Docker Overview

by Sunit & Pratyusha

Cloud Services

- IaaS
- CaaS
- PaaS
- SaaS

Pizza as a Service

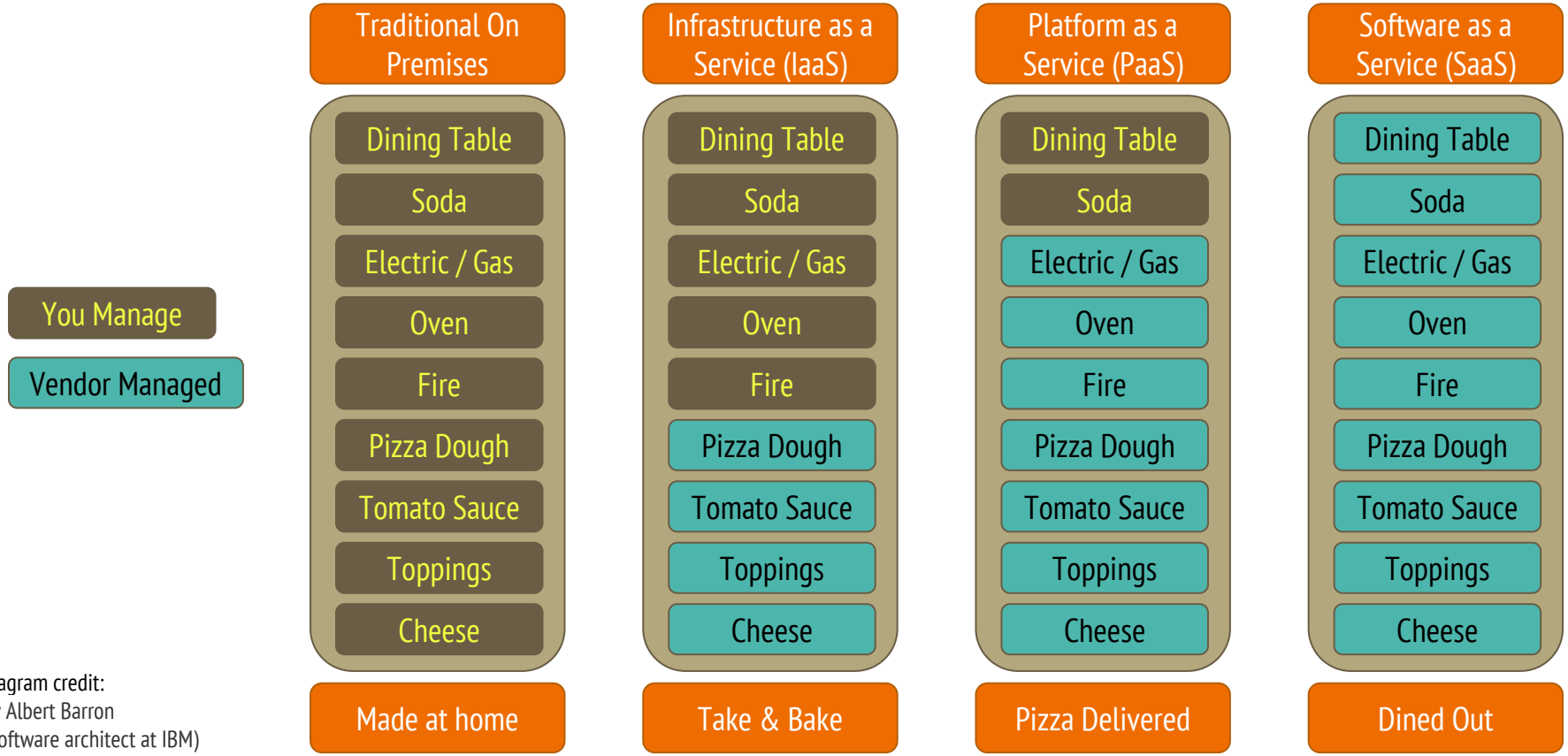
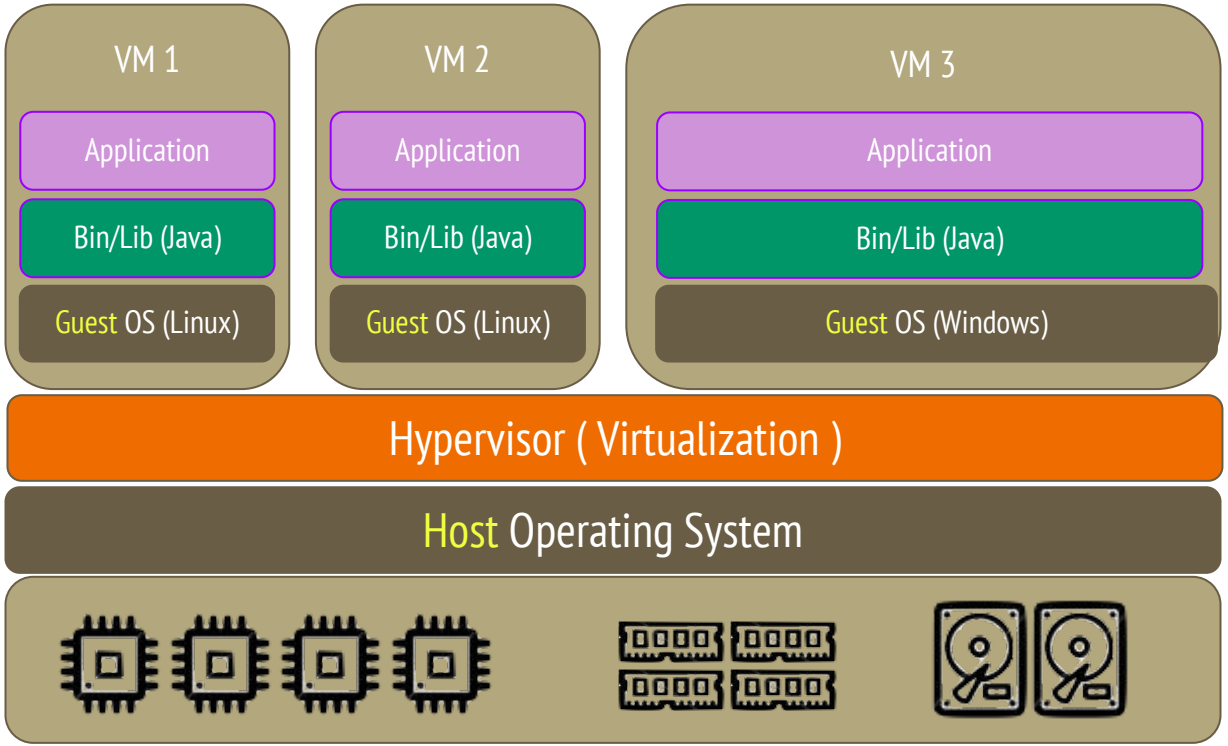
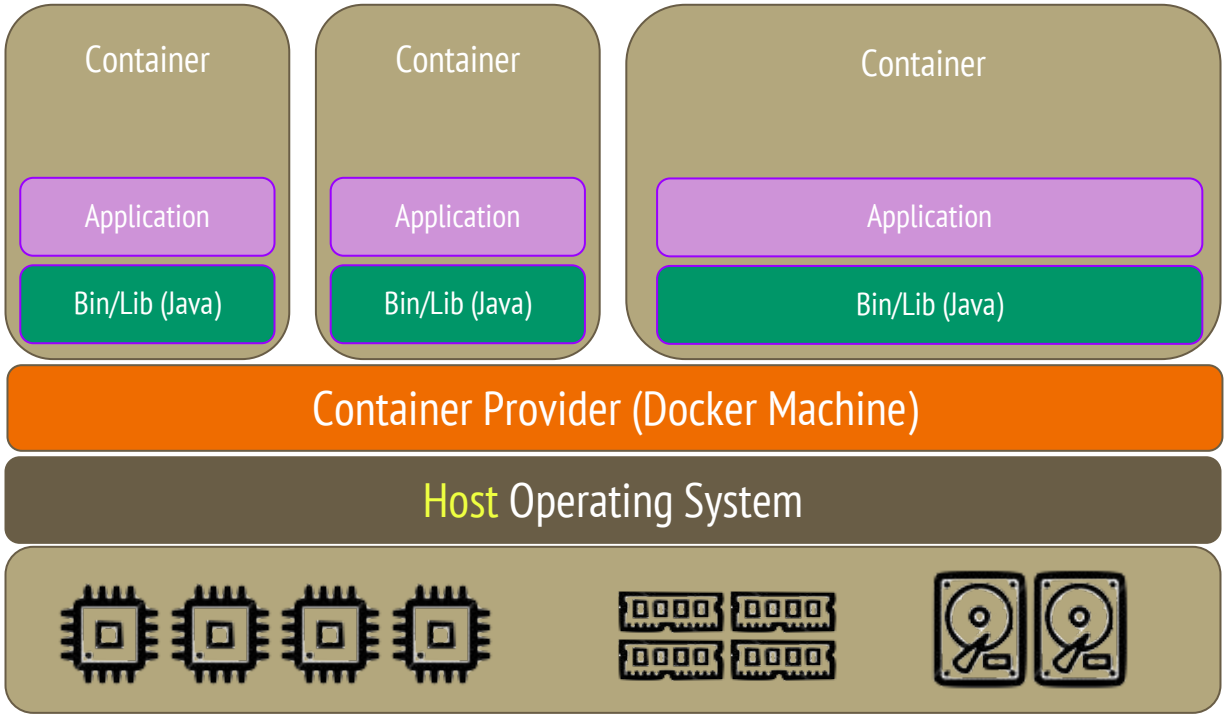


