



Microservices workshop














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 ·

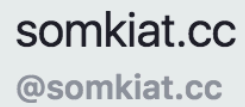

Java and Bigdata



© 2017 - 2018 Siam Chamnankit Company Limited. All rights reserved.

Microservices

3



Photos



+ Add a Button

Agenda

Deploy services with NodeJS
Building images with Docker
Manage containers with docker-compose
Develop deployment pipeline of service



Agenda

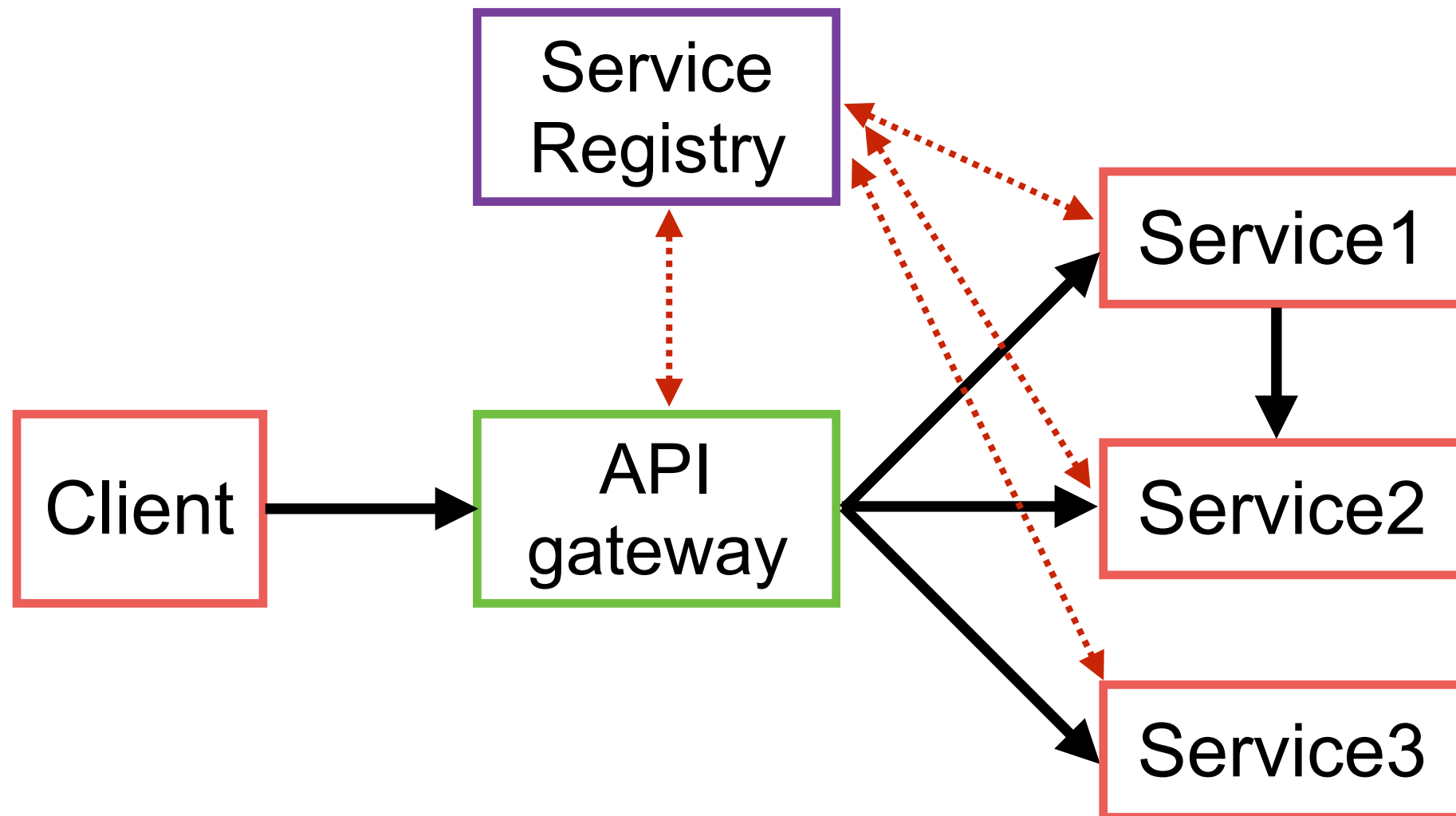
Orchestrate containers with Kubernetes
Rolling update with Kubernetes
Auto-scaling with Kubernetes
Demo deployment with Istio



