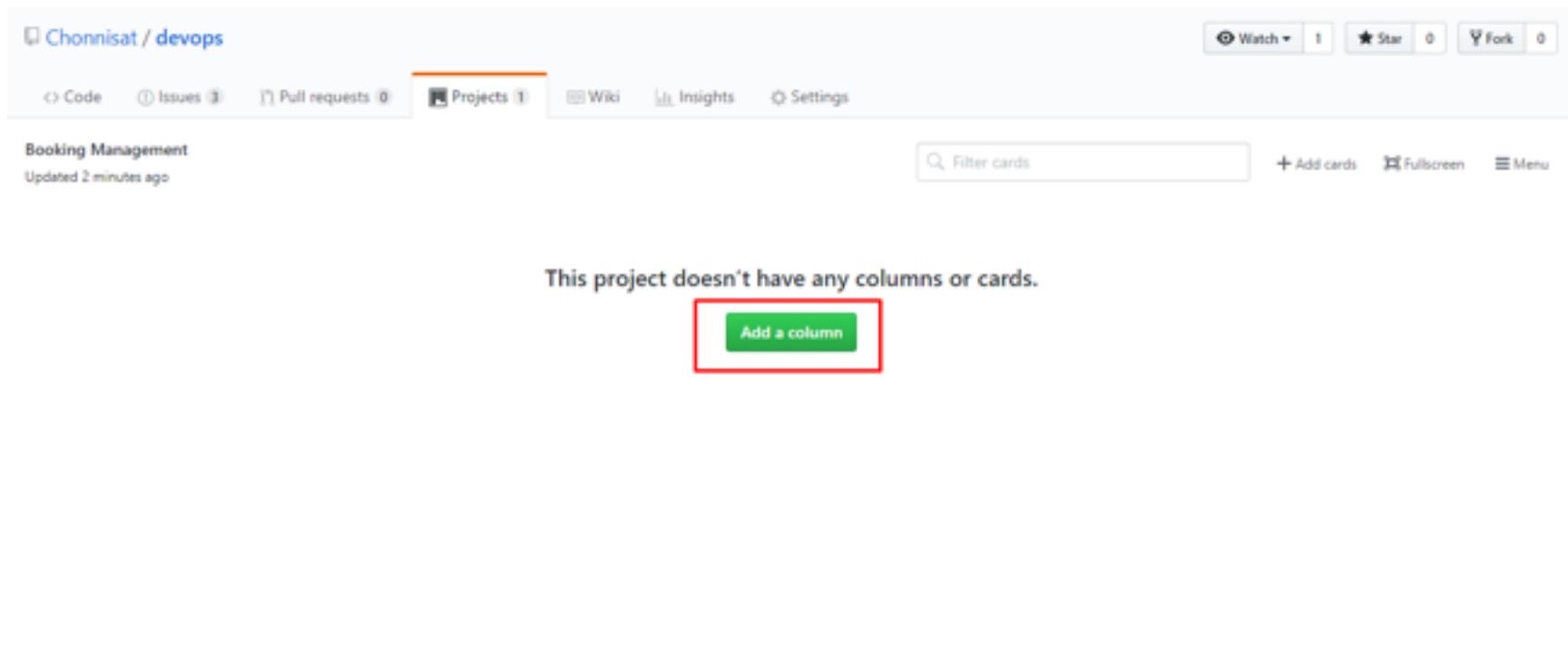
# GITHUB

# New Project in GitHub



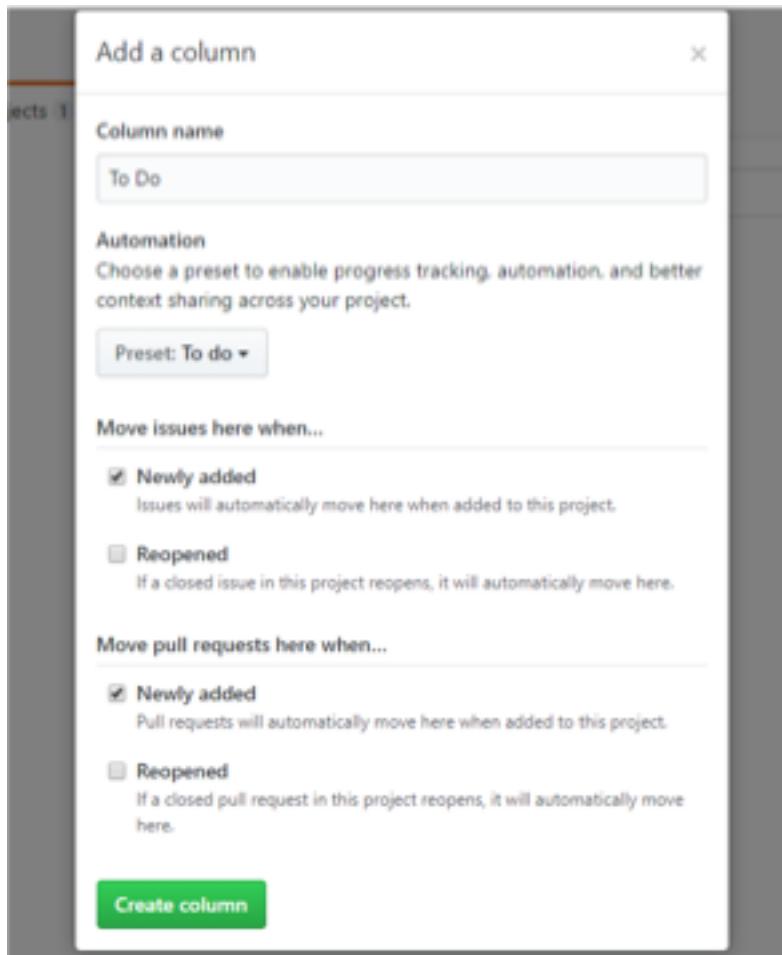
The screenshot shows the GitHub interface for creating a new project. At the top, there is a navigation bar with links: Code, Issues (2), Pull requests (0), Projects (1), Wiki, Insights, and Settings. The Projects link is highlighted with an orange border. Below the navigation bar, the main heading is "Create a new project". A sub-instruction reads: "Coordinate, track, and update your work in one place, so projects stay transparent and on schedule." The first input field is labeled "Project board name" and contains the value "Devops". The second input field is labeled "Description (optional)" and contains the value "Devops Project". This field is highlighted with a blue border. Below these fields is a section titled "Project template" with the sub-instruction "Save yourself time with a pre-configured project board template.". A dropdown menu is open, showing the option "Template: None". At the bottom of the form is a green button labeled "Create project".

# New Project 7u GitHub



The screenshot shows a GitHub project page for 'Chonnisat / devops'. The top navigation bar includes links for Code, Issues (3), Pull requests (0), Projects (1), Wiki, Insights, and Settings. The 'Projects' tab is selected, showing a single card titled 'Booking Management' with the status 'Updated 2 minutes ago'. A search bar labeled 'Filter cards' is present, along with buttons for 'Add cards', 'Fullscreen', and 'Menu'. The main content area displays the message 'This project doesn't have any columns or cards.' with a prominent green 'Add a column' button below it, which is highlighted with a red box.

# New Project in GitHub



# New Project ที่ GitHub

The screenshot shows a GitHub project board for the repository "Chonnisat / devops". The repository name is at the top left, followed by a star count (1), a fork count (0), and a watch button. Below the repository name are navigation links: Code, Issues (2), Pull requests (0), Projects (1), Wiki, Insights, and Settings.

The main area displays a project board for "Booking Management". It has three columns: To Do, In Progress, and Done.

- To Do:** Contains two items:
  - ① Create Master AirCraft #3 opened by Chonnisat enhancement
  - ② Create Master Airport #2 opened by Chonnisat enhancement
- In Progress:** Contains one item:
  - ① สร้างหน้าร้านขายตั๋วเครื่องบินด้วย Promotion Code #4 opened by Chonnisat bug
- Done:** Contains one item:
  - ① New Schedule #1 opened by Chonnisat good first issue

Each card includes a "More options" menu icon (three dots). A search bar at the top right says "Filter cards". On the right side, there are buttons for "Add cards", "Fullscreen", and "Menu". A "Add column" button is also present.

# การใช้งาน Issue ใน GitHub

- Login เข้า GitHub และเลือก tab Issues

The screenshot shows the GitHub interface for the repository 'Chonnisat / devops'. The top navigation bar includes 'Pull requests', 'Issues', 'Marketplace', and 'Explore'. Below the repository name, there are buttons for 'Watch', 'Star', and 'Fork'. The 'Issues' tab is currently selected, showing 0 open issues. A search bar displays the filter 'is:issue is:open'. On the right, there is a green 'New issue' button. At the bottom, there are filters for 'Author', 'Labels', 'Projects', 'Milestones', 'Assignee', and 'Sort', along with a note that 0 Open and 1 Closed issues exist.

There aren't any open issues.

Use the links above to find what you're looking for, or try a [new search query](#). The Filters menu is also super helpful for quickly finding issues most relevant to you.

# การใช้งาน Issue ใน GitHub

- สามารถเพิ่ม Label ตามที่ต้องการได้

The screenshot shows the GitHub interface for the repository 'Chonnisat / devops'. The 'Issues' tab is selected, showing 0 issues. A search bar for labels is present. Below it is a table listing eight labels:

Label	Description	Edit	Delete
bug	Something isn't working		
duplicate	This issue or pull request already exists		
enhancement	New feature or request		
good first issue	Good for newcomers		
help wanted	Extra attention is needed		
invalid	This doesn't seem right		
question	Further information is requested		

A green button labeled 'New label' is located at the top right of the label list area, with a red box drawn around it.

# การใช้งาน Issue ใน GitHub

The screenshot shows the GitHub Labels interface. At the top, there is a search bar labeled "Search all labels", a "Labels" button (which is currently selected), a "Milestones" button, and a green "New label" button. Below this, a modal window is open for creating a new label named "pending". The modal includes fields for "Label name" (containing "pending"), "Description" (containing "Suspend this issue"), and "Color" (set to a pink shade). There are also "Cancel" and "Create label" buttons. A red box highlights this modal window. Below the modal, a list of existing labels is displayed:

Label	Description	Edit	Delete
bug	Something isn't working		
duplicate	This issue or pull request already exists		
enhancement	New feature or request		
good first issue	Good for newcomers		
help wanted	Extra attention is needed		
invalid	This doesn't seem right		
question	Further information is requested		
wontfix	This will not be worked on		

# การใช้งาน Issue ใน GitHub

- สามารถเพิ่ม Milestones ตามที่ต้องการได้

The screenshot shows the GitHub interface for the repository 'Chonnisat / devops'. The 'Issues' tab is selected, showing 0 issues. A red box highlights the green 'New milestone' button in the top right corner of the main content area. Below the button, there are filters for 'Open' (1) and 'Closed' (0) issues, and a 'Sort' dropdown. The 'Schedule Function' section displays a progress bar at 100% complete, with 0 open and 1 closed issue. At the bottom, there are links for 'Edit', 'Close', and 'Delete'.

© 2018 GitHub, Inc. Terms Privacy Security Status Help

Contact GitHub API Training Shop Blog About

# การใช้งาน Issue ใน GitHub

## New milestone

Create a new milestone to help organize your issues and pull requests. Learn more about [milestones and issues](#).

Title

Due date (optional)

Description

**Create milestone**

# การใช้งาน Issue ใน GitHub

The screenshot shows the GitHub interface for the repository 'Chonnisat / devops'. The 'Issues' tab is selected, showing 1 open issue. A milestone titled 'Master Program' is visible, due by July 6, 2018, with 0% completion and 0 open/closed issues. Navigation links include Code, Issues (selected), Pull requests, Projects, Wiki, Insights, Settings, Labels, and Milestones. A green button for 'New milestone' is also present.

Chonnisat / devops

Code Issues (1) Pull requests Projects Wiki Insights Settings

Labels Milestones New milestone

1 Open 0 Closed Sort ▾

**Master Program**  
Due by July 6, 2018 Last updated less than a minute ago  
Master ที่จะมาดูในรุ่น Setup User  
0% complete 0 open 0 closed  
Edit Close Delete

# การใช้งาน Issue ใน GitHub

- New Issues

The screenshot shows the GitHub interface for the repository 'Chonnisat / devops'. The 'Issues' tab is active. At the top right, there are buttons for 'Watch', 'Star', 'Fork', and a green 'New issue' button, which is highlighted with a red box. Below the tabs, there are filters for 'Filters', a search bar containing 'is:issue is:open', and buttons for 'Labels' and 'Milestones'. The main content area displays a message: 'There aren't any open issues.' with an exclamation mark icon. It also includes a 'ProTip!' message: 'Find everything you created by searching author:Chonnisat.' At the bottom left, there is a search bar with the text 'chonnisat/devops'.

# การใช้งาน Issue ใน GitHub

The screenshot shows a GitHub repository named 'Chonnisat / devops'. The 'Issues' tab is selected, showing 0 issues. A new issue is being created with the title 'Create Master Airport'. The issue body is currently empty. On the right side, there are fields for 'Assignees' (Sommalk), 'Labels' (enhancement, highlighted in green), 'Projects' (Booking Management), and 'Milestone' (Master Program). A note at the bottom left says 'Styling with Markdown is supported'. A 'Submit new issue' button is at the bottom right.

# การใช้งาน Issue ใน GitHub

The screenshot shows the GitHub Issues page for the repository Chonnisat / devops. The page has a dark theme. At the top, there is a search bar, navigation links for Pull requests, Issues, Marketplace, and Explore, and user profile icons. Below the header, the repository name 'Chonnisat / devops' is displayed, along with 'Watch 1', 'Star 0', and 'Fork 0'. The 'Issues' tab is selected, showing 3 open issues and 1 closed issue. The issues listed are:

- #4 ค่านาฬิกาไม่ถูกต้องในกรณีใส่ Promotion Code bug
- #3 Create Master AirCraft enhancement
- #2 Create Master Airport enhancement

Each issue card includes a small profile picture of the author, the issue number, title, type, and a link to view the details. A 'New issue' button is located at the top right of the issue list area. A pro tip at the bottom suggests adding 'noassignee' to see unassigned issues.

# การใช้งาน Issue ใน GitHub

- เมื่อมีการ Close Issues หน้า Milestones จะแสดงความคืบหน้าของ Milestones นั้นๆ

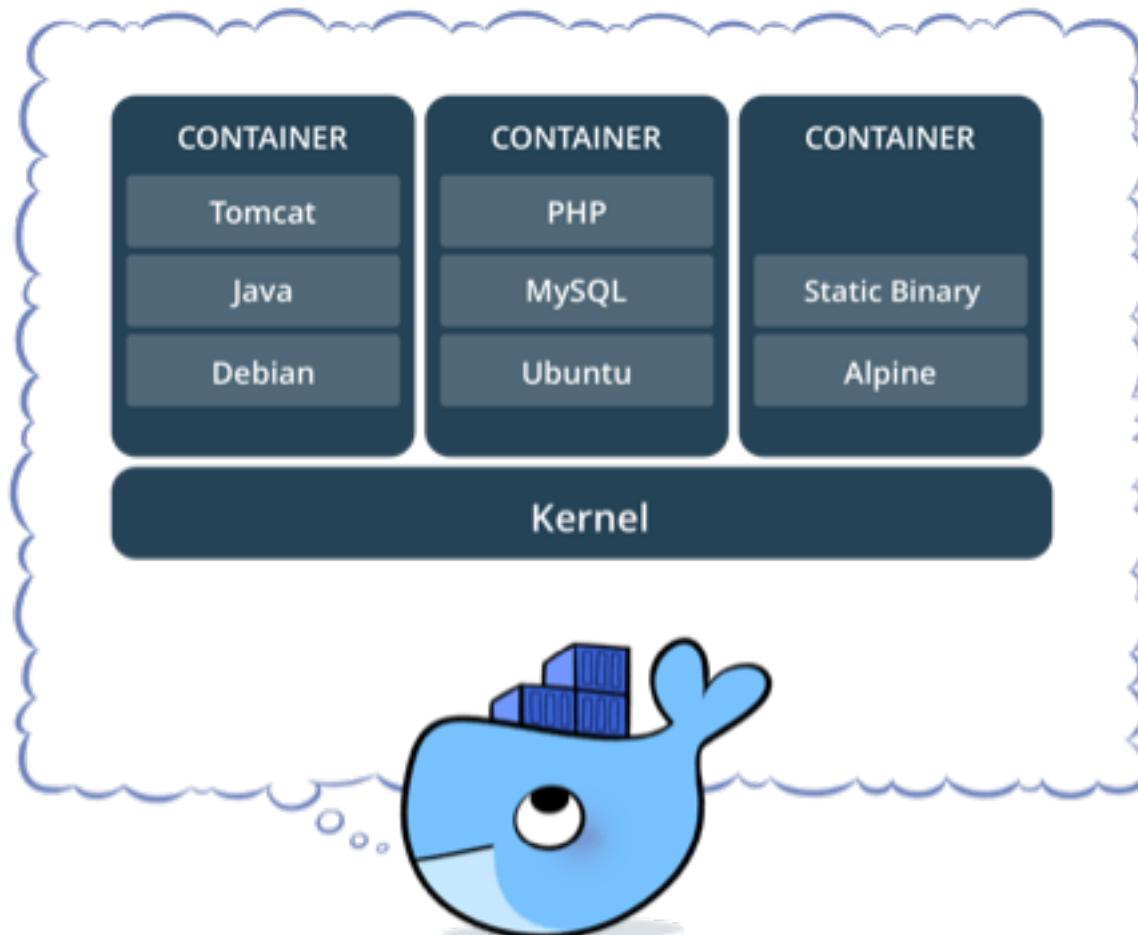
The screenshot shows the GitHub Issues page for the repository 'Chonnisat / devops'. The 'Issues' tab is selected, showing 2 open issues. Two milestones are listed:

- Master Program**: Due by July 6, 2018. Last updated less than a minute ago. Status: 50% complete (1 open, 1 closed). Actions: Edit, Close, Delete.
- Fix Bug**: Due by July 5, 2018. Last updated 8 minutes ago. Status: 0% complete (1 open, 0 closed). Actions: Edit, Close, Delete.

