

DevOps

เครื่องมือสำหรับมืออาชีพ

ยกระดับความรู้และทักษะการใช้เทคโนโลยีที่เกี่ยวกับ DevOps

Agenda

Agenda

Day 1

- What's DevOps
- DevOps tools
- Proxmox
- Container
- Docker
- Install recommended tools

Day 2

- Git version controls
- Gitlab CI/CD
- Dockerfile
- Docker compose
- Portainer.io
- Auto deploy

DevOps

DevOps

Development (Dev) + Operations (Ops)

DevOps

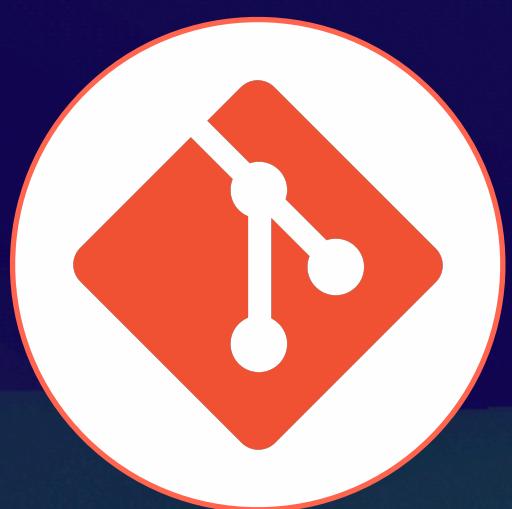
Development (Dev) + Operations (Ops)

คือ แนวคิดและวิธีการที่รวมกระบวนการ “พัฒนา” และ “ปฏิบัติการ”
เข้าด้วยกัน เพื่อส่งมอบซอฟต์แวร์ได้เร็วขึ้น เสถียรขึ้น และอัตโนมัติมากขึ้น

