สยามชำนาญกิจ
SIAM CHAMNAN KIT

Deploy microservices

Somkiat Puisungnoen

Somkiat | Home

Somkiat Puisungnoen

Update Info 1 | View Activity Log 10+ | ...

Timeline | About | Friends 3,138 | Photos | More

When did you work at Opendream?

22 Pending Items

Intro

Software Craftsmanship

Software Practitioner at สยามชำนาญกิจ พ.ศ. 2556

Agile Practitioner and Technical at SPRINT3r

Post | Photo/Video | Live Video | Life Event

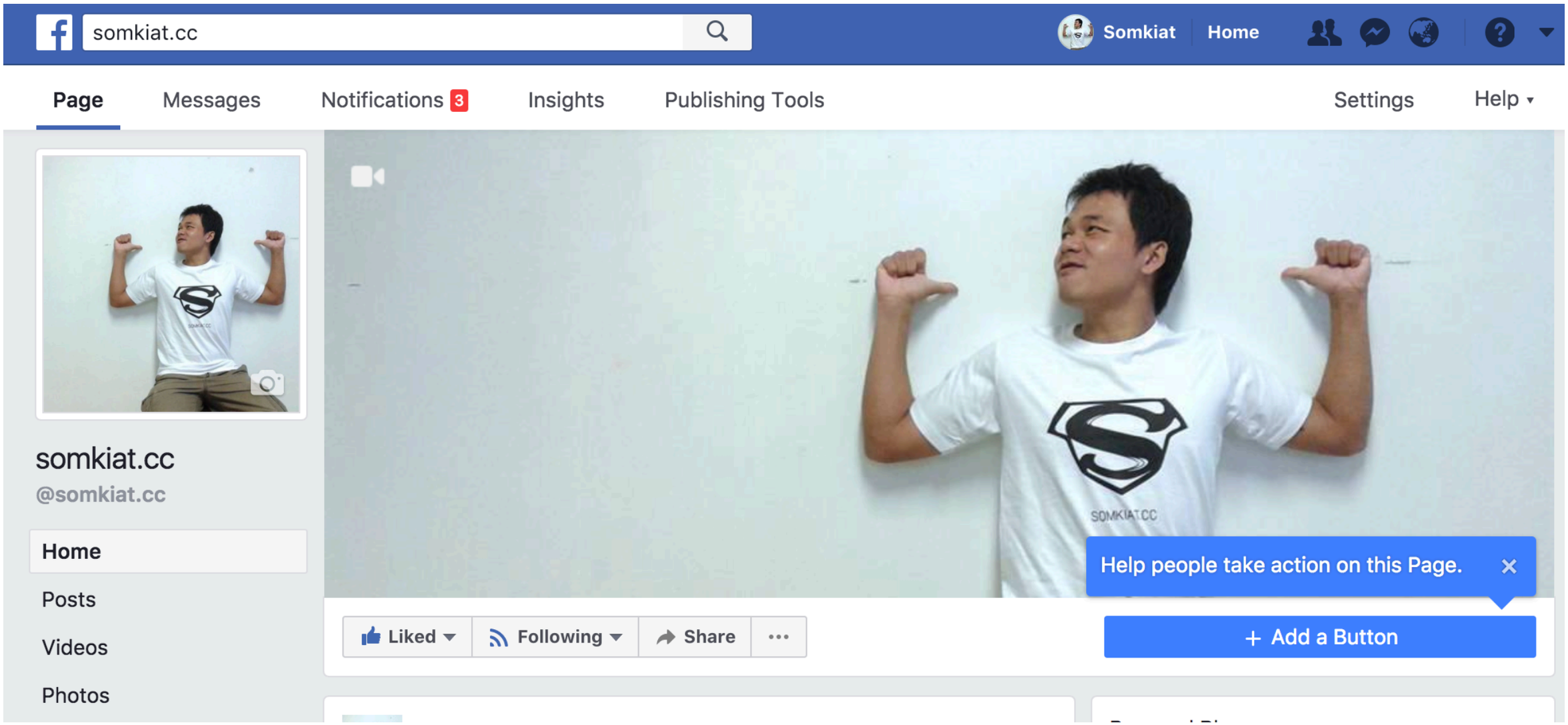
What's on your mind?

Public | Post

Somkiat Puisungnoen

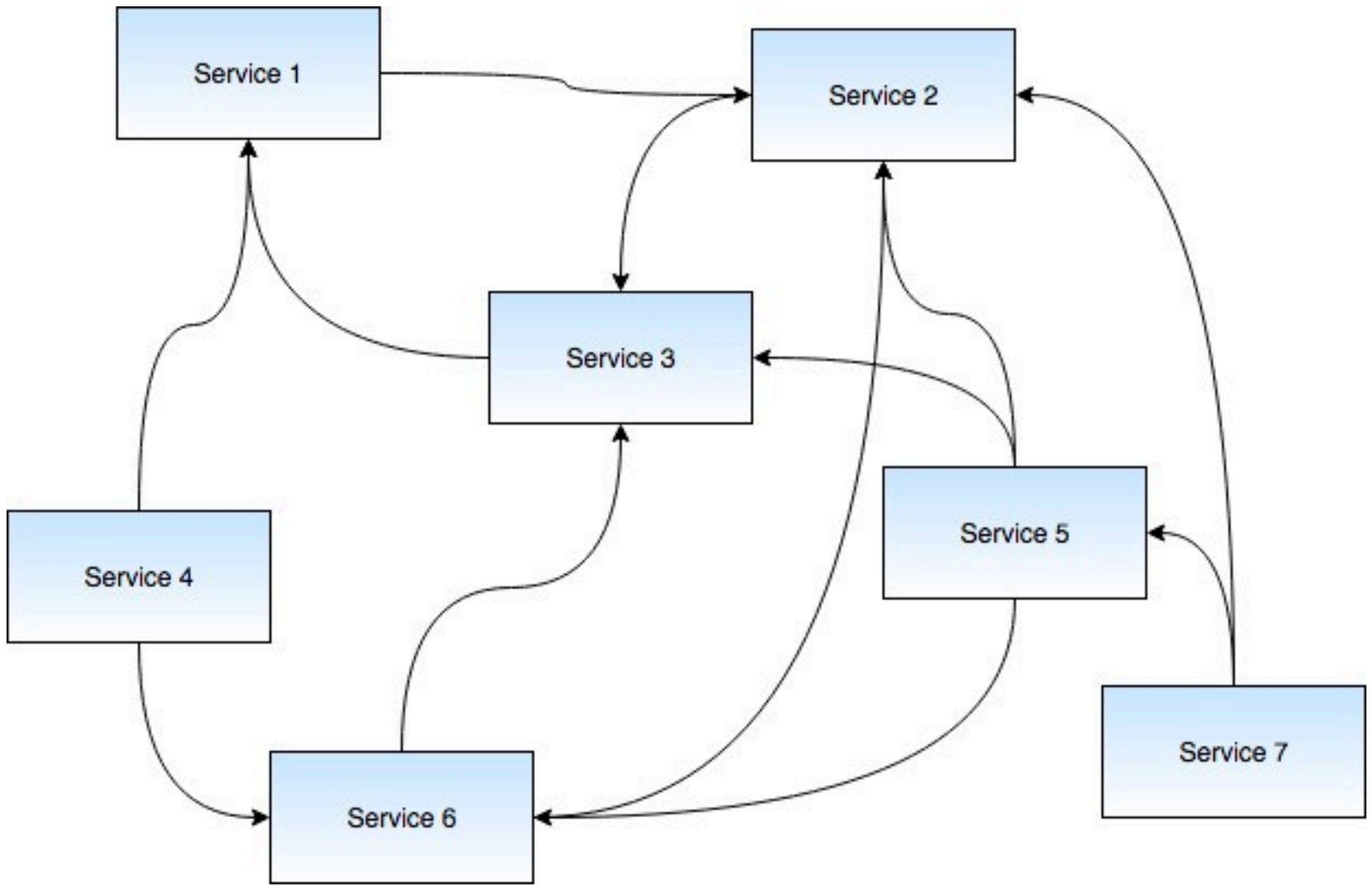
15 mins · Bangkok ·

Java and Bigdata

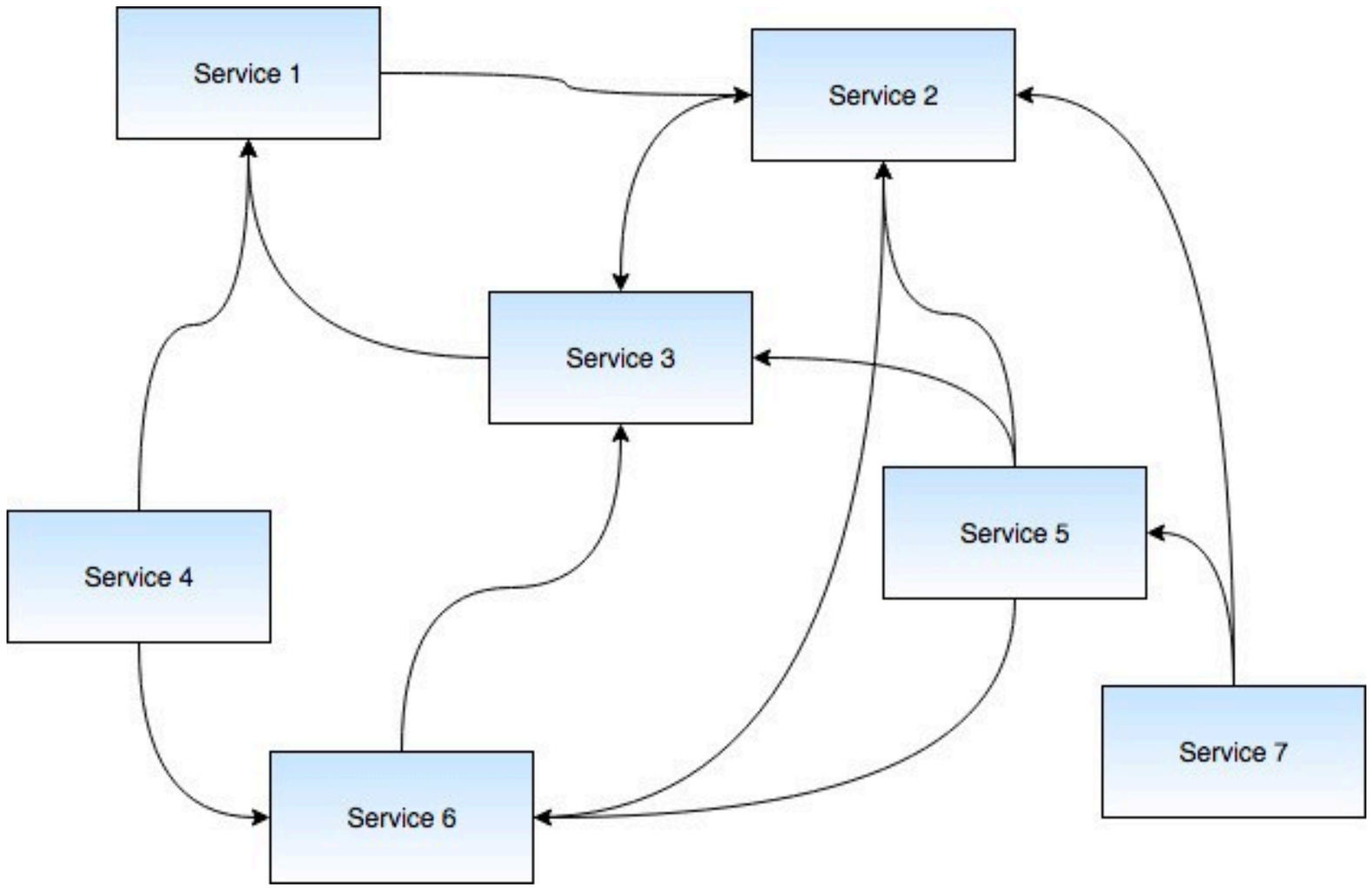


Deploy microservices

Many services !!



