



X



Docker Workshop

with Online Zoom Session

Instructors

Sirinat Paphatsirinatthi (Oam)

Co-Founder @ KubeOps Skills

CNCF Certified Kubernetes Application Developer (CKAD)

CNCF Certified Kubernetes Administrator (CKA)



Saritrat Jirakulphondchai (JoJo)

Co-Founder & CTO @ KubeOps Skills

CNCF Certified Kubernetes Application Developer (CKAD)

CNCF Certified Kubernetes Administrator (CKA)

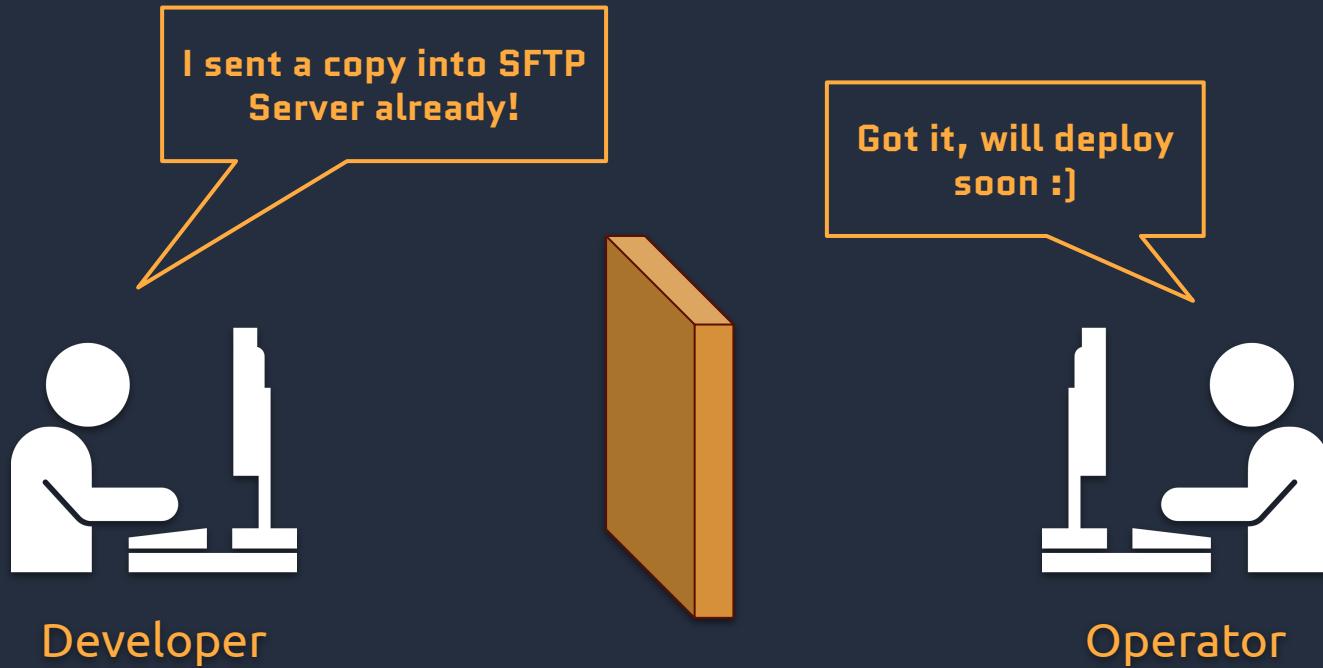
Agenda

- Old School Development Way
- Introduction to Container
- Why we need containers?
- Docker vs Virtual Machine
- Docker overview
- Docker Build
- Docker Registry
- Docker Run

Agenda (Cont.)

- How Distroless
- Jib vs Dockerfile
- Docker Storage
- Docker Network
- Docker Architecture
- Docker Compose
- Container Orchestration
- Docker Stack and Kubernetes
- Best practices
- DevOps world

Old school Way



Deployment Day 🔥

Old school Way



Developer



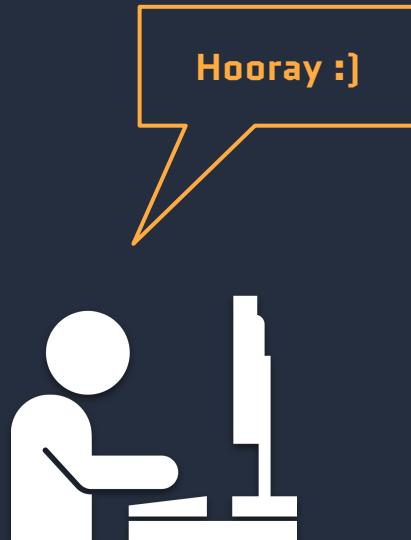
Deploying....



Operator

Deployment Day 🔥

Old school Way



Deployed
Completed!