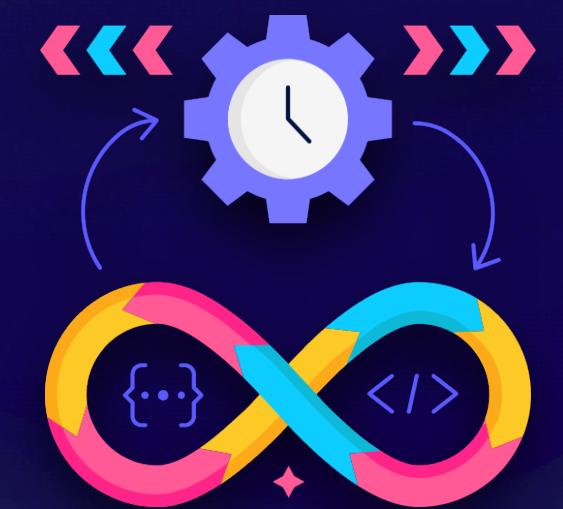
DevOps have to know



DEV



Version
Control



CI / CD



VM &
Container



LINUX



Network &
Security



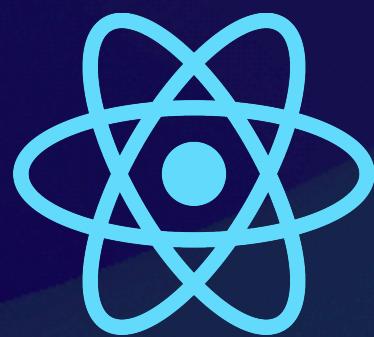
Cloud



Infrastructure

DevOps - dev

FRONTEND



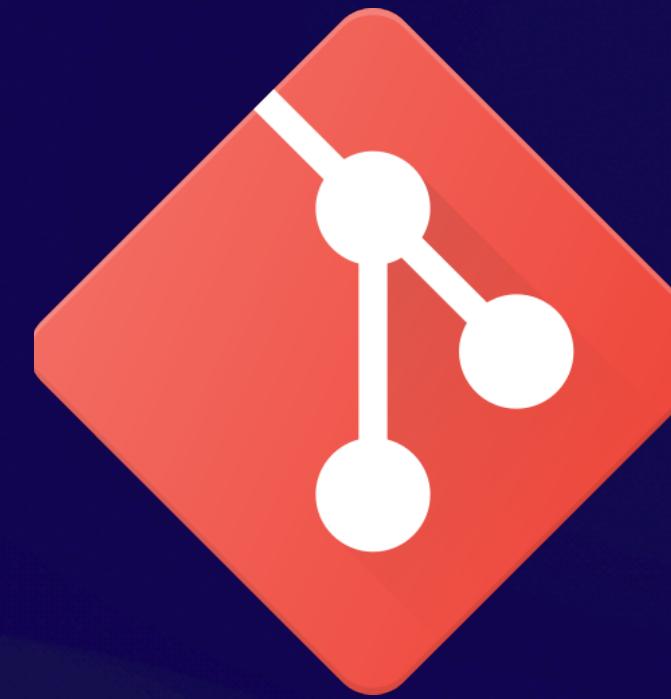
BACKEND



DevOps - Version Control System



Git คือ ระบบจัดการเวอร์ชันของซอฟต์แวร์ (Version Control System)



Trip: git command

git init

เริ่มต้นโปรเจกต์

git status

ดูการเปลี่ยนแปลงของไฟล์

git add .

เตรียมไฟล์

git commit -m “ข้อความ”