How to deploy ?



How to deploy ?

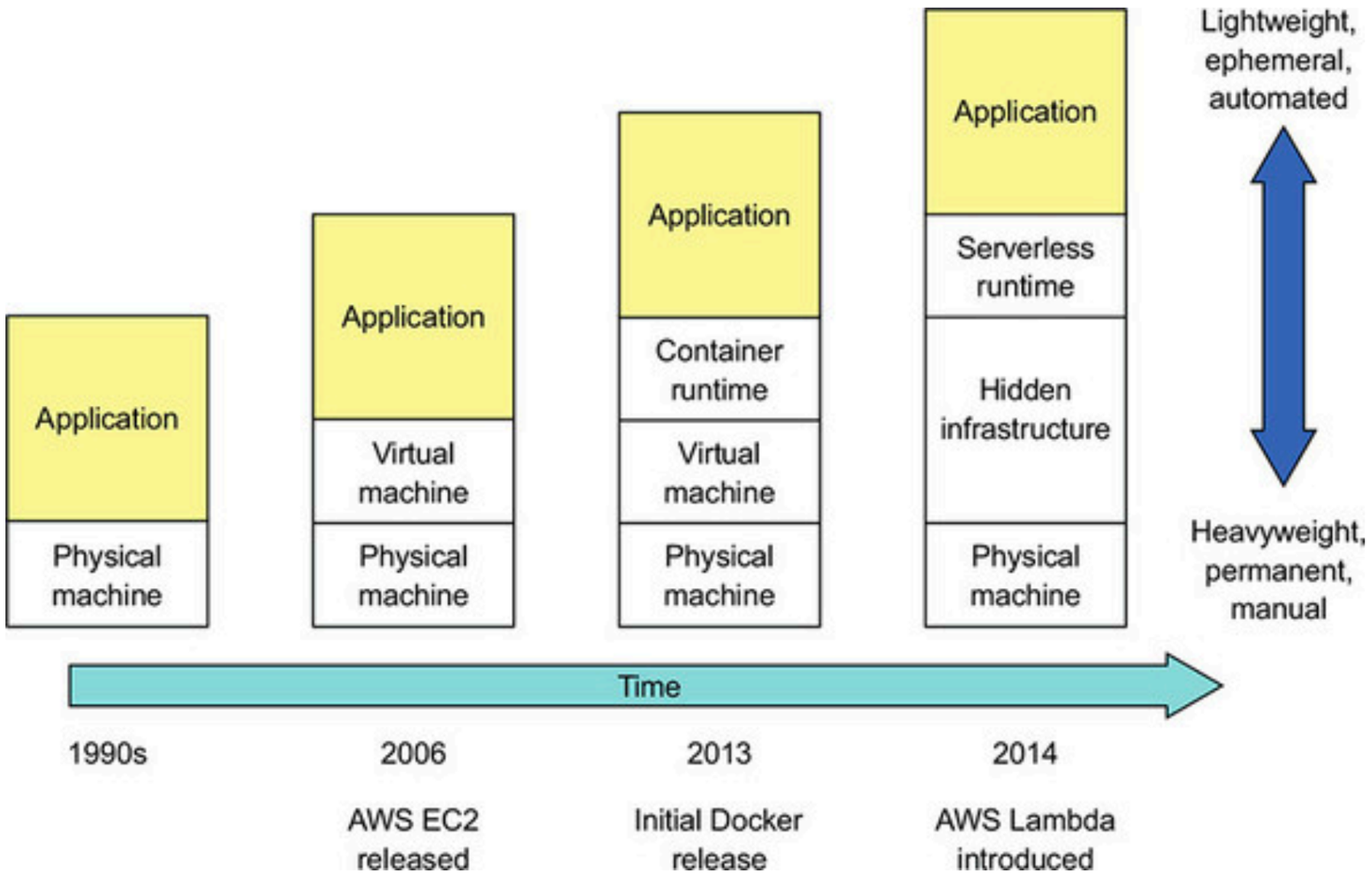
Language-specific packaging format
Deploying a service as a VM
Deploying a service as a Container
Serverless deployment

Deployment strategies

Service per instance
Multiple services per instance
Multiple services in multiple instances

Trend about Deployment !!

Heavyweight to Lightweight
Manual to Automated



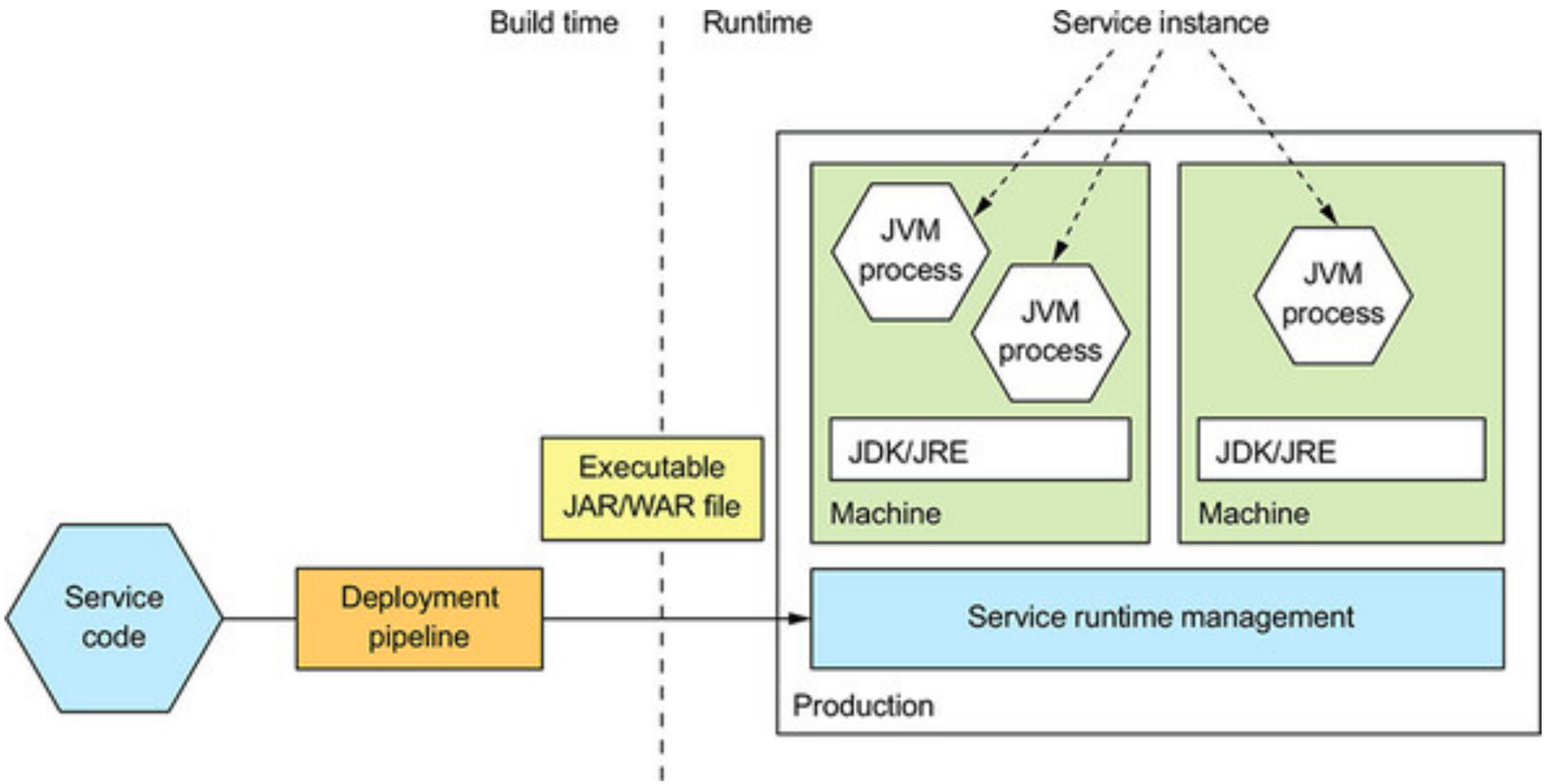
1. Language-specific

Create executable files for deploy
Deploy files to production server
All services are process

