Playing root on a virtual box

CS252

Aug 9 2018

This week

- We experimented with netcat
  - Very powerful utility
- We experimented with scraping static web content and doing things with it
  - Could you do this with nc instead of wget?
- We experimented with basic shell programming
  - Could you write a shell script to clone a static data repository?

Next week

- We will move to CSE 2nd and 3rd floor labs
- We want to mess around with network configurations and set up our own servers
- Not having sudo access is annoying
- Solution: virtualization
  - You will experiment with docker
  - You will get to be root on your docker box, and manage permissions
  - You will figure out how to install ssh and a web server on your docker box

Docker

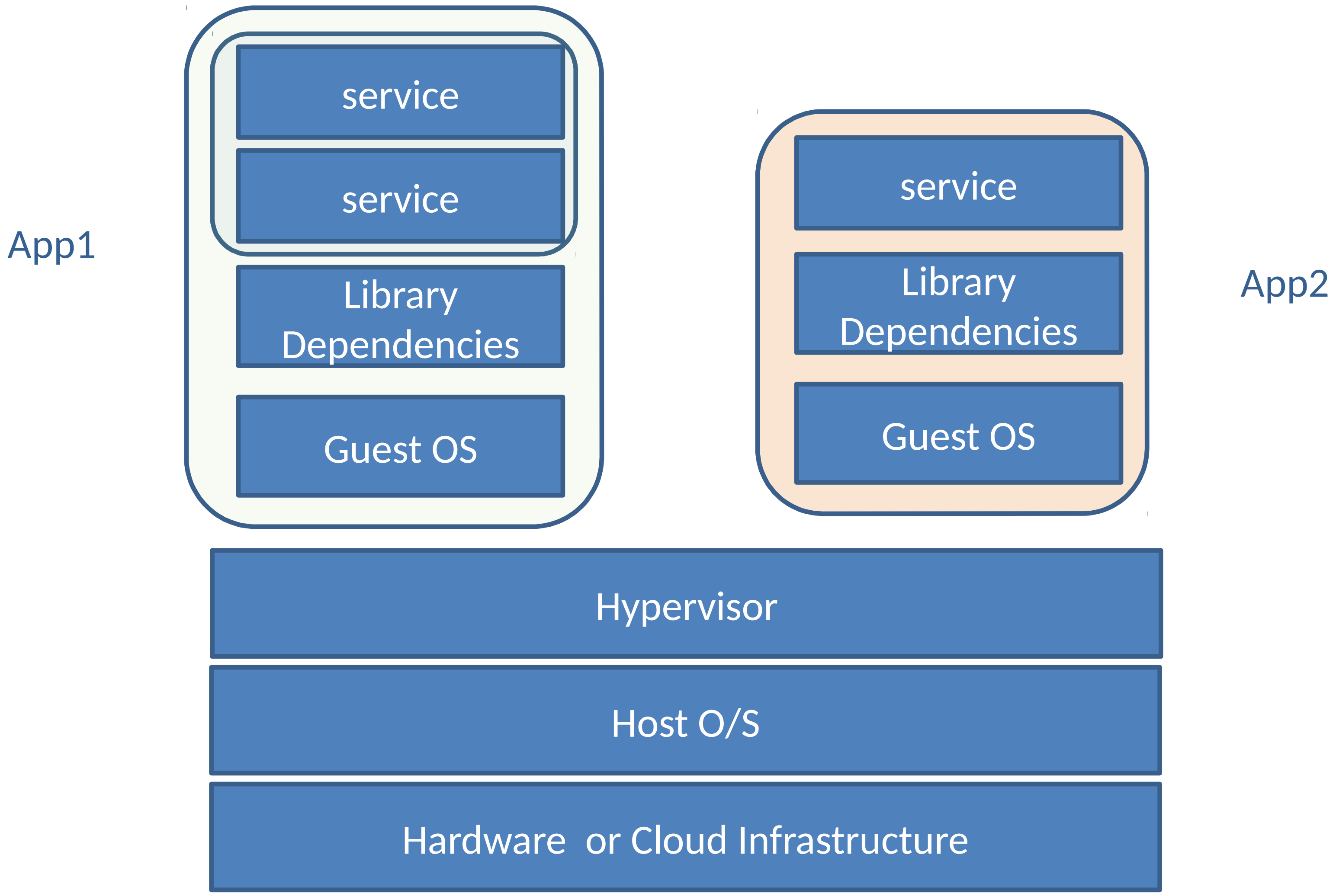
- A framework to help deploy *Linux Applications* packaged as *software containers*