Developer

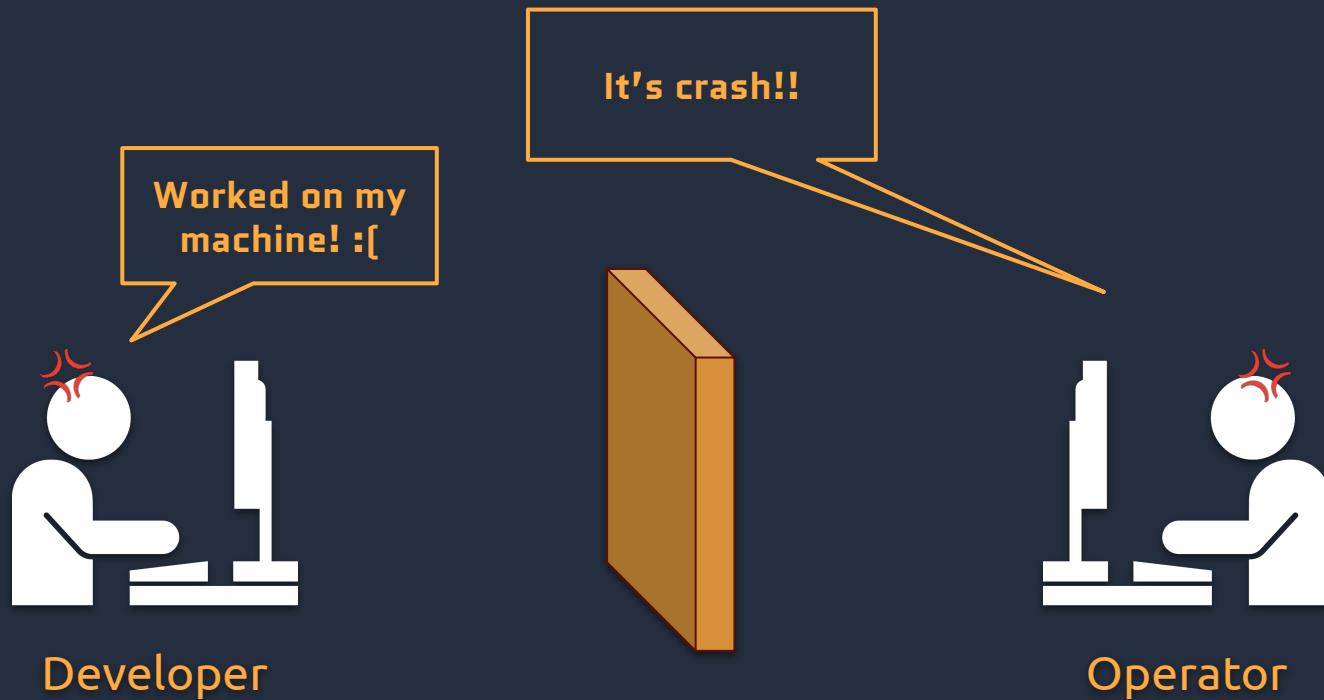
Deployment Day 🔥



Operator

Old school Way

Deployment Day 🔥



What is Container?

Help packaging application's code

- Container helps to bundle an application's code together with the related configuration files and libraries, and with the dependencies required for the app to run.

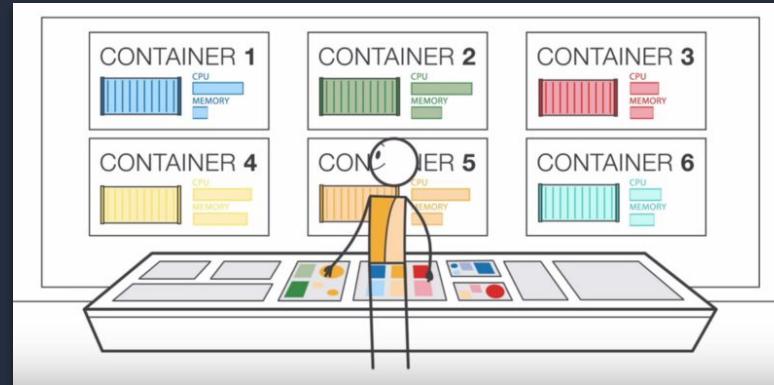
Make sure every environment is the same

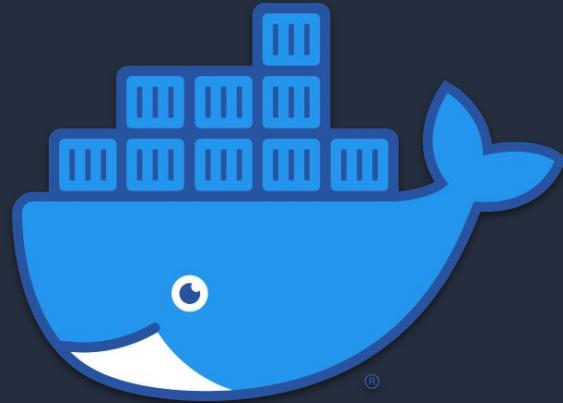
- Containers address the problem of an application failing to run correctly when moved from one environment to another.

What is container? (Cont.)

Portable applications

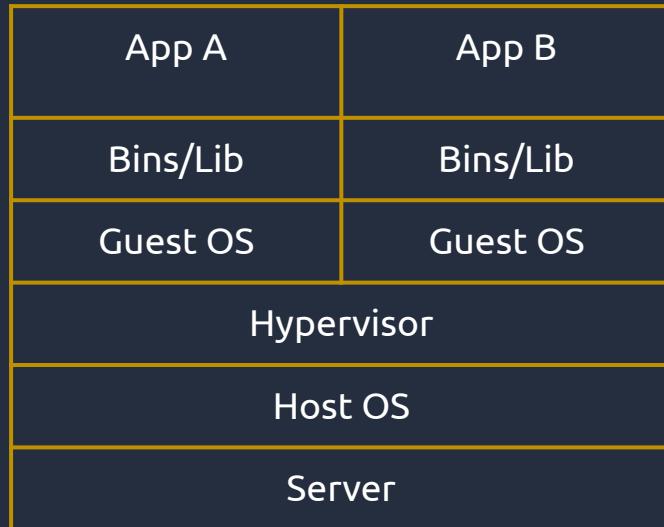
- An application, its dependencies, and its configuration are packaged together as a container image. The container image instance could be deployed to the host OS for each environment.



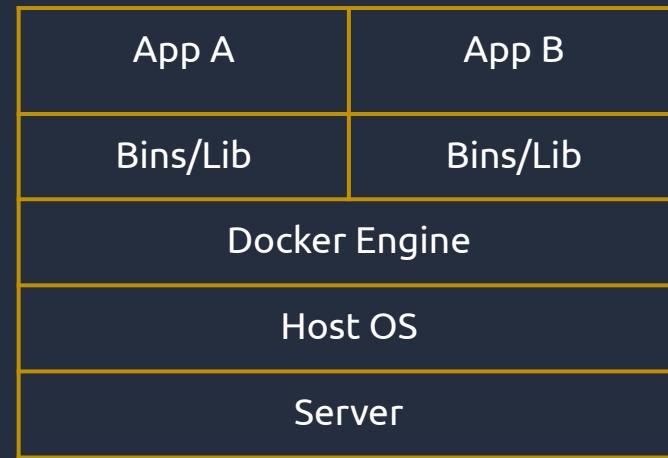


The most popular container manager is Docker!

Docker vs Virtual Machine



VS



vmware



docker



Docker Overview



Docker Build

What is Dockerfile?

- Dockerfile describes step by step instructions of all the commands you need to run to assemble a Docker Image.