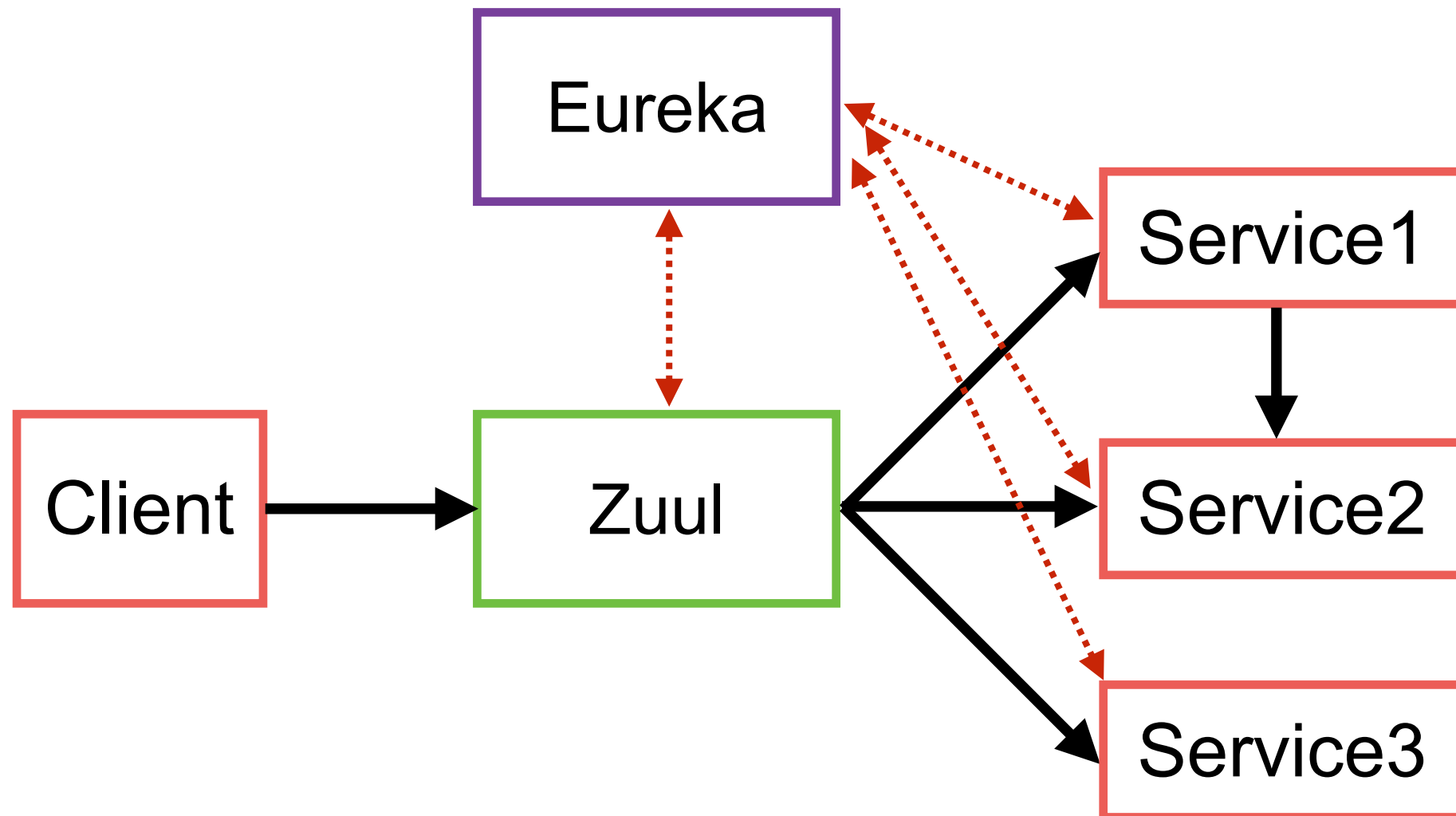
Deploy Microservice



Original Architecture



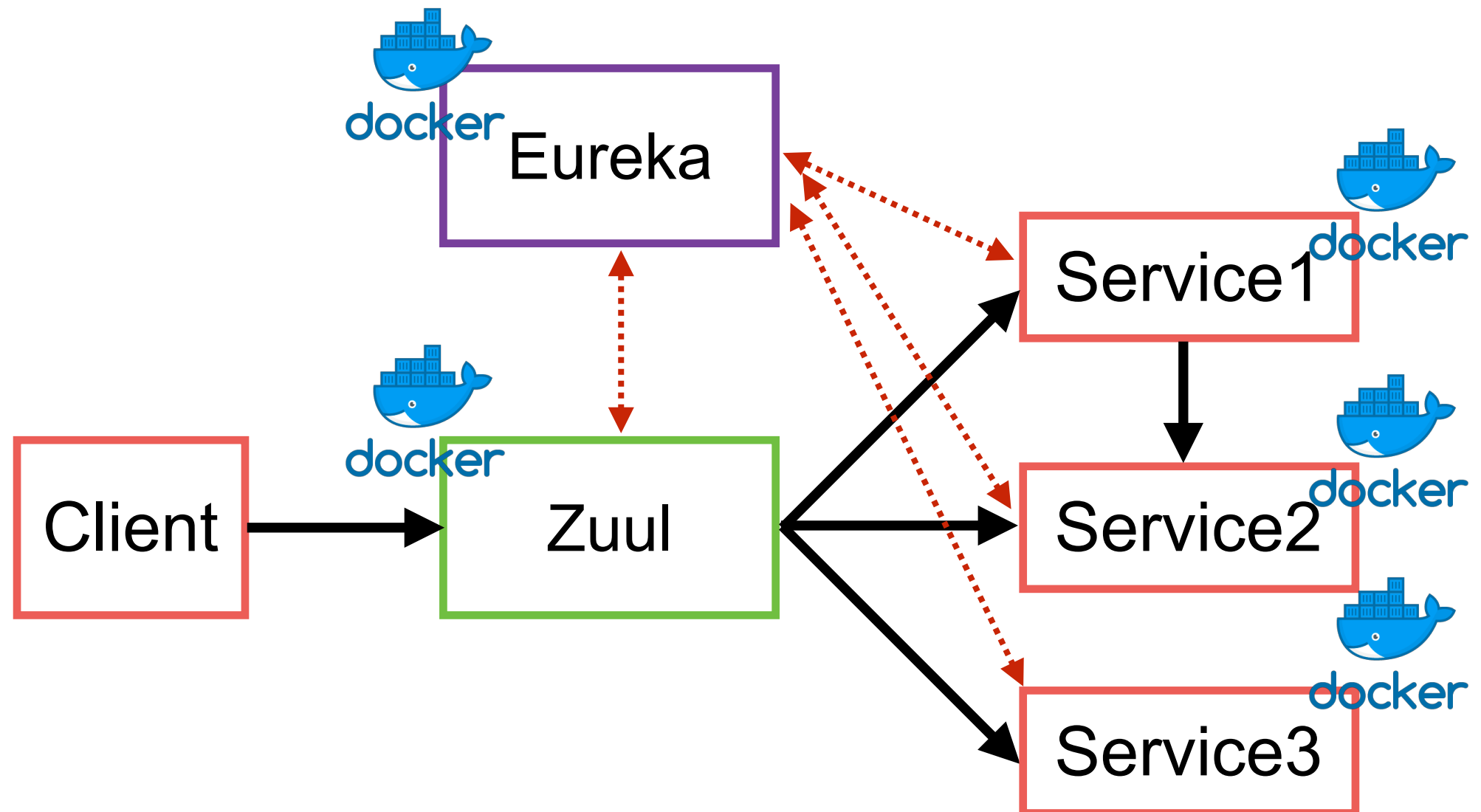
Original Architecture



How to deploy with Docker ?