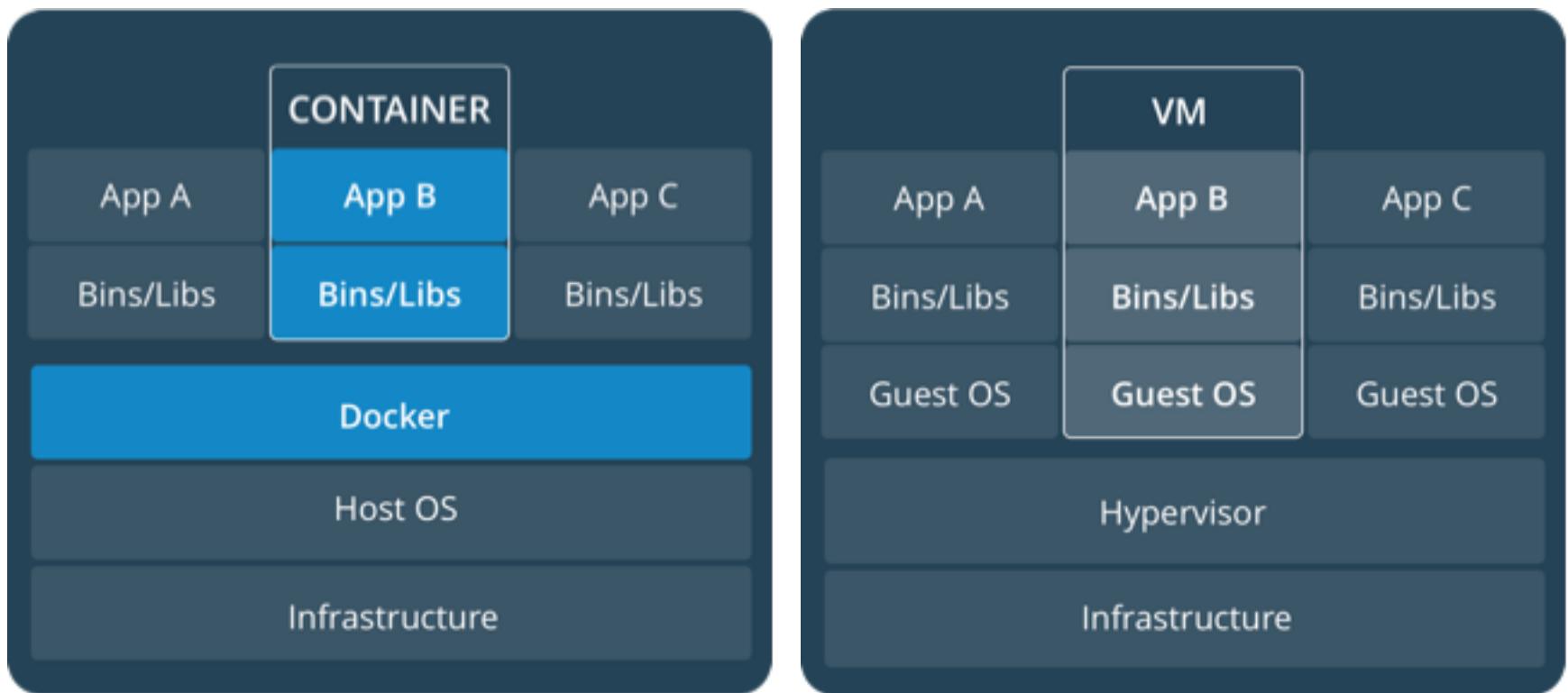
Containers virtualize with

**DOCKER**

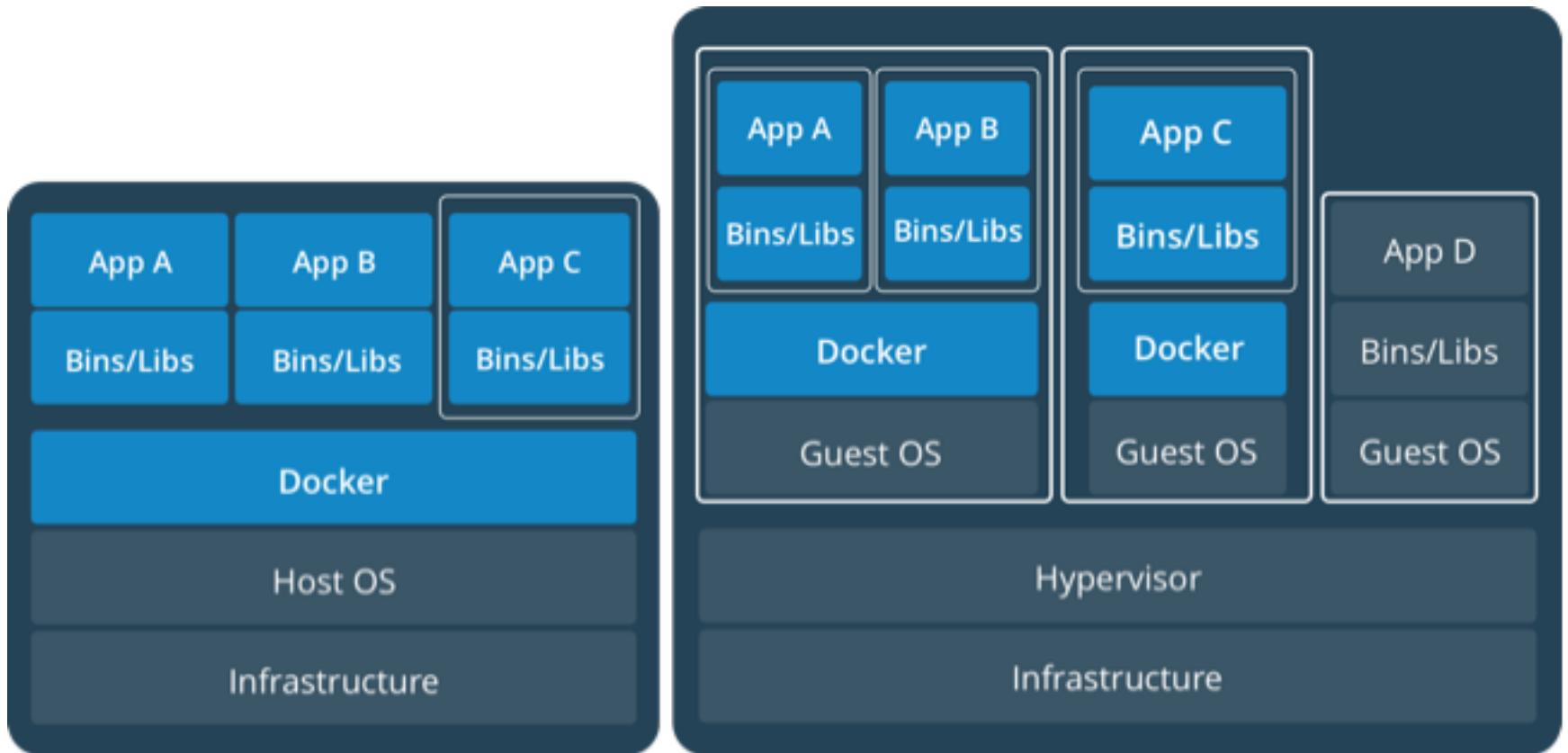
# About Container



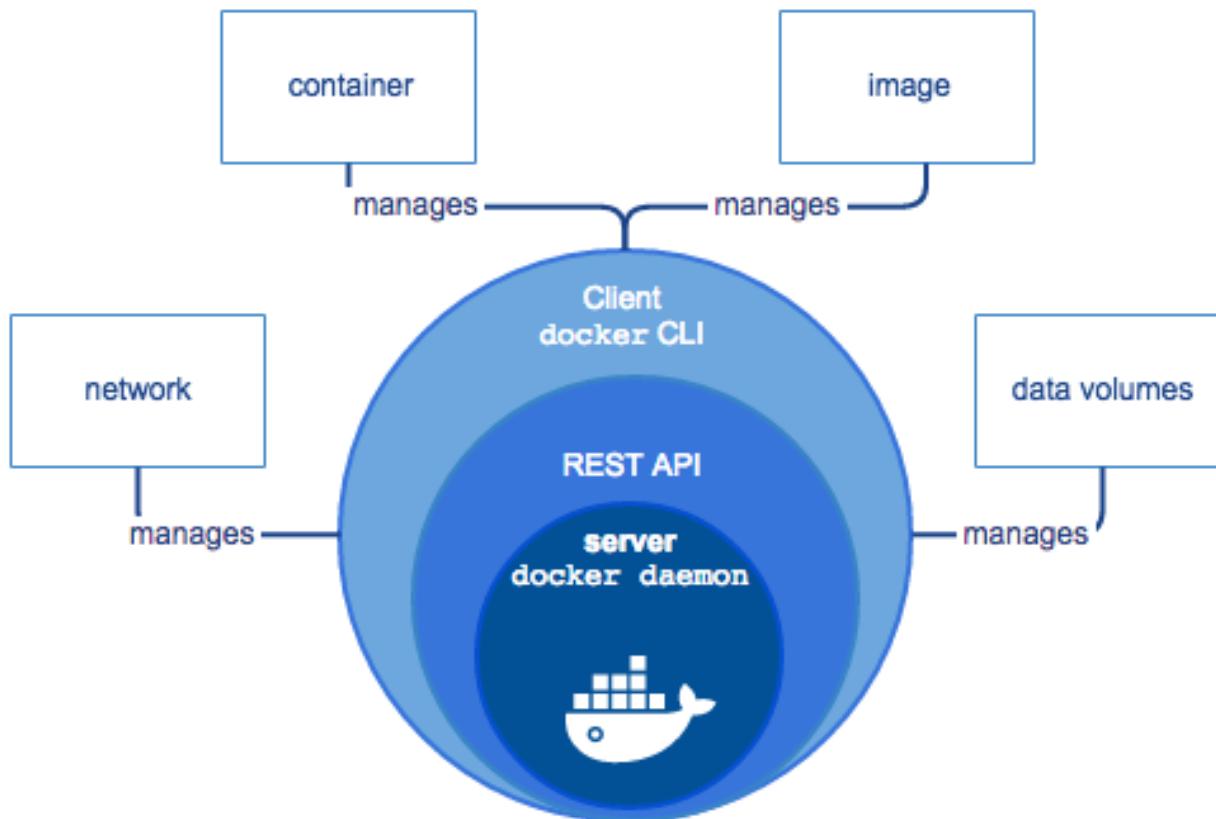
# Comparing Containers and Virtual Machines



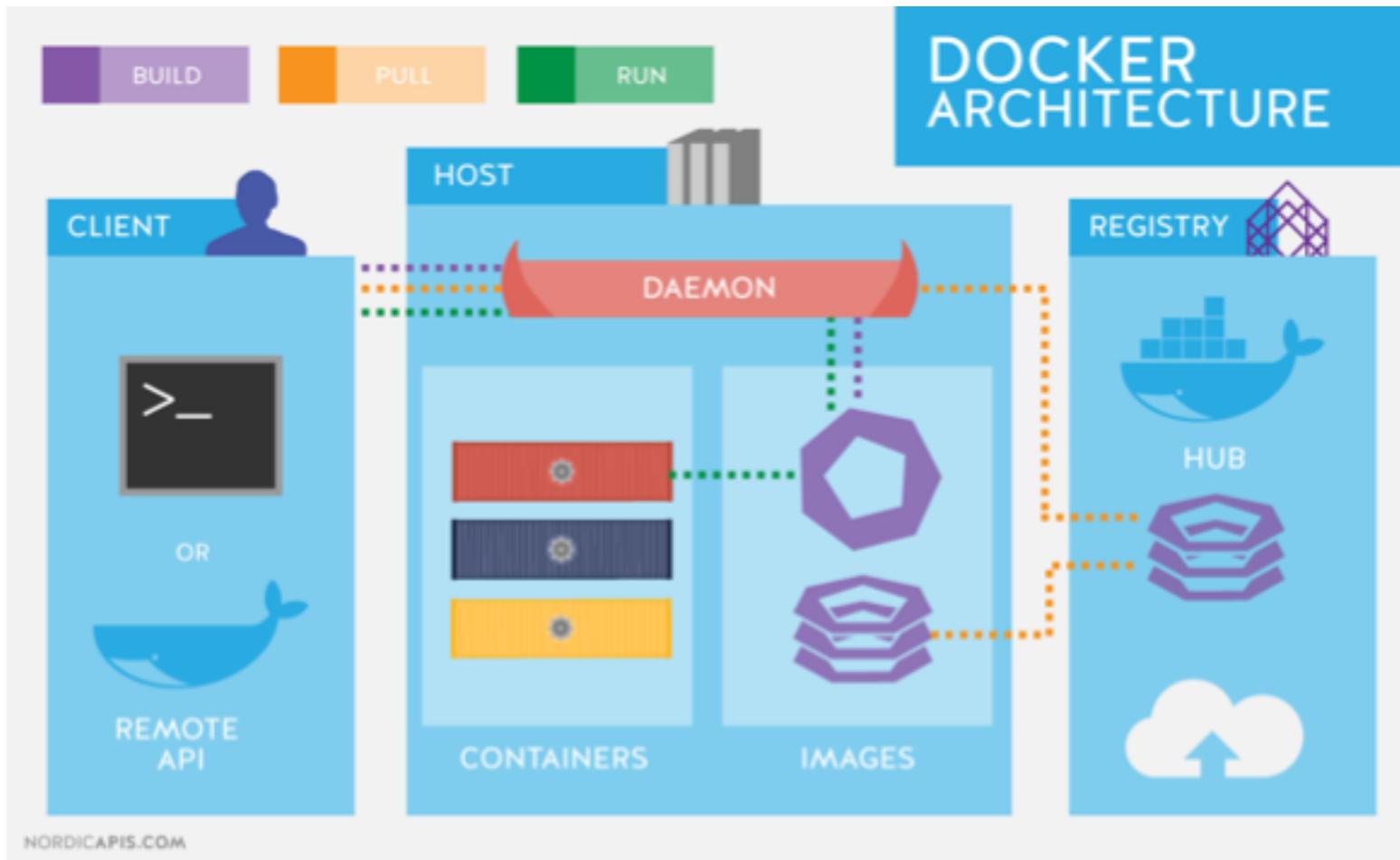
# Containers and Virtual Machines Together