“Docker Build” command

- “docker build” command processes Dockerfile generating a Docker Image in your Local Image Cache, which you can then start-up using the “docker run” command, or push to a permanent Image Repository

```
$ docker build -t <IMAGE_NAME> .
```

Docker Registry



What is Docker Registry?

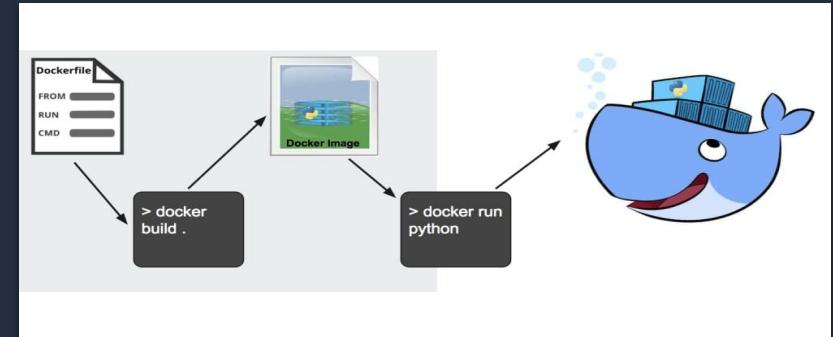
- A registry is a storage and content delivery system, holding named Docker images, available in different tagged versions. eg. Docker Hub, Quay.io, GCR, ECR

“Docker Push” command

- “docker push” command helps you to push Docker images to the Docker registry.

```
$ docker push [IMAGE:TAG]
```

Docker Run



“Docker Run” command

- “`docker run`” command helps you to run Docker image as a container instance.

General Command Form

- The basic “`docker run`” command takes this form:

```
$ docker run [OPTIONS] IMAGE[:TAG|@DIGEST] [COMMAND] [ARG...]
```

Docker Build (Demo)



Dockerfile

build
→



image

Docker Push - (Demo)



Docker Basic

with Real Action Simulator

[Click Here](#)



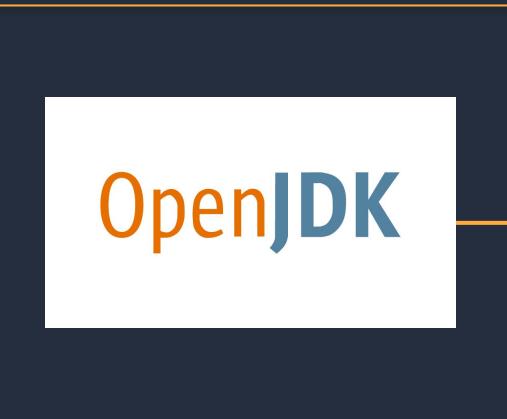
Is it good to run a container with Alpine application runtime image?

JDK Alpine Image



Distroless Image

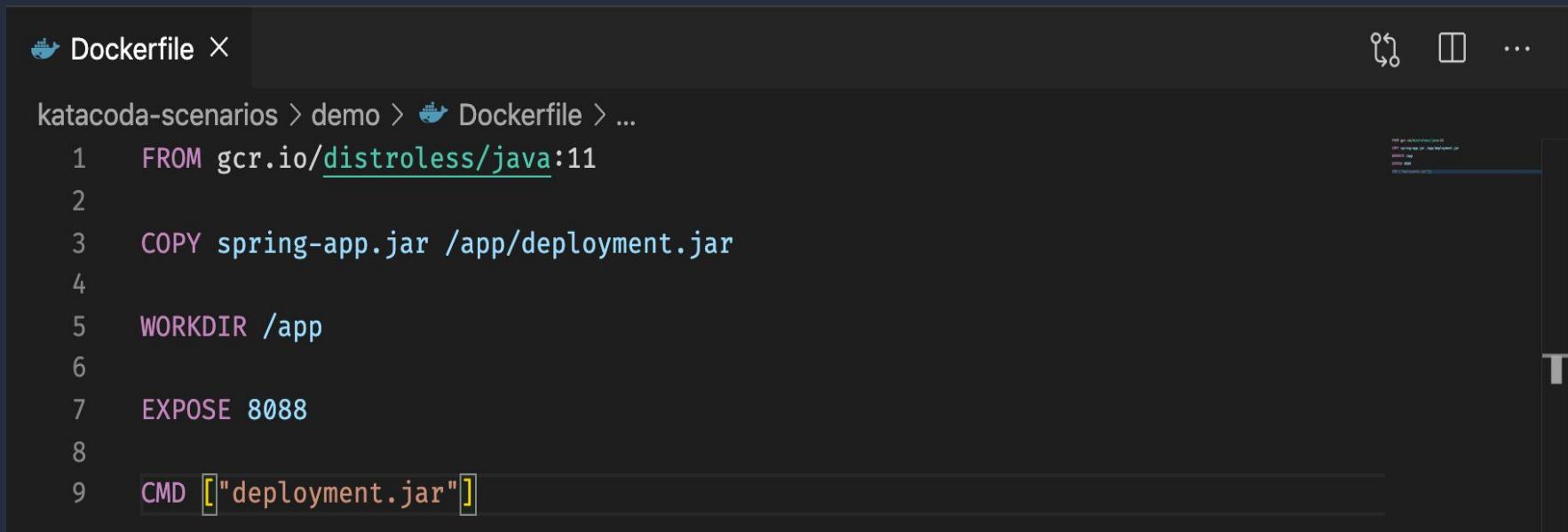
Container



Contains only JDK and it does not provide any OS packages even shell commands.

JDK (Application Runtime)

Distroless Image Docker file



A screenshot of a Dockerfile editor interface. The title bar says "Dockerfile X". The path is "katacoda-scenarios > demo > Dockerfile > ...". The Dockerfile content is:

```
1 FROM gcr.io/distroless/java:11
2
3 COPY spring-app.jar /app/deployment.jar
4
5 WORKDIR /app
6
7 EXPOSE 8088
8
9 CMD ["deployment.jar"]
```

The code editor has syntax highlighting for Docker commands. A status bar at the bottom right shows "T" and "KubeOps skills".