diagram credit:
by Albert Barron
(Software architect at IBM)

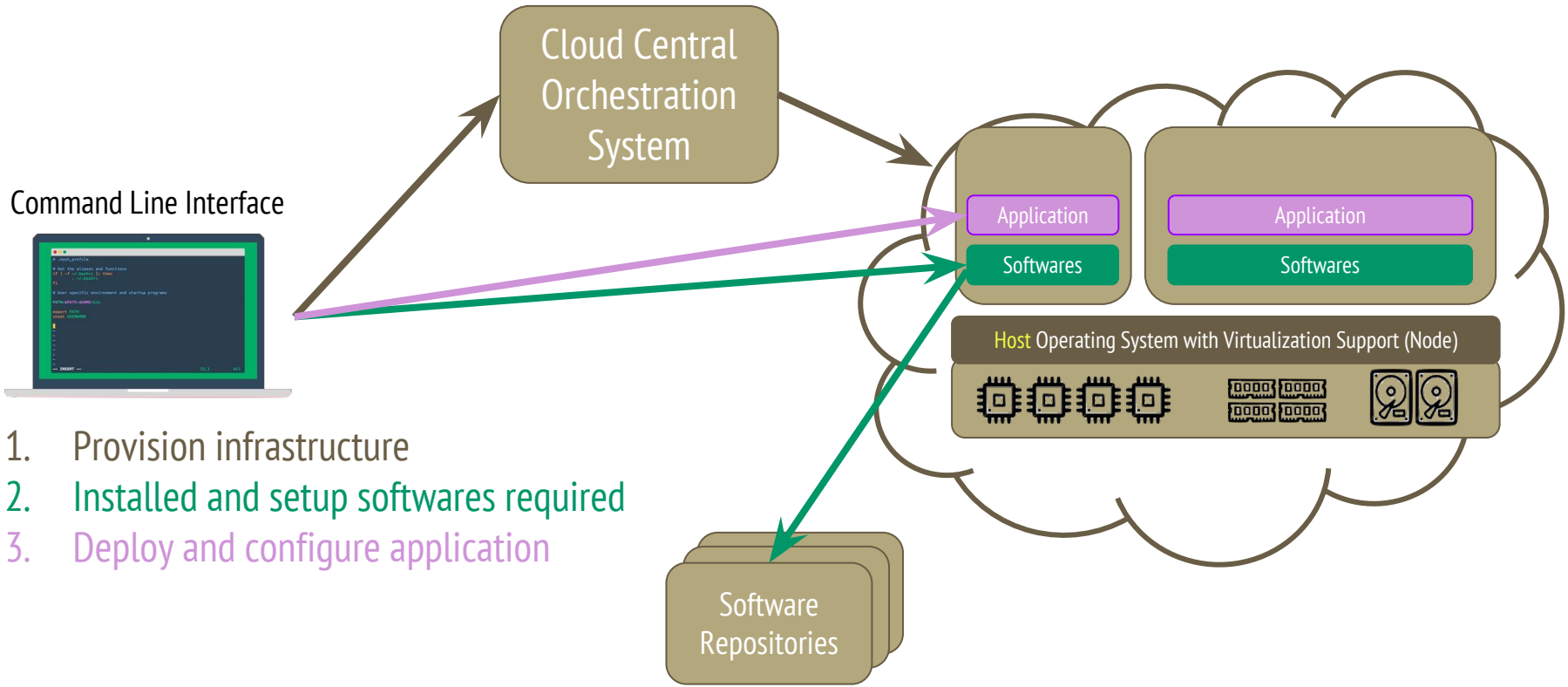
Traditional virtualization



Container based virtualization



Cloud with DevOps

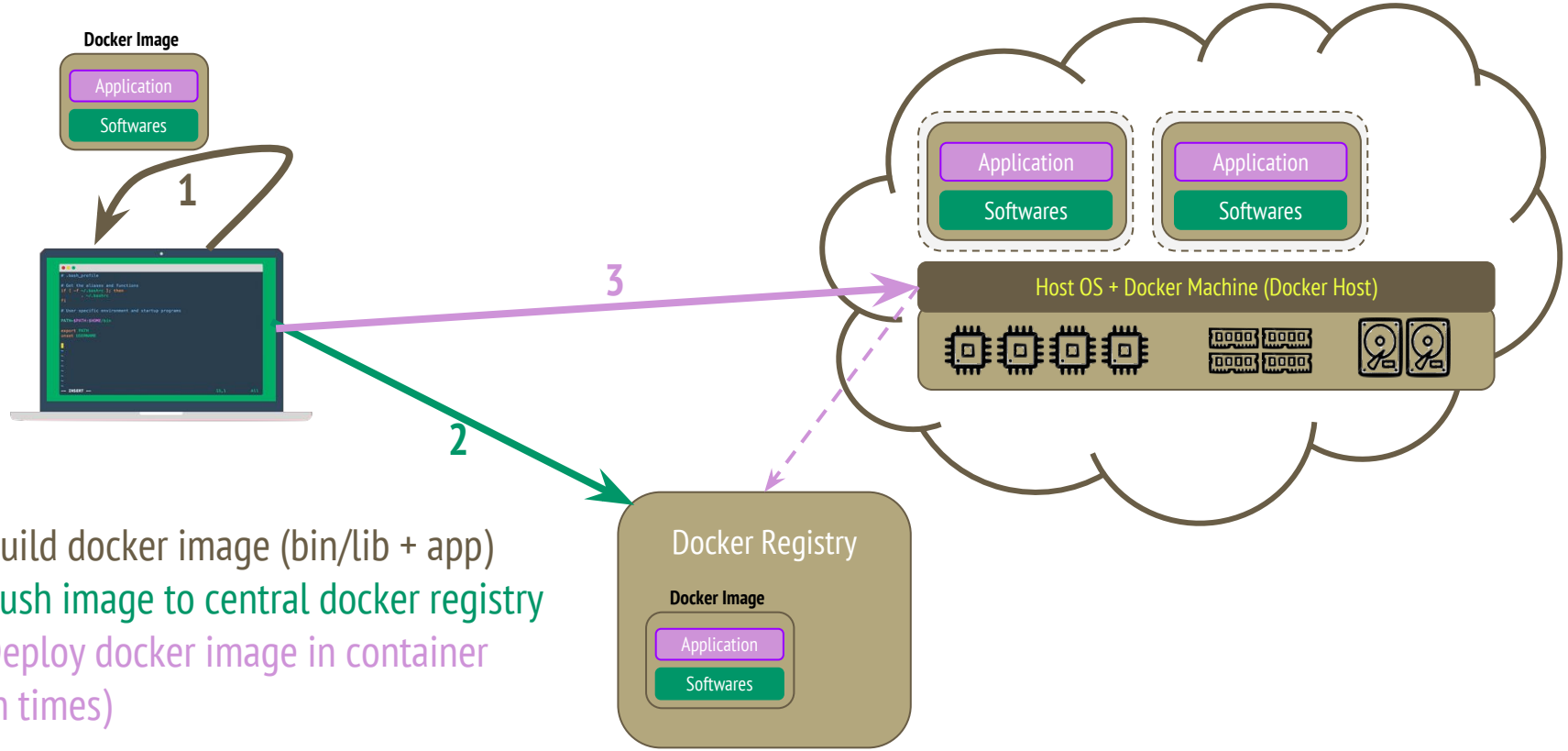


1. Provision infrastructure
2. Installed and setup softwares required
3. Deploy and configure application

Docker....