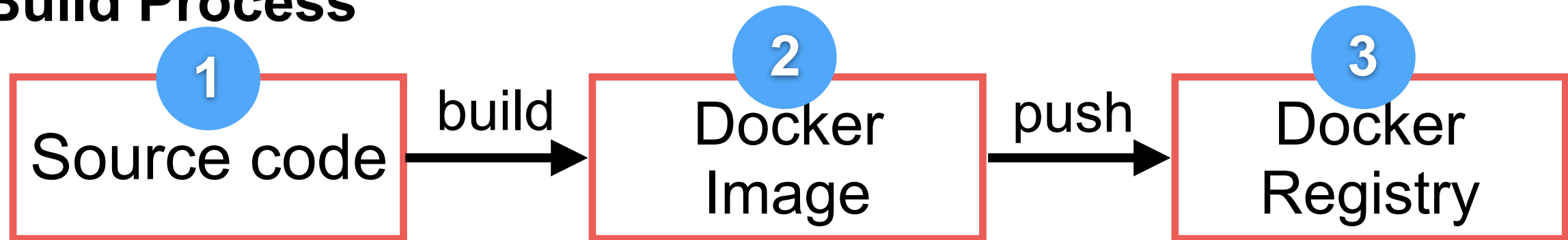


Deploy with Docker

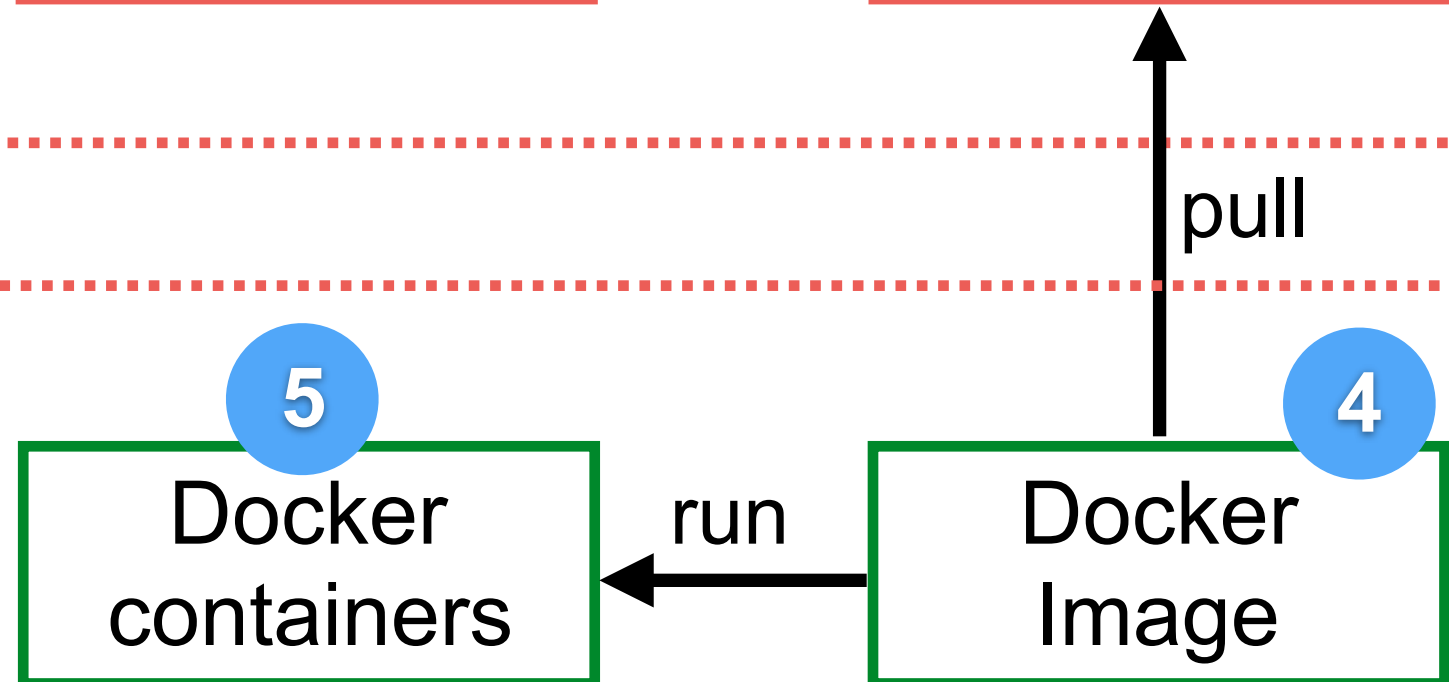


Process to build and deploy

Build Process



Deploy Process



Step 1 :: Organize source code



Step 1 :: Organize source code ?