- Avoids the traditional way of installation via *configuration scripts*

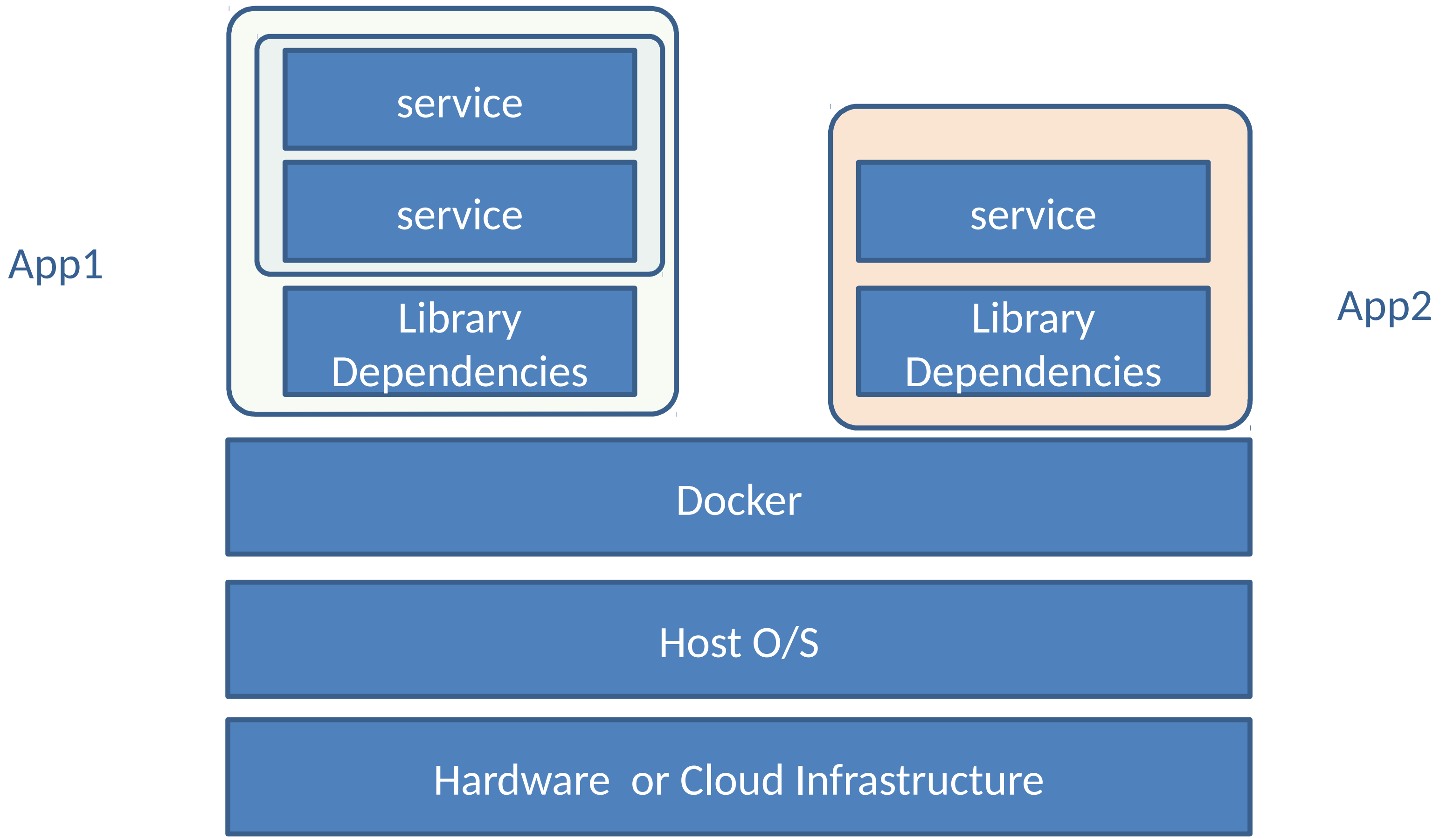
- Can be used for ordinary Linux applications as well as web applications