- Build Docker **images** that hold your applications.
- Create Docker **containers** from Docker images to run your applications.
- Share those Docker images via Docker **registry**.

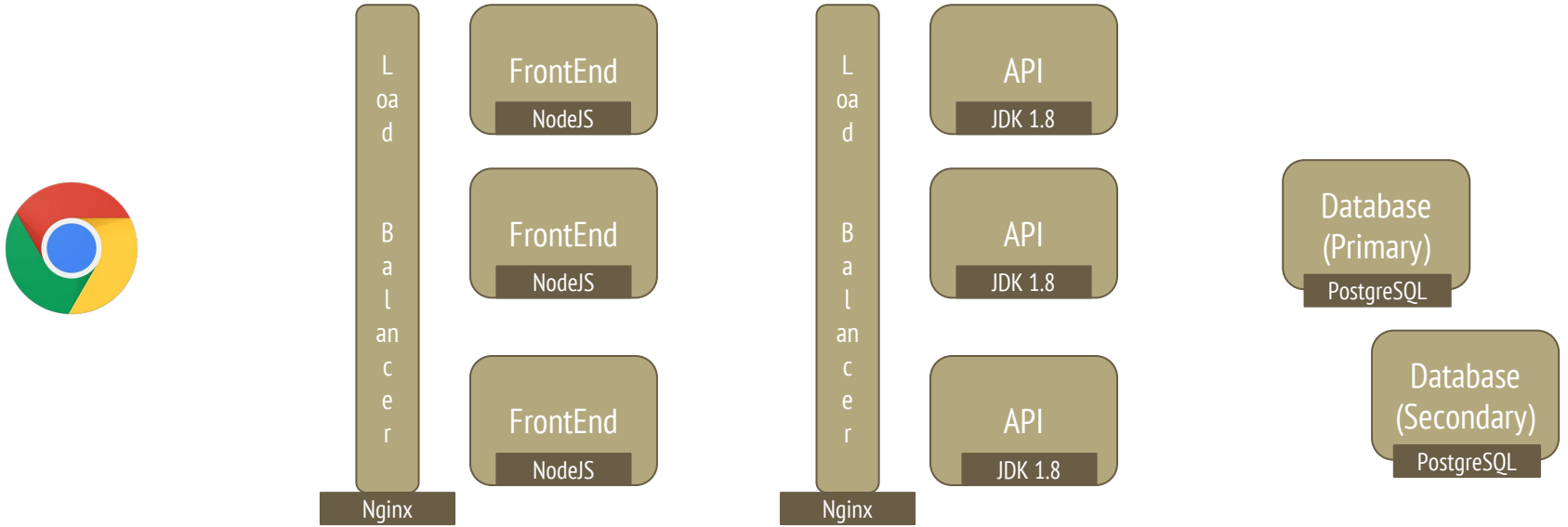
How Docker works?



1. Build docker image (bin/lib + app)
2. Push image to central docker registry
3. Deploy docker image in container (n times)

Demo

Environment setup is more complex

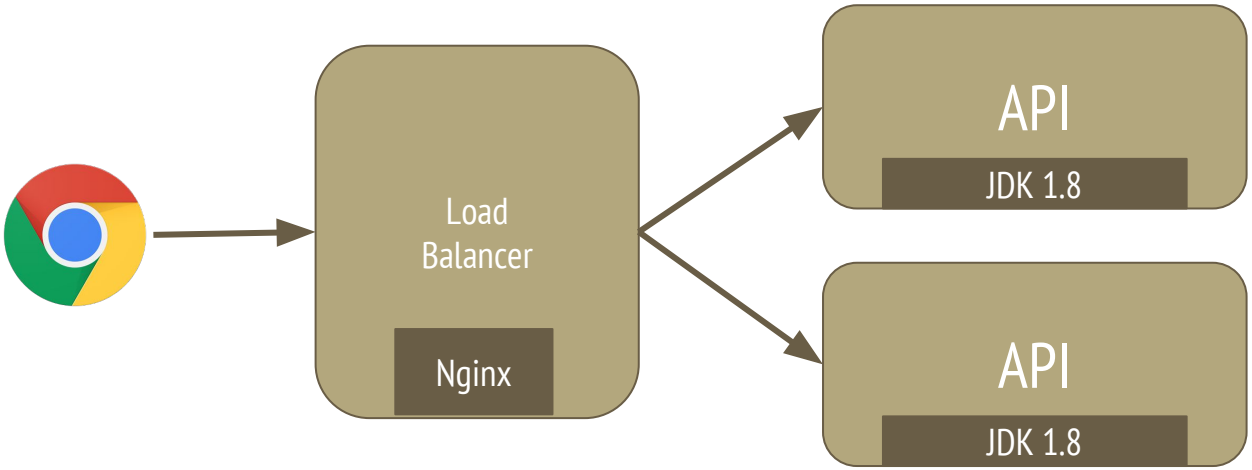


Full environment setup is much more complex than single container, frontends, api servers, load balancers, database cluster...