Single source code repository

Single source code + subdirectory

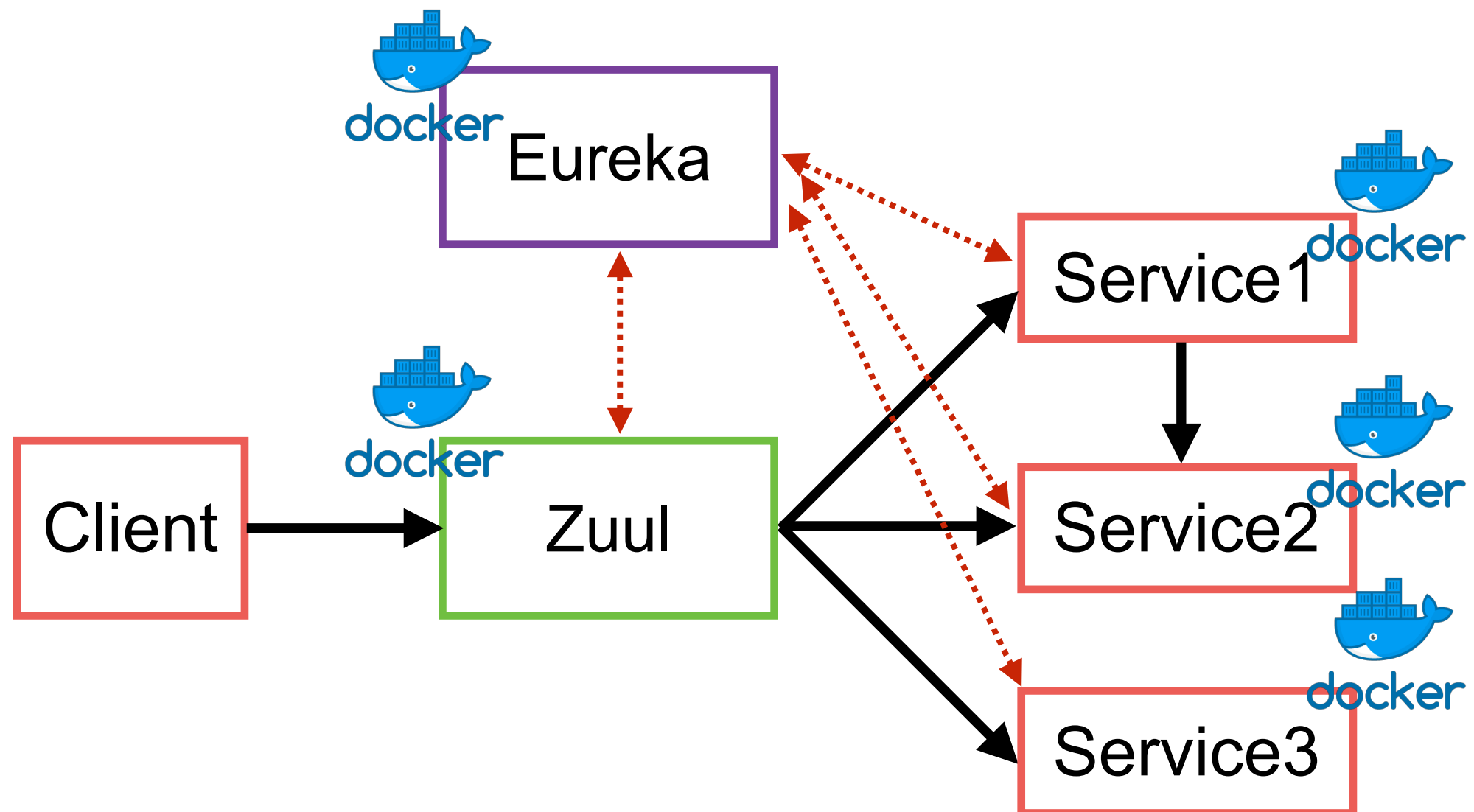
Source code repository per service



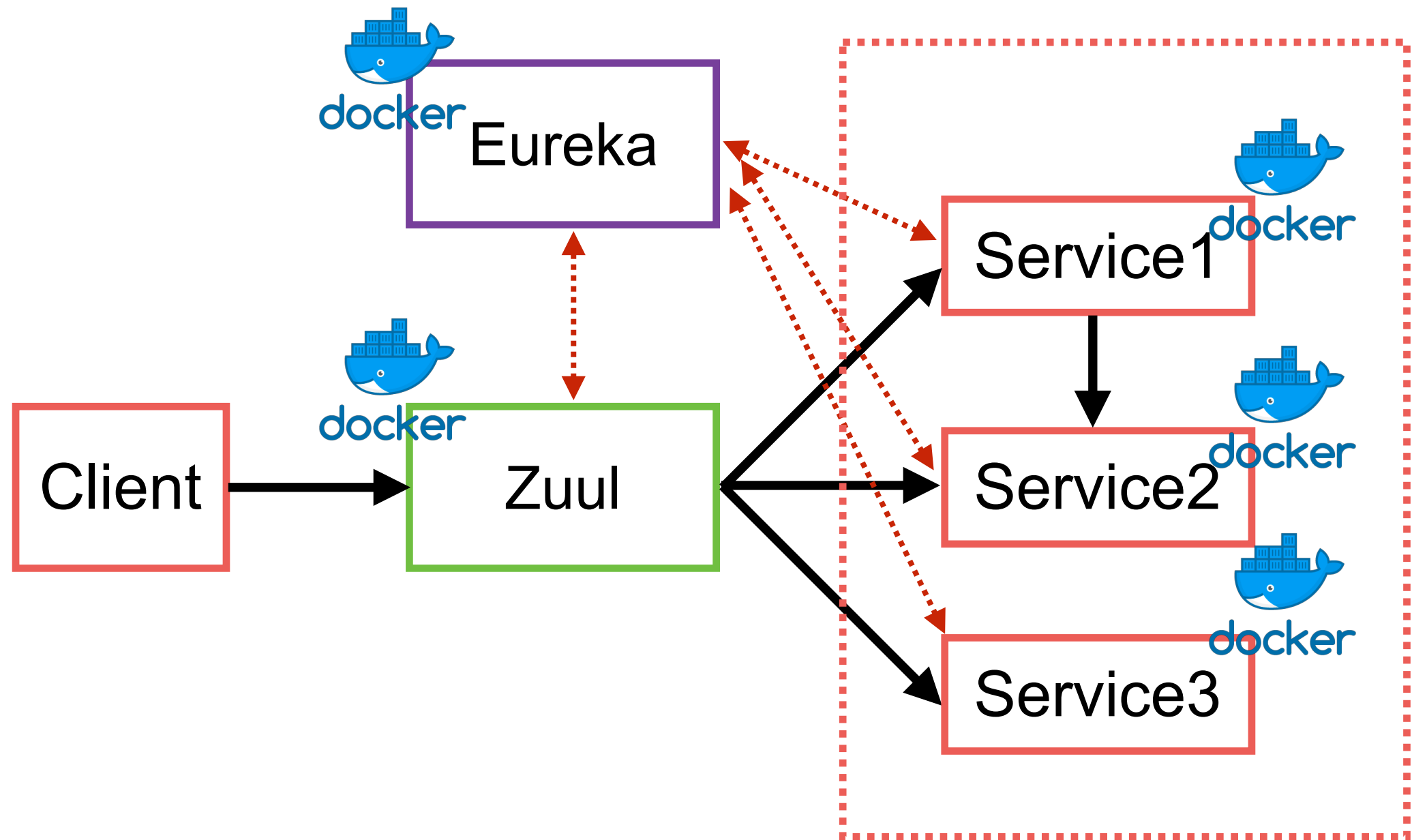
Step 2 :: Build Docker Image



Step 2 :: Build Docker Image



Step 2 :: Build Docker Image



Build image of services

Dockerfile

Docker command

Docker compose



Dockerfile

```
FROM node:10.14.2
WORKDIR /src
COPY package.json .
RUN npm install
COPY . .
EXPOSE 3002
CMD [ "npm", "start" ]
```



Dockerfile

```
FROM node:10.14.2
```

Base Image

```
WORKDIR /src
```

```
COPY package.json .
```

```
RUN npm install
```

```
COPY . .
```