Image: Wikipedia Commons

Typical VM - Architecture



Docker - Architecture



Docker : Purpose

- A programmer creates an application on a development machine
- Package all dependencies for the applications (like libraries) into a “container”
- Deploy the container into a hub
- The container can be downloaded and deployed into any network or machine or cloud

Docker: Linux App example

```
$ docker run ubuntu /bin/echo 'Hello world’
Hello World
$
```

- `docker run` runs the container `ubuntu` which has to run the command `/bin/echo` with the argument `‘Hello World’`
- The container is pulled from a ***hub***

https://docs.docker.com/engine/tutorials/dockerizing/

Docker: Linux App example

```
$ docker run -t -i ubuntu /bin/bash
```

https://docs.docker.com/engine/tutorials/dockerizing/

Docker and Web Apps

A typical work-flow

- Develop your web application on your machine
- Test
- Deploy on a server or the cloud
- Change
- Test
- Redeploy

Docker and a node.js app

- In previous lectures we have seen a node.js application, and added a self-signed certificate to it.
- We build a docker image
- Test and run the docker image

Omitted (due to large size of the image)

- Deploy it to docker hub

Docker Documentation

A more sophisticated example

If the application consists of multiple containers. (e.g. a node container, a redis container etc.)

- Build each container separately using separate *Dockerfiles*
- Docker *Compose files* specify how the app is made from multiple containers.
- run `docker-compose up`

A sample compose file

```
node:
   build: ./node
   links:
      - redis
   ports:
      - “8080”

redis:
   image: redis
   ports:
      - “3001”
```

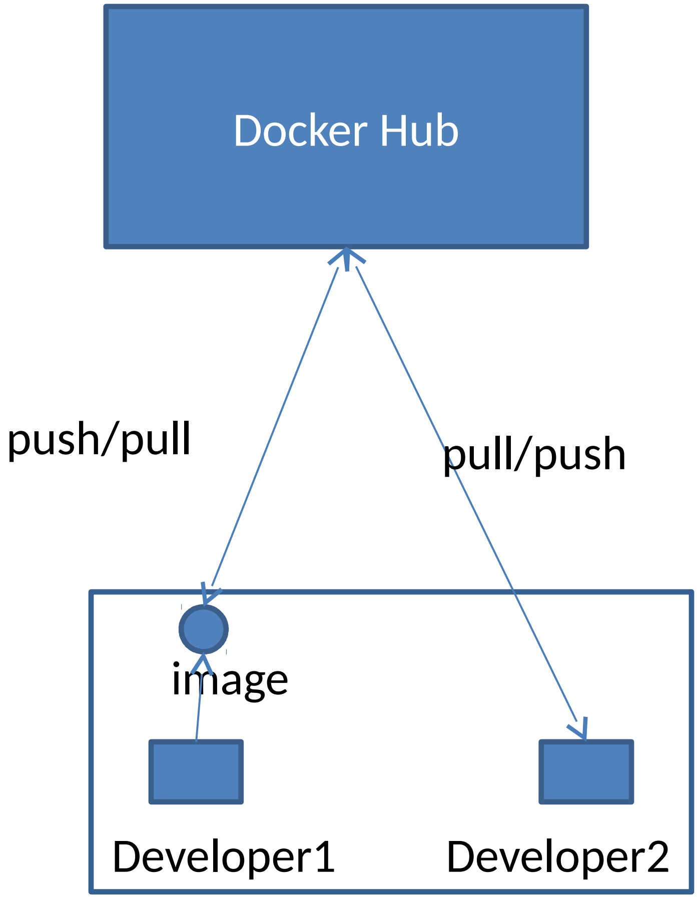
- You can scale dynamically:

```
    docker-compose scale node=4
```