Distroless Image Demo

with Real Action Simulator

[Click Here](#)

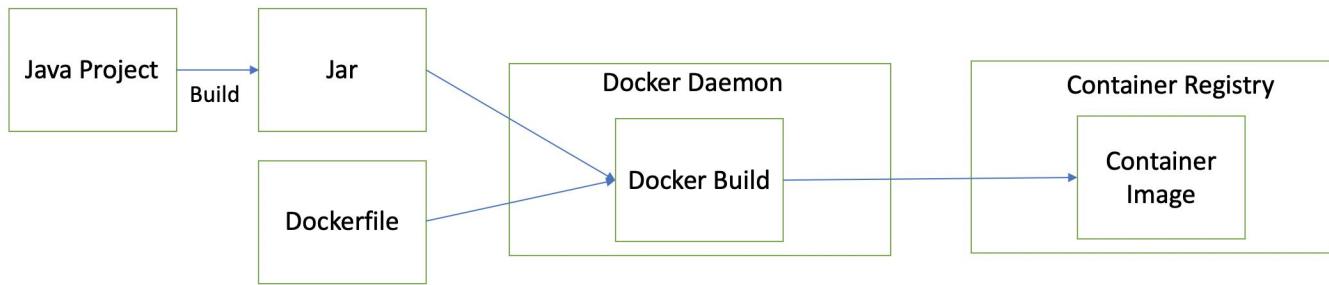


But, Are there any other ways to optimize these?

Can developers focus on what matter most? (writing code)

Jib Plugin

Docker Build Flow



Jib Build Flow



Jib Plugin Demo

with Real Action

By IntelliJ



Docker Storage

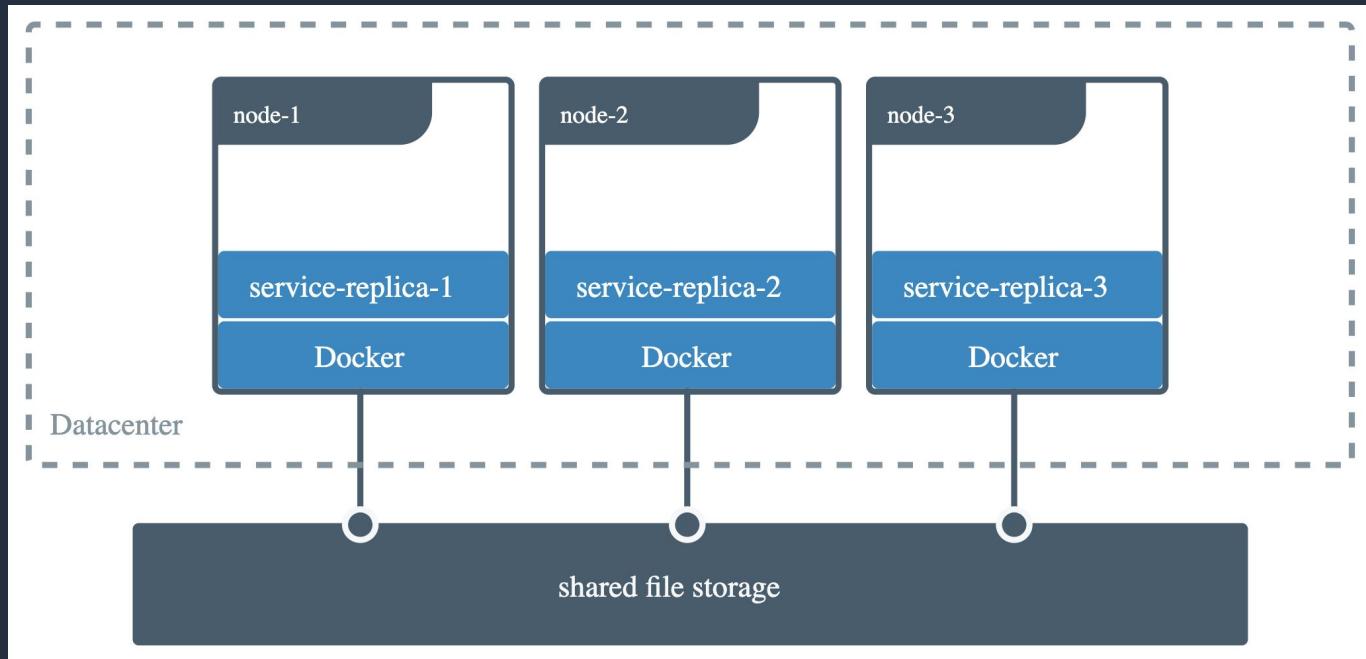
Storage By default

- all files created inside a container are stored on a writable container layer.

That mean:

- The data doesn't persist when that container no longer exists, and it can be difficult to get the data out of the container if another process needs it.
- A container's writable layer is tightly coupled to the host machine where the container is running. You can't easily move the data somewhere else.

Docker Storage



Docker Network

“Docker network” command

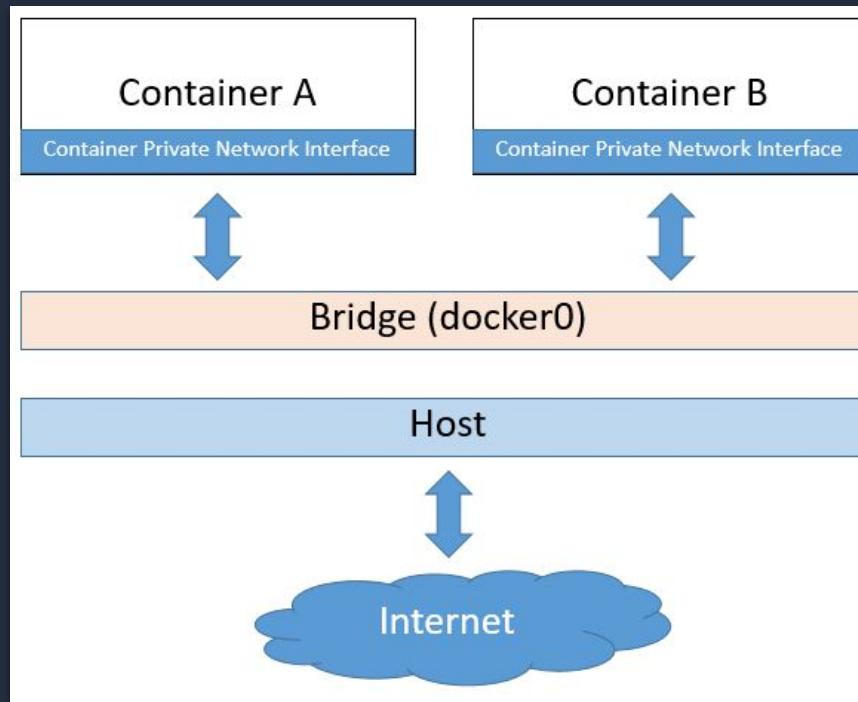
- “docker network” command helps you to connect, disconnect, create, list, inspect network in the docker environment.