บันทึกเวอร์ชัน

git push

ส่งการเปลี่ยนแปลงไปยัง server

git pull

ดึงการเปลี่ยนแปลงจาก server

git branch

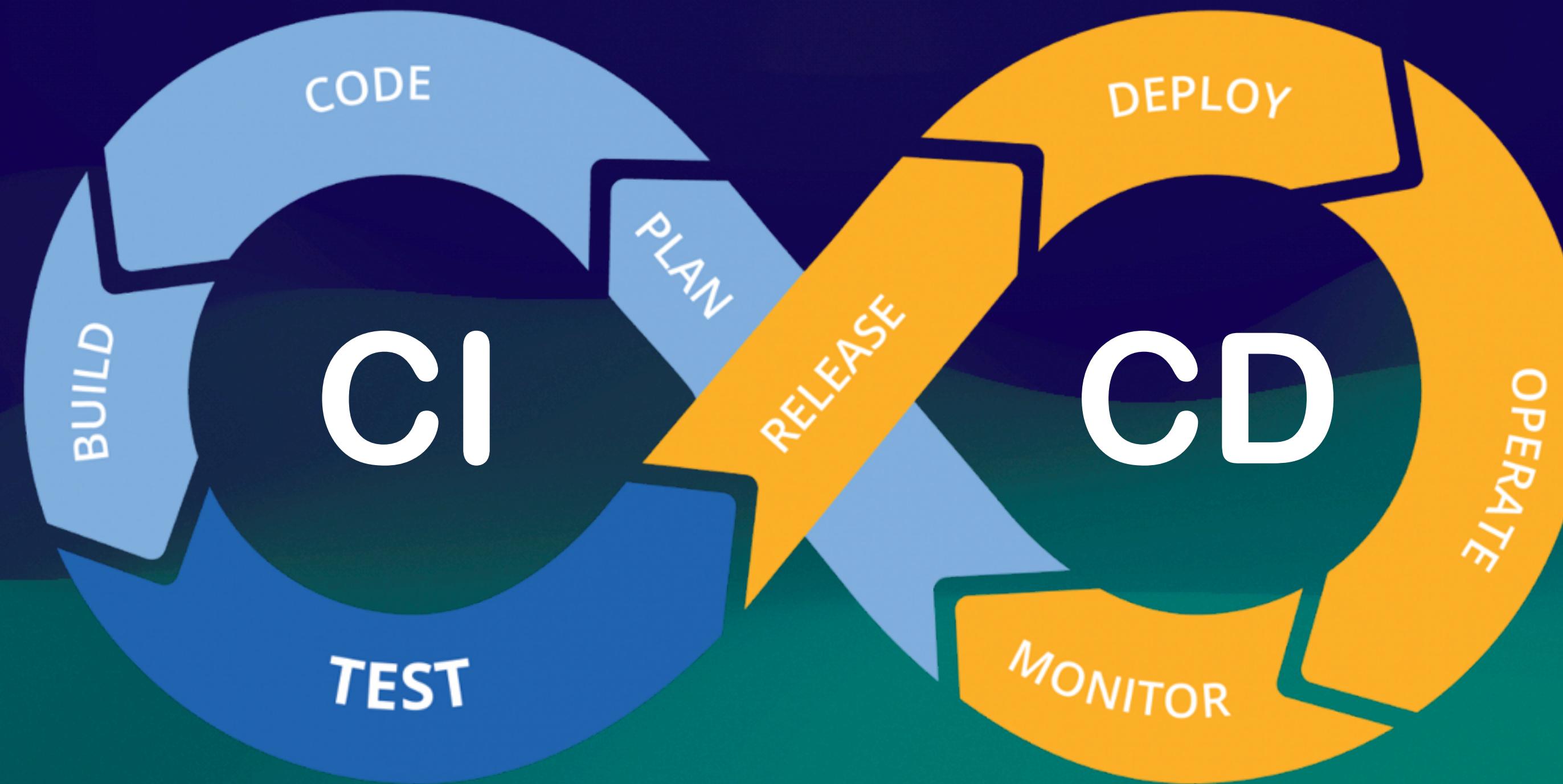
ตรวจสอบ branch ที่มี

git checkout branch_name # เปลี่ยน branch

ดึงข้อมูลใน git repo มาใช้

git clone

DevOps - CI/CD



Continuous
Integration

Continuous
Delivery

DevOps - CI/CD

Best CI/CD Tools



GitHub Actions



CI CD



Jenkins



argo



Bamboo



circleci



axify



Buddy



Spinnaker



TeamCity



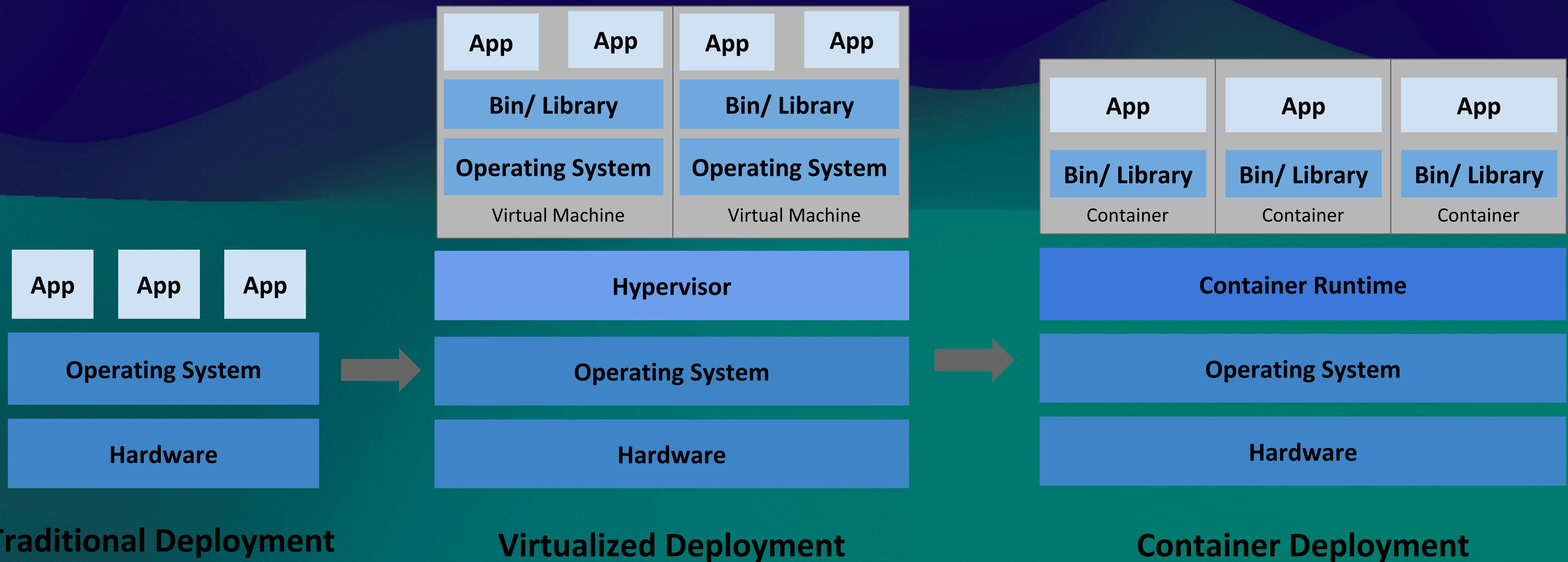
Azure



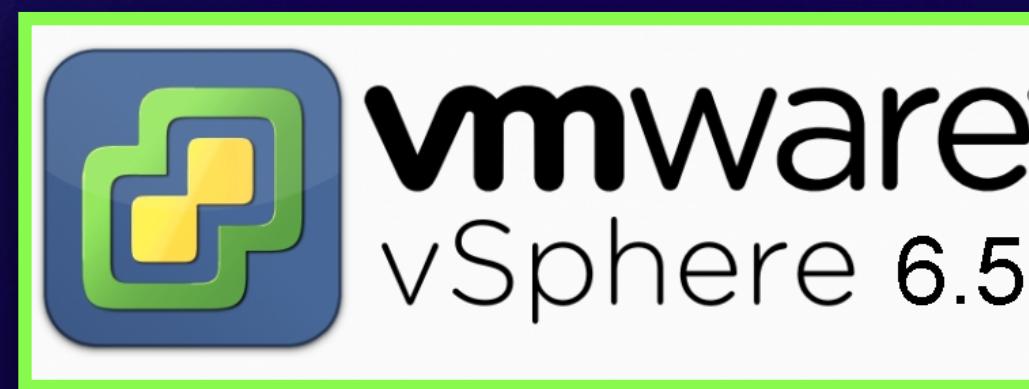
Concourse

DevOps - VM & Container

“ก่อนจะมี Container เราใช้ Virtual Machine (VM) มาก่อน”



DevOps - VM & Container | virtualisations



Linux OS | ทำไมต้องใช้ Linux กับ DevOps?

มากกว่า 90% ของเซิร์ฟเวอร์บน Cloud

- เช่น AWS, Azure, GCP ใช้ Linux (เช่น Ubuntu, CentOS, Alpine, Debian)