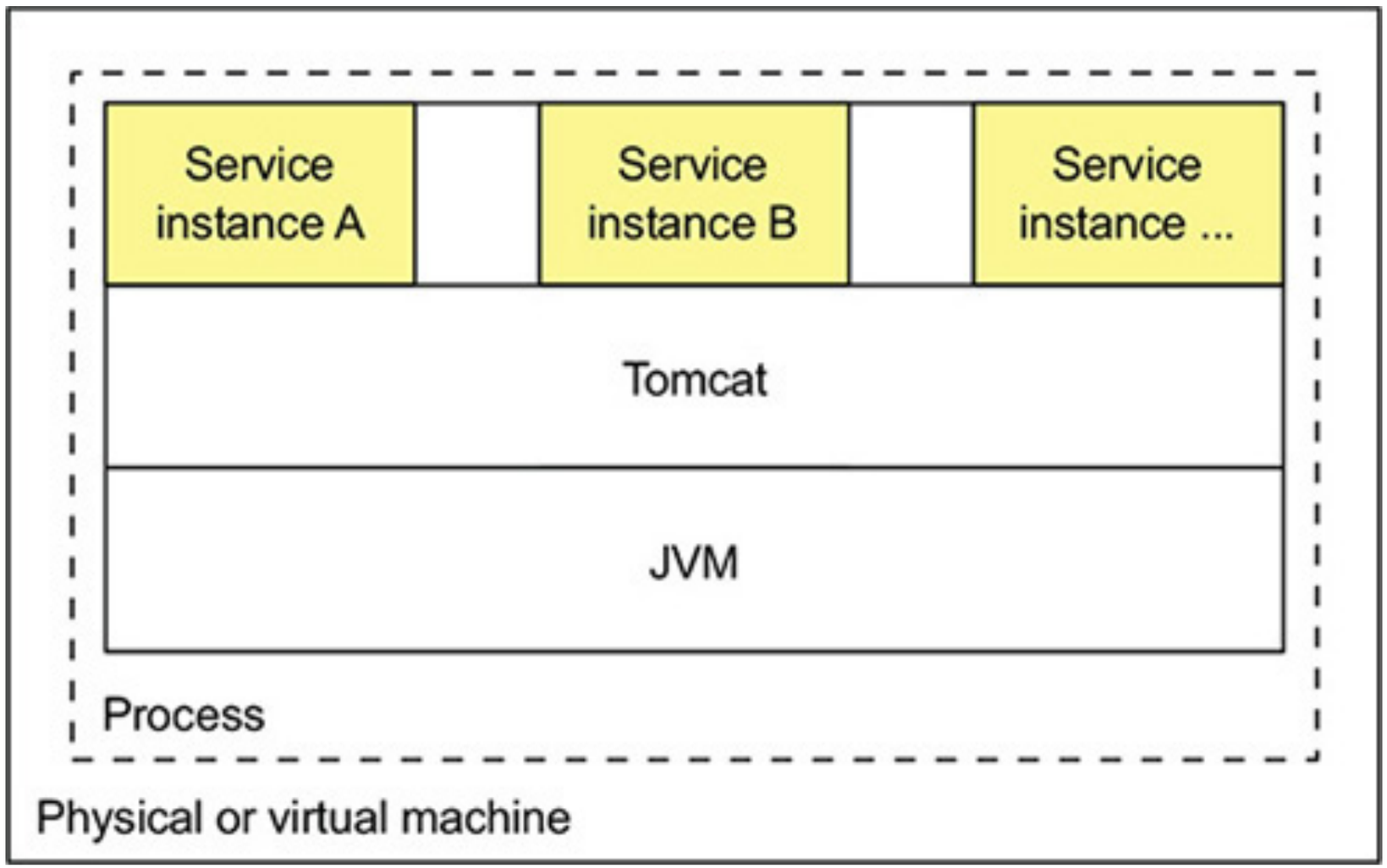
Executable files

Programming Language	Executable file format
Java	JAR/WAR/EAR
Golang	Binary
NodeJS, JavaScript	Directory of source code

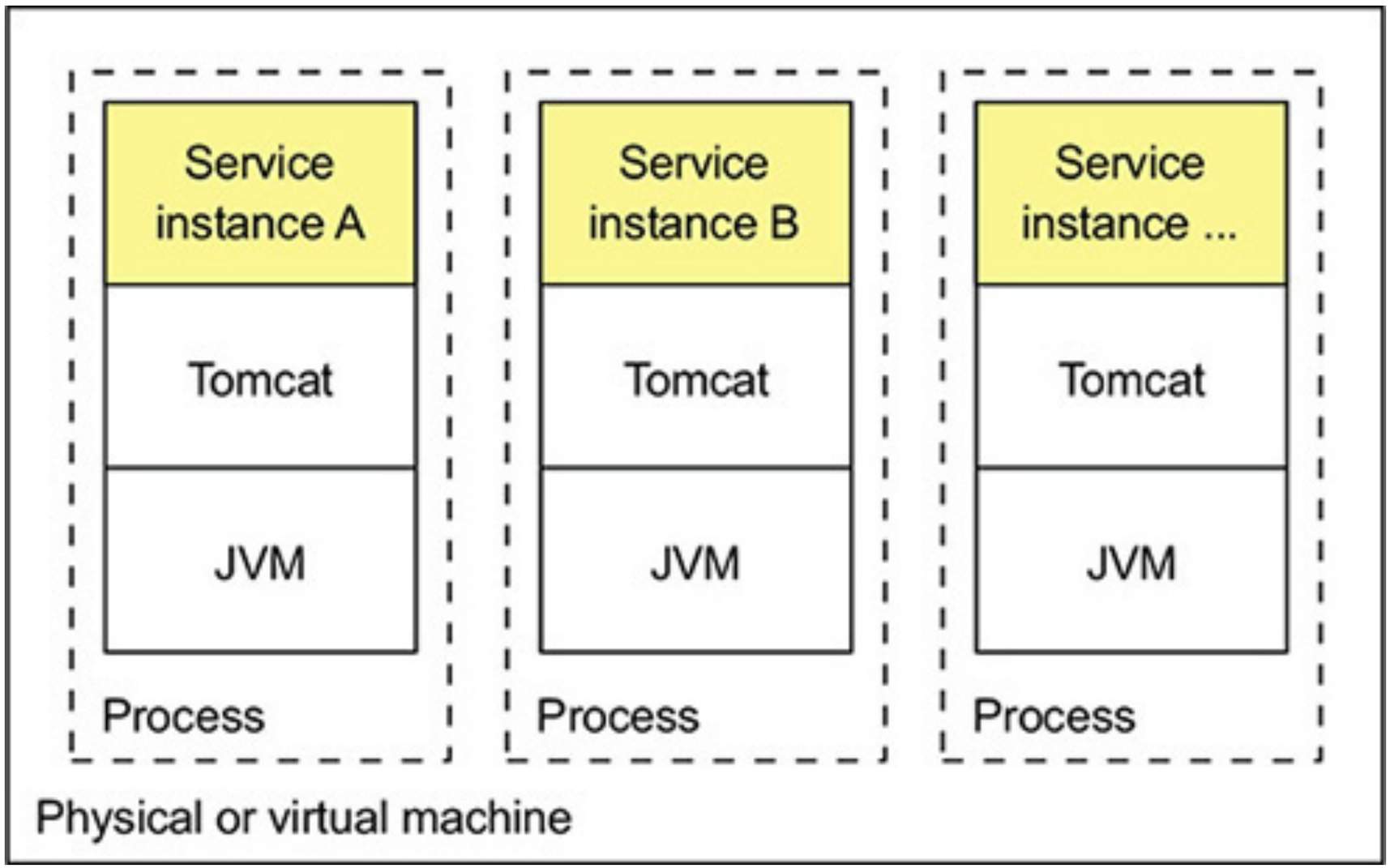
Example :: Deploy Java



Deploy multiple services per web server



Deploy multiple services per machine



Benefits

Fast deployment
Efficient resource utilisation

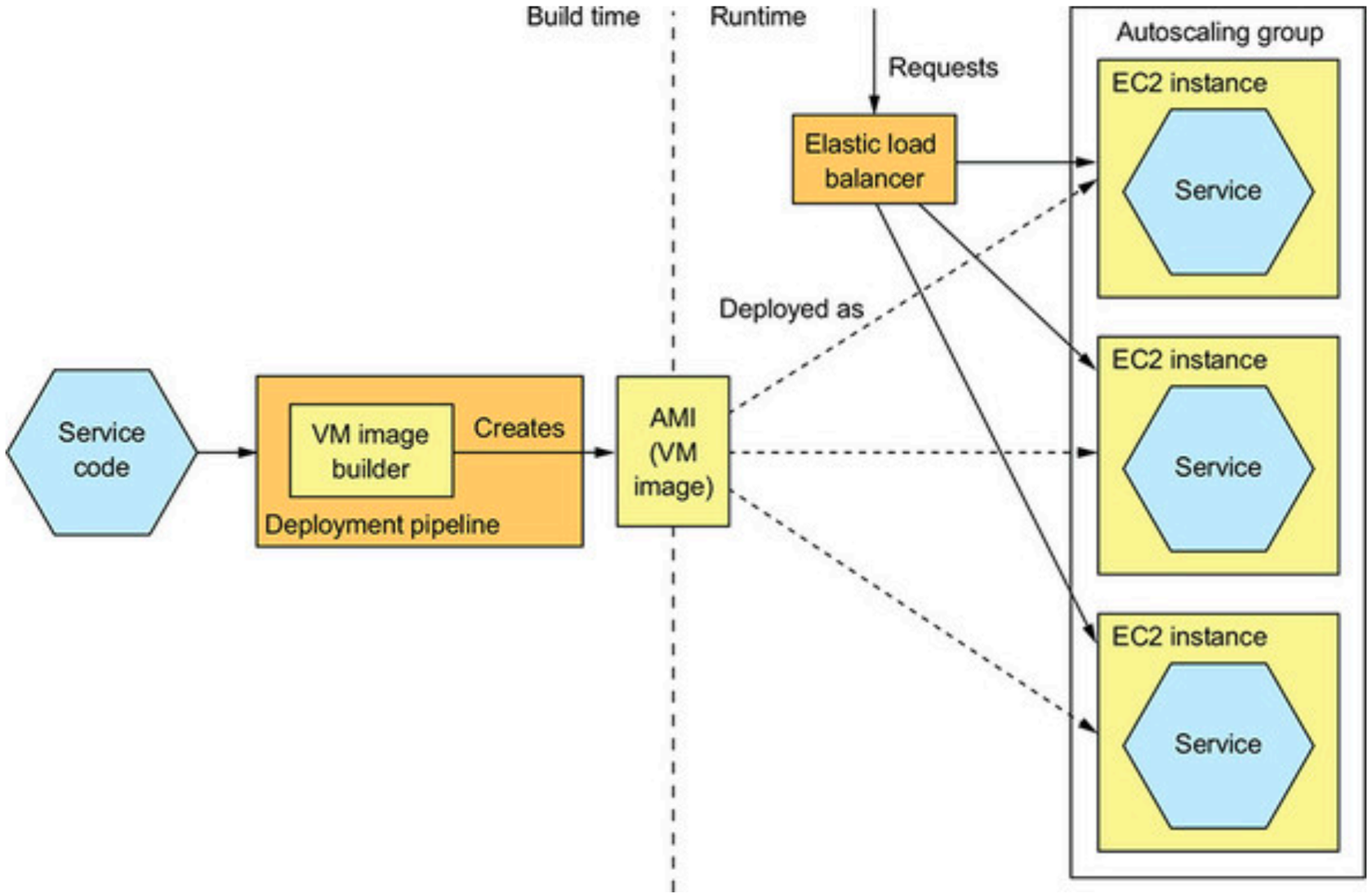
Drawbacks

Lack of encapsulation tech stack
Can’t limit resources per service
Lack of isolation