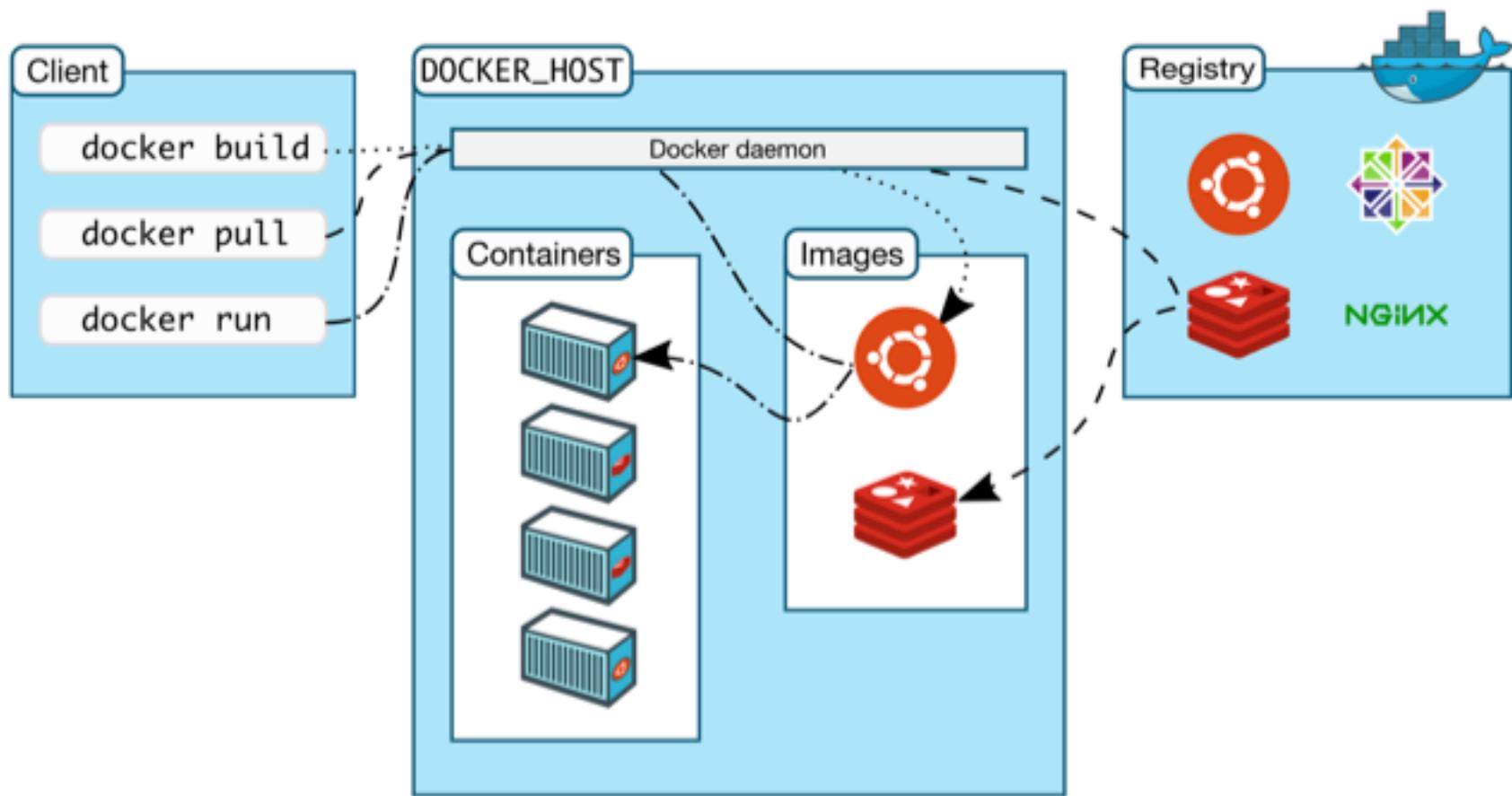
# Docker engine



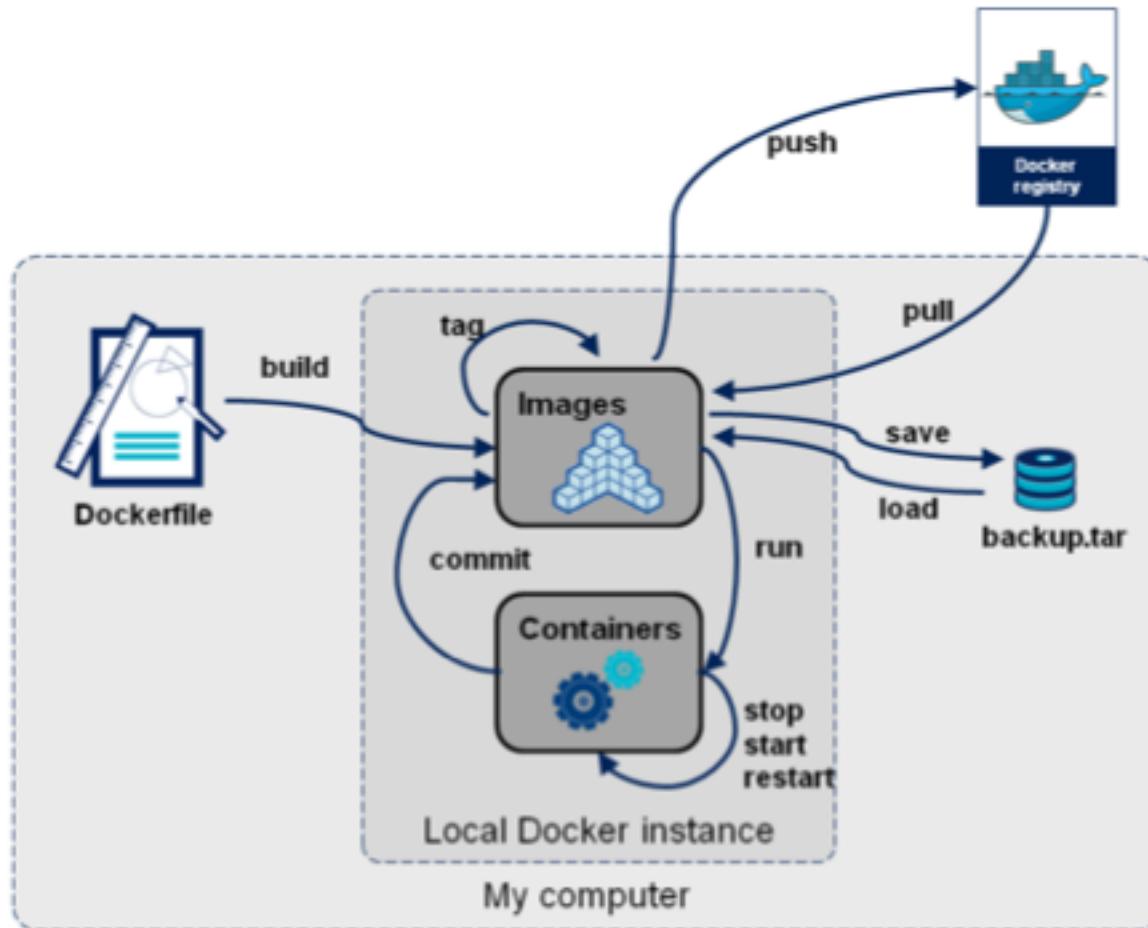
# Docker Architecture



# Docker Architecture#2



# Docker Architecture#3



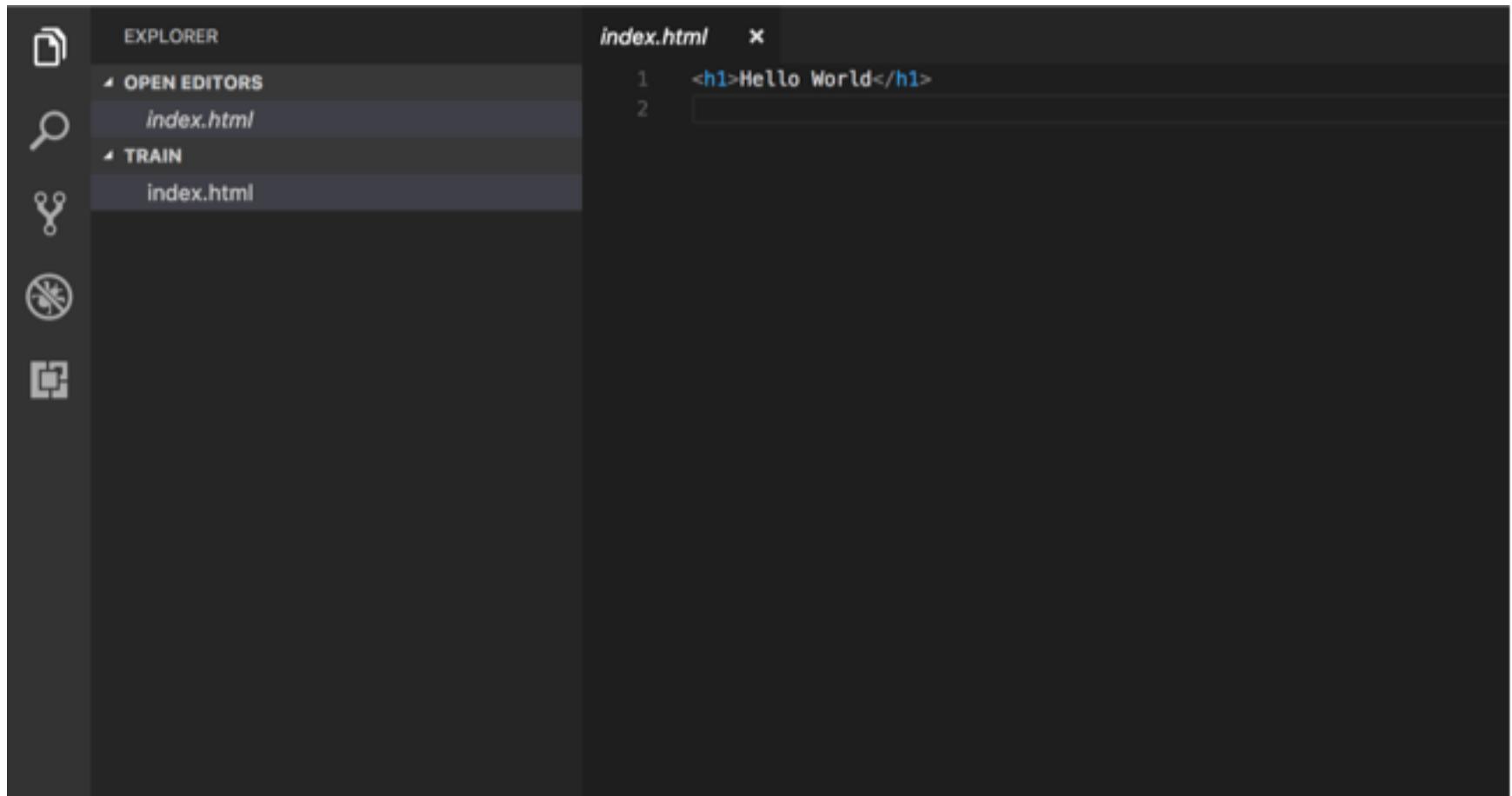
# Hello World Docker

- เปิด cmd และรันคำสั่งดังนี้

```
docker run --name some-nginx \
-v /your_path:/usr/share/nginx/html:ro \
-p 80:80 \
-d nginx
```

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

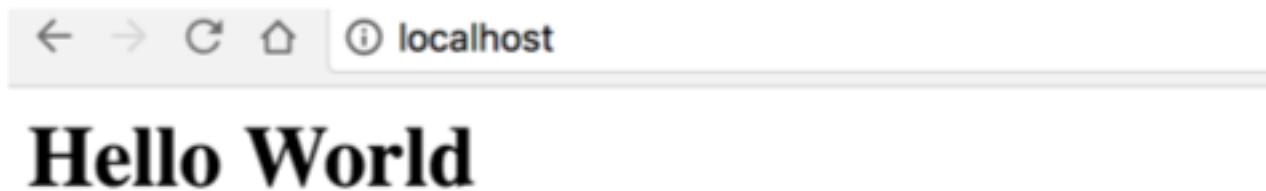
# สร้าง file index.html



The screenshot shows a dark-themed code editor interface. On the left is a vertical toolbar with icons for file operations like Open, Save, Find, and Copy/Paste. To its right is the Explorer sidebar, which lists 'OPEN EDITORS' containing 'index.html' and 'TRAIN' also containing 'index.html'. The main workspace is titled 'index.html' and contains the following code:

```
1  <h1>Hello World</h1>
2
```

# เข้า url `http://localhost`



# 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>

# 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

# Dockerfile

## FROM

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

## RUN

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

## EXPOSE

- EXPOSE <port>
- EXPOSE 8080

# 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 –build
- docker-compose up --force-recreate
- docker-compose ps
- docker-compose scale web=5
- docker-compose stop
- docker-compose rm

Automate Configuration with

# VAGRANT

# Introduction

- Vagrant - the command line utility for managing the lifecycle of virtual machines
- Vagrant is a tool for building and managing virtual machine environments in a single workflow
- With an easy-to-use workflow and focus on automation, Vagrant lowers development environment setup time, increases production parity, and makes the "**works on my machine**" excuse a relic of the past.

# Why Vagrant

- Vagrant provides easy to configure, reproducible, and portable work environments built on top of industry-standard technology and controlled by a single consistent workflow to help maximize the productivity and flexibility of you and your team.
- Vagrant gives you a disposable environment and consistent workflow for developing and testing infrastructure management scripts.
- Vagrant is designed for everyone as the easiest and fastest way to create a virtualized environment!

# Project Setup

- Run command
  - mkdir `vagrant_getting_started`
  - cd `vagrant_getting_started`
  - `vagrant init`
- \*\*\* `vagrant_getting_started` គឺជោងទីសម្រាប់ការរាយការណ៍ project

# Boxes

- ຕັດຕັ້ງ image (box)
  - vagrant box add **ubuntu/xenial64**

\*\*\* ສາມາດຮັບ image(box) ໄດ້ຈີ່ website  
<https://app.vagrantup.com/boxes/search>

# Using a box

- Open Vagrantfile add following command

```
Vagrant.configure("2") do |config|
  config.vm.box = "ubuntu/xenial64"
end
```

\*\*\* config คือ ชื่อตัวแปรเอาไว้อ้างอิงใน script สามารถตั้งชื่ออะไรก็ได้

\*\*\* เลข 2 หมายถึง version configure ของ vagrant ที่จะใช้

# Up And SSH

- Run command for start vm
    - vagrant up <NAME>
  - Run command for remote to vm
    - vagrant ssh <NAME>
- \*\*\* <NAME> ใช้ในกรณีที่มีการ manage มากกว่า 1 vm

# Synced Folders

- Vagrant will automatically sync your files to and from the guest machine.
- `/vagrant` is synced directory
- Run command
  - `vagrant up`
  - `vagrant ssh`
  - `ls /vagrant`

## Output

Vagrantfile

# Provisioning

- Provisioners in Vagrant allow you to automatically install software, alter configurations, and more on the machine as part of the vagrant up process.

```
Vagrant.configure("2") do |config|
  # ... other configuration

  config.vm.provision "shell", inline: "echo hello"
end
```

# Shell Provisioner

## SCRIPTING

```
$script = <<-SCRIPT
echo I am provisioning...
date > /etc/vagrant_provisioned_at
SCRIPT

Vagrant.configure("2") do |config|
  config.vm.provision "shell", inline: $script
end
```

# Multi-Machine

- Vagrant is able to define and control multiple guest machines per Vagrantfile.
- This is known as a "multi-machine" environment.

# Define machine

```
Vagrant.configure("2") do |config|
  config.vm.provision "shell", inline: "echo Hello"

  config.vm.define "web" do |web|
    web.vm.box = "apache"
  end

  config.vm.define "db" do |db|
    db.vm.box = "mysql"
  end
end
```

# Networking

- In order to access the Vagrant environment created, Vagrant exposes some high-level networking options for things such as forwarded ports, connecting to a public network, or creating a private network.

# Defining a Forwarded Port

## FORWARD PORT

```
Vagrant.configure("2") do |config|
  config.vm.network "forwarded_port", guest: 80, host: 8080
end
```

## FORWARD PORT PROTOCOL

```
Vagrant.configure("2") do |config|
  config.vm.network "forwarded_port", guest: 2003, host: 12003, protocol: "tcp"
  config.vm.network "forwarded_port", guest: 2003, host: 12003, protocol: "udp"
end
```

# Private networks

## DHCP

```
Vagrant.configure("2") do |config|
  config.vm.network "private_network", type: "dhcp"
end
```

## STATIC IP

```
Vagrant.configure("2") do |config|
  config.vm.network "private_network", ip: "192.168.50.4"
end
```

# Public networks

## DHCP

```
Vagrant.configure("2") do |config|
  config.vm.network "public_network"
end
```

## Using the DHCP Assigned Default Route

```
Vagrant.configure("2") do |config|
  config.vm.network "public_network",
    use_dhcp_assigned_default_route: true
end
```

## STATIC IP

```
config.vm.network "public_network", ip: "192.168.0.17"
```

# Command-Line Interface

**box** manages boxes: installation, removal, etc.

**destroy** stops and deletes all traces of the vagrant machine

**global-status** outputs status Vagrant environments for this user

**halt** stops the vagrant machine

**help** shows the help for a subcommand

**init** initializes a new Vagrant environment by creating a Vagrantfile

**login** log in to HashiCorp's Vagrant Cloud

**package** packages a running vagrant environment into a box

**plugin** manages plugins: install, uninstall, update, etc.

**port** displays information about guest port mappings

**powershell** connects to machine via powershell remoting

**provision** provisions the vagrant machine

# Command-Line Interface

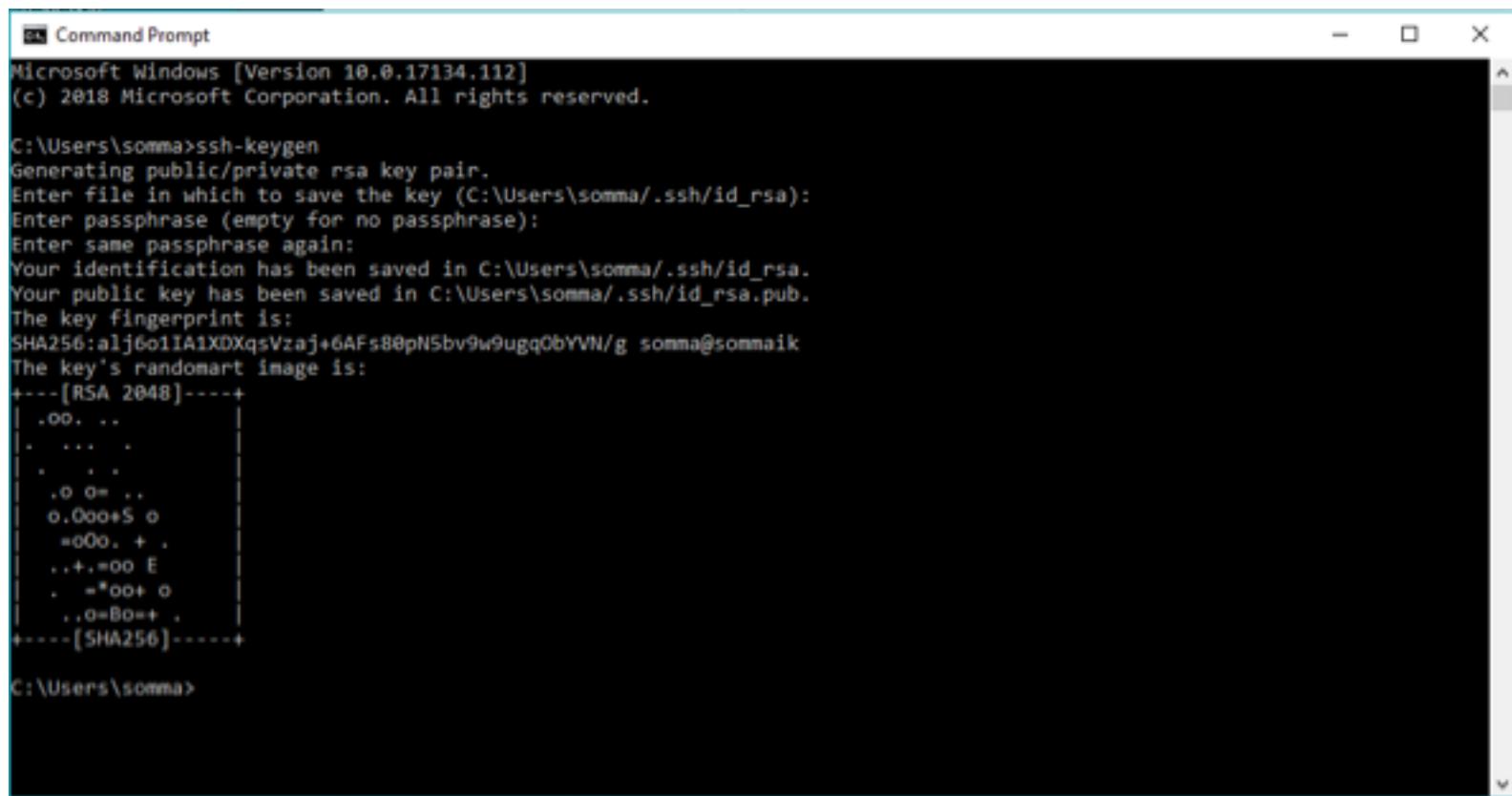
<b>push</b>	deploys code in this environment to a configured destination
<b>rdp</b>	connects to machine via RDP
<b>reload</b>	restarts vagrant machine
<b>resume</b>	resume a suspended vagrant machine
<b>snapshot</b>	manages snapshots: saving, restoring, etc.
<b>ssh</b>	connects to machine via SSH
<b>ssh-config</b>	outputs OpenSSH valid configuration
<b>status</b>	outputs status of the vagrant machine
<b>suspend</b>	suspends the machine
<b>up</b>	starts and provisions the vagrant environment
<b>validate</b>	validates the Vagrantfile
<b>version</b>	prints current and latest Vagrant version

การใช้ vagrant with

# SSH KEY

# ການສ້າງ SSH KEY

- ວິທີ ssh-keygen

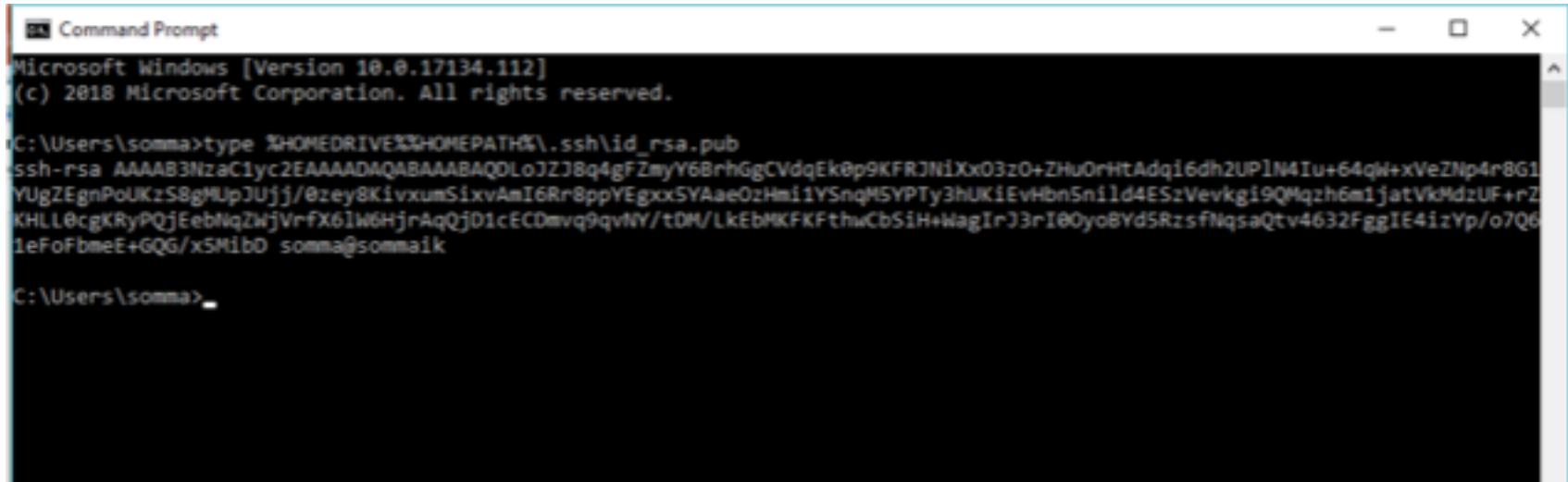


```
Command Prompt
Microsoft Windows [Version 10.0.17134.112]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\somma>ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (C:\Users\somma/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in C:\Users\somma/.ssh/id_rsa.
Your public key has been saved in C:\Users\somma/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:alj6o1IA1XDXqsVzaj+6AFs80pN5bv9w9ugq0bYVN/g somma@sommaik
The key's randomart image is:
+---[RSA 2048]---+
| .0. ...
| ...
| ...
| .0 0= ..
| 0.0oo+S o
| =oo. + .
| ..+=oo E
| . -*oo+ o
| ..o=Bo==+
+---[SHA256]---+
C:\Users\somma>
```

# การสร้าง SSH KEY

- ตรวจสอบโดยการพิมพ์คำสั่งนี้ type  
%HOMEDRIVE%%HOMEPATH%\ssh\id\_rsa.pub



```
Command Prompt
Microsoft Windows [Version 10.0.17134.112]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\somma>type %HOMEDRIVE%%HOMEPATH%\ssh\id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQDLoJZJ8q4gFZmyY6BrhGgCVdqEk0p9KFRJNiXxO3z0+ZHuOrHtAdqi6dh2UPlN4Iu+64qi+xVeZNp4r8G1
YUgZEgnPoUKzS8gMUpJUjj/0zey8KivxumSixvAmI6Rr8ppYEgxSYAaeOzHmi1YSnqMSYPTy3hUKiEvHbnSnild4ESzVevkgi9QMqzh6m1jatVkmzdUF+r2
KHLL0cgKRyPQjEebNqZkjVrFX6lW6HjnAqQjD1cECDmvq9qvNY/tDM/LkEbMKFKfhwCbSiH+WagIrJ3rI0Oyo8Yd5RzsfnqsaQtv4632FggIE4izYp/o7Q6
1eFoFbmeE+GQG/x5MibO somma@sommaik

C:\Users\somma>
```

# การ Run Vagrant

- สร้างโปรเจคที่ต้องการ แล้ว run คำสั่ง vagrant init จะได้ Vagrantfile

```
C:\WINDOWS\system32>e:  
E:\>cd Nooti3w  
E:\Nooti3w>cd workspace_train  
E:\Nooti3w\workspace_train>mkdir vagrant_getting_started  
E:\Nooti3w\workspace_train>cd vagrant_getting_started  
E:\Nooti3w\workspace_train\vagrant_getting_started>vagrant init  
A 'Vagrantfile' has been placed in this directory. You are now  
ready to 'vagrant up' your first virtual environment! Please read  
the comments in the Vagrantfile as well as documentation on  
'vagrantup.com' for more information on using Vagrant.  
E:\Nooti3w\workspace_train\vagrant_getting_started>dir  
Volume in drive E has no label.  
Volume Serial Number is 6A49-8E1C  
  
Directory of E:\Nooti3w\workspace_train\vagrant_getting_started  
  
06/27/2018  04:48 PM    <DIR>      .  
06/27/2018  04:48 PM    <DIR>      ..  
06/27/2018  04:48 PM           3,081 Vagrantfile  
                  1 File(s)       3,081 bytes  
                  2 Dir(s)  302,528,217,088 bytes free  
E:\Nooti3w\workspace_train\vagrant_getting_started>
```

