Jumpstarting Docker

Arup Nanda

VP, Data Services

Priceline.com

My application worked in Dev but not in QA

Will it work in production?

I need an environment right now

No, I can’t wait for 2 weeks

I just need it for a day to test my hypothesis. What do you mean I should have budgeted it?.

priceline®

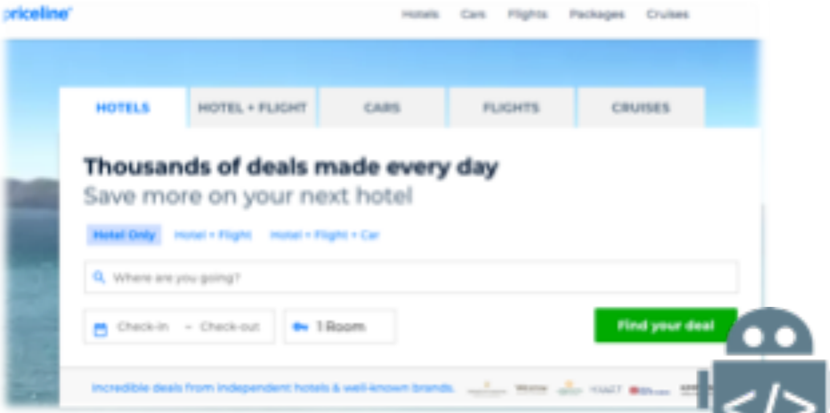
BOOKING HOLDINGS

booking.com|priceline.com|kayak.com|agoda.com|rentalcars.com|opentable.com

WWW

Affiliates

White labels



Thousands of deals made every day

Save more on your next hotel

Where are you going?

Incredible deals from independent hotels & well-known brands.