General Command Form

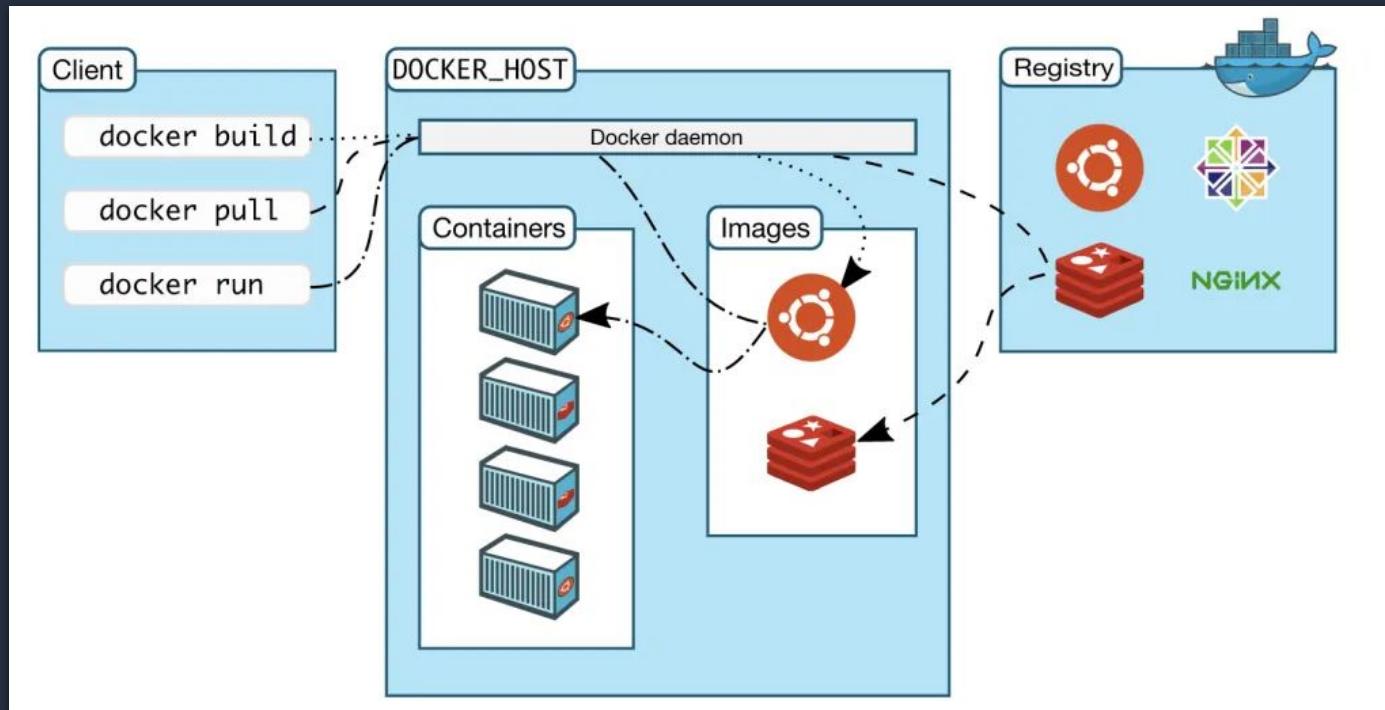
- The basic “docker network” command takes this form:

```
$ docker network [COMMAND]
```

Docker Network



Docker Architecture



Docker Integration

with Real Action Simulator

[Click Here](#)