2. Deploy as a Virtual Machine

Create VM on machine
Deploying a service on VM

Example :: Deploy Java

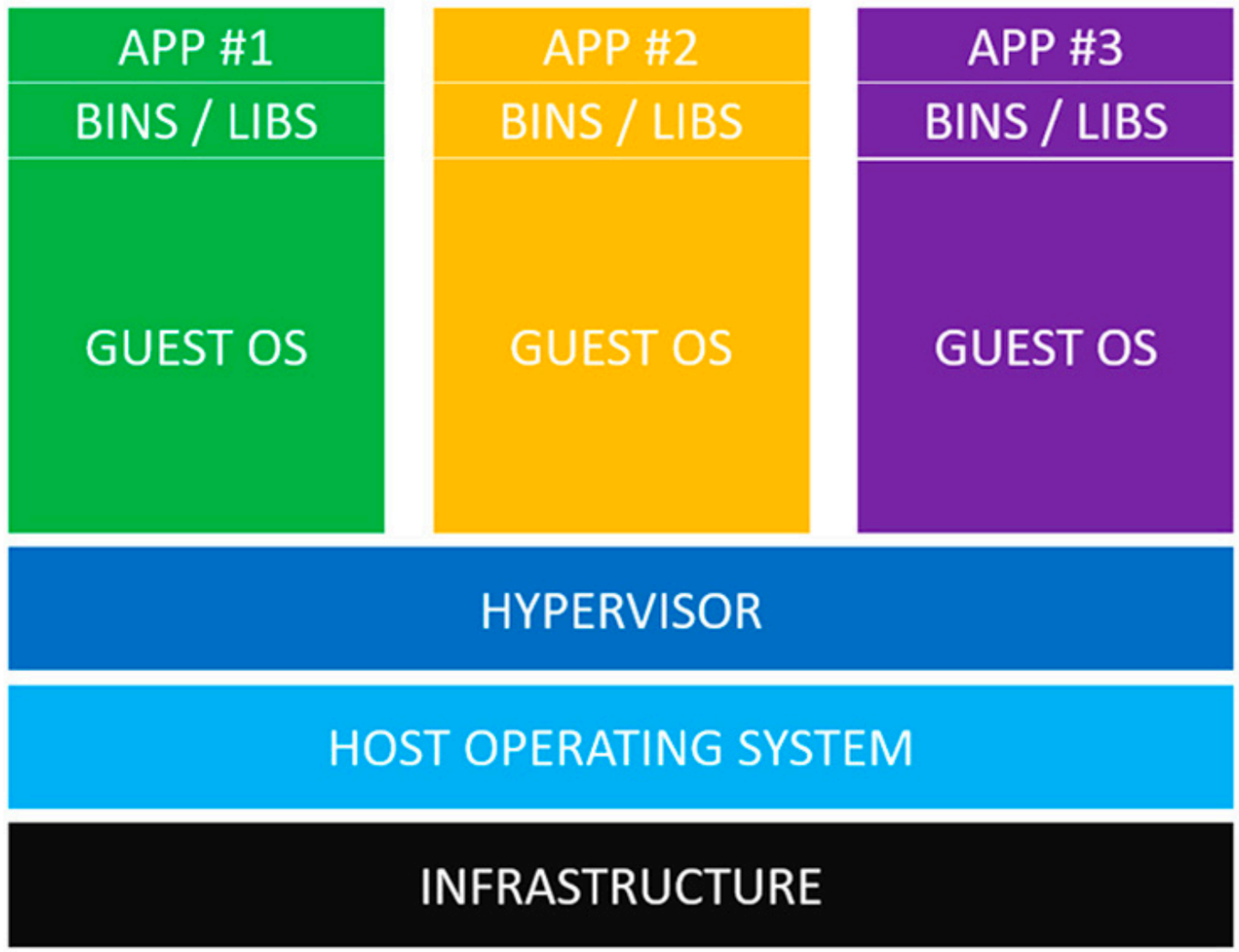


Benefits

Encapsulation tech stack
Isolated service instances
Use mature cloud infrastructure

Drawbacks

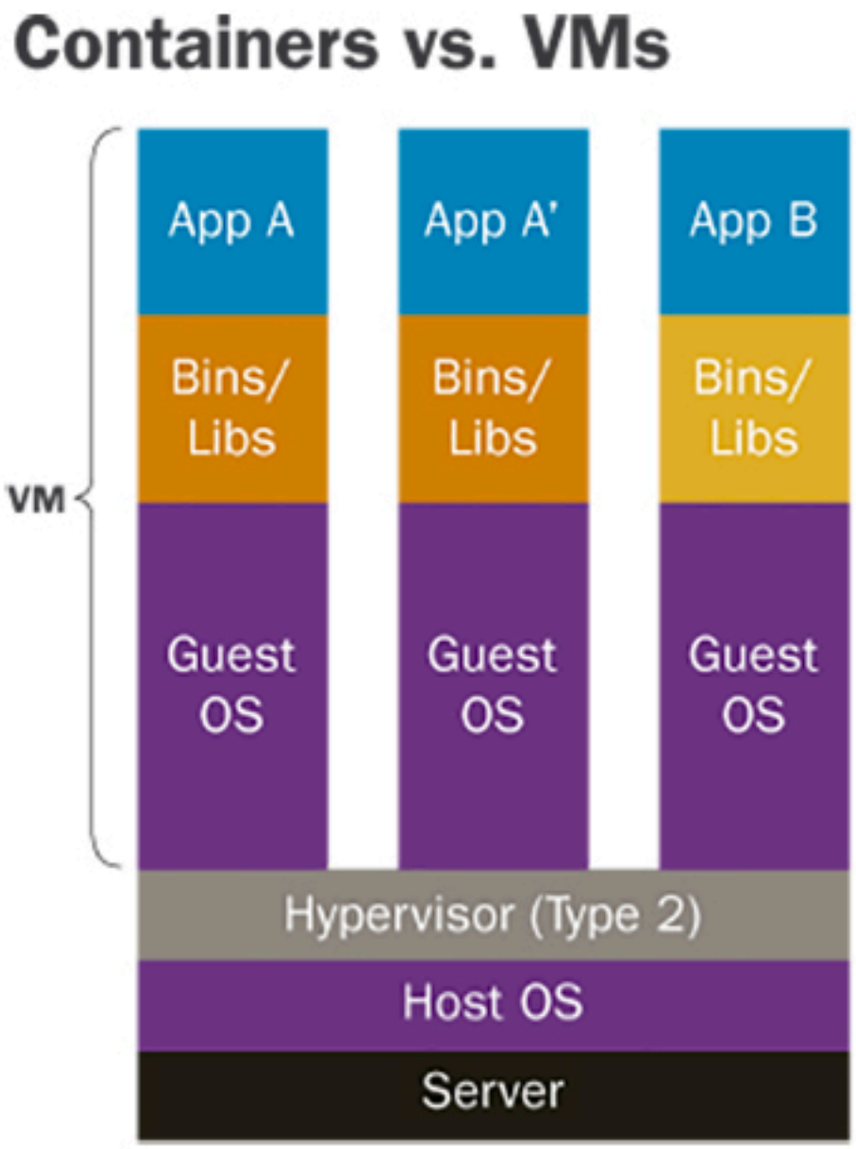
Less efficient resources utilisation



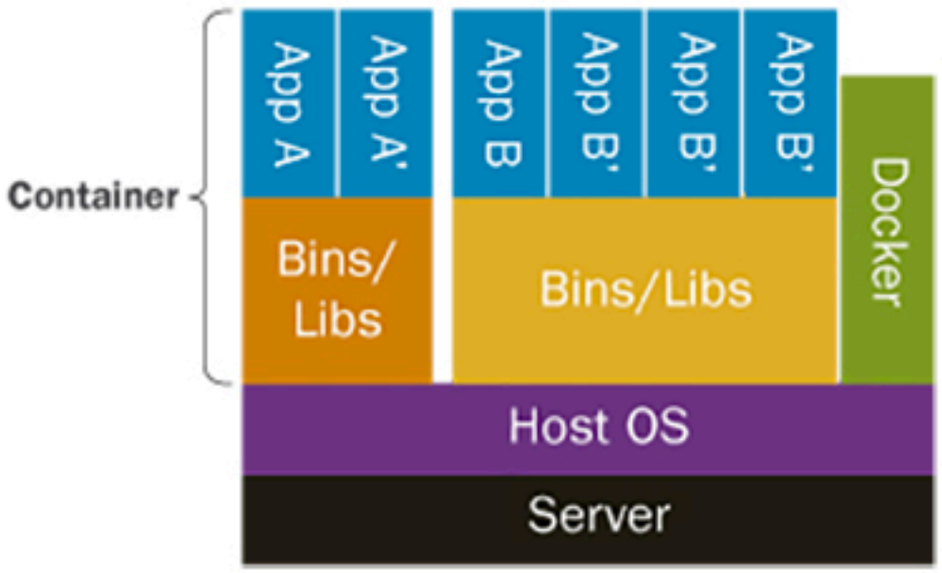
Drawbacks

Less efficient resources utilisation
Slow deployment
System administration overhead