Web Server, Database, Container Platform อย่าง Kubernetes ก็ใช้ Linux เป็นฐาน

ระบบ Embedded / Edge / IoT / AI ส่วนใหญ่ใช้ Linux ด้วยเช่นกัน

Linux OS | ตระกูลของ Linux

Debian-based : ใช้งานง่าย, community ใหญ่ (apt/dpkg)

Red Hat-based : เลสี่ยรระดับองค์กร, enterprise (dnf/yum)

Arch-based : Rolling release, เร็ว, เบา (pacman)

Alpine Linux : เบามาก (~5MB), Hardened Security (apk)

Slackware-based : ดั้งเดิม, ไม่มี dependency auto (slackpkg)

Gentoo : Compile ทุกอย่างเอง, ยืดหยุ่นสูง (emerge (Portage))

Void Linux : เบา, ไม่มี systemd (xbps)

NixOS : ใช้ declarative config, reproducible (nix-env)

Linux OS | ที่ควรรู้



Linux OS | ทำไม DevOps ต้องเรียนรู้ Linux

- ✓ ใช้ในระบบจริงทุก Production
- ✓ เป็นฐานของทุก Container และ Cloud
- ✓ Automate ได้ด้วย Bash และ Shell
- ✓ รองรับ Tools ทุกชนิด ในสาย DevOps
- ✓ คุ้มค่า เรียนครั้งเดียว ใช้ได้ตลอดชีวิตสายงานไอที