Docker Compose

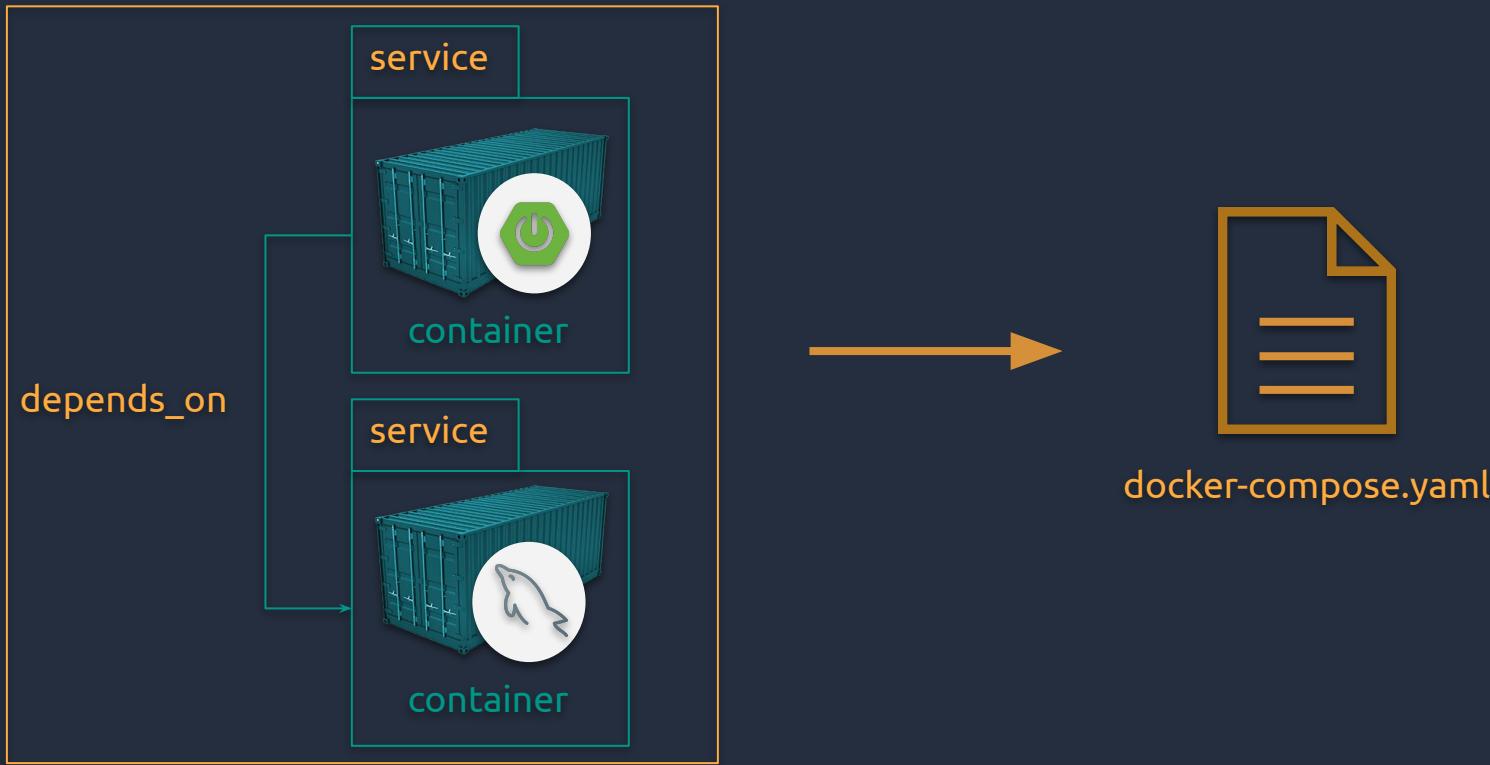
Docker Compose

- The Compose file is a YAML file defining services, networks and volumes.
The default path for a Compose file is `./docker-compose.yml`

Docker Compose Tip

Tip: You can use either a `.yml` or `.yaml` extension for this file. They both work.

Docker Compose - (Demo)



Docker Compose

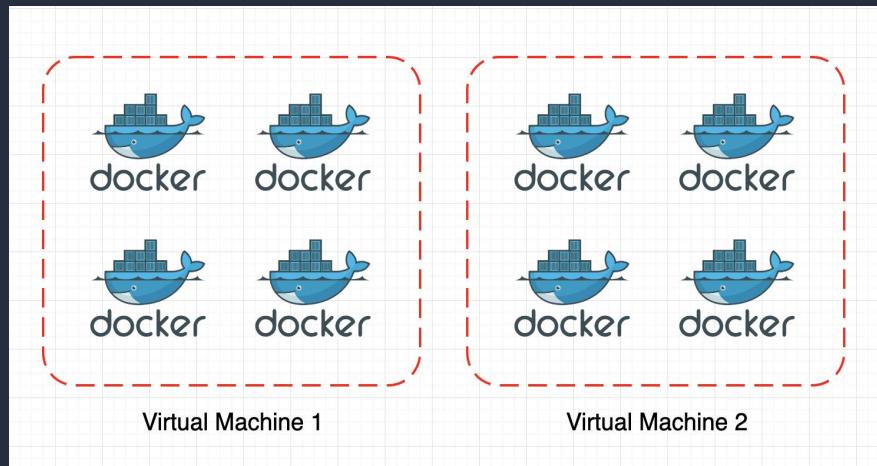
with Real Action Simulator

[Click Here](#)



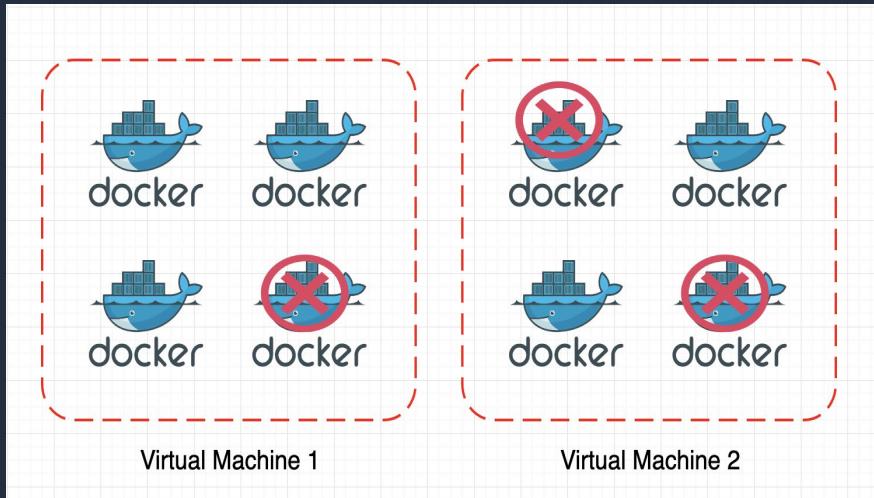
Is it good enough to use only Docker
containers for high scalability design?

Imagine in this situation !



How do you handle scheduling your containers based on suitable cpu and memory requirements across your virtual machines?

Imagine in this situation !



How do you handle your container when it down?