# การ Run Vagrant

- Run คำสั่ง vagrant box add ubuntu/xenial64

```
E:\Nooti3w\workspace_train\vagrant_getting_started>vagrant box add ubuntu/xenial64
--> box: Loading metadata for box 'ubuntu/xenial64'
    box: URL: https://vagrantcloud.com/ubuntu/xenial64
--> box: Adding box 'ubuntu/xenial64' (v20180622.0.0) for provider: virtualbox
The box you're attempting to add already exists. Remove it before
adding it again or add it with the '--force' flag.

Name: ubuntu/xenial64
Provider: virtualbox
Version: 20180622.0.0
```

- จากนั้นให้แก้ไข file Vagrantfile ในโปรเจค

```
Vagrant.configure("2") do |config|
  config.vm.box = "ubuntu/xenial64"
end
```

# เรียกใช้ ssh key ใน Vagrantfile

## แทรกเข้าไปใน Vagrantfile

```
config.ssh.insert_key = false
config.ssh.private_key_path = ["~/.ssh/id_rsa", "~/.vagrant.d/insecure_private_key"]
config.vm.provision "file", source: "~/.ssh/id_rsa.pub", destination: "~/.ssh/authorized_keys"
config.vm.provision "file", source: "~/.ssh/id_rsa", destination: "~/.ssh/id_rsa"
config.vm.provision "file", source: "~/.ssh/id_rsa.pub", destination: "~/.ssh/id_rsa.pub"
config.vm.provision "shell", inline: <<-EOC
  sudo sed -i -e "\#PasswordAuthentication yes# s#PasswordAuthentication yes#PasswordAuthentication no#g" /etc/ssh/sshd_config
  sudo service ssh restart
EOC
```

# การ Run Vagrant

- Run คำสั่ง vagrant up

```
E:\Nooti3w\workspace_train\vagrant_getting_started>vagrant up
Bringing machine 'default' up with 'virtualbox' provider...
--> default: Checking if box 'ubuntu/xenial64' is up to date...
--> default: Clearing any previously set forwarded ports...
--> default: Clearing any previously set network interfaces...
--> default: Preparing network interfaces based on configuration...
--> default: Adapter 1: nat
--> default: Forwarding ports...
--> default: 22 (guest) => 2222 (host) (adapter 1)
--> default: Running 'pre-boot' VM customizations...
--> default: Booting VM...
--> default: Waiting for machine to boot. This may take a few minutes...
--> default: SSH address: 127.0.0.1:2222
--> default: SSH username: vagrant
--> default: SSH auth method: private key
--> default: Warning: Connection aborted. Retrying...
--> default: Warning: Connection reset. Retrying...
--> default: Warning: Connection aborted. Retrying...
--> default: Warning: Connection reset. Retrying...
--> default: Warning: Connection aborted. Retrying...
--> default: Warning: Connection reset. Retrying...
--> default: Warning: Remote connection disconnect. Retrying...
--> default: Warning: Connection aborted. Retrying...
--> default: Warning: Connection reset. Retrying...
--> default: Warning: Remote connection disconnect. Retrying...
--> default: Warning: Connection aborted. Retrying...
--> default: Warning: Connection reset. Retrying...
--> default: Warning: Connection aborted. Retrying...
--> default: Warning: Remote connection disconnect. Retrying...
--> default: Warning: Connection aborted. Retrying...
--> default: Warning: Connection reset. Retrying...
--> default: Warning: Connection aborted. Retrying...
--> default: Warning: Remote connection disconnect. Retrying...
--> default: Warning: Connection aborted. Retrying...
--> default: 
--> default: Vagrant insecure key detected. Vagrant will automatically replace
--> default: this with a newly generated keypair for better security.
--> default: 
--> default: Inserting generated public key within guest...
--> default: Removing insecure key from the guest if it's present...
--> default: Key inserted! Disconnecting and reconnecting using new SSH key...
```

# การ Run Vagrant

- Run คำสั่ง vagrant box list

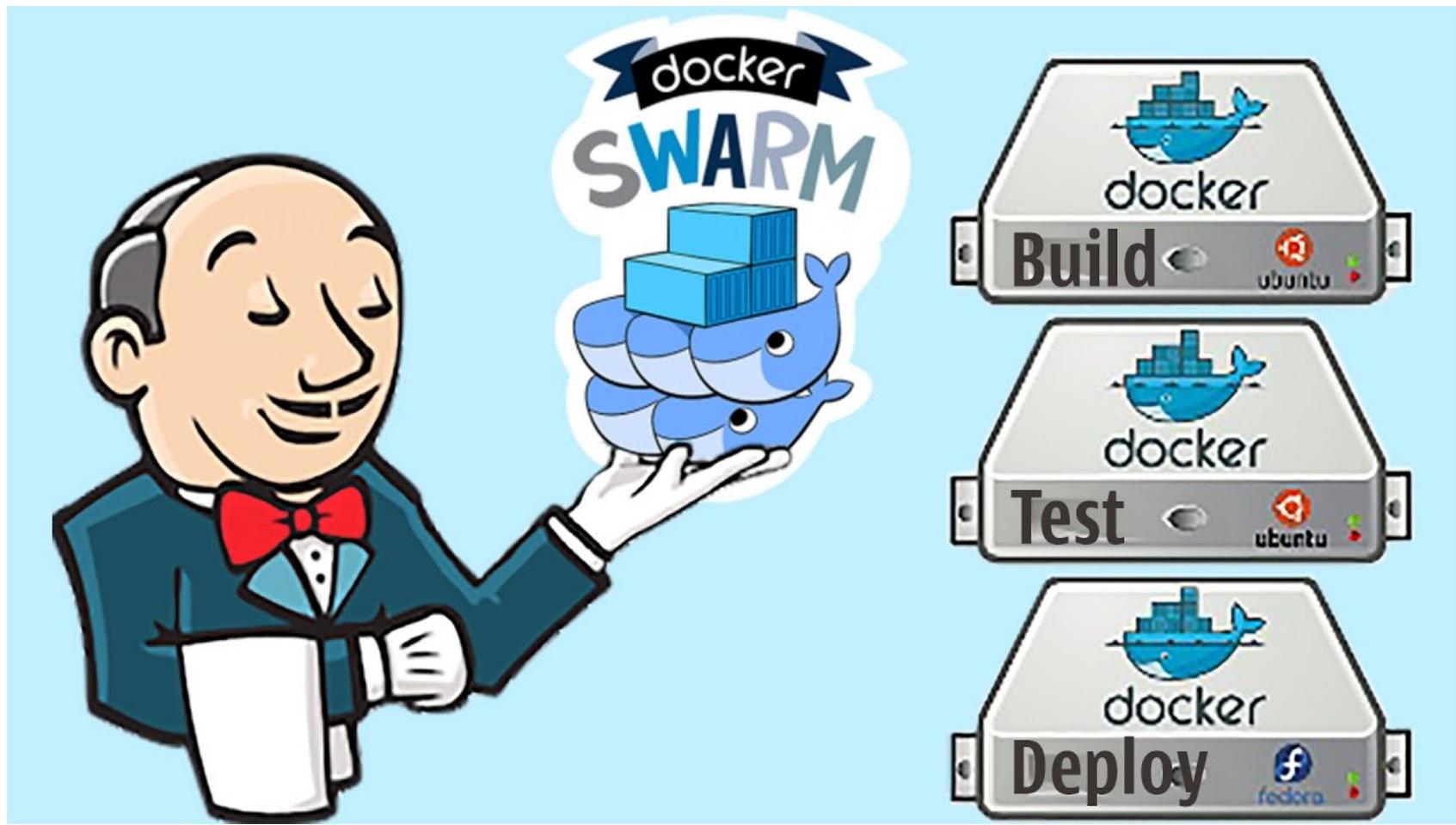
```
E:\Nooti3w\workspace_train\vagrant_getting_started>vagrant box list  
ubuntu/xenial64 (virtualbox, 20180622.0.0)
```

- Run คำสั่ง vagrant ssh

```
E:\Nooti3w\workspace_train\vagrant_getting_started>vagrant ssh  
Welcome to Ubuntu 16.04.4 LTS (GNU/Linux 4.4.0-128-generic x86_64)  
  
 * Documentation: https://help.ubuntu.com  
 * Management: https://landscape.canonical.com  
 * Support: https://ubuntu.com/advantage  
  
 Get cloud support with Ubuntu Advantage Cloud Guest:  
 http://www.ubuntu.com/business/services/cloud  
  
0 packages can be updated.  
0 updates are security updates.  
  
vagrant@ubuntu-xenial:~$
```

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 กั้งหนด
- 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 กั้งหนด

# เริ่มสร้าง docker swarm (manager)

- run คำสั่ง ใน terminal ดังนี้

```
docker swarm init --advertise-addr=192.168.33.12
```

```
vagrant@egr1:~$ docker swarm init --advertise-addr=192.168.33.12
Swarm initialized: current node (izupnbin1ra7cv2a9ur4k95ve) is now a manager.
```

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

```
    docker swarm join --token SWMTKN-1-3dip03ya9iixhhsk1aegfs3ptd30pwuzpk7413i0okg6nco99-egnikmox8hvhh8fok2d8zfppex 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 ด่อไป

# Join เข้ากลุ่ม swarm

- หลังจากที่ได้สร้างเครื่อง manager ไปแล้วก็ต้องนำคำสั่งที่ได้จาก console ของเครื่อง manager มารันในเครื่อง worker ดังตัวอย่าง

```
vagrant@node1:~$ docker swarm join --token SWMTKN-1-3dip03yo9iixhhsk1oegfs3ptd30pwuzpk7413lookg6nca99-egmkmlo8vhvh8fol2d8zfppmx 192.168.33.12:2377
This node joined a swarm as a worker.
```

# ตรวจสอบจำนวน node ทั้งหมด

- run คำสั่ง ใน terminal ดังนี้  
docker node list

```
vagrant@mgr1:~$ docker node ls
ID           HOSTNAME  STATUS  AVAILABILITY  MANAGER STATUS  ENGINE VERSION
izupnb1ra7cv2a9ur4k95ve *  mgr1      Ready   Active        Leader        18.03.1-ce
mdi3i3cjpknx8d58ujrfc9ujz  node1     Ready   Active        Active        18.03.1-ce
```

# สร้าง service บนเครื่อง manager

- run คำสั่ง ใน terminal ดังนี้

```
docker service create <><> nginx
```

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

\*\* option ที่ควรบี

```
--replicas 2
--name web_server
--constraint "node.role != manager"
--publish 8080:80
```

replicas คือ จำนวน node ที่ต้องการสร้าง  
name ชื่อที่เอาไว้อ้างอิง  
constraint เขื่อนไขในการเลือกเครื่อง  
publish คือ port ที่จะเปิดให้บริการจากภายนอก

# ดู service ทั้งหมด

- run คำสั่ง ใน terminal ดังนี้  
docker service ls

```
verify: Service converged
[vagrant@vagr1:~$ 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@mgr1:~$ 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

- run คำสั่ง ใน terminal ดังนี้  
docker service ps <service\_name>

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

- run คำสั่ง 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
```

# au service

- runคำสั่ง ใน terminal ดังนี้

```
docker service rm <service_name>
```

```
vagrant@mg1:~$ 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

## Customize Jenkins

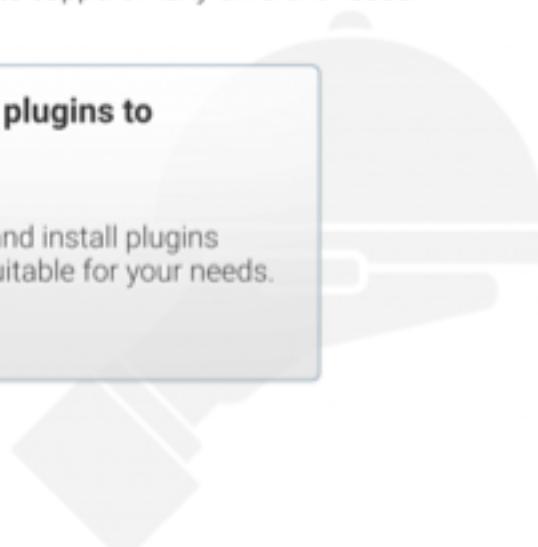
Plugins extend Jenkins with additional features to support many different needs.

**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 ** Pipelines: Multibranch ** Authentication Tokens API Plugin ** Docker Commons Plugin ** Durable Task Plugin ** Pipelines: Nodes and Processes ** Docker Pipeline ** Pipelines: Stage Tags Metadata ** Pipelines: 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

# Create Admin User

Getting Started

## Create First Admin User

Username:  (highlighted)

Password:

Confirm password:

Full name:

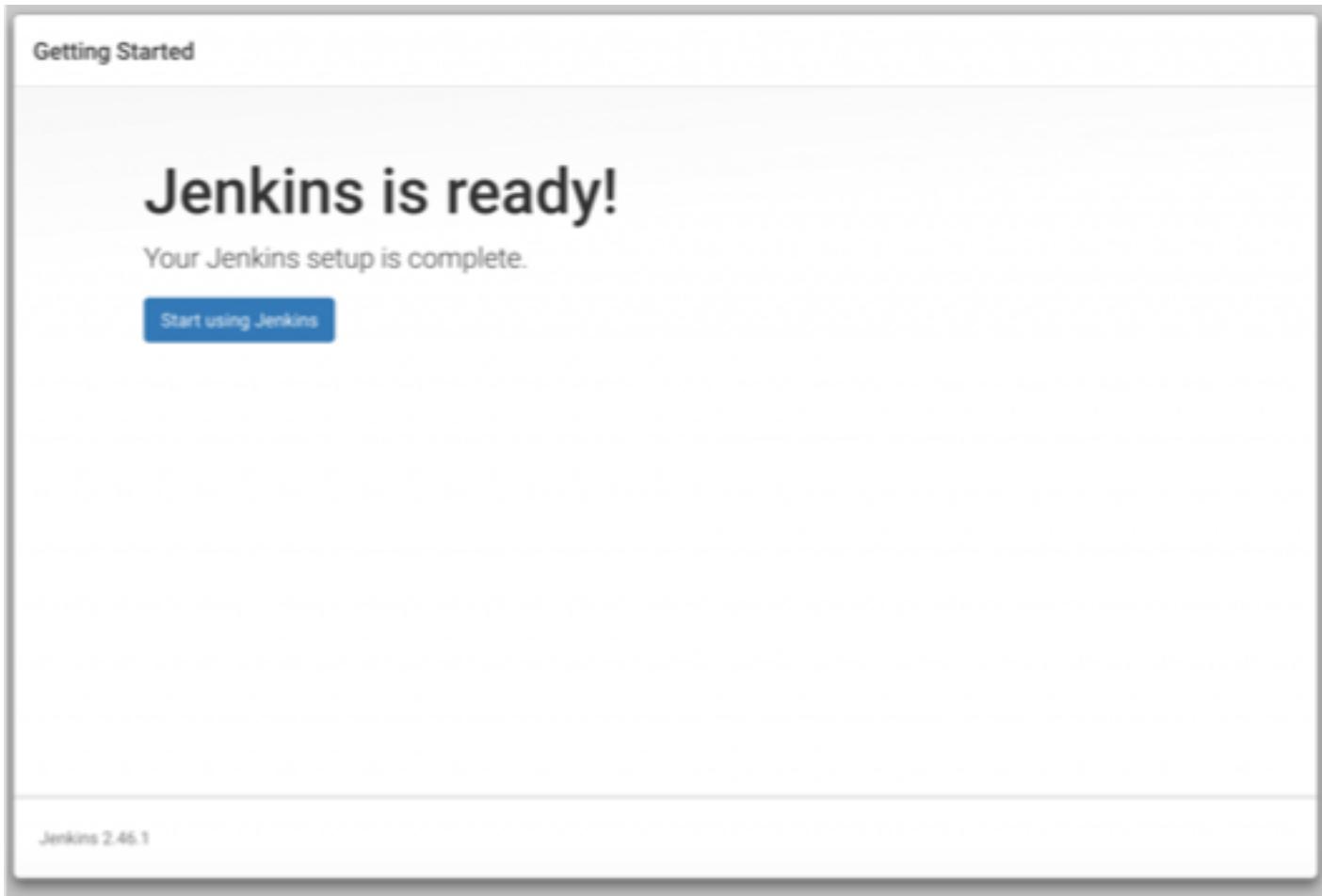
E-mail address:

Jenkins 2.46.1

Continue as admin

Save and Finish

# Setup Complete



The screenshot shows a Jenkins setup completion page. At the top left, it says "Getting Started". In the center, the text "Jenkins is ready!" is displayed in large, bold, dark font. Below it, a smaller text says "Your Jenkins setup is complete.". A blue button labeled "Start using Jenkins" is centered below the message. At the bottom left of the main content area, the text "Jenkins 2.46.1" is visible. The entire page has a light gray background with a white central content area.

Getting Started

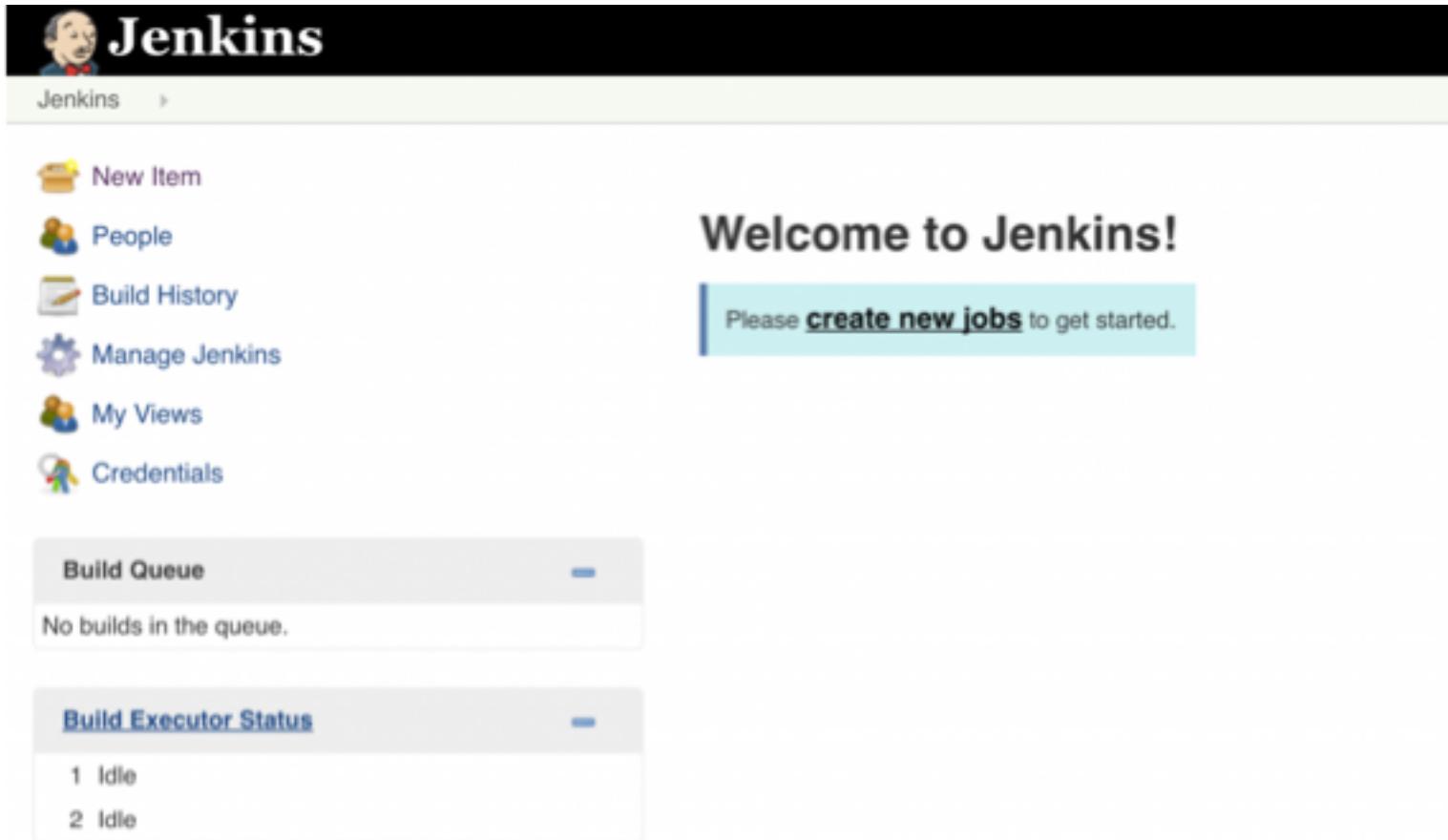
# Jenkins is ready!