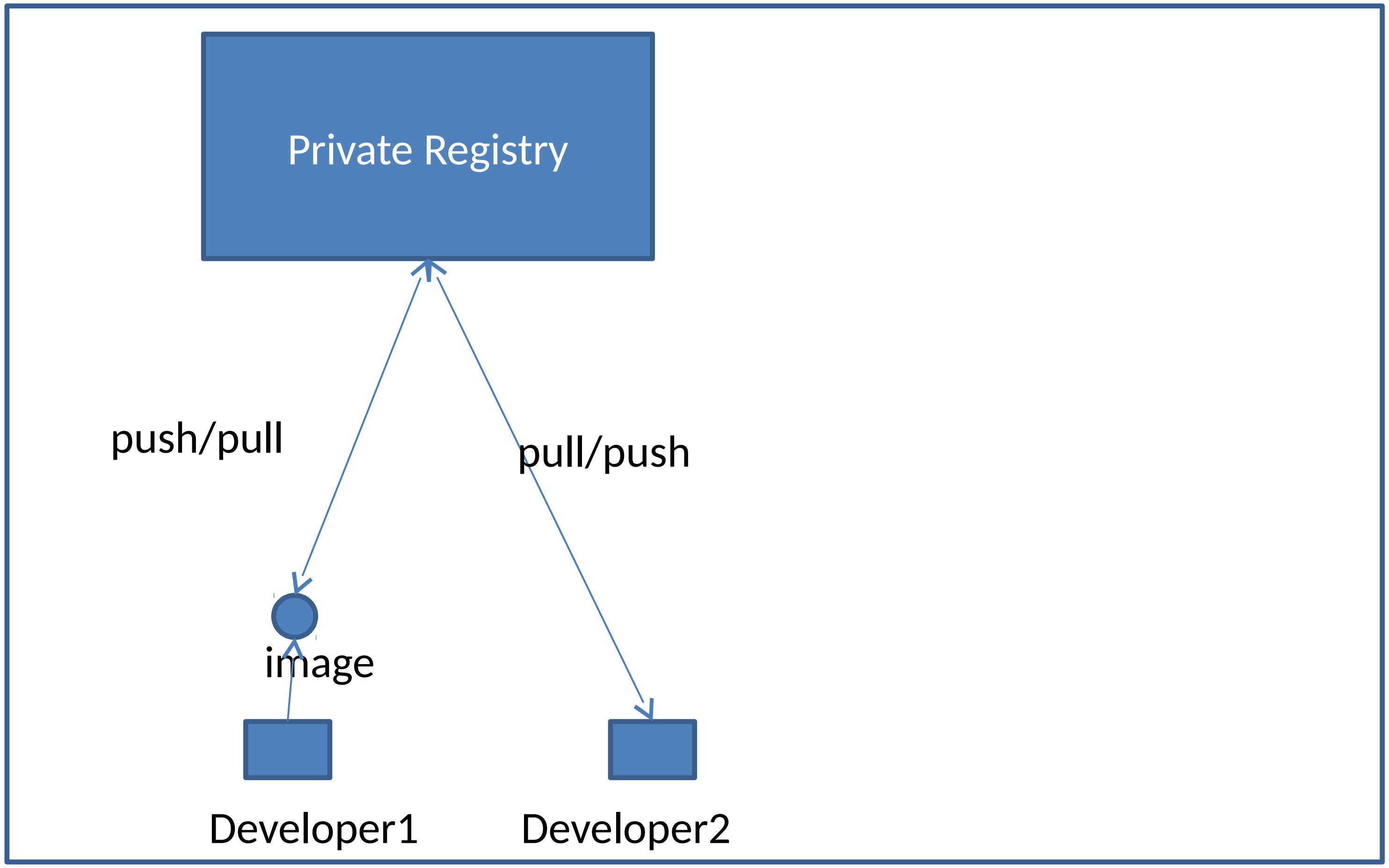
A simple real-world workflow

- Internal to the team
  - Build images and compose application
  - Upload builds to a *registry* [Docker Hub]
- External Deployment
  - Deploy the application to the cloud [Docker Cloud]