<https://www.proxmox.com/en/downloads>

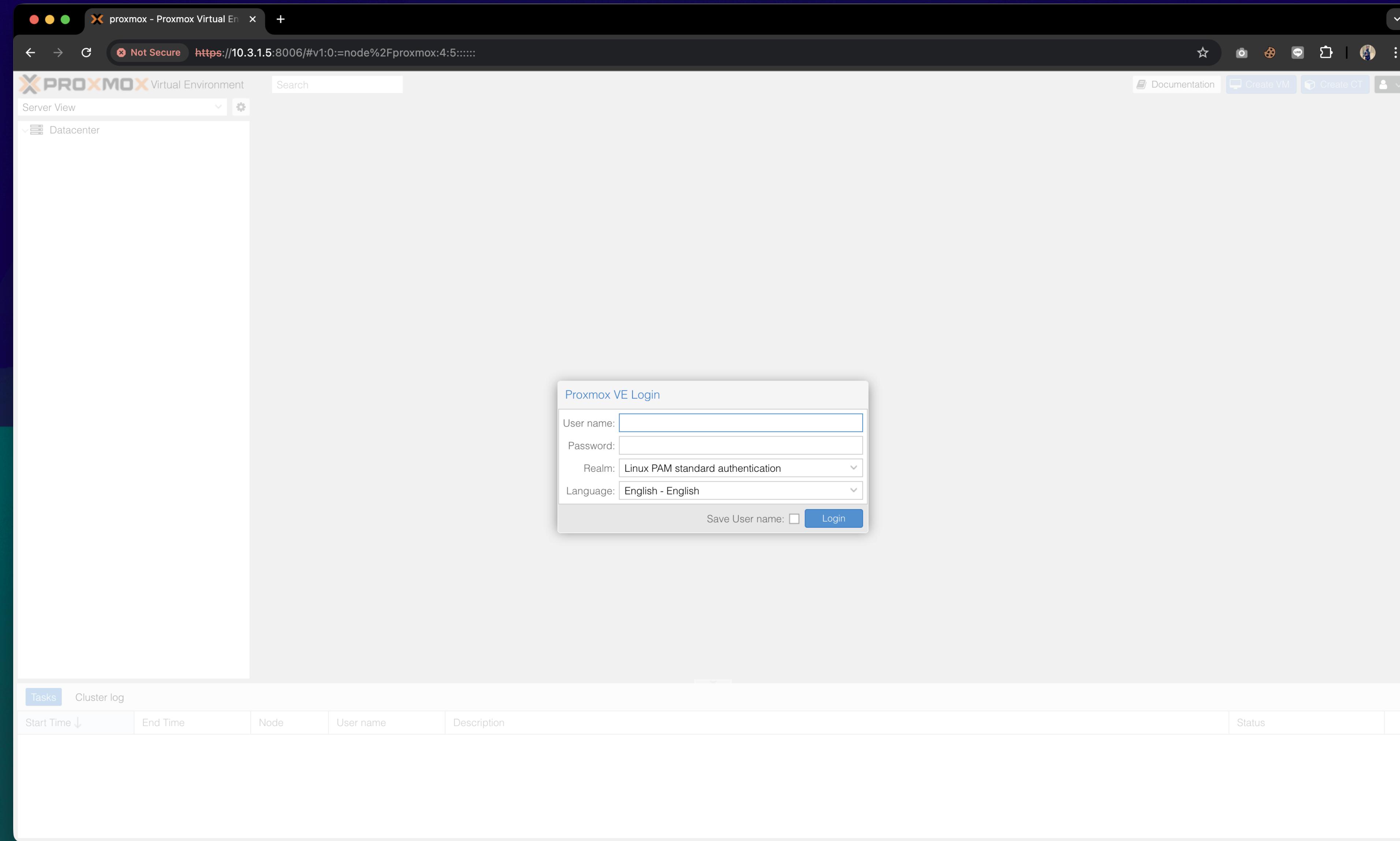
The screenshot shows the Proxmox Downloads page with a dark background. At the top, there's a navigation bar with links for Home, Products, Downloads (which is highlighted in orange), Services, Partners, and About. There are also links for Get Help, Buy now, a search icon, and a globe icon. Below the navigation, there's a section titled "Downloads" with three buttons: "Proxmox Virtual Environment", "Proxmox Backup Server", and "Proxmox Mail Gateway". A section titled "Latest Releases" follows, featuring three items:

- Proxmox VE 9.0 ISO Installer**
Version: 9.0-1 | File Size: 1.64 GB | Last Updated: August 05, 2025
SHA256SUM: 228f948ae696f2448460443f4b619157cab78ee69802acc0d06761ebd4f51c3e
Download | Torrent
- Proxmox Backup Server 4.0 ISO Installer**
Version: 4.0-1 | File Size: 1.34 GB | Last Updated: August 06, 2025
SHA256SUM: 208607b250164863b5731a29dd89569a123e6f385c5ec0939a4942357bf731e2
Download | Torrent
- Proxmox Mail Gateway 8.2 ISO Installer**
Version: 8.2-1 | File Size: 1.58 GB | Last Updated: February 27, 2025
SHA256SUM: 2a348db5bf588450d1d47a8746d3713efc390c7fe1fc5ddf379045542999b549
Download | Torrent



Installation

PROXMOX Login





Dashboards

Not Secure https://10.3.1.5:8006/#v1.0:=node%2Fproxmox:4.5::::

Documentation Create VM Create CT root@pam

Reboot Shutdown Shell Bulk Actions Help

Hour (average)