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 in large letters. Below the title is a navigation menu with the following items:

- New Item
- People
- Build History
- Manage Jenkins
- My Views
- Credentials

In the center, a large "Welcome to Jenkins!" message is displayed, followed by a call-to-action: "Please [create new jobs](#) to get started." Below this, there are two sections: "Build Queue" (which shows "No builds in the queue.") and "Build Executor Status" (which shows "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

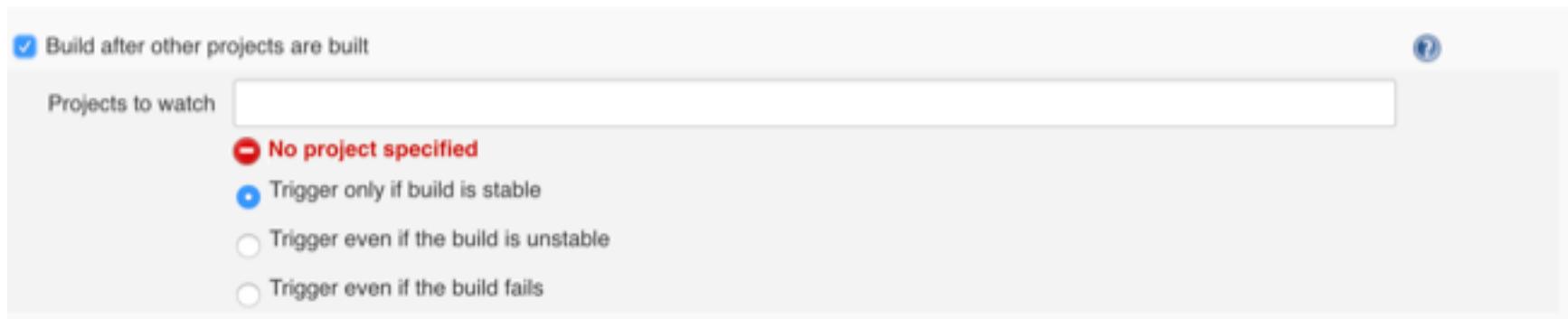
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

# 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



Build after other projects are built

Projects to watch

No project specified

Trigger only if build is stable

Trigger even if the build is unstable

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 H/12 \* \* \* \*

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.

สั่งให้ทำงานอัตโนมัติตามช่วงเวลาที่กำหนด โดยสามารถกำหนดเวลาได้อยู่ในรูปแบบ crontab format

# Poll SCM



สั่งให้ทำงานอัตโนมัติตามช่วงเวลาที่กำหนด โดยสามารถกำหนดเวลาได้อยู่ในรูปแบบ crontab format

โดยมีลักษณะการทำงานคือจะไปทำการตรวจสอบบดูก่อนว่า 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) (?)

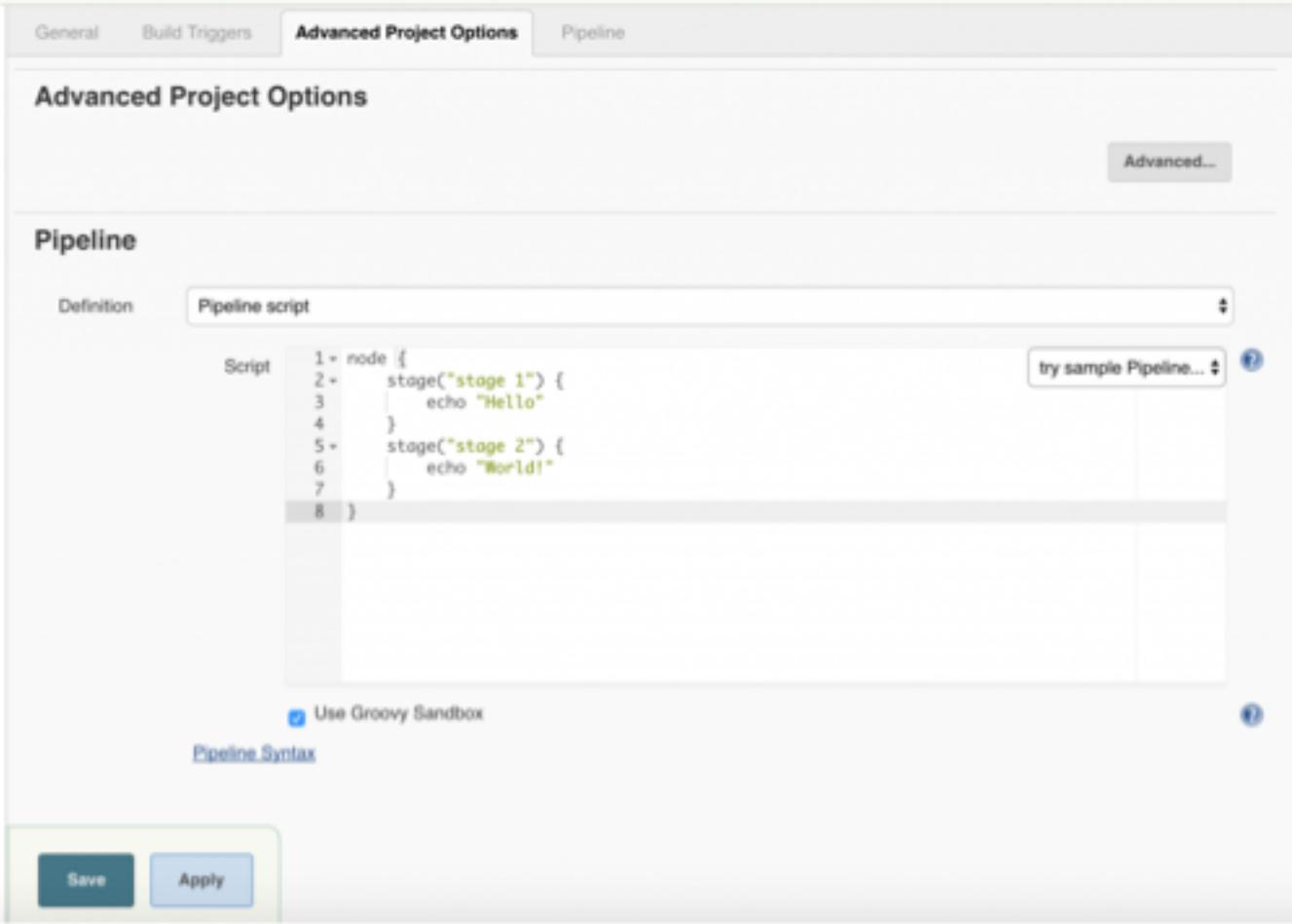
Authentication Token **0817544663**

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`

# Pipeline script



The screenshot shows the Jenkins Pipeline configuration interface. At the top, there are tabs for General, Build Triggers, Advanced Project Options (which is selected), and Pipeline. Below the tabs, the title "Advanced Project Options" is displayed. On the right side of the main area, there is a "Advanced..." button. The main content area is titled "Pipeline" and contains a "Definition" section. The "Definition" dropdown is set to "Pipeline script". Below the dropdown is a code editor containing a Groovy script:

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

To the right of the code editor is a "try sample Pipeline..." button with a question mark icon. Below the code editor, there is a checkbox labeled "Use Groovy Sandbox" with a checked status. At the bottom of the configuration screen, there 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
●	☀	aci.onosw.com	1 mo 0 days - #22	4 mo 25 days - #13	23 sec	
●	☀	aci.wisdomways.com	5 days 10 hr - #25	8 days 11 hr - #20	2 min 8 sec	
●	☁	aci.zone@scort.com	4 mo 14 days - #11	5 mo 4 days - #3	45 sec	
●	☀	birt.onosw.com	2 mo 17 days - #8	N/A	57 sec	
●	☀	booking.wisdomways.com	5 days 10 hr - #5	N/A	1 min 46 sec	
●	🌧	checker.onosw.com	5 mo 10 days - #2	5 mo 10 days - #2	50 sec	

กดเครื่องหมาย + เพื่อกำการเพิ่ม tab เพื่อการกรองข้อมูลของ job

# Set List View

View name:

List View       My View

Show simple list format. You can choose which jobs are to be displayed in which view.

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

Name:

Description:

[Plain text] [Preview](#)

Filter build queue

Filter build executors

**Job Filters**

Status Filter:  All selected jobs

Recurse in subfolders

Jobs:

- api.pnpse.com
- api.wisdomaiways.com
- api.zonereport.com
- birt.pnpse.com
- booking.wisdomaiways.com
- checker.pnpse.com
- checkerapi.pnpse.com
- demo.pnpse.com
- issue.pnpse.com
- issueapi.pnpse.com
- train-api.pnpse.com
- trainee.wisdomaiways.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
132ms	145ms

#1 Apr 12, 2017 8:51 AM No Changes

BSS for all RSS for failures

Permalinks



# Logs

The screenshot shows the Jenkins interface. On the left, there's a sidebar with various navigation links: Back to Dashboard, Status, Changes, Build Now, Delete Pipeline, Configure, Full Stage View, and Pipeline Syntax. Below that is the Build History section, which shows a single build (#1) from April 12, 2017, at 8:51 AM with 'No Changes'. At the bottom of the sidebar are RSS feed links for 'RSS for all' and 'RSS for failures'.

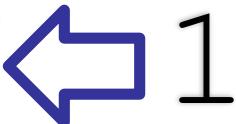
The main content area has two tabs: 'Stage Logs (stage 2)' and 'Stage View'. The 'Stage Logs' tab is active, showing a log entry: 'Print Message (self time 19ms)' followed by the output 'World!'. There's also a link to 'Recent Changes'.

The 'Stage View' tab displays a timeline of stages. It shows 'stage 1' with an average time of 132ms, 'stage 2' with an average time of 132ms, and a final stage with an average time of 145ms. A blue arrow points to the 'Logs' button for the 'stage 2' box.

At the bottom, there's a 'Permalinks' section.

# 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	
	Jobball	



2

# Add Credentials #2



The screenshot shows the Jenkins Global credentials interface. At the top, there's a navigation bar with links: Jenkins, Credentials, System, and Global credentials (unrestricted). Below the navigation, there are two buttons: "Back to credential domains" (with a green arrow icon) and "Add Credentials" (with a blue key icon). A large blue arrow points from the "Add Credentials" button towards the right side of the screen, indicating where the new credential configuration dialog will appear.



A configuration dialog box is displayed on the right side of the screen. It has the following fields:

- Kind: Username with password
- Scope: Global (Jenkins, nodes, items, all child items, etc)
- Username: [empty field]
- Password: [empty field]
- ID: [empty field]
- Description: [empty field]

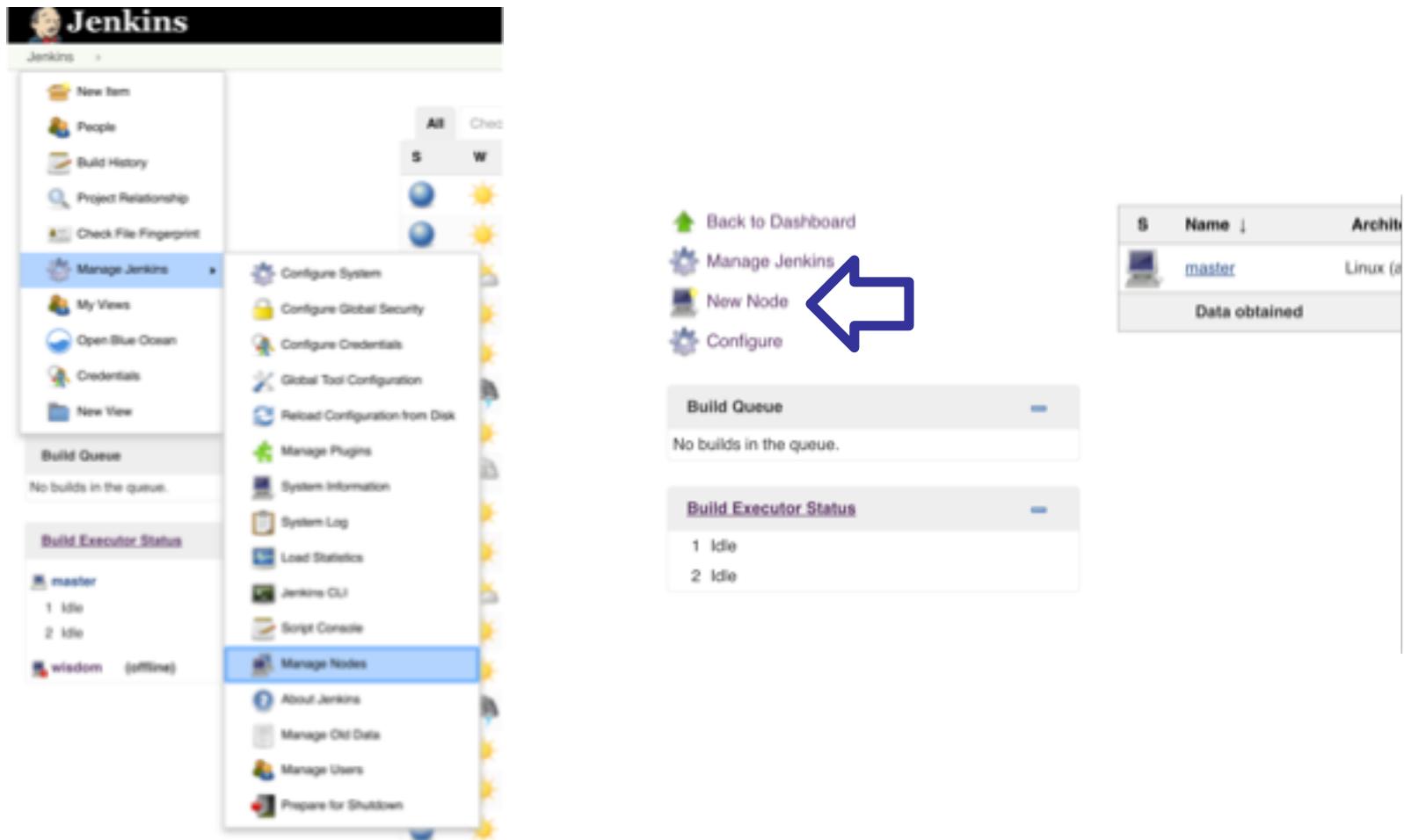
At the bottom of the dialog is an "OK" button.

# Install Plugins

The screenshot shows the Jenkins dashboard with the 'Manage Jenkins' menu open. Under 'Manage Jenkins', the 'Manage Plugins' option is highlighted. On the right, a list of available Jenkins plugins is displayed, categorized under 'Updates'.

Name	Version	Installed
Apache HttpClient 4.x API	4.5.5-0.0	4.5.5-0.0
Autowire for Blue Ocean	1.2.2	1.2.1
Bitsbucket Branch Source	2.2.11	2.2.8
Blue Ocean	1.8.0	1.8.0
Blue Ocean Core API	1.8.0	1.8.0
Blue Ocean Pipeline Editor	1.8.0	1.8.0
Branch API	2.0.20	2.0.18
Common API for Blue Ocean	1.8.0	1.8.0
Goals API for Blue Ocean	1.8.0	1.8.0
Credentials Pipeline	1.16	1.14

# Add new node



The screenshot shows the Jenkins dashboard. On the left, there's a sidebar with various options like 'New Item', 'People', and 'Manage Jenkins'. Under 'Manage Jenkins', 'Manage Nodes' is highlighted with a blue box and a blue arrow points to it from the main content area. The main content area has a header 'All' and 'W' (Workload) status indicators. It includes links for 'Back to Dashboard', 'Manage Jenkins', 'New Node' (which is also highlighted with a blue box and has a blue arrow pointing to it), and 'Configure'. Below these are sections for 'Build Queue' (No builds in the queue) and 'Build Executor Status' (1 Idle, 2 Idle). To the right, there's a table showing a single node named 'master' with details like 'Architecture: Linux (x86\_64)' and 'Data obtained'.

S	Name	Architecture
master	Linux (x86_64)	Data obtained

# Add new node

Node name

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

Name	mynode
Description	
# of executors	1
Remote root directory	/home/user
Labels	
Usage	Use this node as much as possible
Launch method	Launch slave agents via SSH
Host	
Credentials	root <input type="button" value="Add"/>
Host Key Verification Strategy <input type="button" value="Known hosts file"/> <input type="button" value="Verification Strategy"/>	
Availability	Keep this agent online as much as possible
Node Properties	
<input type="checkbox"/> Environment variables	
<input type="checkbox"/> Tool Locations	
<input type="button" value="Save"/>	