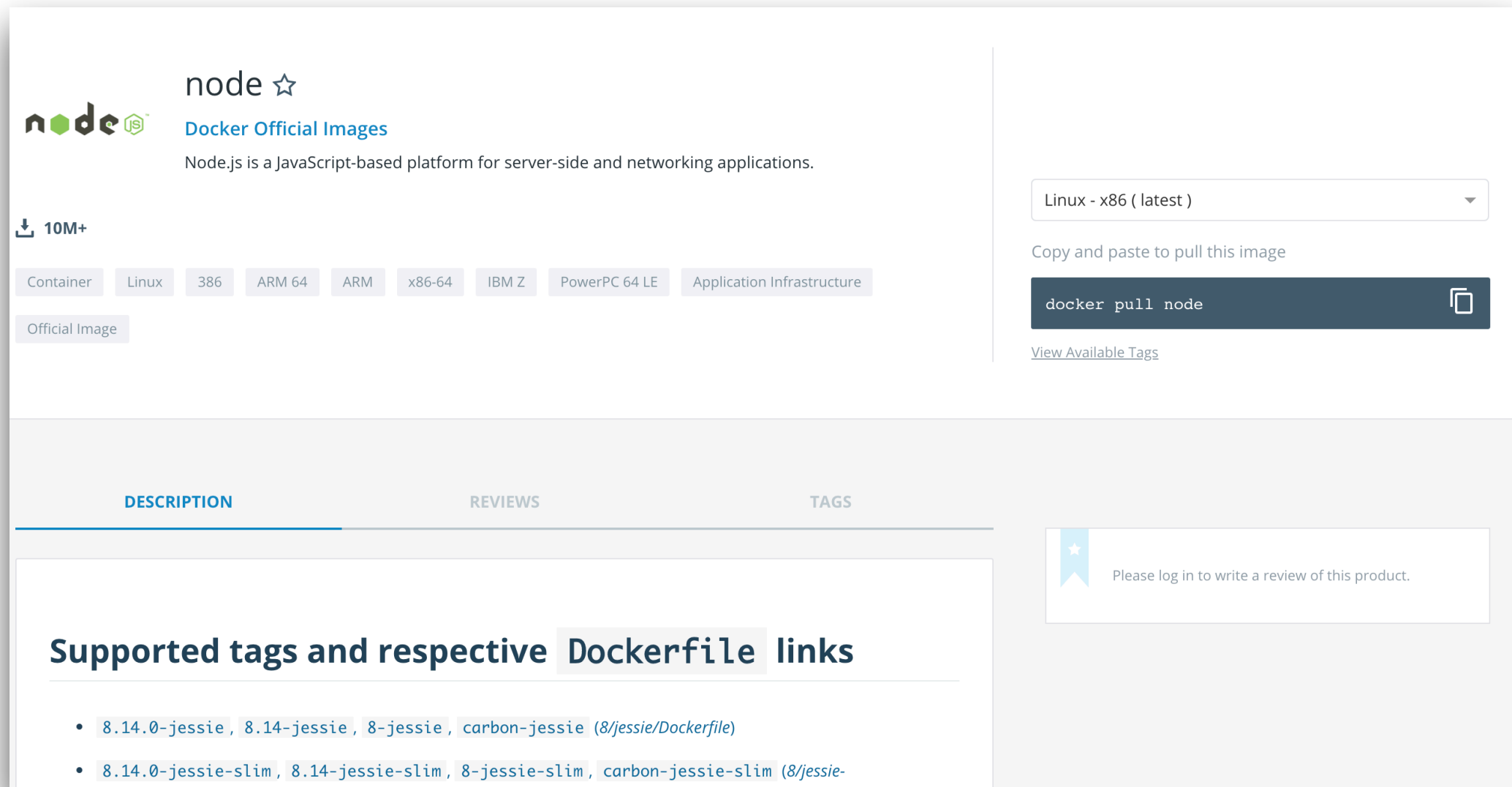
```
EXPOSE 3002
```

```
CMD [ "npm", "start" ]
```



Base Image

FROM node:10.14.2



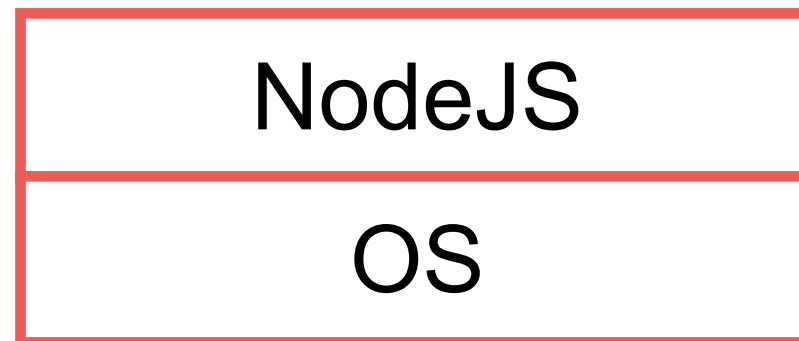
The screenshot shows the Docker Hub page for the 'node' image. The page header includes the 'node' logo, a star icon, and the text 'Docker Official Images'. Below this, it states 'Node.js is a JavaScript-based platform for server-side and networking applications.' and shows a download count of '10M+'. A horizontal bar lists various architectures: Container, Linux, 386, ARM 64, ARM, x86-64, IBM Z, PowerPC 64 LE, and Application Infrastructure. A button labeled 'Official Image' is also present. On the right side, there is a dropdown menu set to 'Linux - x86 (latest)', a text field with the instruction 'Copy and paste to pull this image', and a dark blue button with the command 'docker pull node' and a copy icon. Below the button is a link 'View Available Tags'. The main content area has three tabs: 'DESCRIPTION', 'REVIEWS', and 'TAGS'. The 'DESCRIPTION' tab is active, showing the title 'Supported tags and respective Dockerfile links' and a list of tags: '8.14.0-jessie', '8.14-jessie', '8-jessie', 'carbon-jessie' (with a link to the Dockerfile), and '8.14.0-jessie-slim', '8.14-jessie-slim', '8-jessie-slim', 'carbon-jessie-slim' (with a link to the Dockerfile). A sidebar on the right contains a message: 'Please log in to write a review of this product.'

https://hub.docker.com/_/node/



Base Image

FROM node:10.14.2



Dockerfile

```
FROM node:10.14.2
```

```
WORKDIR /src
```

```
COPY package.json .
```

```
RUN npm install
```

```
COPY . .
```

```
EXPOSE 3002
```