Server View Node 'proxmox'

Datacenter proxmox

- localnetwork (proxmox)
- local (proxmox)
- local-lvm (proxmox)

Search Package versions

Summary Notes Shell System Network Certificates DNS Hosts Options Time System Log Updates Repositories Firewall Disks LVM LVM-Thin Directory ZFS Ceph Replication Task History Subscription

proxmox (Uptime: 1 day 03:16:25)

CPU usage 0.00% of 16 CPU(s) IO delay 0.00%

Load average 0.08, 0.07, 0.02

RAM usage 3.30% (2.06 GiB of 62.21 GiB) KSM sharing 0 B

/ HD space 3.14% (2.95 GiB of 93.93 GiB) SWAP usage 0.00% (0 B of 8.00 GiB)

CPU(s) 16 x AMD Ryzen 9 5900HX with Radeon Graphics (1 Socket)

Kernel Version Linux 6.8.12-9-pve (2025-03-16T19:18Z)

Boot Mode EFI

Manager Version pve-manager/8.4.0/ec58e45e1bcdff2ac

Repository Status Production-ready Enterprise repository enabled Enterprise repository needs valid subscription

CPU usage

0.16
0.14
0.12
0.10
0.08
0.06
0.04
0.02
0 B

02:28:00 02:38:00 02:48:00 02:58:00 03:08:00 03:18:00 03:28:00 03:37

Server load

Load average 0.2
0.18
0.16
0.14
0.12
0.10
0.08
0.06
0.04
0.02
0 B

02:28:00 02:38:00 02:48:00 02:58:00 03:08:00 03:18:00 03:28:00 03:37

Memory usage

64 Gi
56 Gi
48 Gi
40 Gi
32 Gi
24 Gi
16 Gi
8 Gi
0 Bytes

02:28:00 02:38:00 02:48:00 02:58:00 03:08:00 03:18:00 03:28:00 03:37

Network traffic

netin netout 30 k

Tasks Cluster log

Start Time	End Time	Node	User name	Description	Status
Jul 06 07:21:42	Jul 06 07:21:42	proxmox	root@pam	Bulk start VMs and Containers	OK
Jul 06 00:24:43	Jul 06 00:24:52	proxmox	root@pam	Shell	OK



VM and LXC

 ส่วนประกอบ:

- Hypervisor (เช่น KVM ใน Proxmox)
- Guest OS (เช่น Ubuntu, Windows)
- Virtual CPU, RAM, Disk, Network

 จุดเด่น:

- รัน OS ได้หลายแบบ (Linux, Windows, BSD)
- แยกจาก Host อย่างปลอดภัย
- เหมาะกับงานที่ต้องการ Isolation สูง เช่น Production Server, OS ที่ไม่มี Container

 ข้อเสีย:

- ใช้ทรัพยากรากกว่า LXC
- Boot ช้ากว่า Container

 ส่วนประกอบ:

- แชร์ Kernel กับ Host (ไม่ต้อง boot kernel เอง)
- มี Root FS, Process, Network แยกจาก Host
- ทำงานเหมือน VM แต่เบากว่า

 จุดเด่น:

- เบากว่า VM มาก (Boot ภายใน 1-2 วิ)
- ใช้ RAM/CPU น้อยกว่า
- เหมาะสมกับงานที่ต้องการ Isolation สูง เช่น Lab นักศึกษา, Dev/Test Environment, ระบบเบา ๆ เช่น GitLab Runner, Web Service

 จุดที่ควรระวัง:

- Kernel ต้องเหมือน Host (เช่น Host เป็น Linux → LXC ก็ต้อง Linux)
- Isolation ไม่เท่า VM (เหมาะสมกับงานที่ควบคุมได้)



VM (Virtual Machine)

Create VM: Ubuntu 24.04.3 LTS

- Download ISO: <https://releases.ubuntu.com/24.04.3/ubuntu-24.04.3-live-server-amd64.iso>
- Cloud Image: <https://cloud-images.ubuntu.com/noble/current/noble-server-cloudimg-amd64.img>
- Create VM with Cloud-Init drive
- Specification (HA Docker Swarm)
 - 3 nodes, vCPU 2-4, RAM 4-8GB, Disk1 20GB, Disk2 100GB