# Add new node

- Remote root directory : ใส่ชื่อ 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') {  
    ...  
}
```

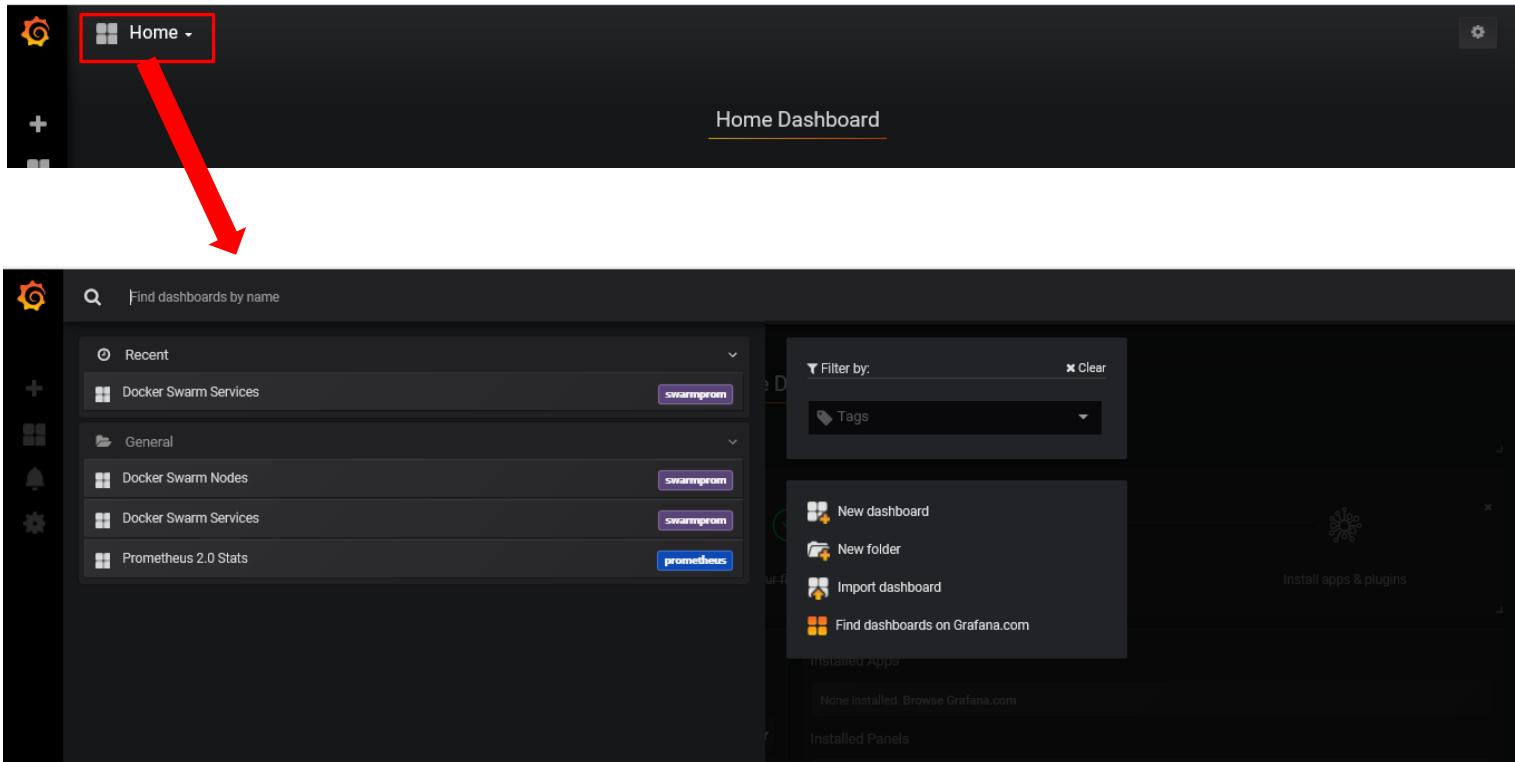
Monitoring with

# GRAFANA

# การใช้งาน Grafana

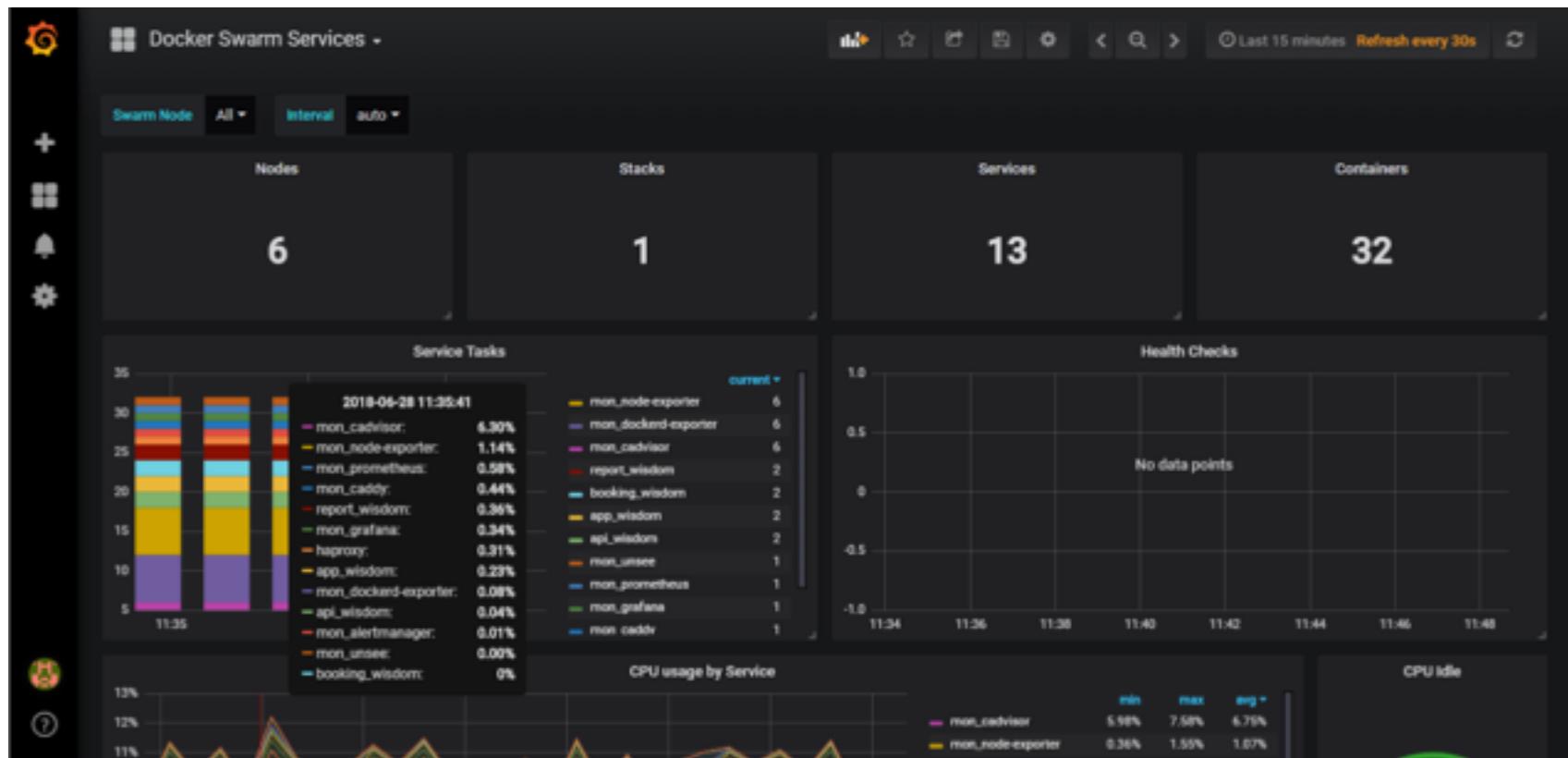


# การใช้งาน Grafana

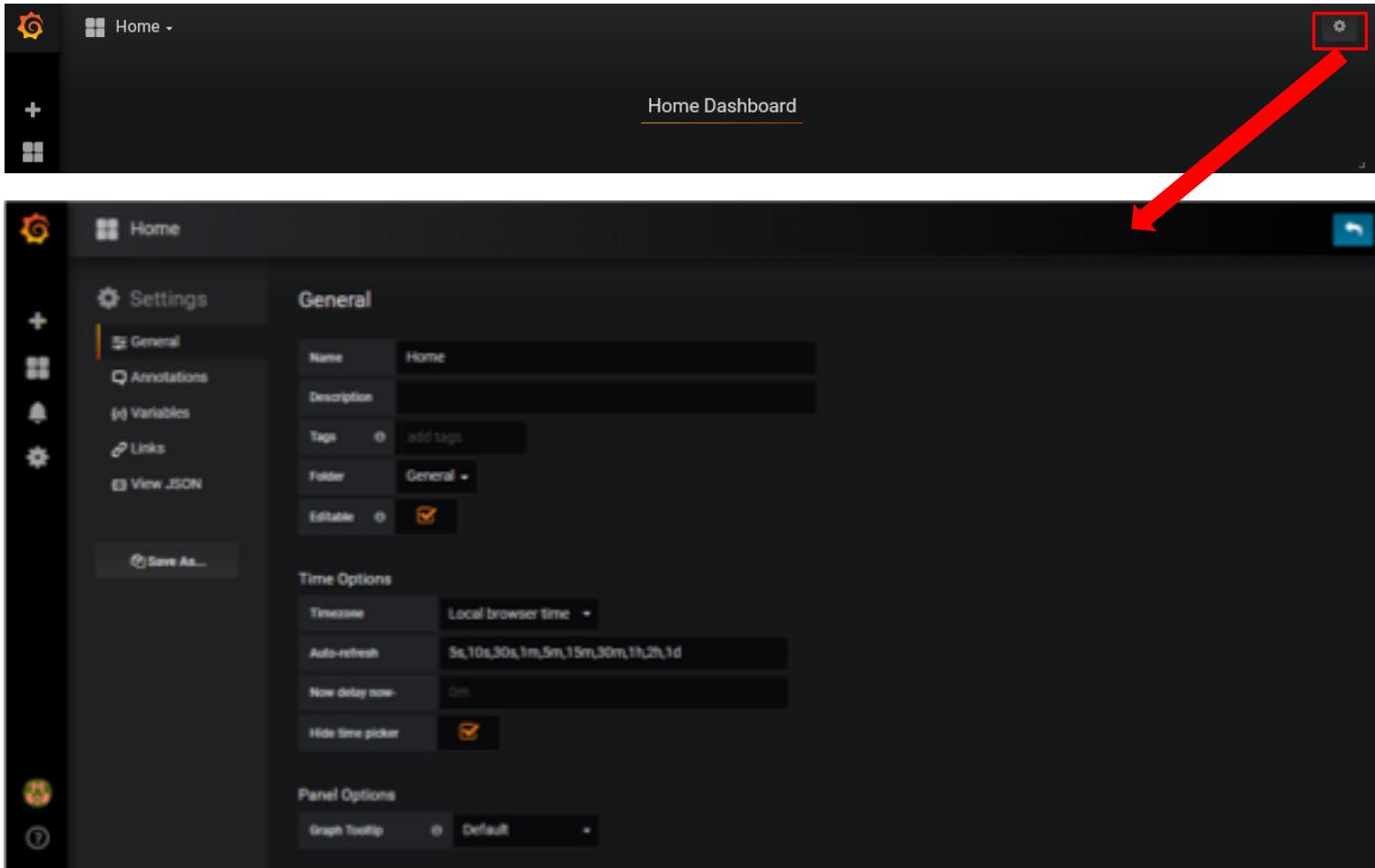


# การใช้งาน Grafana

- หน้าจอแสดงรายละเอียดแต่ละ Service



# การใช้งาน Grafana

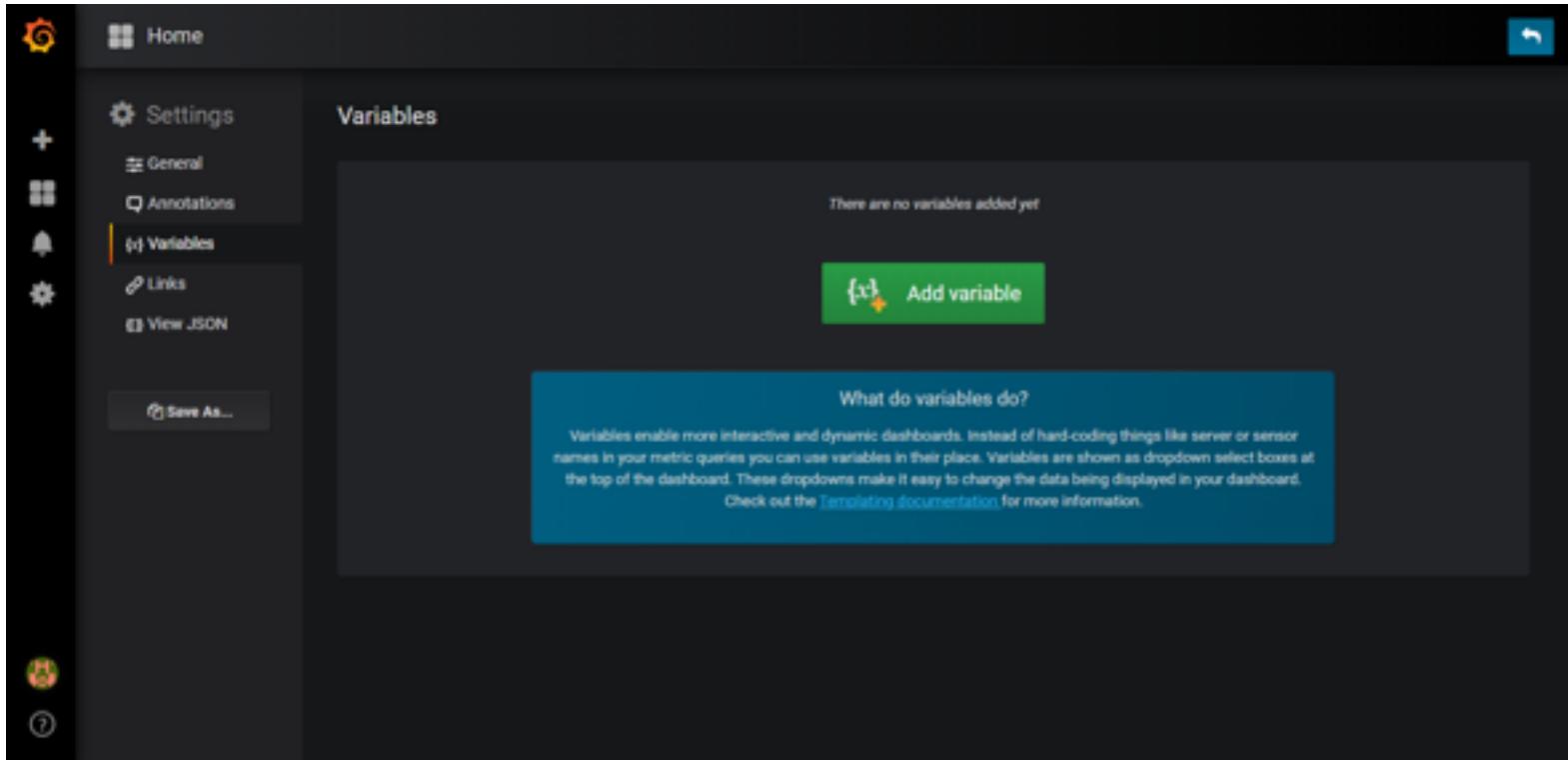


The screenshot shows the Grafana interface. At the top, there's a navigation bar with icons for Home, Settings (highlighted with a red box), Annotations, Variables, Links, and View JSON. The main title is "Home Dashboard". In the center, there's a "General" settings panel for the "Home" dashboard. It includes fields for Name (Home), Description, Tags (with a placeholder "add tags"), Folder (General), and Editable (with a checked checkbox). Below this are "Time Options" with Timezone set to "Local browser time", Auto-refresh set to "5s,10s,30s,1m,5m,15m,30m,1h,2h,1d", Now delay now- set to "0m", and Hide time picker checked. At the bottom are "Panel Options" with Graph Tooltip set to "Default". A large red arrow points from the bottom right towards the Settings icon in the top right corner of the dashboard header.

# การใช้งาน Grafana

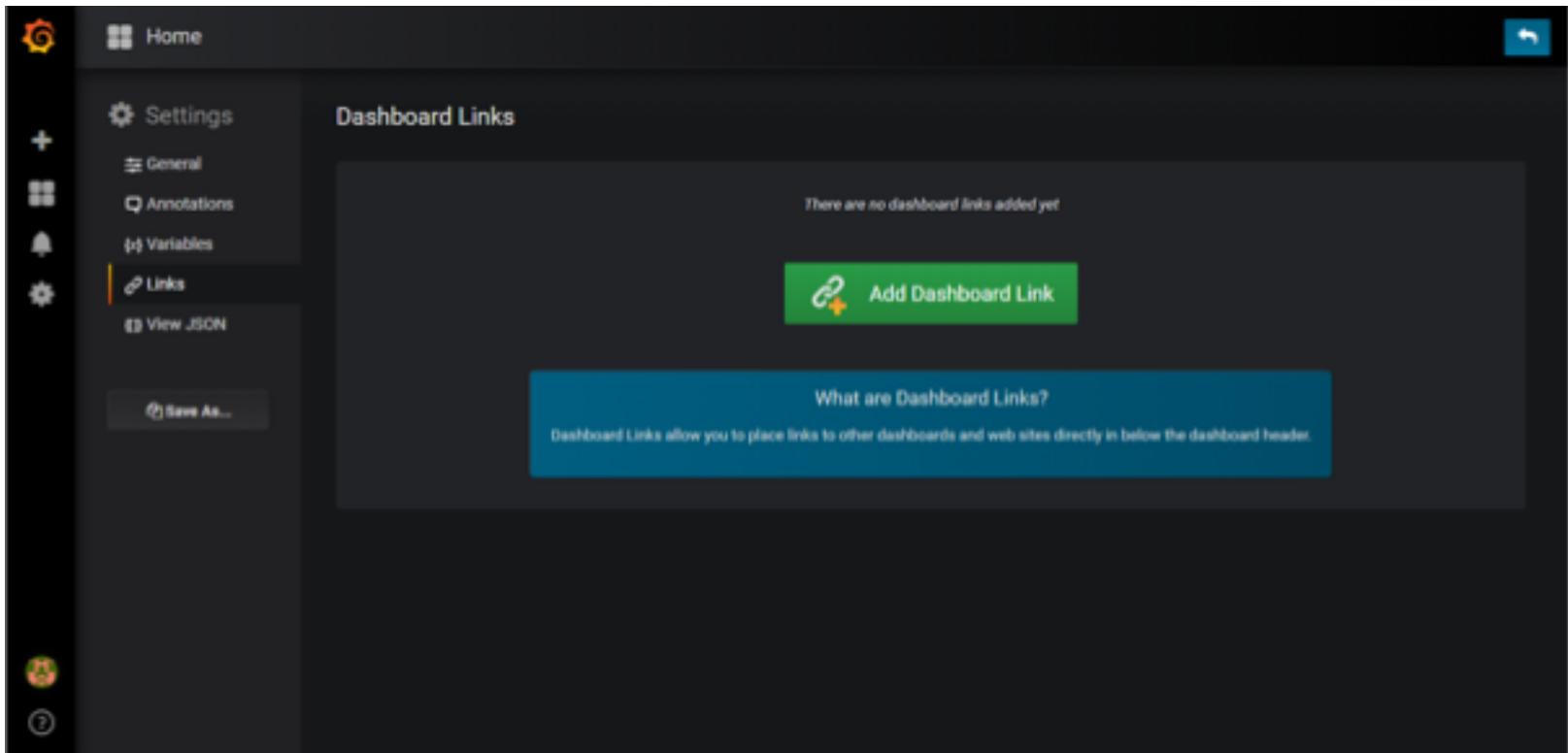
The screenshot shows the Grafana interface for managing annotations. On the left, a sidebar menu includes options like Home, Settings (General, Annotations, Variables, Links, View JSON), and Save As... The main content area is titled "Annotations" and displays a section for "Query name" with a dropdown for "Data source" set to "Grafana". A sub-section titled "Annotations & Alerts (Built-in)" is shown. Below this, a message states "There are no custom annotation queries added yet". A prominent green button at the bottom right says "Add Annotation Query" with a plus icon. A blue callout box provides information about what annotations are: "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 the "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 with white text provides information about what variables do: '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

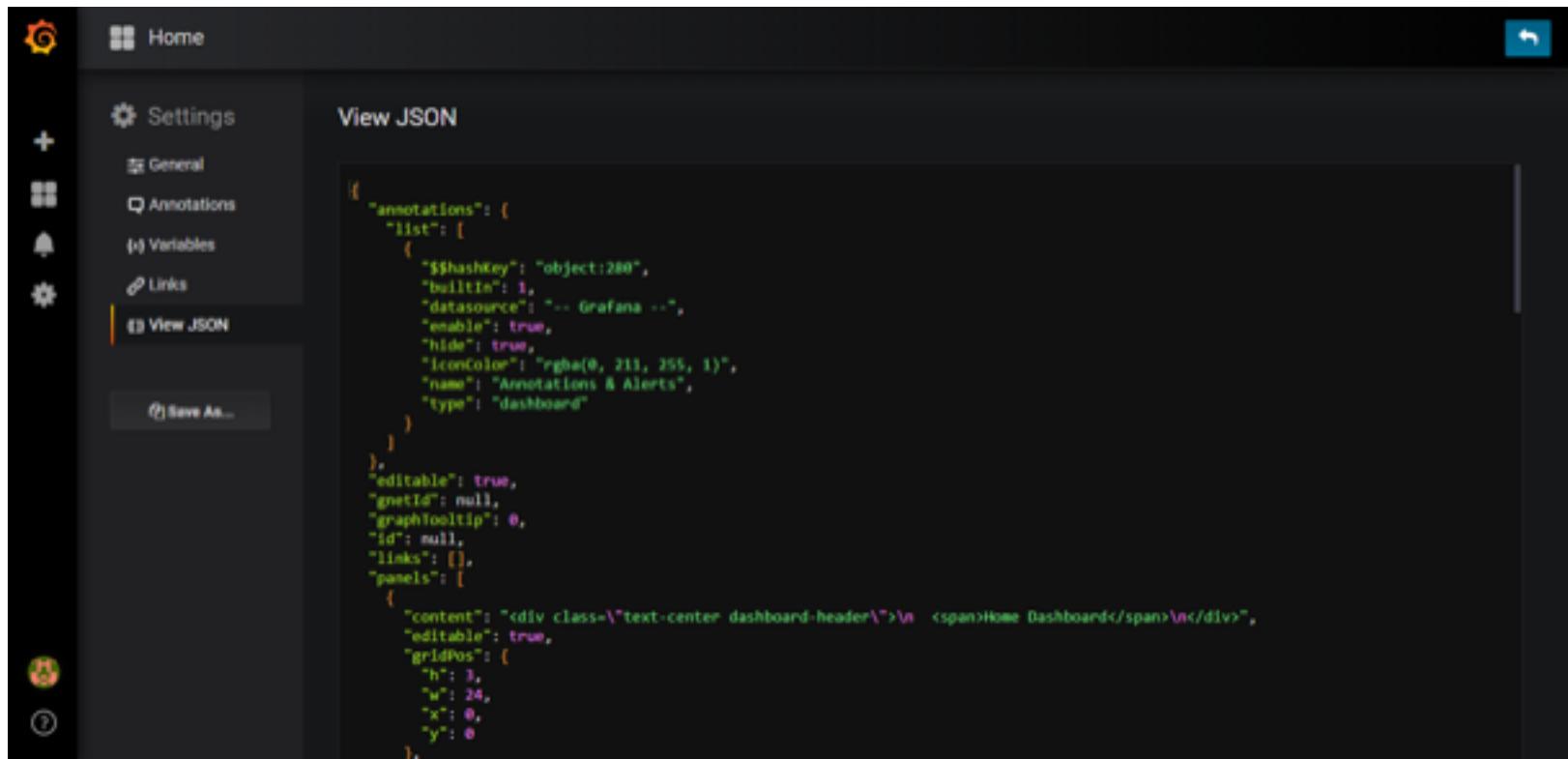