Docker Compose

Tool for deploying multi-container docker applications.

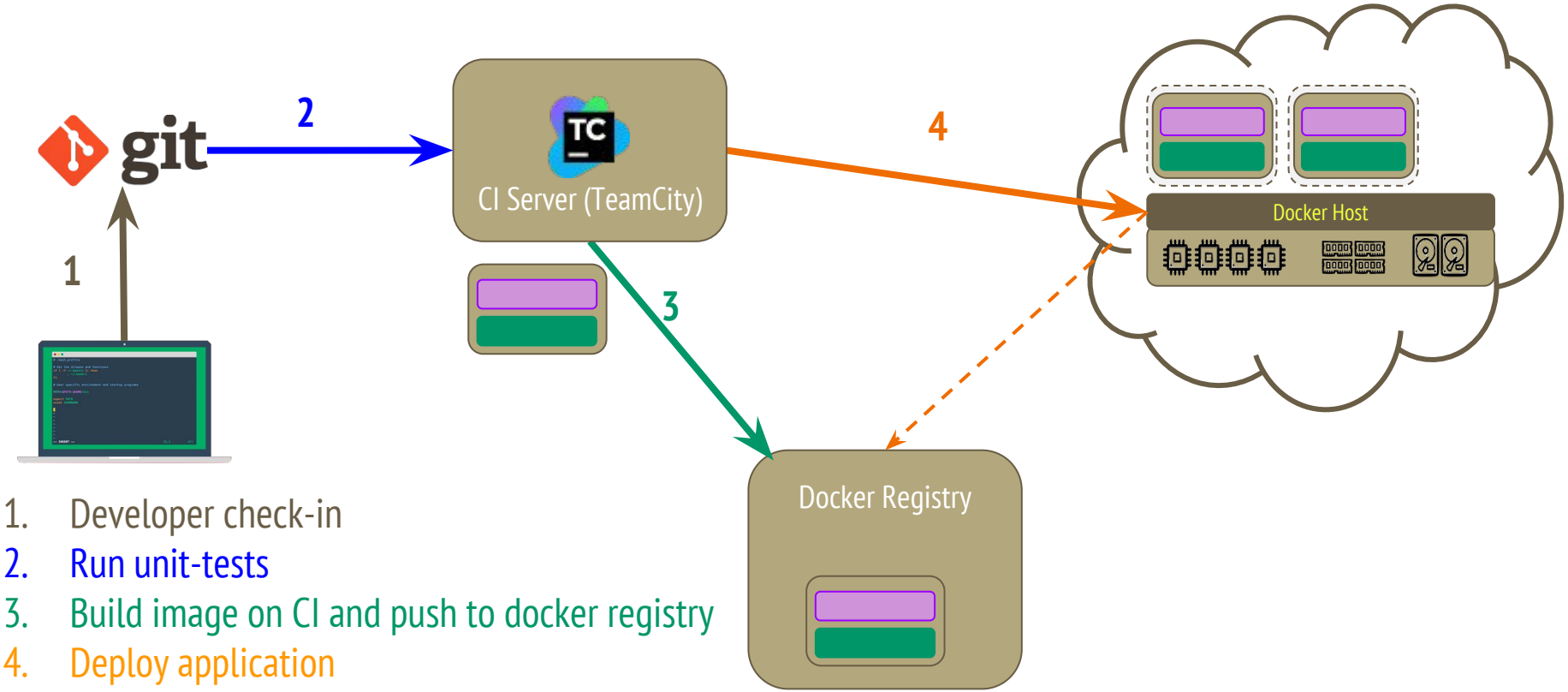
API with Load Balancer



Docker Compose Demo

How?

How it looks end to end?



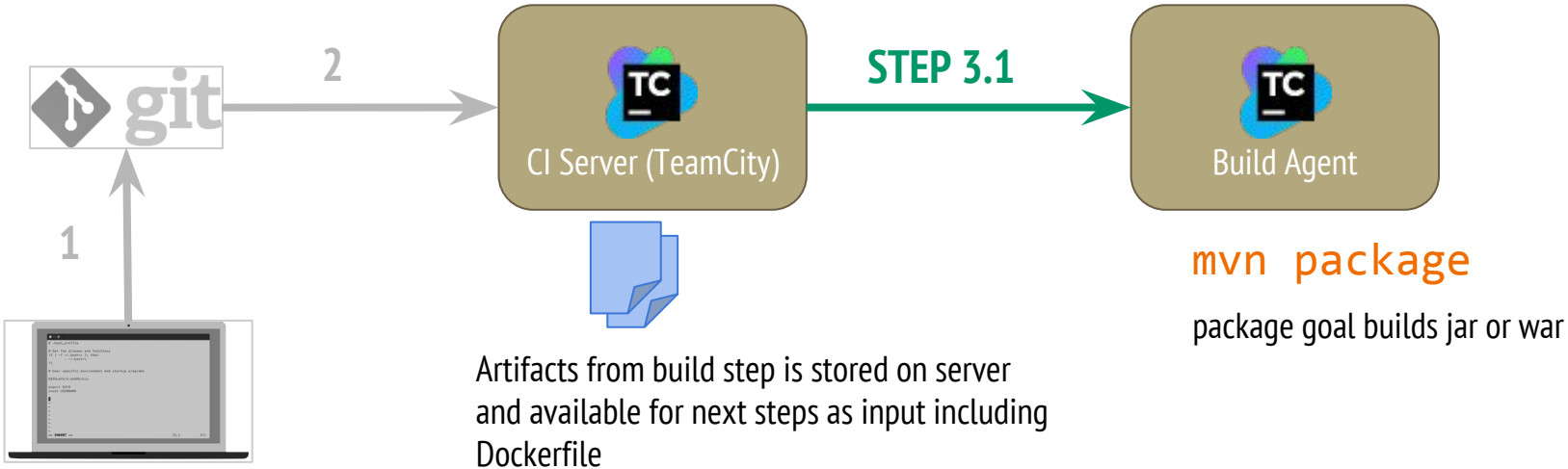
1. Developer check-in
2. Run unit-tests
3. Build image on CI and push to docker registry
4. Deploy application