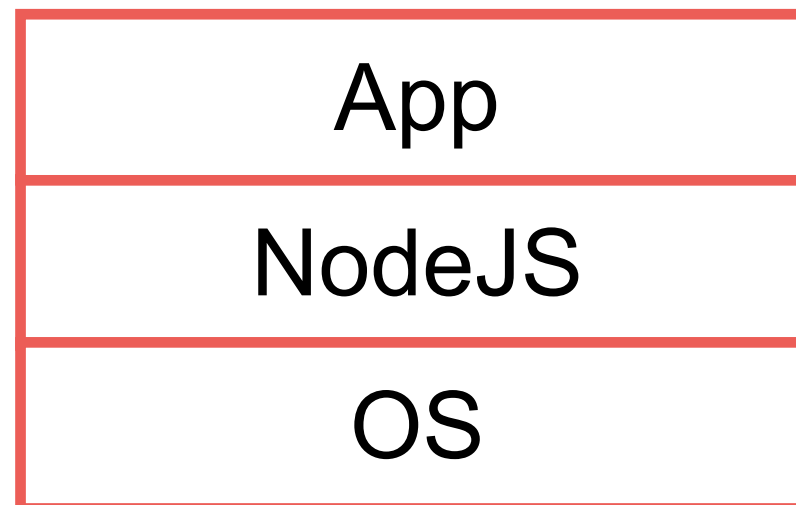
```
CMD [ "npm", "start" ]
```

Application code
and library



Application

```
FROM node:10.14.2  
WORKDIR /src  
COPY package.json .  
RUN npm install  
COPY . .
```



Dockerfile

```
FROM node:10.14.2
WORKDIR /src
COPY package.json .
RUN npm install
COPY . .
EXPOSE 3002
CMD [ "npm", "start" ]
```

Start process of service



Build Docker Image

`$docker image build -t <name> .`



Step 3 :: Keep Docker Image into Docker Registry



Keep Docker Image

Docker registry
Default at Docker Hub



Format name of Docker Image

Name	Meaning
hello-world	Official image at Docker Hub
somkiat/hello-world	Community/User image at Docker Hub
<u>http://ip:port/hello-world</u>	Custom Docker Registry Server to keep image



Push Docker Image to Registry

```
$docker image push <name>
```



Step 4 :: Deploy



Deploy services

Docker command

Docker compose



Docker compose

```
version: '3.7'  
services:
```

```
  service1:  
    image: somkiat/service1  
    build:  
      context: ./service1  
    ports:  
      - "3001:3000"
```

```
  service2:  
    image: somkiat/service2  
    build:  
      context: ./service2
```



Docker compose

```
version: '3.7'  
services:
```

```
  service1:  
    image: somkiat/service1  
    build:  
      context: ./service1  
    ports:  
      - "3001:3000"
```

```
  service2:  
    image: somkiat/service2  
    build:  
      context: ./service2
```



Docker compose

```
version: '3.7'  
services:
```

```
  service1:  
    image: somkiat/service1  
    build:  
      context: ./service1  
    ports:  
      - "3001:3000"
```

```
  service2:  
    image: somkiat/service2  
    build:  
      context: ./service2
```