The screenshot shows the Grafana Settings page with a dark theme. On the left, there's a sidebar with icons for Home, Settings, Annotations, Variables, Links (which is selected and highlighted in orange), and View JSON. Below the sidebar is a 'Save As...' button. The main content area is titled 'Dashboard Links' and contains a message: 'There are no dashboard links added yet.' A large green button with a plus sign and the text 'Add Dashboard Link' is centered. At the bottom, a blue callout box provides information about what Dashboard Links are and how they work.

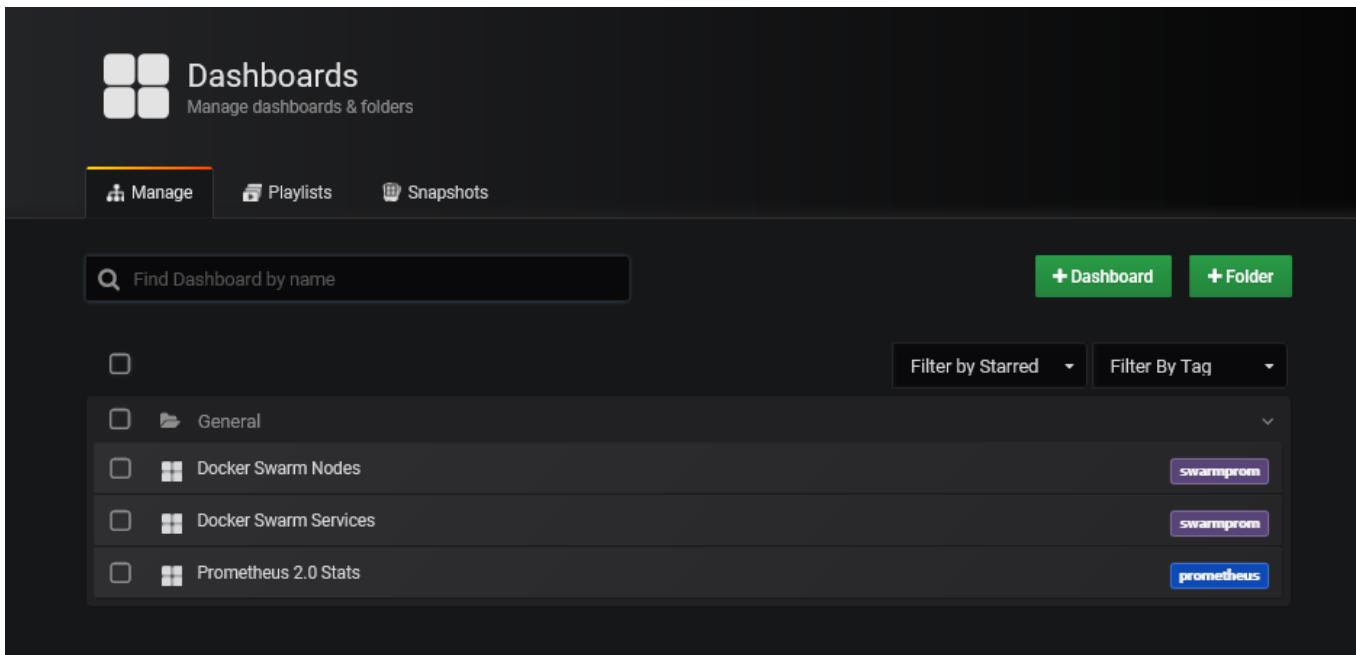
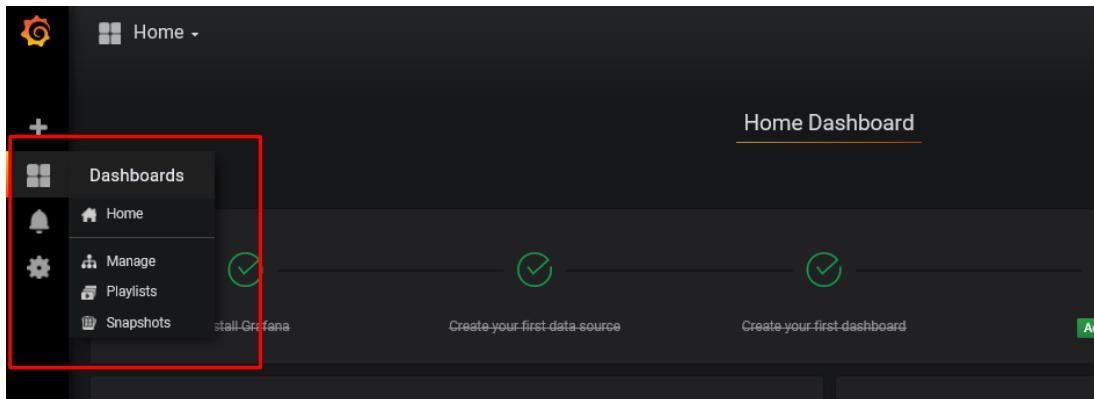
What are Dashboard Links?

Dashboard Links allow you to place links to other dashboards and web sites directly in below the dashboard header.

# การใช้งาน Grafana

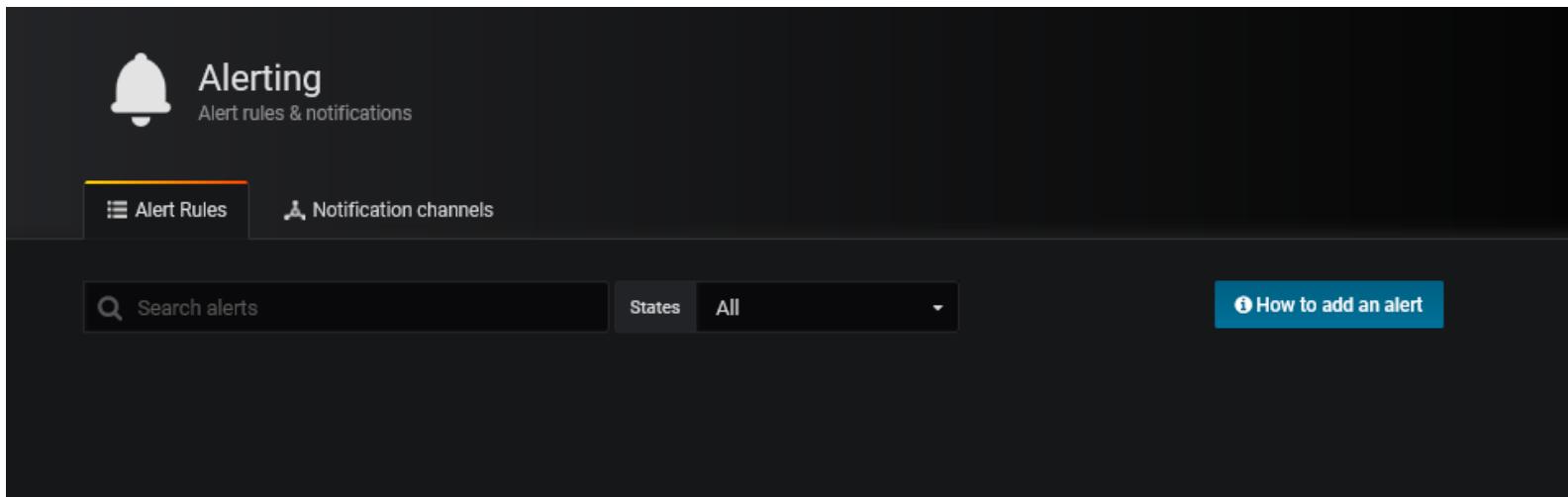
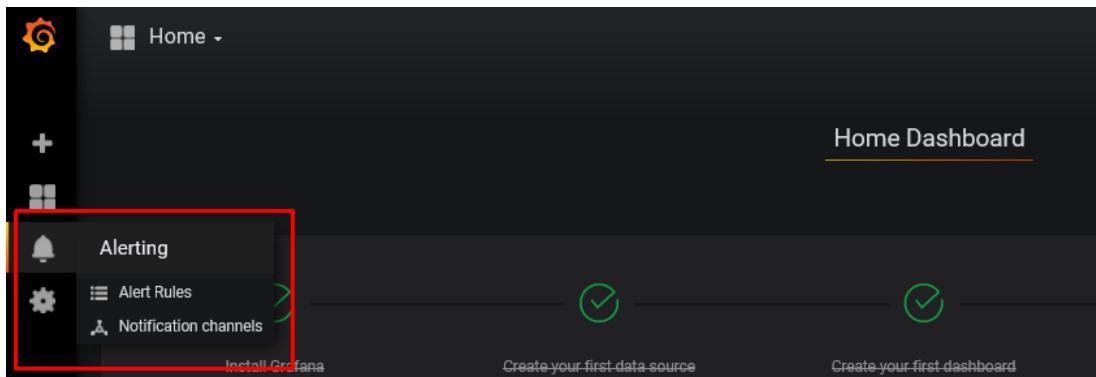


# การใช้งาน Grafana



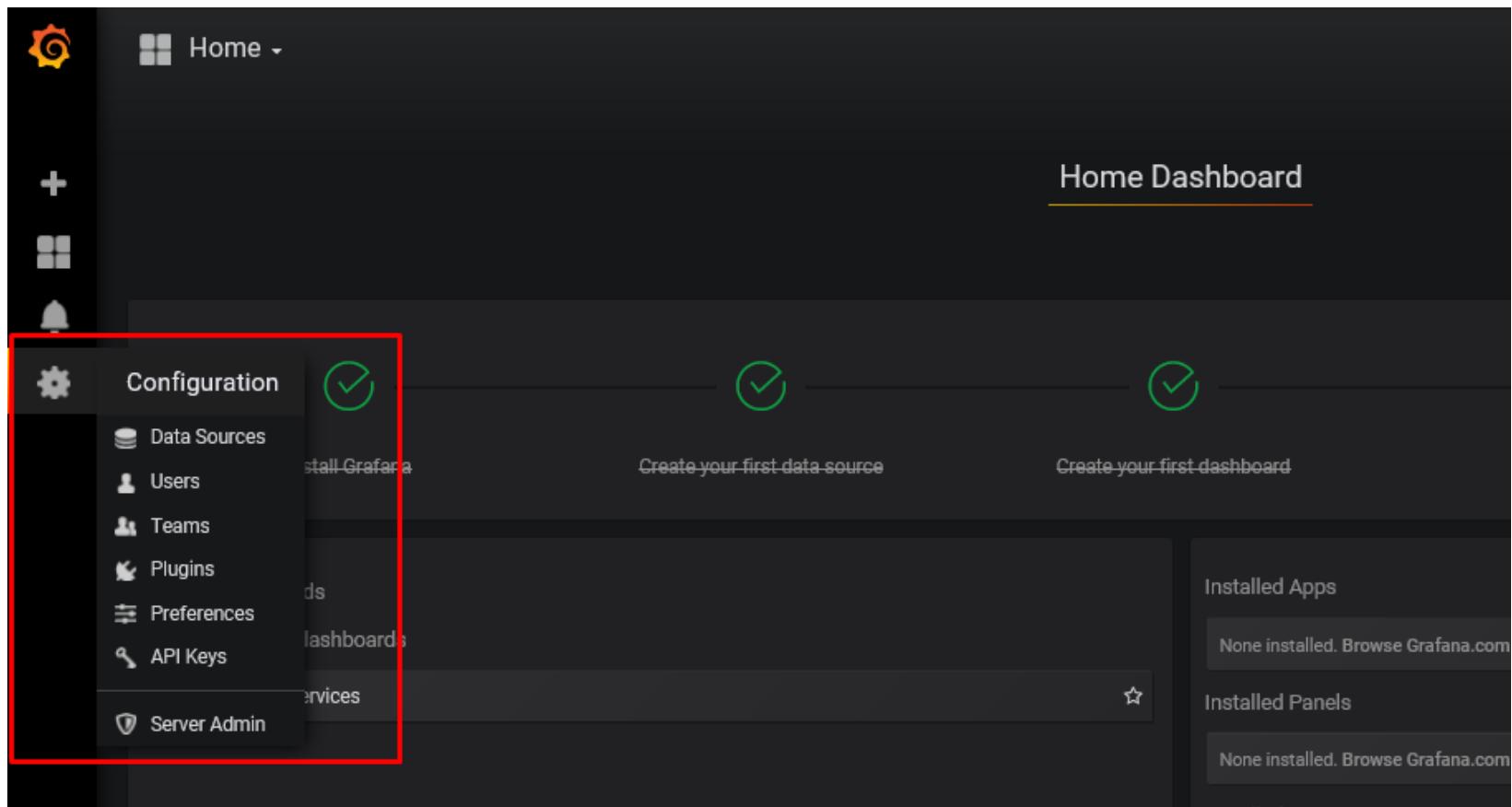
The screenshot shows the 'Dashboards' management interface. It features a sidebar with 'Manage', 'Playlists', and 'Snapshots' buttons. Below is a search bar with 'Find Dashboard by name' and two green buttons: '+ Dashboard' and '+ Folder'. The main content area lists dashboards under 'General': 'Docker Swarm Nodes', 'Docker Swarm Services', and 'Prometheus 2.0 Stats'. Each dashboard has a small icon and a purple tag labeled 'swarmprom' or 'prometheus' to its right. There are also filters for 'Filter by Starred' and 'Filter By Tag'.

# การใช้งาน Grafana



The screenshot shows the Grafana Alerting page. It features a header with a bell icon and the title 'Alerting'. Below the header are tabs for 'Alert Rules' and 'Notification channels', with 'Alert Rules' being active. There's a search bar labeled 'Search alerts', a dropdown for 'States' set to 'All', and a blue button labeled 'How to add an alert'.

# การใช้งาน Grafana



Home Dashboard

Configuration

- Data Sources
- Users
- Teams
- Plugins
- Preferences
- API Keys

Install Grafana

Create your first data-source

Create your first dashboard

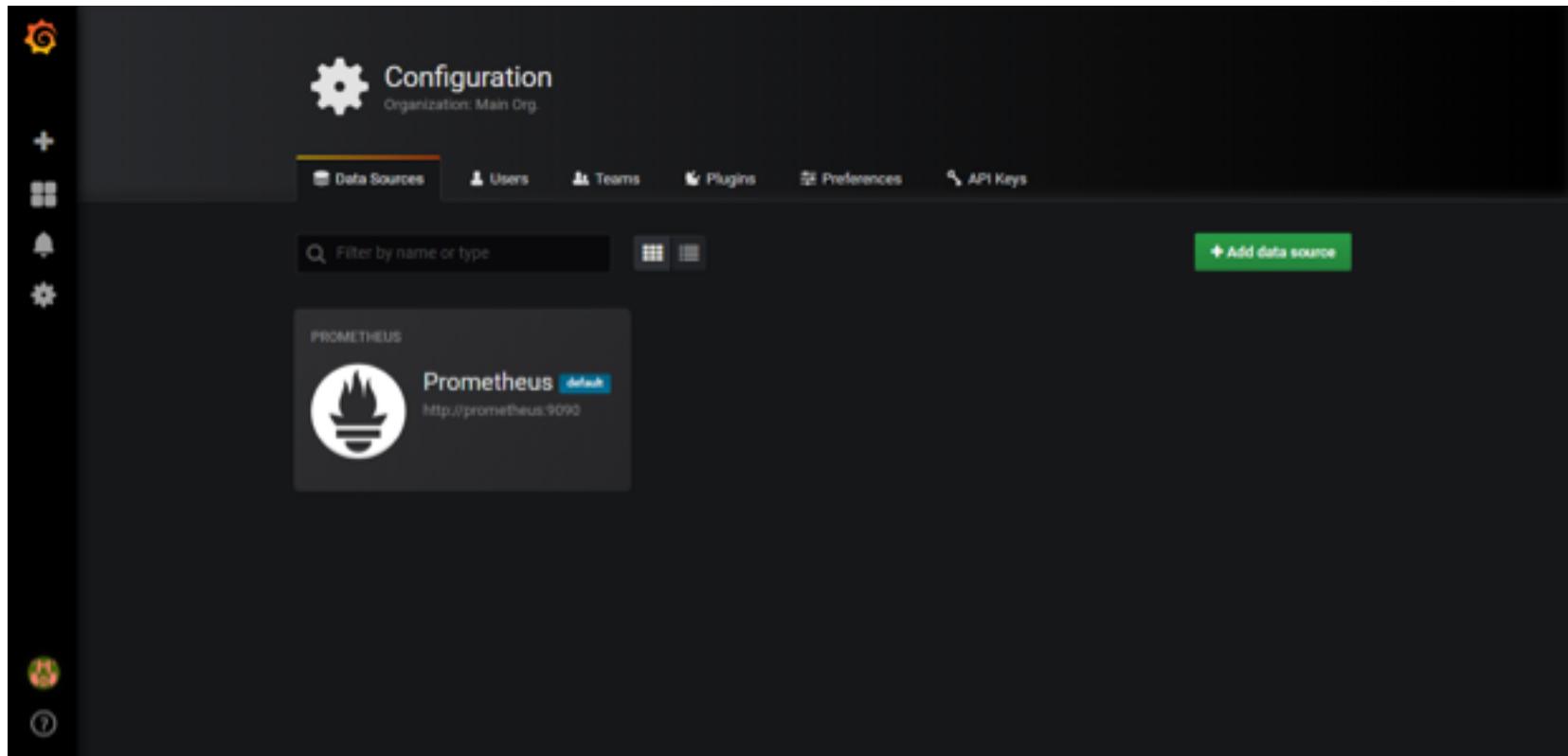
Installed Apps

None installed. Browse Grafana.com

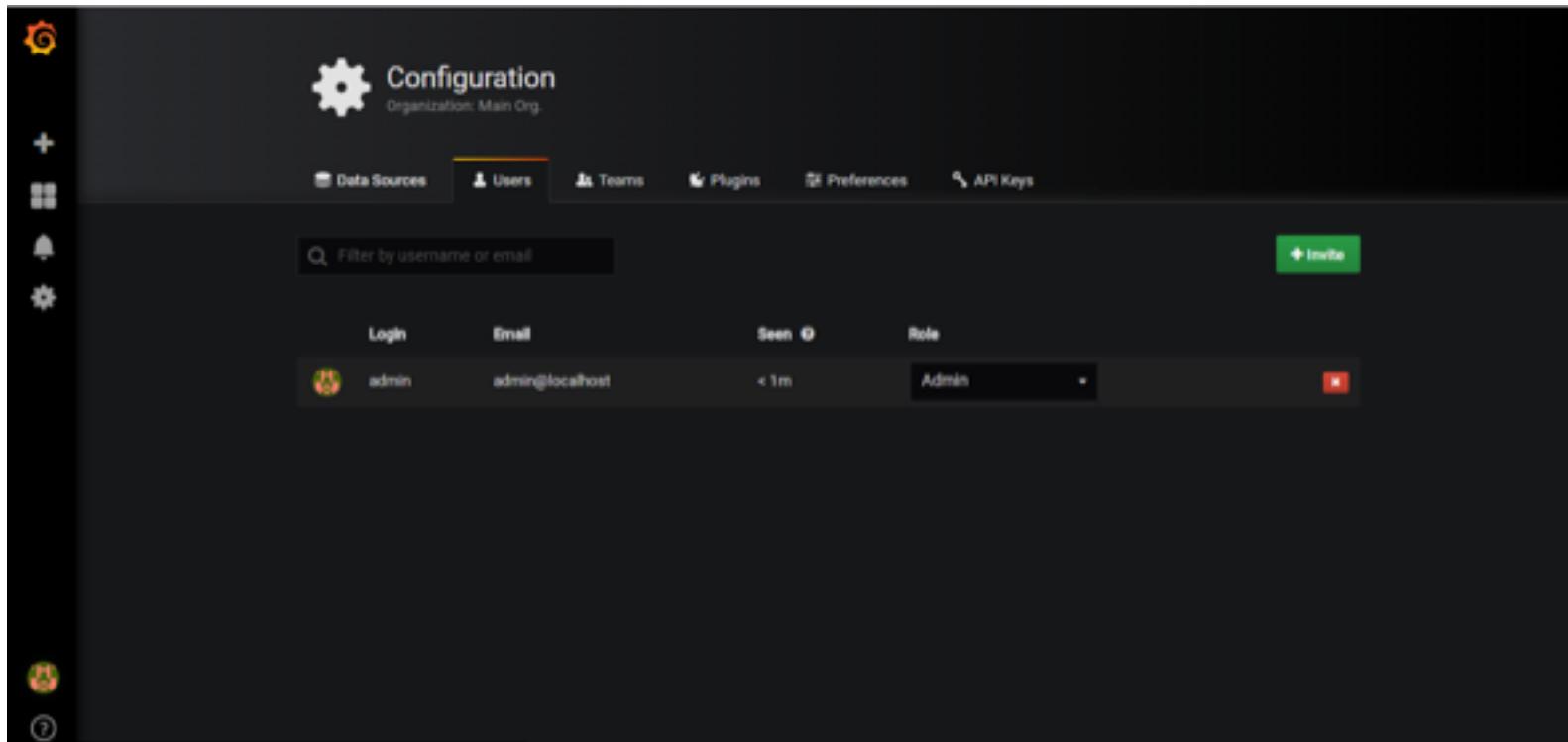
Installed Panels

None installed. Browse Grafana.com

# การใช้งาน Grafana

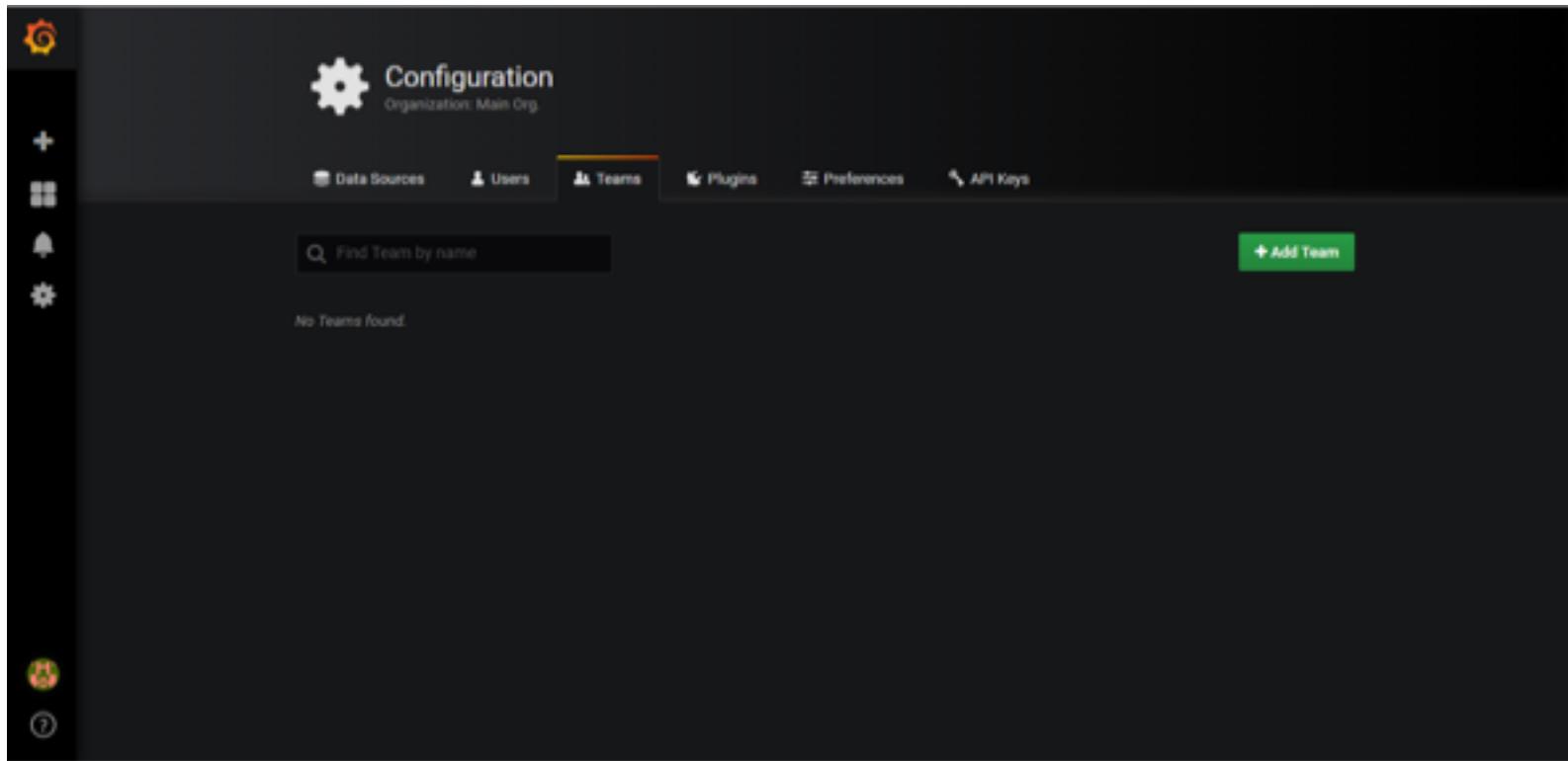


# การใช้งาน Grafana

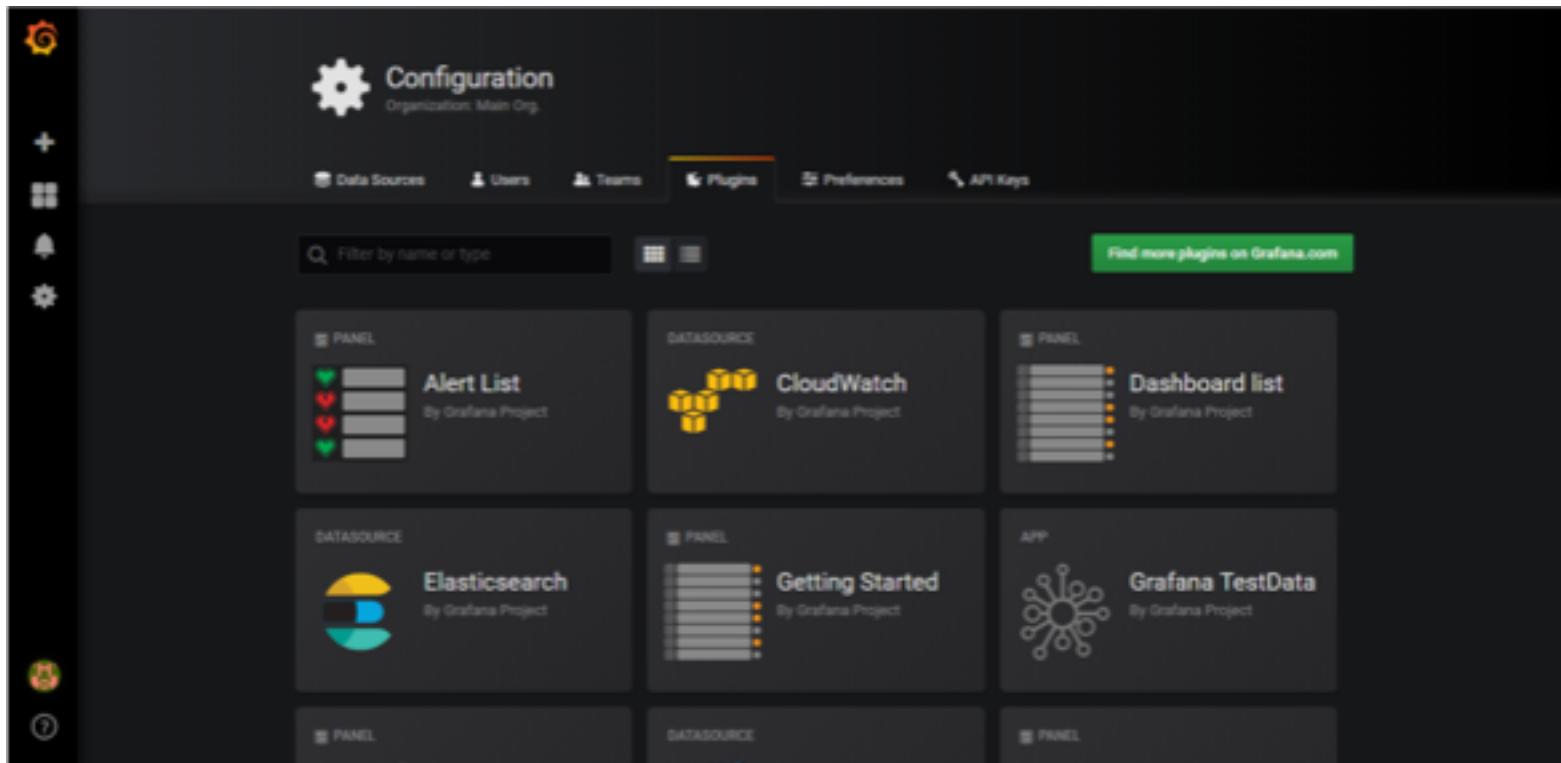


The screenshot shows the Grafana Configuration interface. The top navigation bar includes links for Data Sources, Users (which is currently selected), Teams, Plugins, Preferences, and API Keys. A search bar allows filtering by username or email, and a green "Invite" button is visible. The main content area displays a table with columns for Login, Email, Seen, and Role. One user entry is shown: "admin" (Email: admin@localhost, Last Seen: < 1m, Role: Admin). A red "Delete" button is located to the right of the role dropdown for this user.