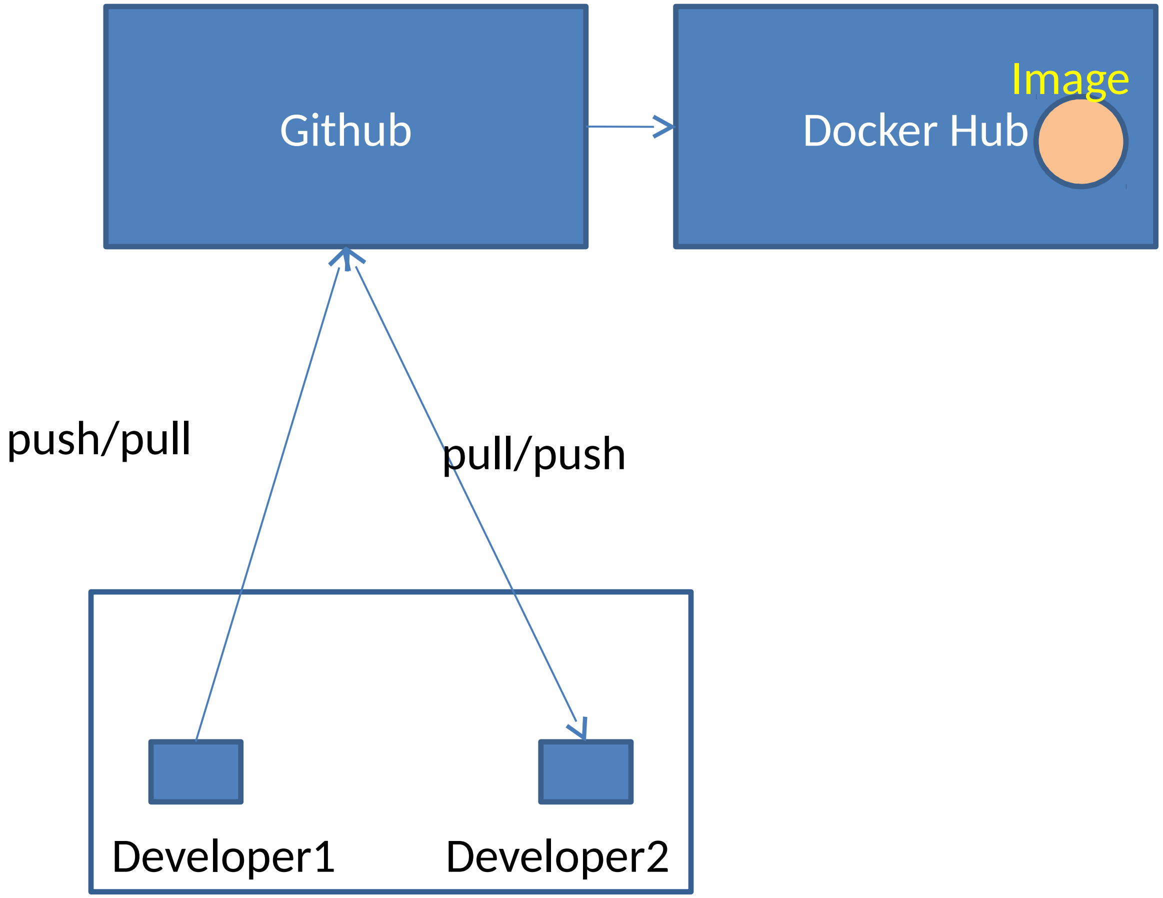
Development : Scenario 1



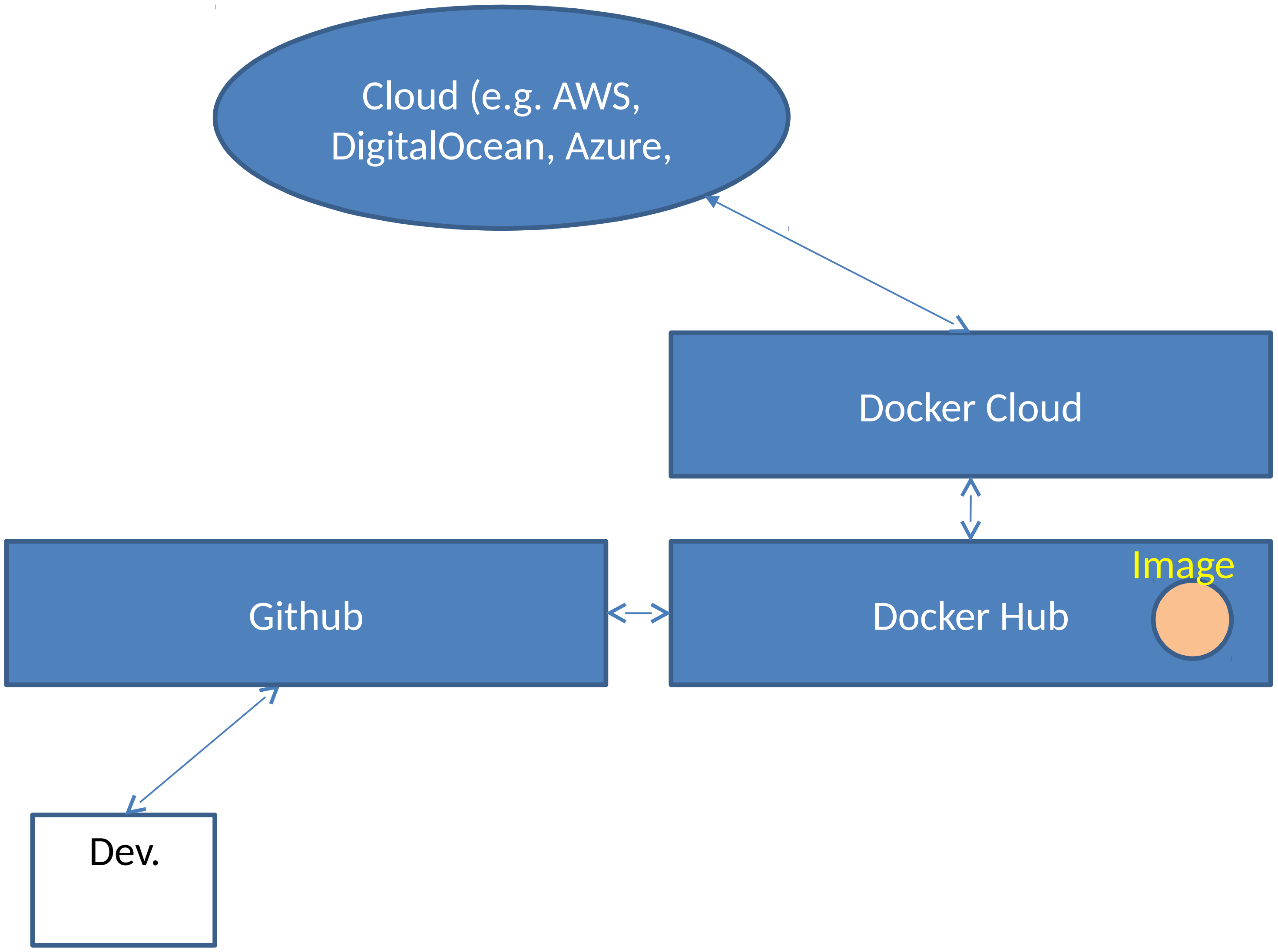
Development : Scenario 2



Development : Scenario 3



Deployment: Docker Cloud



Technologies which enable Docker

In the next lab you will

- Learn how to use docker
  - How to run and stop containers
  - How to commit changes to containers as images
- Learn how to handle file permissions for a multi-user Unix system
  - Let multiple users remotely access your docker box
- Serve a web page from your docker box