Next station



Container Runtime และเครื่องมือ DevOps ยอดนิยม





<https://www.docker.com>

<https://github.com/yep-piggysmile/devops-docker-trainings.git>



Docker

Docker คือ แพลตฟอร์มซอฟต์แวร์แบบโอเพนซอร์ส ที่ช่วยให้การสร้าง ทดสอบ และปรับใช้อัปพลิเคชันเป็นไปอย่างรวดเร็วและง่ายดาย โดย Docker จะรวมซอฟต์แวร์ทั้งหมดที่จำเป็นสำหรับการรันแอปพลิเคชันไว้ใน "คอนเทนเนอร์" (Container) ซึ่งเป็นหน่วยมาตรฐานที่มีทุกอย่างที่แอปพลิเคชันต้องการ เช่น ไลบรารี โค้ด และการตั้งค่าต่างๆ ทำให้อัปพลิเคชันสามารถทำงานได้เหมือนกันในทุกสภาพแวดล้อม โดยไม่ต้องกังวลกับปัญหาความเข้ากันได้ของระบบปฏิบัติการ



Docker | Installation & Join cluster

Manager node: swarm01

```
~# curl -fsSL https://get.docker.com | sh  
~# sudo usermod -aG docker $USER  
~# docker swarm init --advertise-addr <MANAGER-IP>  
* <MANAGER-IP> ให้เปลี่ยนเป็น IP ของ swarm01
```

Worker node: swarm02, swarm03

```
~# curl -fsSL https://get.docker.com | sh  
~# sudo usermod -aG docker $USER  
~# docker swarm join --token <TOKEN> <MANAGER-IP>:2377  
* <TOKEN> ให้เปลี่ยนเป็น token ที่มาจากการเครื่อง swarm01  
** <MANAGER-IP> ให้เปลี่ยนเป็น IP ของ swarm01
```



Docker | Install portainer

- Web UI สำหรับ Docker/Swarm
- บริหาร Container/Stack ได้ง่าย

Manager node: swarm01

```
~# docker volume create portainer_data  
~# docker run -d -p 9443:9443 --name portainer \  
--restart=always -v /var/run/docker.sock:/var/run/docker.sock \  
-v portainer_data:/data portainer/portainer-ce:latest
```



Docker | Install Private DNS : AdGuard Home

ตัวเลือก:

- dnsmasq (VM/LXC) → เบา ติดตั้งง่าย
- Pi-hole / AdGuard Home (Docker) → มี UI, รองรับ DHCP
- CoreDNS (Docker/Swarm) → Production grade

Manager node: swarm01

```
~# docker run -d --name adguardhome \
-v /opt/adguardhome/work:/opt/adguardhome/work \
-v /opt/adguardhome/conf:/opt/adguardhome/conf \
-p 53:53/tcp -p 53:53/udp -p 3000:3000/tcp adguard/adguardhome
```



Docker | Install Traefik

- Reverse Proxy + Load Balancer + ACME
- Map private domain (*.local.lab)

Manager node: swarm01

```
~# docker network create --driver=overlay traefik-net
~# docker service create --name traefik --constraint=node.role==manager \
--publish 80:80 --publish 443:443 \
--mount type=bind,src=/var/run/docker.sock,dst=/var/run/docker.sock \
--network traefik-net traefik:v3.1 --api.insecure=true --providers.docker
```



Docker | Install GitLab + Registry + Runner

```
version: "3.8"
services:
  gitlab:
    image: gitlab/gitlab-ce:latest
    hostname: gitlab.local.lab
    ports:
      - "80:80"
      - "443:443"
    volumes:
      - gitlab_data:/var/opt/gitlab
      - gitlab_logs:/var/log/gitlab
      - gitlab_config:/etc/gitlab
    labels:
      - "traefik.enable=true"
      - "traefik.http.routers.gitlab.rule=Host(`gitlab.local.lab`)"
      - "traefik.http.services.gitlab.loadbalancer.server.port=80"

  registry:
    image: registry:2
    ports:
      - "5000:5000"
    volumes:
      - registry_data:/var/lib/registry

  runner:
    image: gitlab/gitlab-runner:latest
    volumes:
      - /var/run/docker.sock:/var/run/docker.sock
      - gitlab_runner:/etc/gitlab-runner

volumes:
  gitlab_data:
  gitlab_logs:
  gitlab_config:
  registry_data:
  gitlab_runner:
```

Q & A