STEP 3: Build image and push

Lets divide it into different sub-steps,

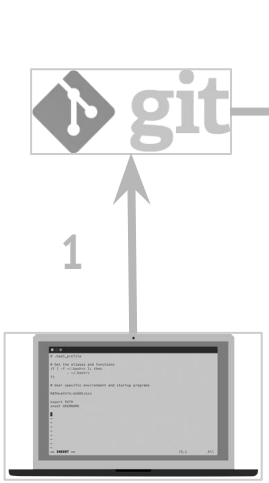
1. Create project artifacts (war, jar, configs ...)
2. Build image and push to Docker registry

Create project artifacts



Sample Spring Boot Project with Dockerfile:
https://github.com/sunitparekh/spring-boot-hello-world

Build image and push to docker registry



1

2

CI Server (TeamCity)

STEP 3.1

Build Agent

mvn package

STEP 3.2

Docker Agent

```
docker build -t project/app:tag -f Dockerfile
```

Build docker image using Dockerfile and artifacts available from previous steps

Docker Registry

```
docker push project/app:tag
```

Sends the newly build image to central docker registry

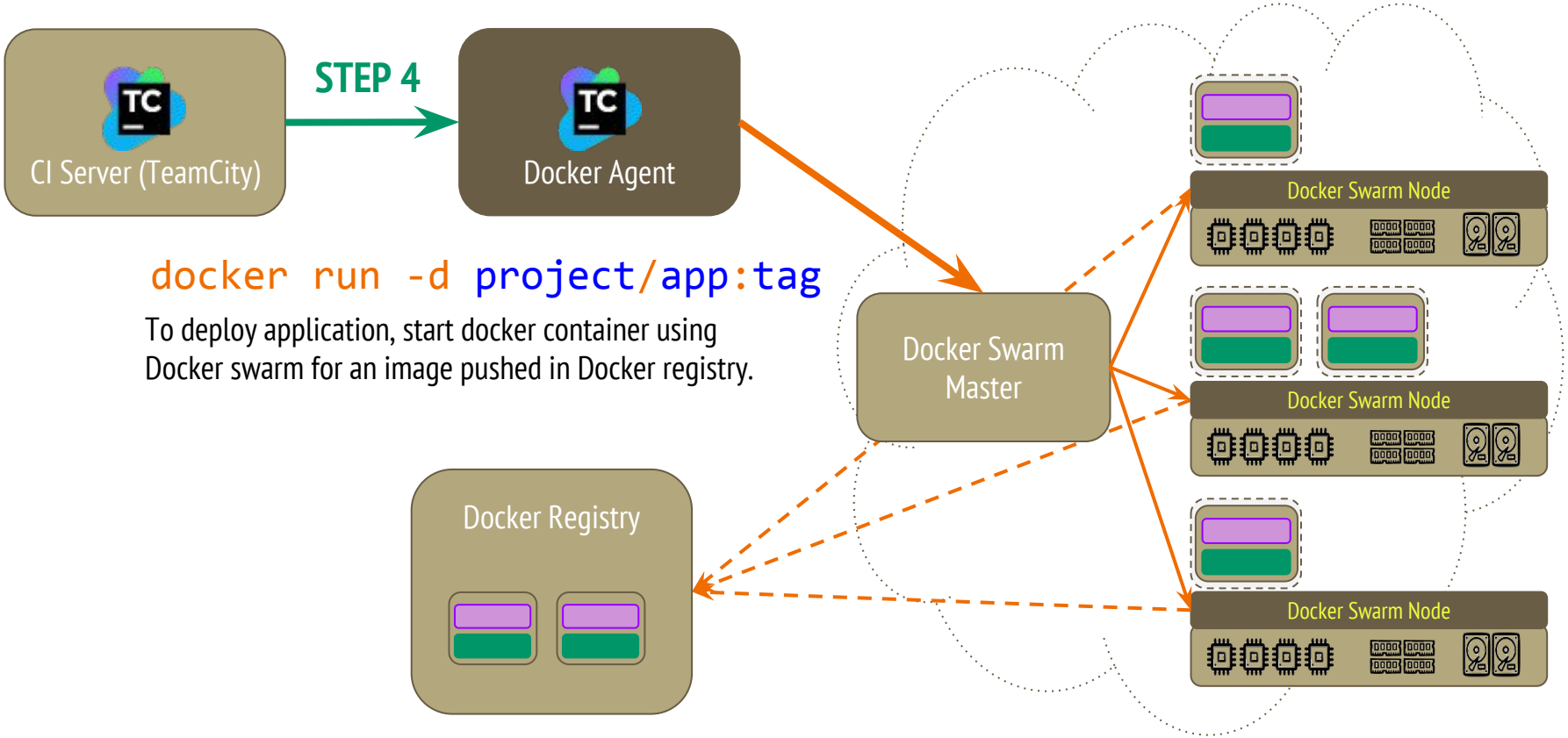
Dockerfile

4 lines (4 sloc) | 169 Bytes

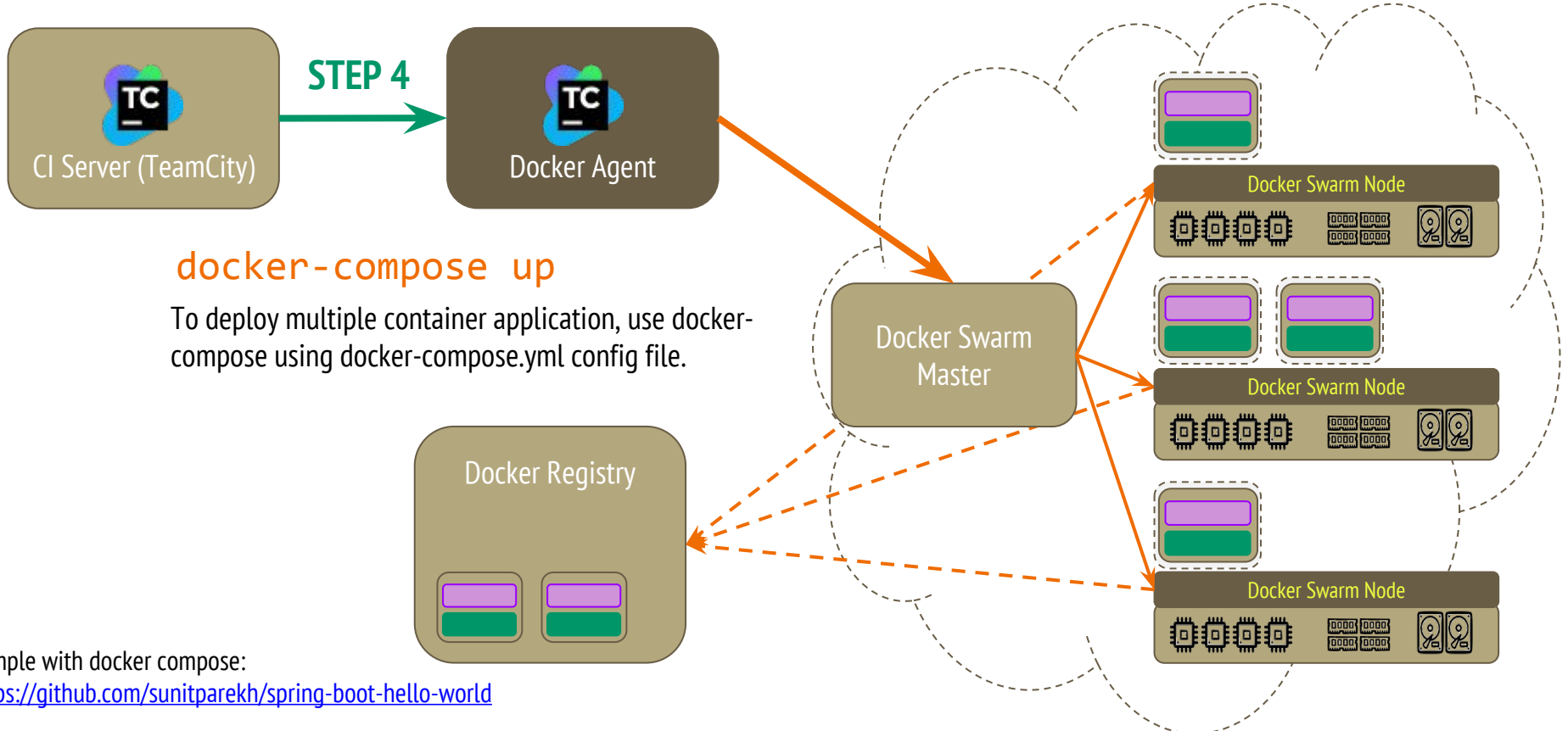
```
1  FROM frolvlad/alpine-oraclejdk8:slim
2  ADD target/hello-world-*RELEASE.jar hello-world.jar
3  EXPOSE 8080
4  ENTRYPOINT ["java","-jar","-Xms256m", "-Xmx512m","/hello-world.jar"]
```

STEP 4: Deploy image in container

Deploy image in container



Deploy multi container apps with docker-compose



`docker-compose up`

To deploy multiple container application, use docker-compose using docker-compose.yml config file.

Sample with docker compose:
https://github.com/sunitparekh/spring-boot-hello-world