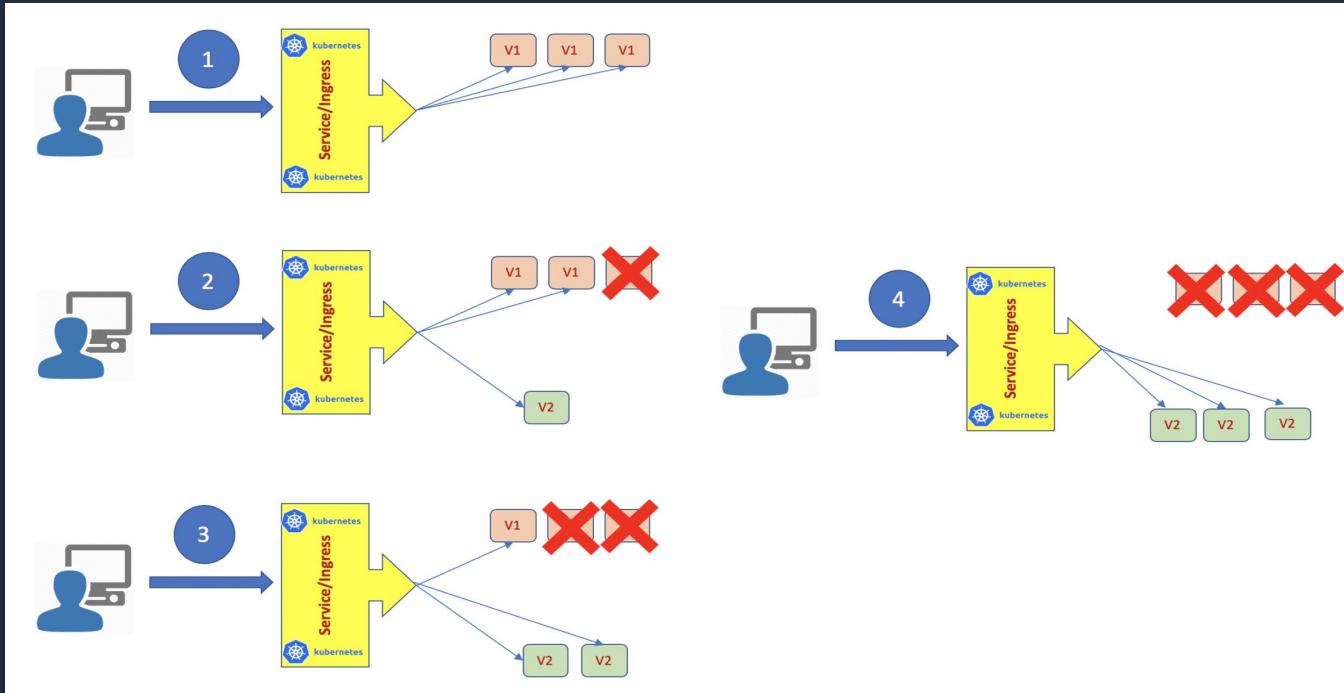
Kubernetes comes to rescue !

- Kubernetes helps you to scale your containers based on cpu and memory requirements across your virtual machines
- Kubernetes supports container self healing with liveness probes
- Kubernetes supports blue green deployment as default container deployment strategy

Kubernetes comes to rescue !

- Kubernetes provides centralized configuration and secret management
- Kubernetes easily to integrate Cloud services with Cloud providers

Kubernetes Blue / Green Deployment



Rollback Strategy

Rolling undo to previous deployment version

```
$ kubectl rollout history deployment <DEPLOY_NAME>
```

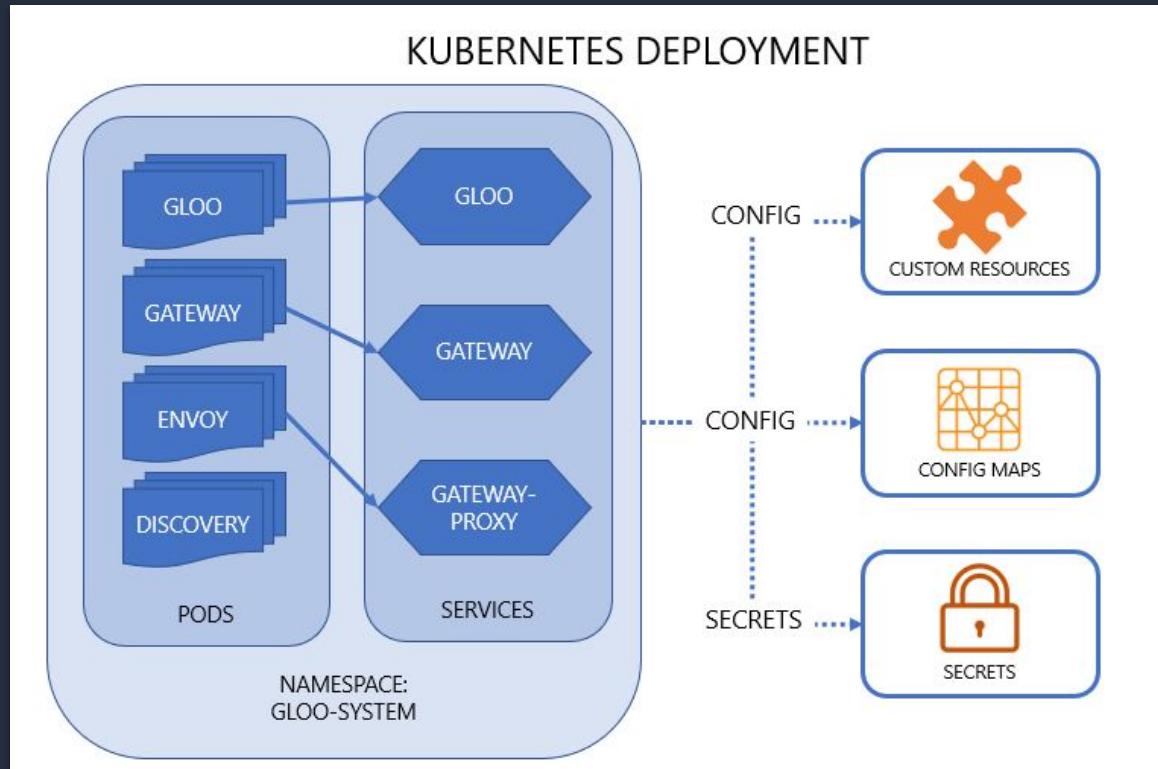
Rolling undo to previous deployment version

```
$ kubectl rollout undo deployment <DEPLOY_NAME>
```

Rolling undo with revision

```
$ kubectl rollout undo deployment <DEPLOY_NAME> --to-revision=<REVISION>
```