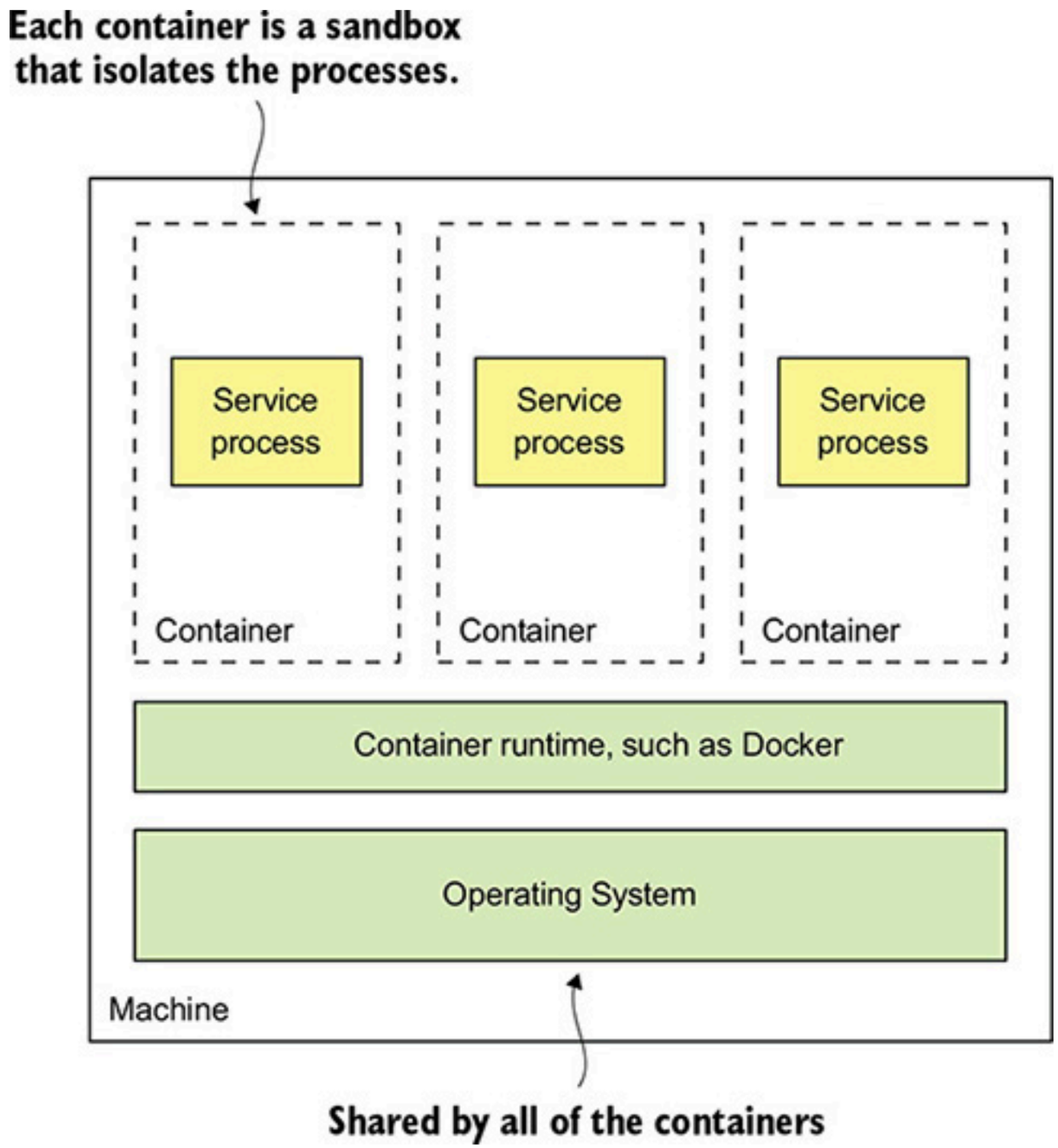
3. Deploy as a Container



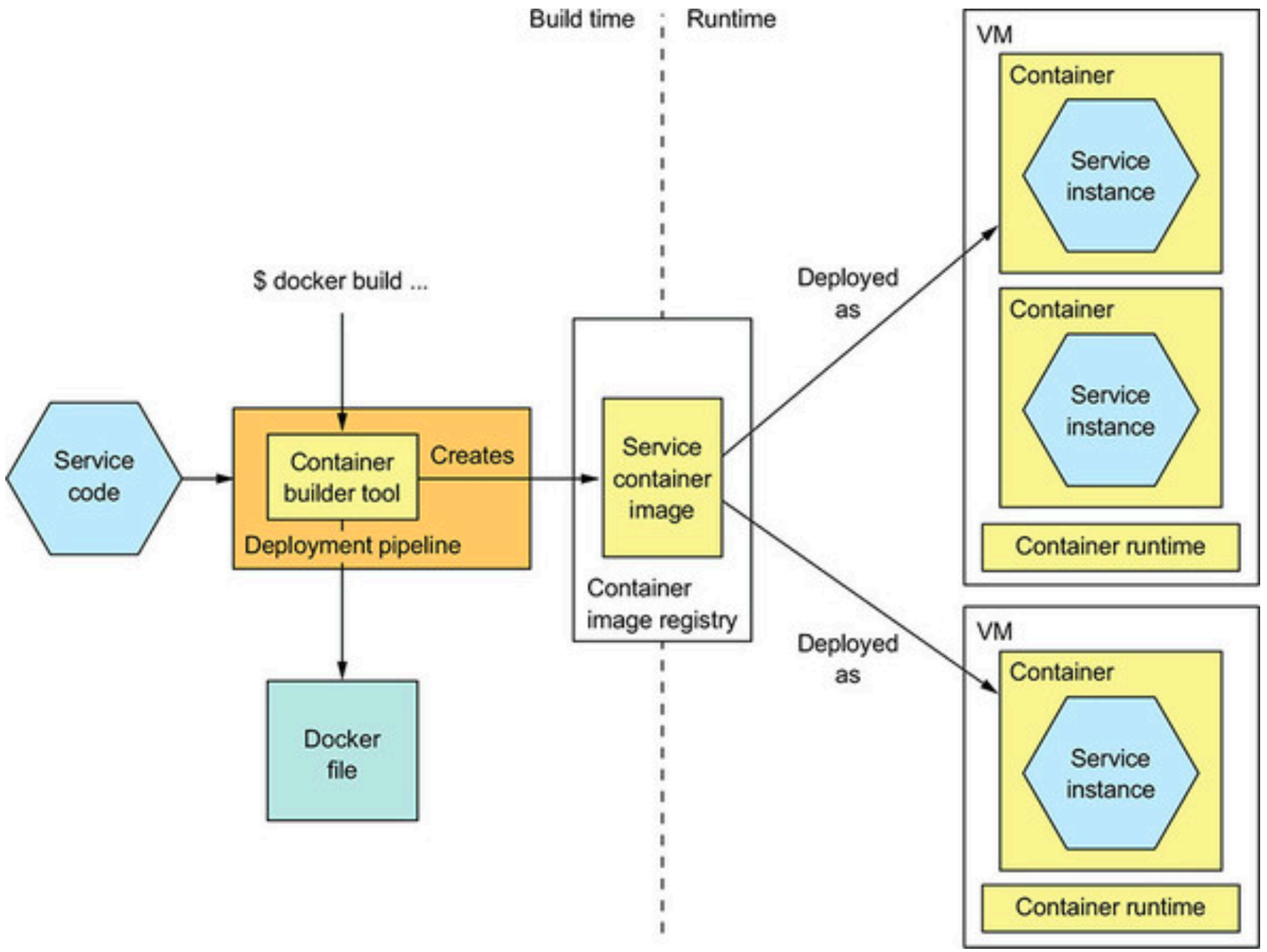
Containers are isolated, but share OS and, where appropriate, bins/libraries



Deploy as a Container



Example :: Deploy Java

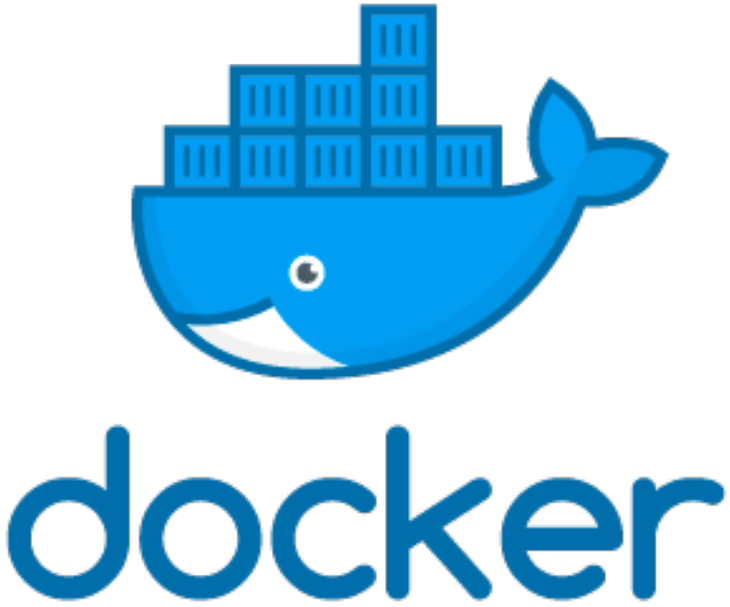


Deploy as a container

Create image
Push image to Container image registry
Deploy service as a container

Deploy as a container

Create image
Push image to Container image registry
Deploy service as a container



Benefits

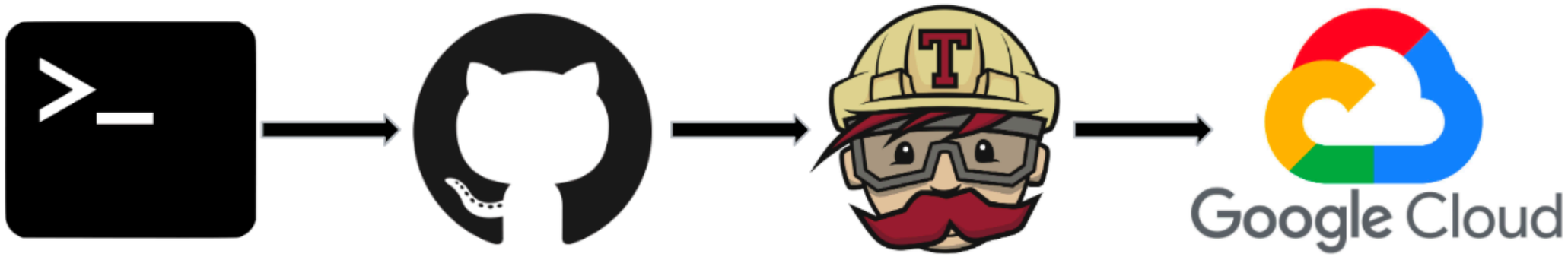
Encapsulation tech stack
Isolated service instances
Service instance’s resources are constrained