docker-compose.yml

17 lines (15 sloc) | 267 Bytes

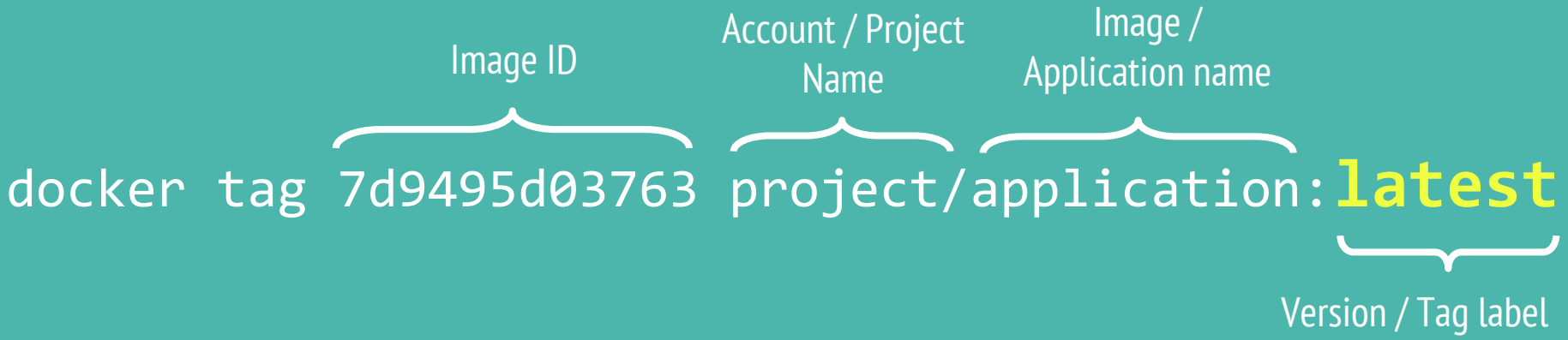
```
1  app1:
2      image: sunitparekh/spring-boot-hello
3      ports:
4          - "8080"
5  app2:
6      image: sunitparekh/spring-boot-hello
7      ports:
8          - "8080"
9
10 nginx1:
11     image: sunitparekh/spring-boot-nginx
12     ports:
13         - "80"
14     links:
15         - app1
16         - app2
```

Multiple Environments

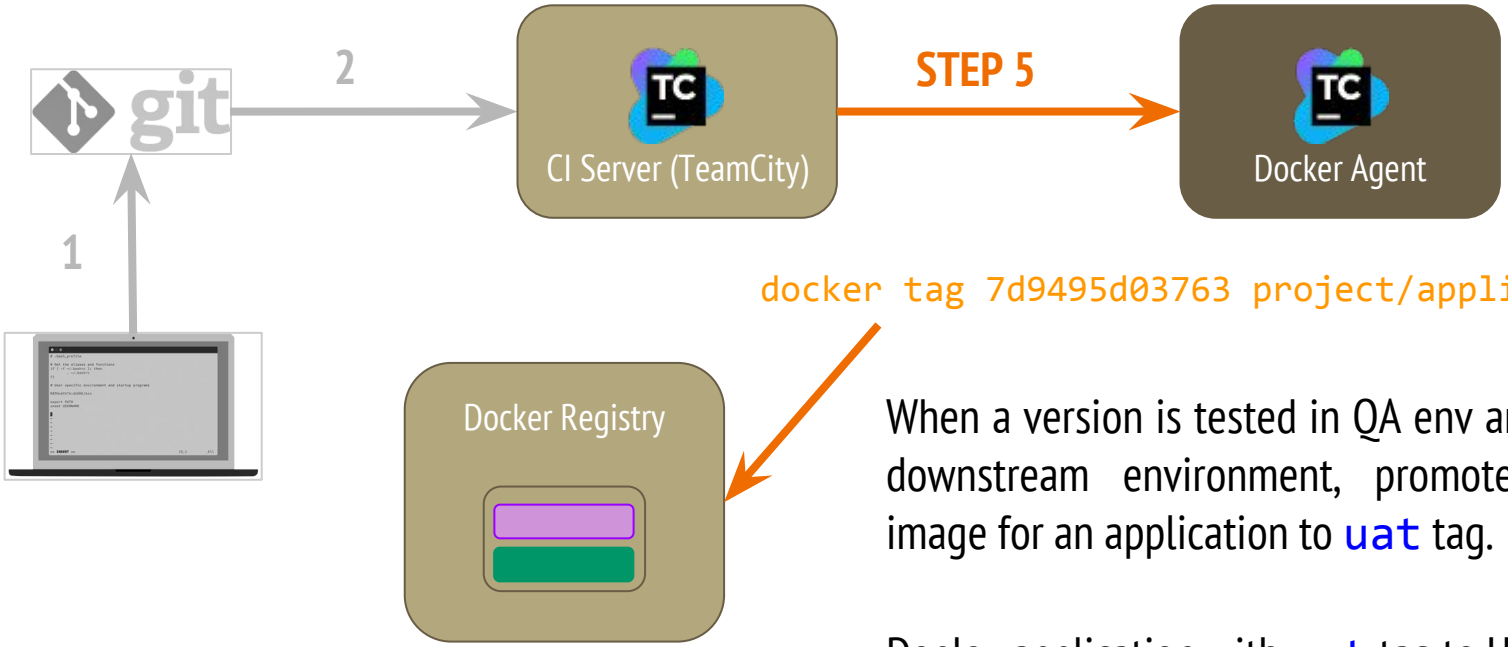
Now we figured out how to deploy multiple container application.

Next thing is, how to manage multiple environments such as SIT, QA, UAT and PROD

Use image **tags** to support multiple environments



Promote docker image with tag for an environment



When a version is tested in QA env and ready for next downstream environment, promote tested docker image for an application to `uat` tag.

Deploy application with `uat` tag to UAT environment.

What we need?

1. Docker CI/**TeamCity Agent** to build, push and deploy images
    - https://www.docker.com/products/docker-compose
2. Docker Trusted **Registry**
    - https://www.docker.com/products/docker-trusted-registry
3. Docker **Swarm** with Docker Nodes
    - https://www.docker.com/products/docker-swarm
4. Docker Universal Control Plane
    - https://www.docker.com/products/docker-universal-control-plane

Kubernetes

Another alternative to Docker Swarm and Docker Compose is Kubernetes.

http://kubernetes.io/

Questions ?

Further Resources

- Docker
  - https://training.docker.com/self-paced-training
  - https://docs.docker.com/v1.8/introduction/understanding-docker/
- uDemy course on Docker

Thanks!

Sunit Parekh
@sunitparekh