# การใช้งาน Grafana



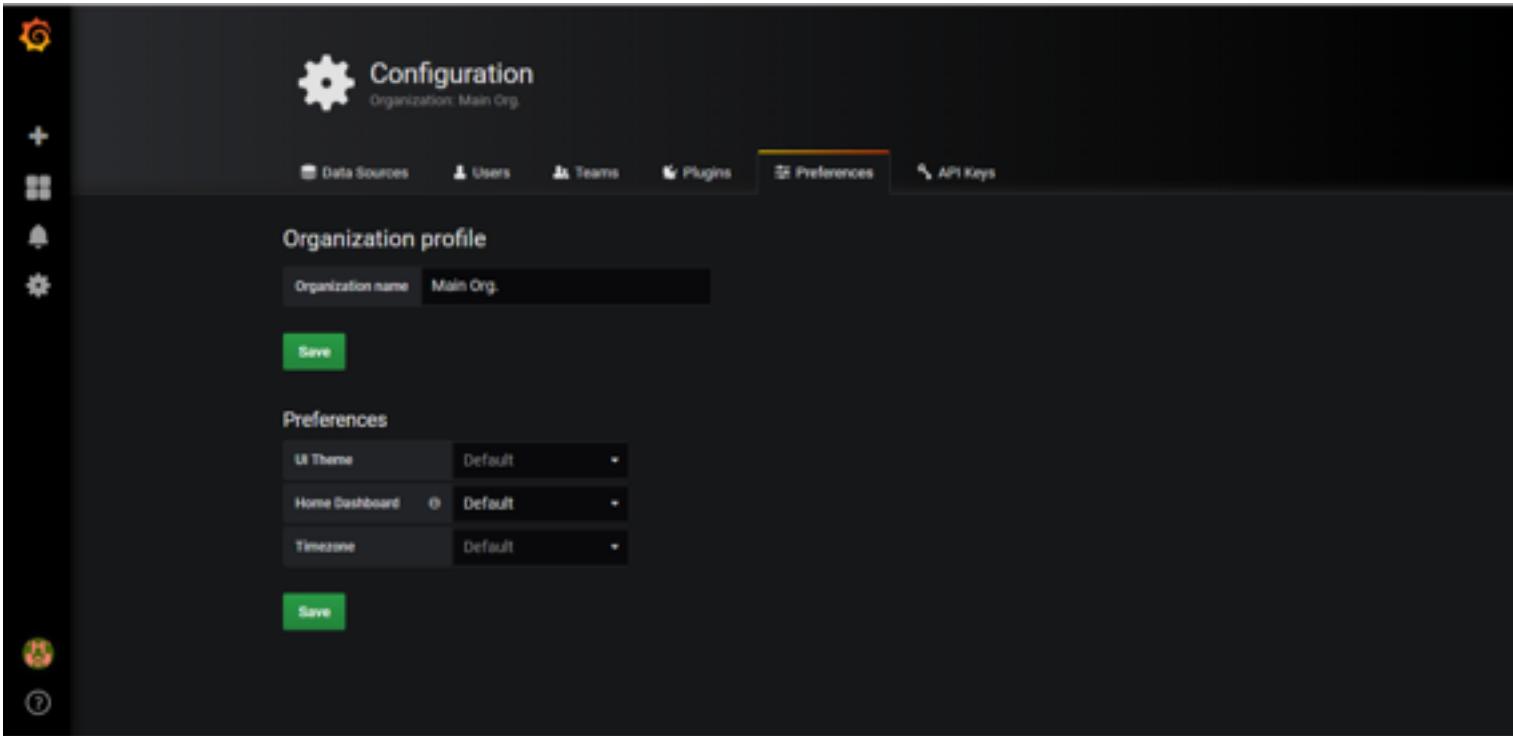
# การใช้งาน Grafana



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

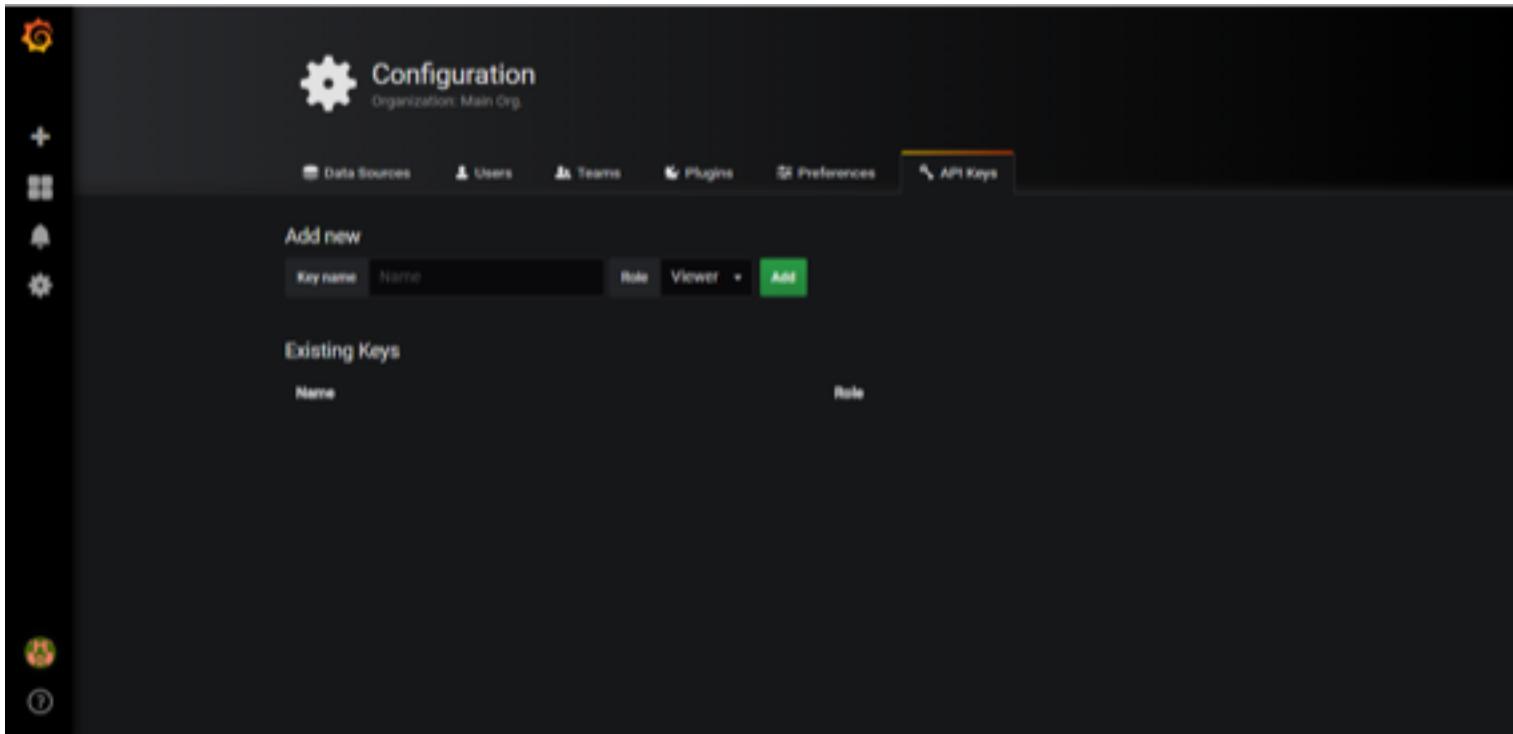
- Organization:** Main Org.
- Navigation:** Data Sources, Users, Teams, Plugins (selected), Preferences, API Keys.
- Search:** Filter by name or type.
- Buttons:** Find more plugins on Grafana.com.
- Grid of Plugins:**
  - PANEL:** Alert List (By Grafana Project)
  - DATASOURCE:** CloudWatch (By Grafana Project)
  - PANEL:** Dashboard list (By Grafana Project)
  - DATASOURCE:** Elasticsearch (By Grafana Project)
  - PANEL:** Getting Started (By Grafana Project)
  - APP:** Grafana TestData (By Grafana Project)

# การใช้งาน Grafana



The screenshot shows the Grafana Configuration interface. At the top, there's a sidebar with icons for Data Sources, Users, Teams, Plugins, Preferences (which is selected), and API Keys. The main area has a dark header with a gear icon and the word "Configuration". Below it, it says "Organization: Main Org.". There's a section for "Organization profile" with a "Save" button. Under "Preferences", there are dropdown menus for "UI Theme" (set to Default), "Home Dashboard" (set to Default), and "Timezone" (set to Default). Each preference has its own "Save" button. On the far left of the main content area, there are two small circular icons: one with a question mark and another with a gear.

# การใช้งาน Grafana



The screenshot shows the Grafana Configuration interface with the "API Keys" tab selected. On the left, there's a sidebar with icons for Data Sources, Users, Teams, Plugins, Preferences, and API Keys. The main area has a dark header with "Configuration" and "Organization: Main Org.". Below the header are tabs for Data Sources, Users, Teams, Plugins, Preferences, and API Keys. The "API Keys" tab is highlighted with a yellow background. A sub-header "Add new" is followed by input fields for "Key name" (with placeholder "Name") and "Role" (set to "Viewer"). A green "Add" button is next to the role dropdown. Below this is a section titled "Existing Keys" with columns for "Name" and "Role".

# 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/>

# Any questions?