Drawbacks

System administration overhead
Infrastructure for containers

Demo
Deploy to Google Cloud

Deploy to Google Cloud



Code

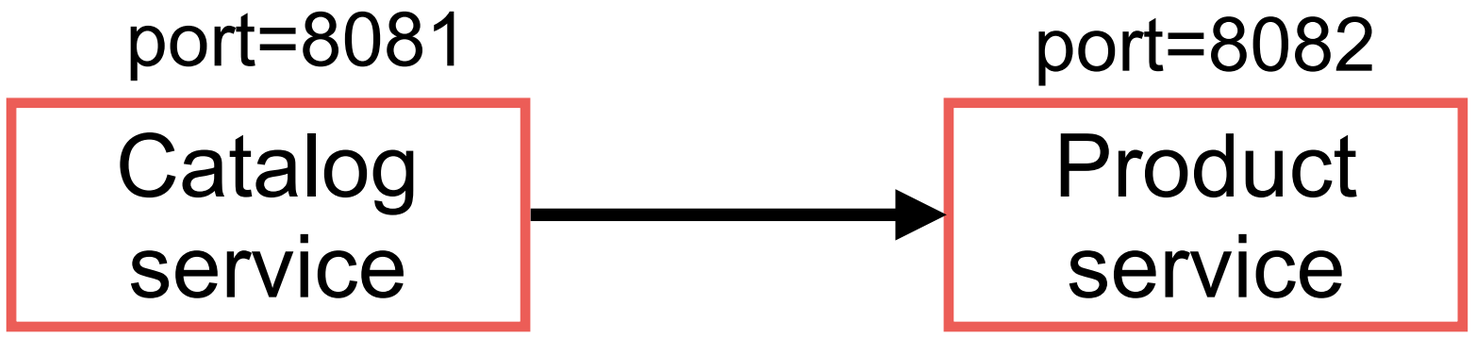
Push to GitHub

Run tests on Travis CI

Deploy to Google Cloud Platform and run end-to-end tests

https://github.com/up1/workshop-depoy-microservice-java

Demo project



Steps

Working with Docker
Auto-build with Travis CI
Deploy to Google Cloud

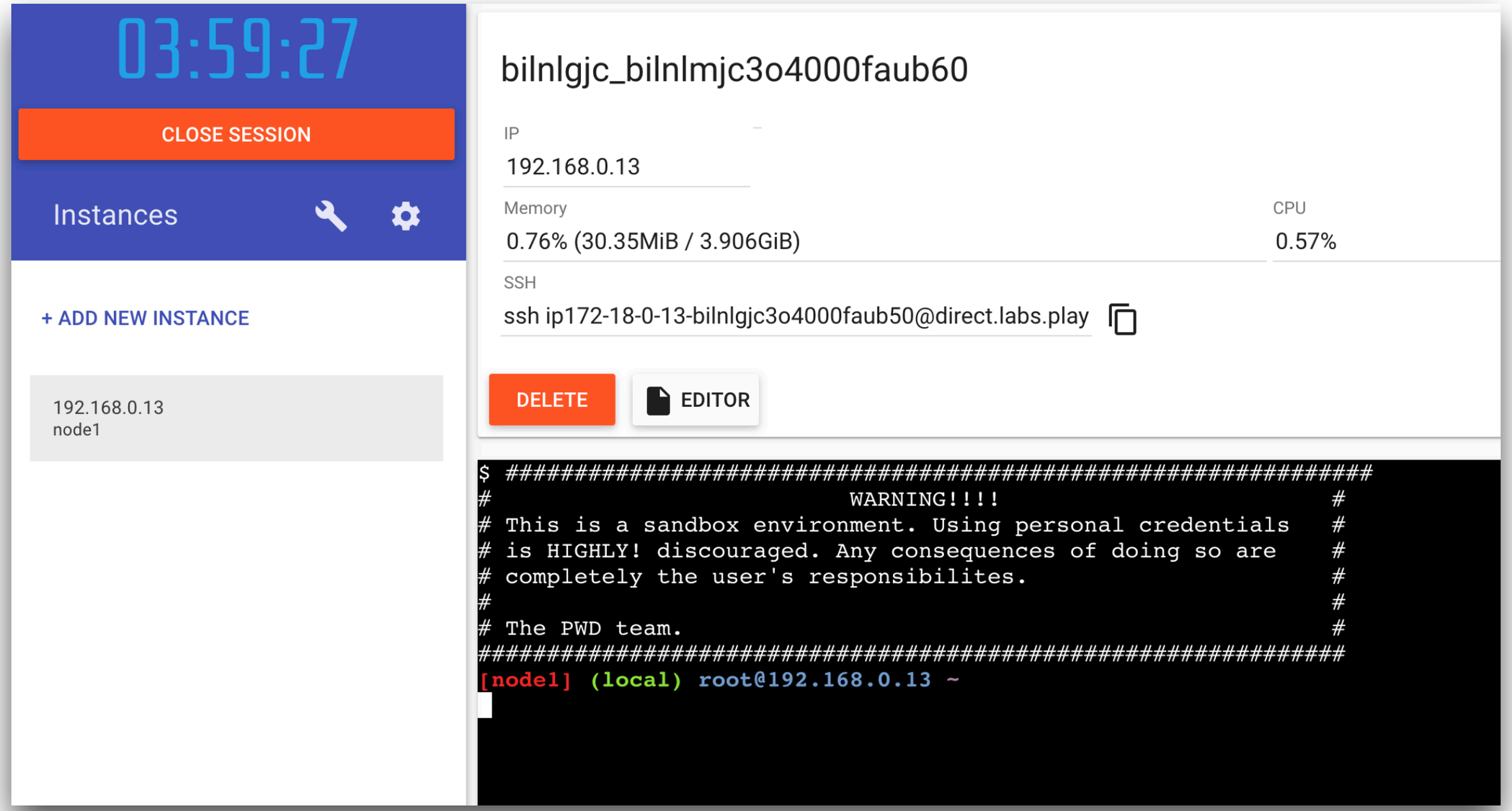
Step 1 :: Working with Docker

Using Play with Docker

https://labs.play-with-docker.com/

Using Play with Docker

Add new instance



1. Clone project

$git clone https://github.com/up1/workshop-depoy-microservice-java.git

2. Build docker image

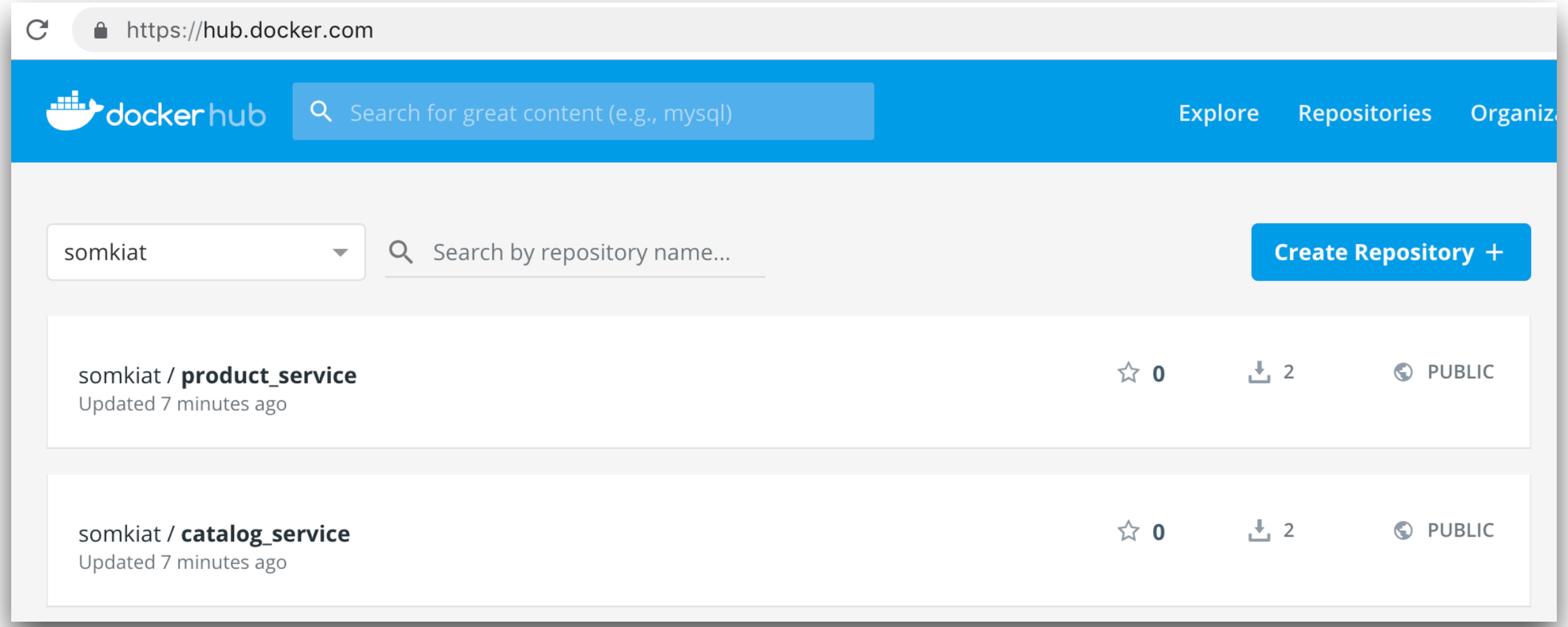
$cd catalog
$docker image build -t catalog_service:1.0 .

$cd product
$docker image build -t product_service:1.0 .

3. Push docker image

$docker login
$docker image push catalog_service:1.0 .
$docker image push product_service:1.0 .