Kubernetes Centralized Configuration



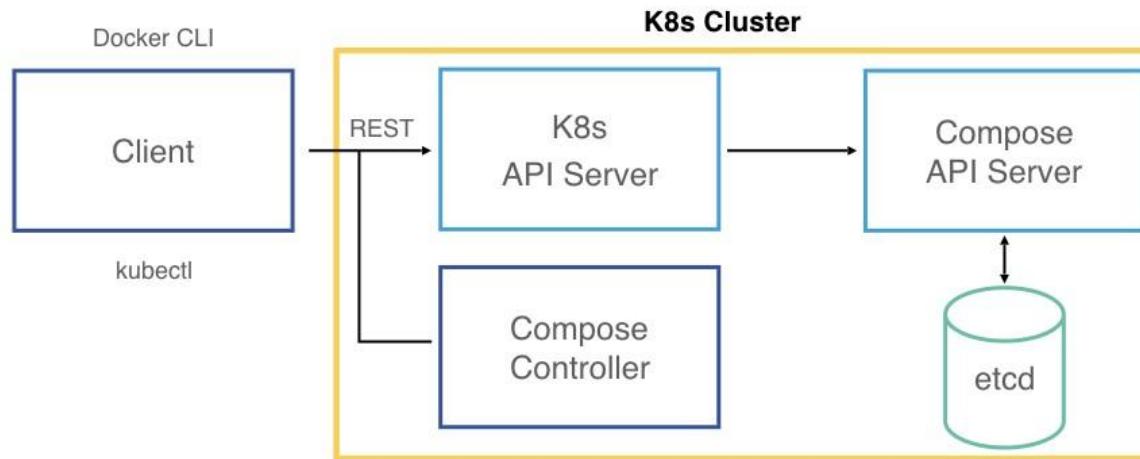
But ... How do you manage / deploy your
containers on Kubernetes with the same
experience you're working with Docker?

Docker Stack comes to rescue !

- Docker Stack **helps you** to manage / deploy your containers on Kubernetes with the same configuration as Docker Compose
- Docker Stack **reduces** Kubernetes learning curve / complexity by leveraging the ease of Docker Compose
- Docker Stack **supports** immutable infrastructure, you can use only single Docker Compose file for all environments

How Docker Stack Works?

Compose on Kubernetes Architecture



docker
con18
EUROPE

Docker Stack: Kubernetes

with Real Action Simulator

[Click Here](#)



Best Practices

1 **Prefer minimal base images**

prefer Distroless images

2 **Least privilege user**

create a dedicated user and group on the image, with minimal permissions to run the application

3 **Use Copy instead of ADD**

ADD could result in MITM attacks, sources of malicious data

Best Practices

4 Don't leak sensitive data to Docker Images

Separate sensitive data out of application source code and keep them in Vault, hence, docker build phase won't contains any sensitive data

5 Don't Use the latest tags

Docker image owners should not use the latest image tags, which may result in application breaking changes if the latest image tags would not compatible with the application

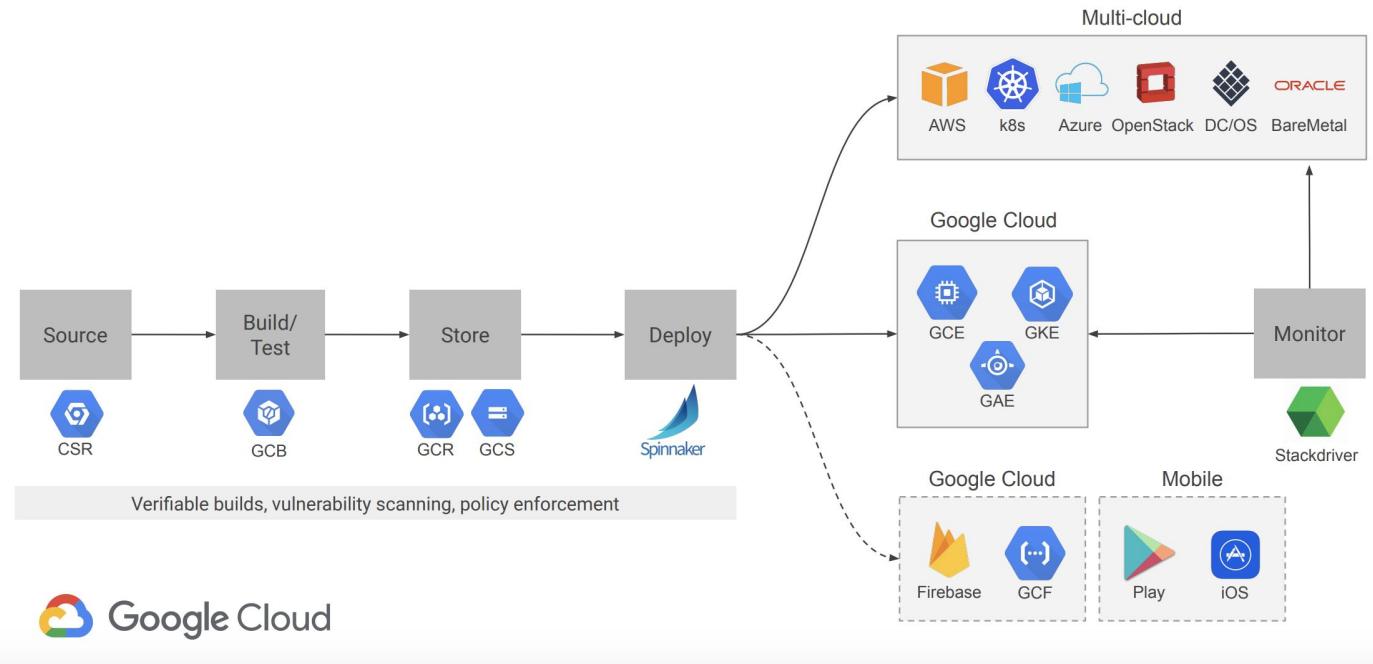
e.g. FROM kubeops:8-alpine → <version>-<operating system>

6 Use labels for metadata

Labels with metadata for images provide useful information for users

DevOps World

CI/CD made easy with Google Cloud



Any Questions ?





KubeOps Skills

KubeOps Skills
@kubeOpsSkills · Consulting Agency

[Send Message](#)

สมาคมโปรแกรมเมอร์ไทย

โครงการสนับสนุนธุรกิจ และ SMEs ทั่วไทย
เพื่อประยุกต์ใช้เทคโนโลยี จำนวนเงิน **10,000 บาท**

สมัครสมาชิก อบรมเชิงปฏิบัติการ เจรจาต่อรองทางดิจิทัล ขอรับทุนบริการ ตู้อัตโนมัติ 10,000 บาท

สอบถามเพิ่มเติม ทางเพจ Facebook ของ @ThaiProgrammerSociety

<https://voucher.thaiprogrammer.org> | 085 878 3637 | depa | THAI PROGRAMMER

สมาคมโปรแกรมเมอร์ไทย
@ThaiProgrammerSociety - Nonprofit Organization

[Contact Us](#)

thaiprogrammer.org