Run with docker compose

\$docker-compose build

\$docker-compose up

\$docker-compose down



Docker compose file by process

Build

docker-compose.yml

Deploy

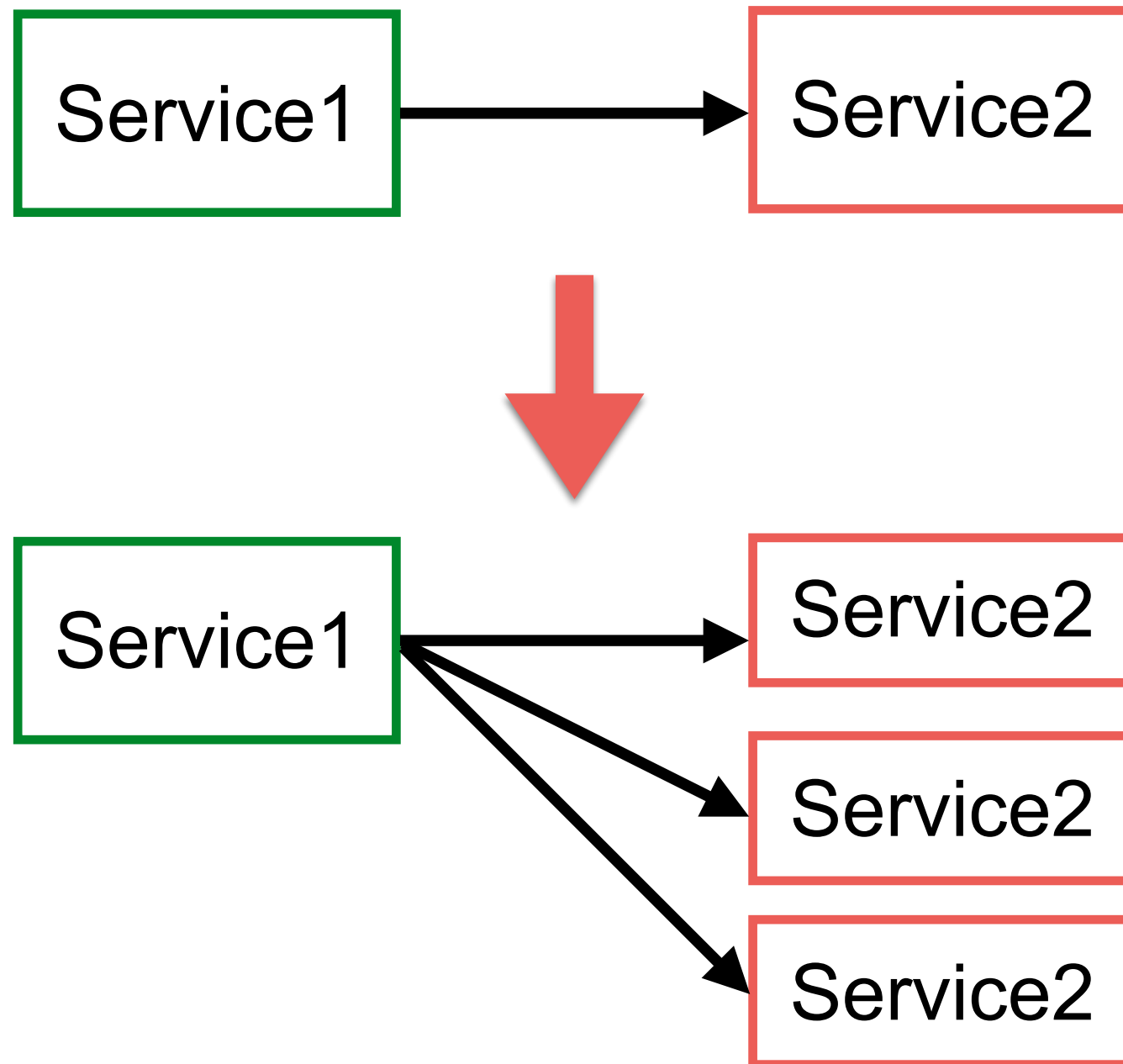
docker-compose-deploy.yml



Step 5 :: Scale services



Scale services

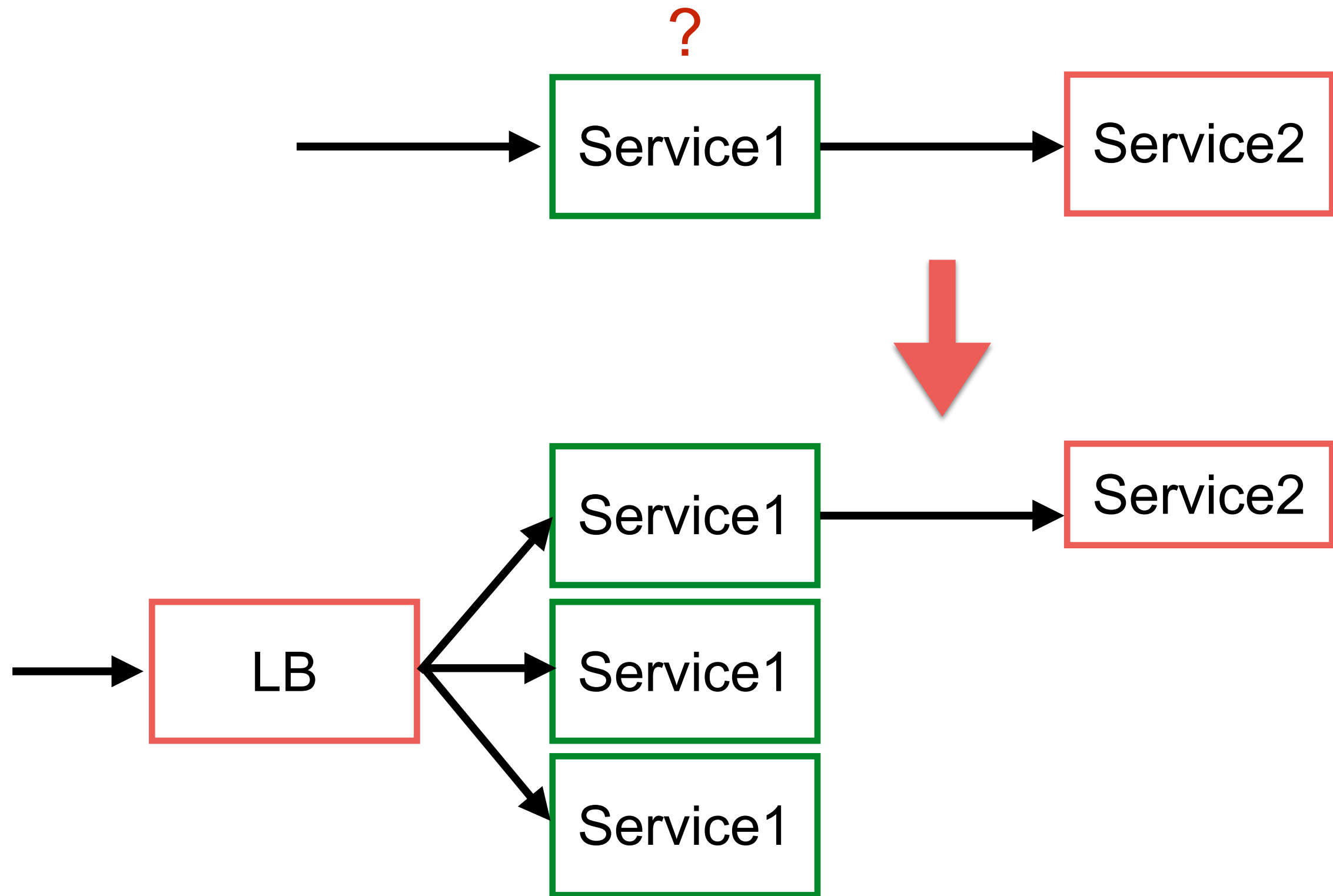


Run with docker compose

```
$docker-compose up --scale service2=3
```



Scale services



Run with docker compose

```
$docker-compose -f docker-compose-lb.yml  
up --scale service1=3
```