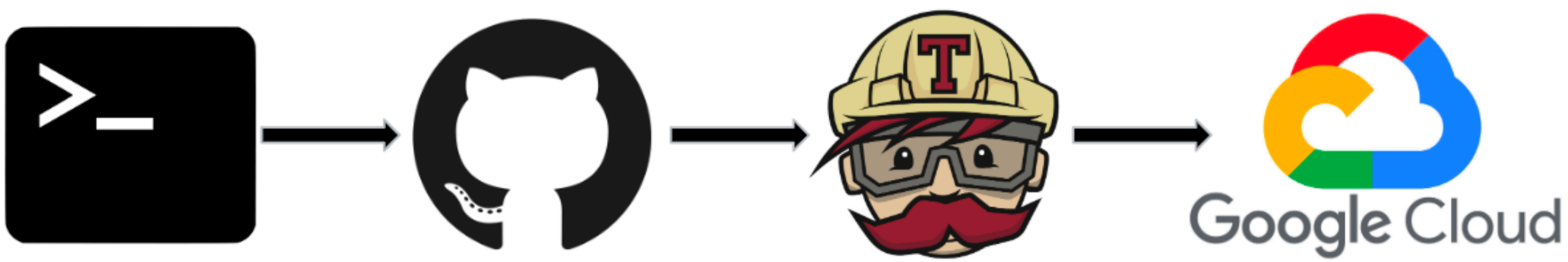
3. Push docker image

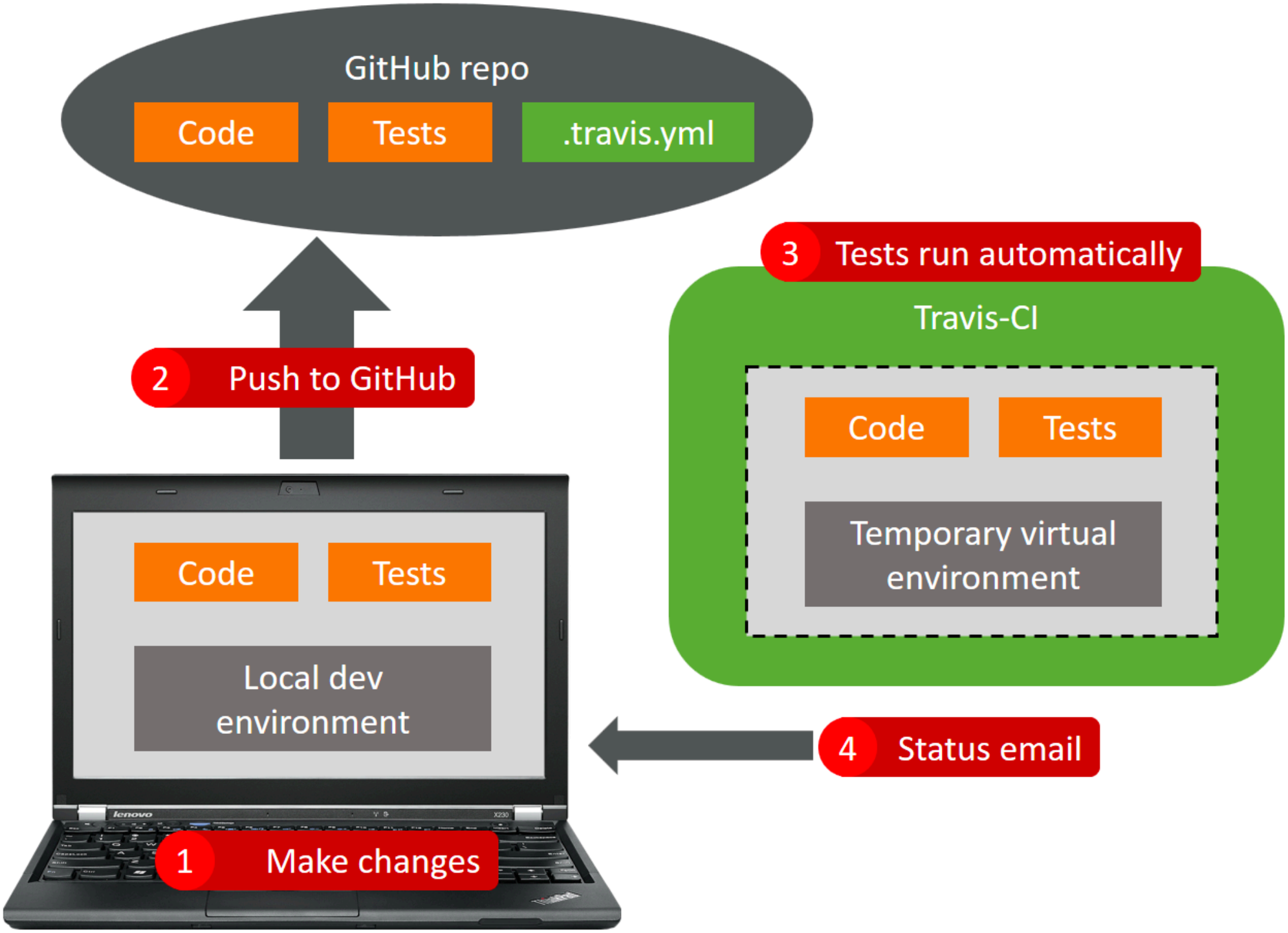


4. Deploy/create container

$docker-compose up -d
$docker-compose ps

Step 2 :: Build with Travis CI



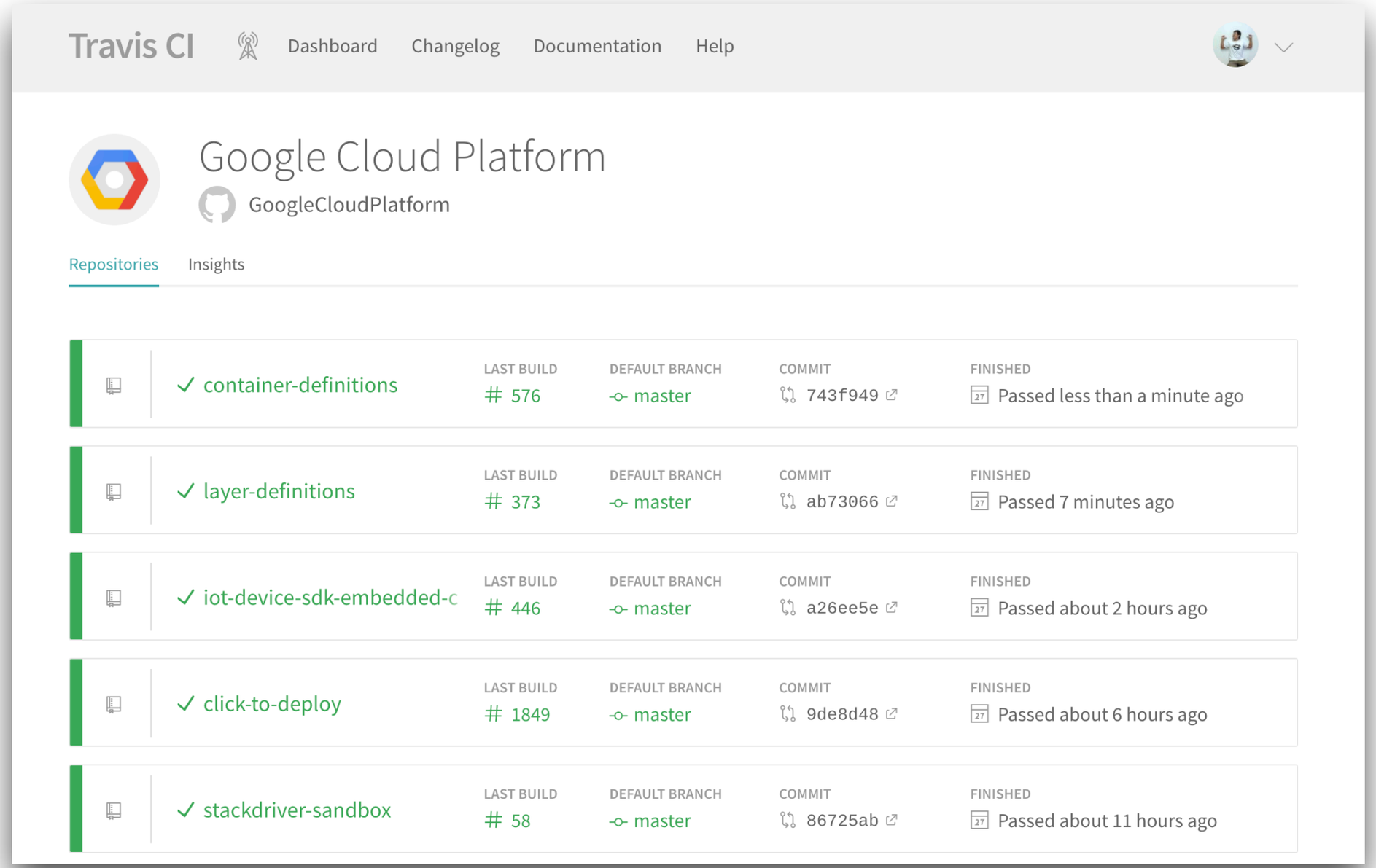


Create file .travis.yml in project

```yaml
language: java
jdk: oraclejdk8
services:
  - docker
```

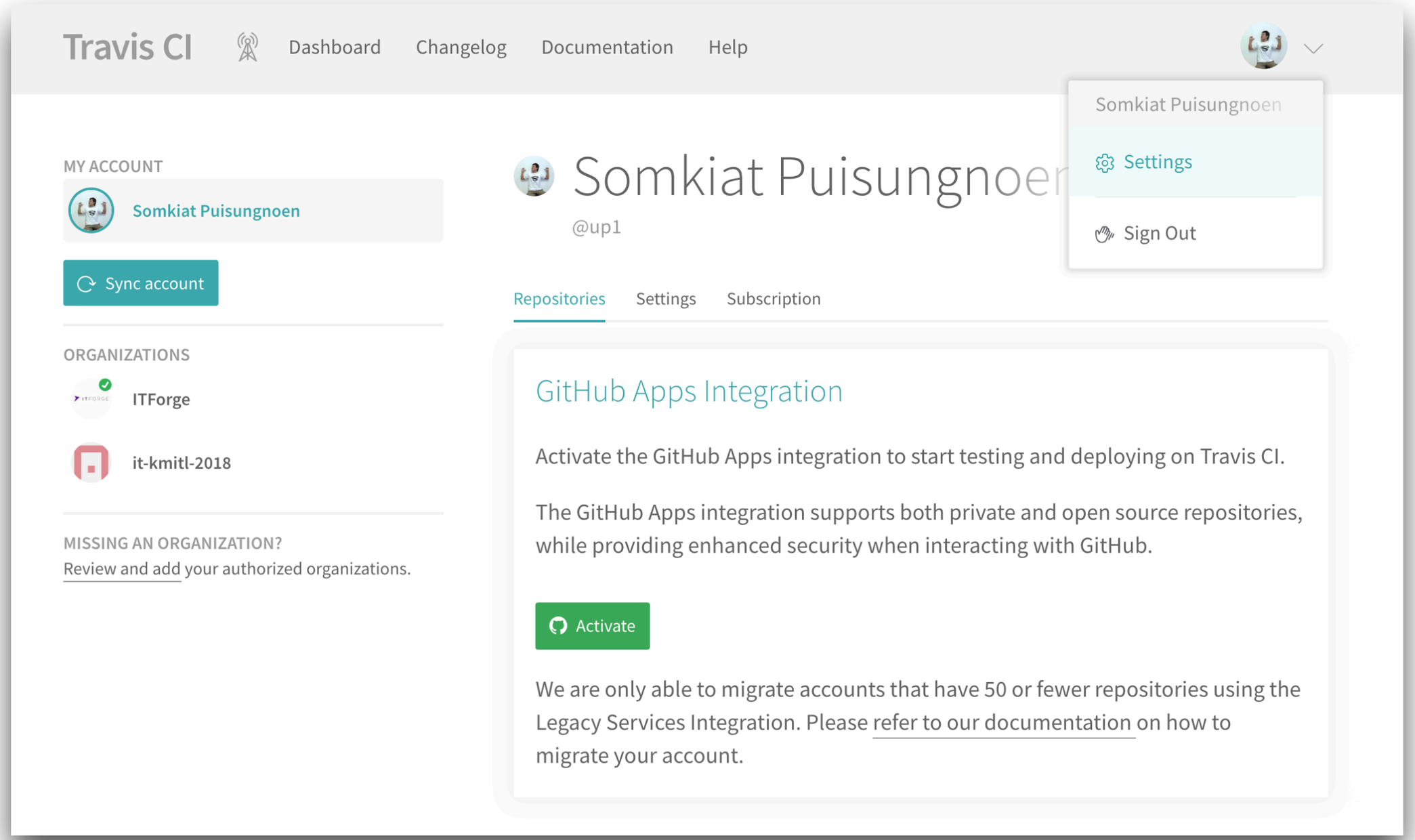
Setting GitHub + travis ci

https://travis-ci.com/GoogleCloudPlatform



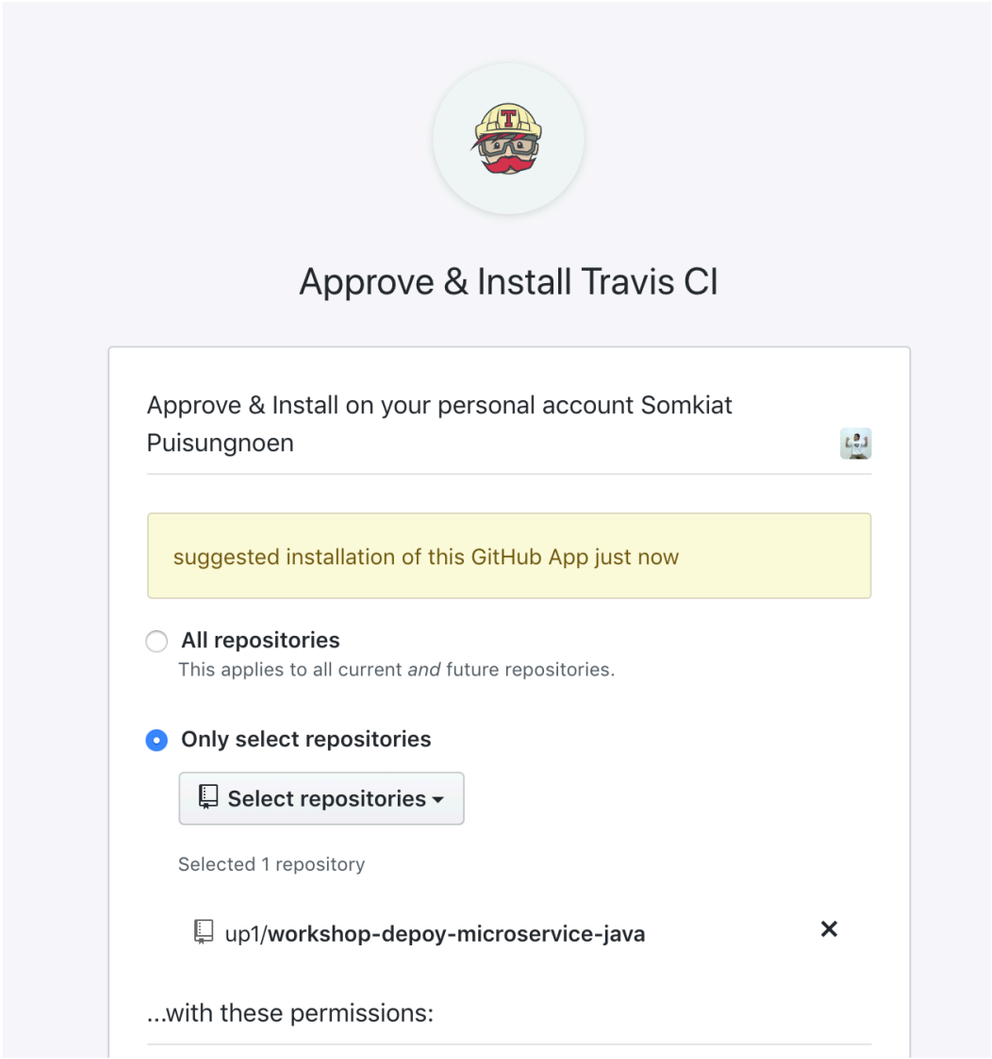
Setting GitHub + travis ci

https://travis-ci.com/GoogleCloudPlatform



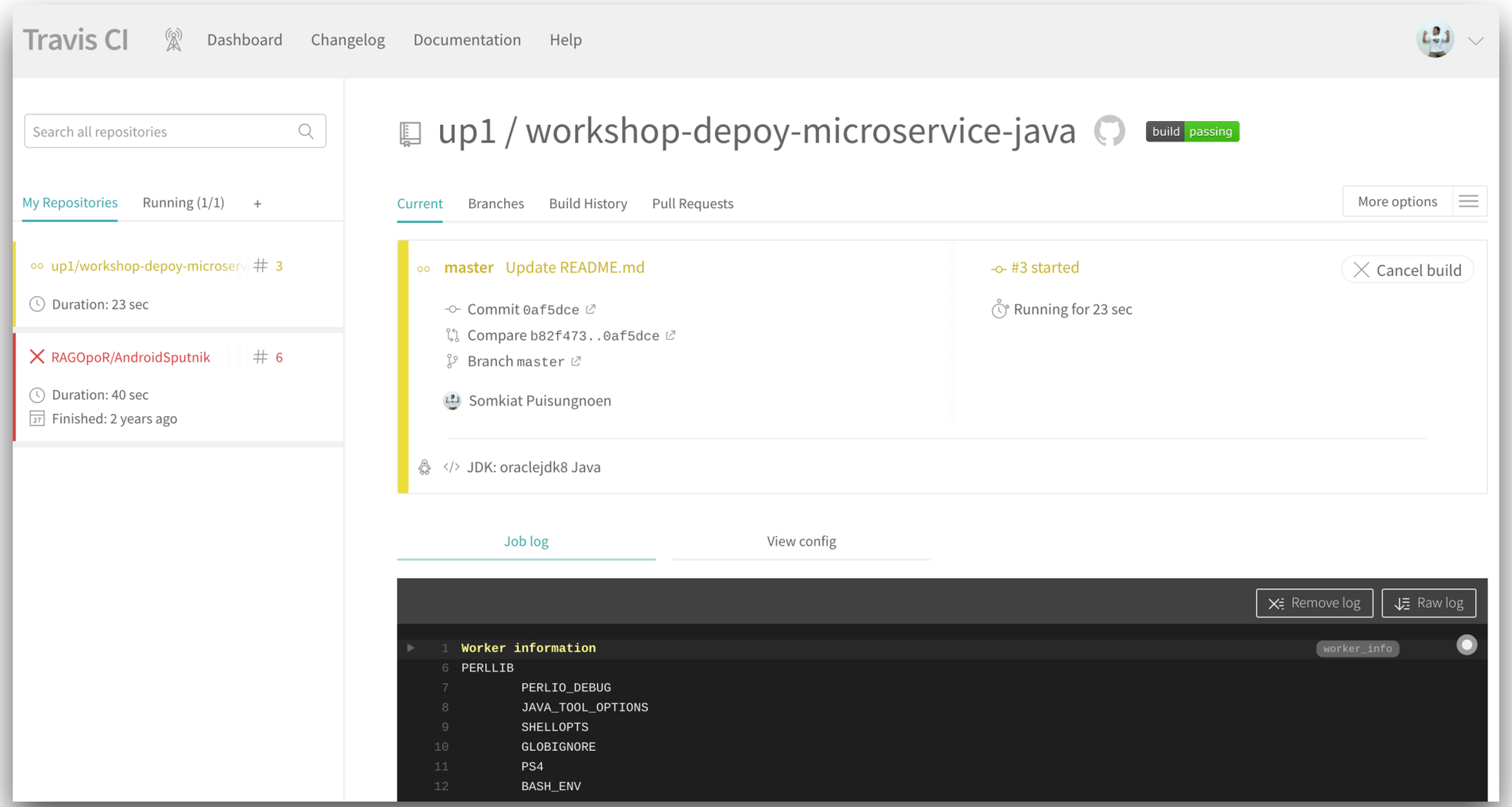
Setting GitHub + travis ci

https://travis-ci.com/GoogleCloudPlatform

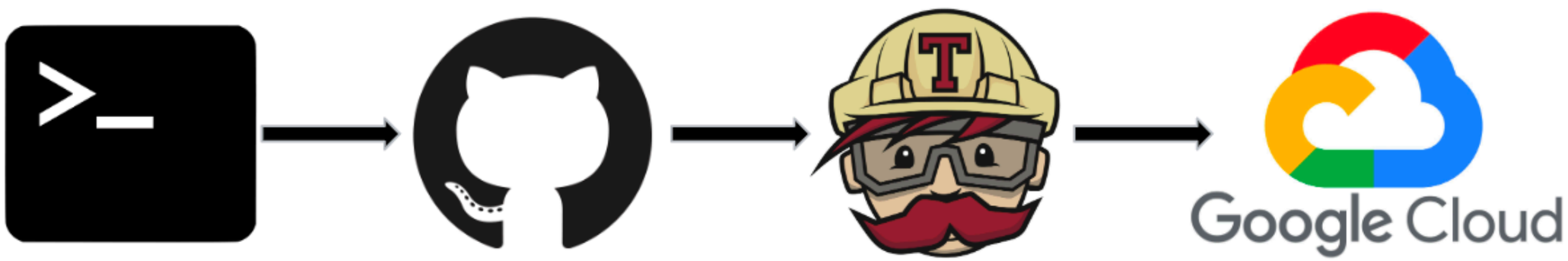


Result

https://travis-ci.com/up1/workshop-depoy-microservice-java



Step 3 :: Deploy to Google Cloud Engine



Code

Push to GitHub

Run tests on Travis CI

Deploy to Google Cloud Platform and run end-to-end tests

Homework

Run test
Run test/code coverage
Push docker image to Container image registry
Deploy to Google Cloud

Travis CI