Bots

Providers

GDS

Affiliates

Demand *Technology* *Supply*

WWW

Affiliates

White labels

Inventory

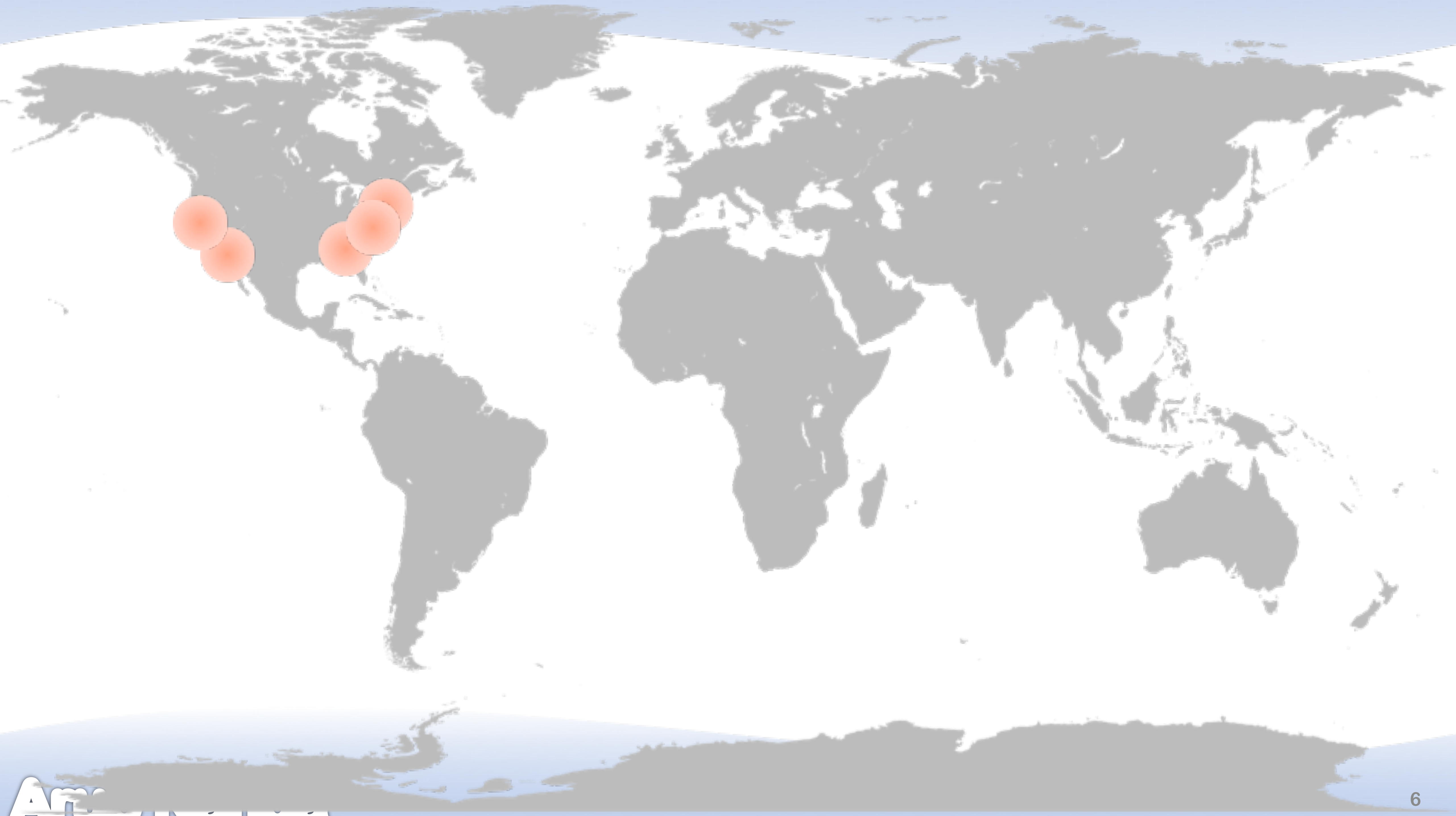
Providers

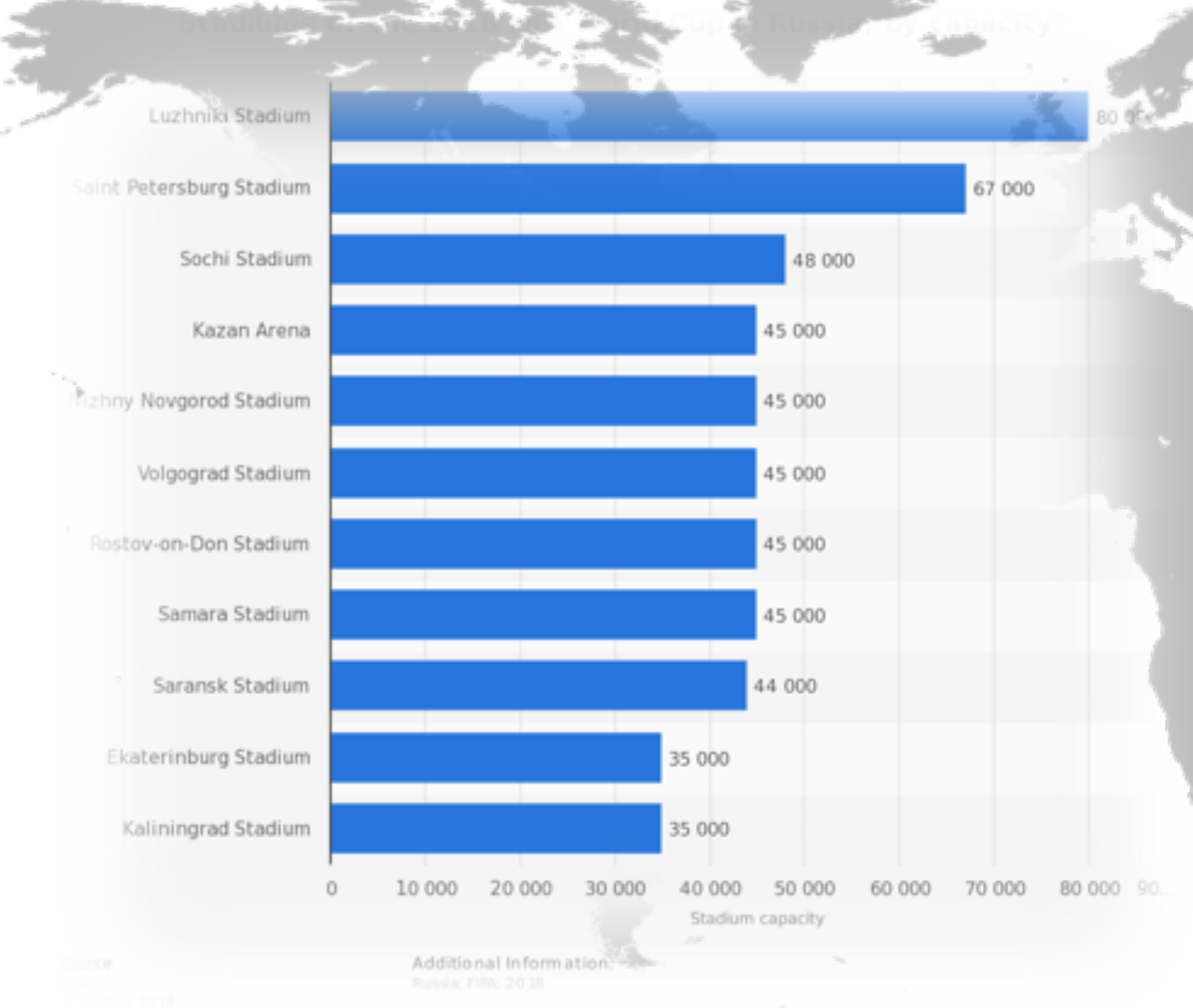
GDS

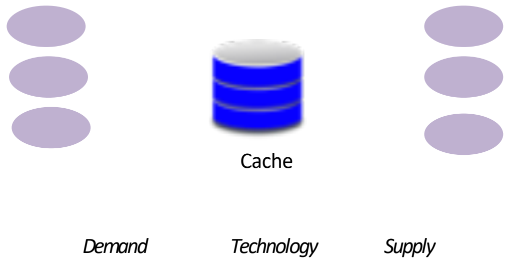
Affiliates

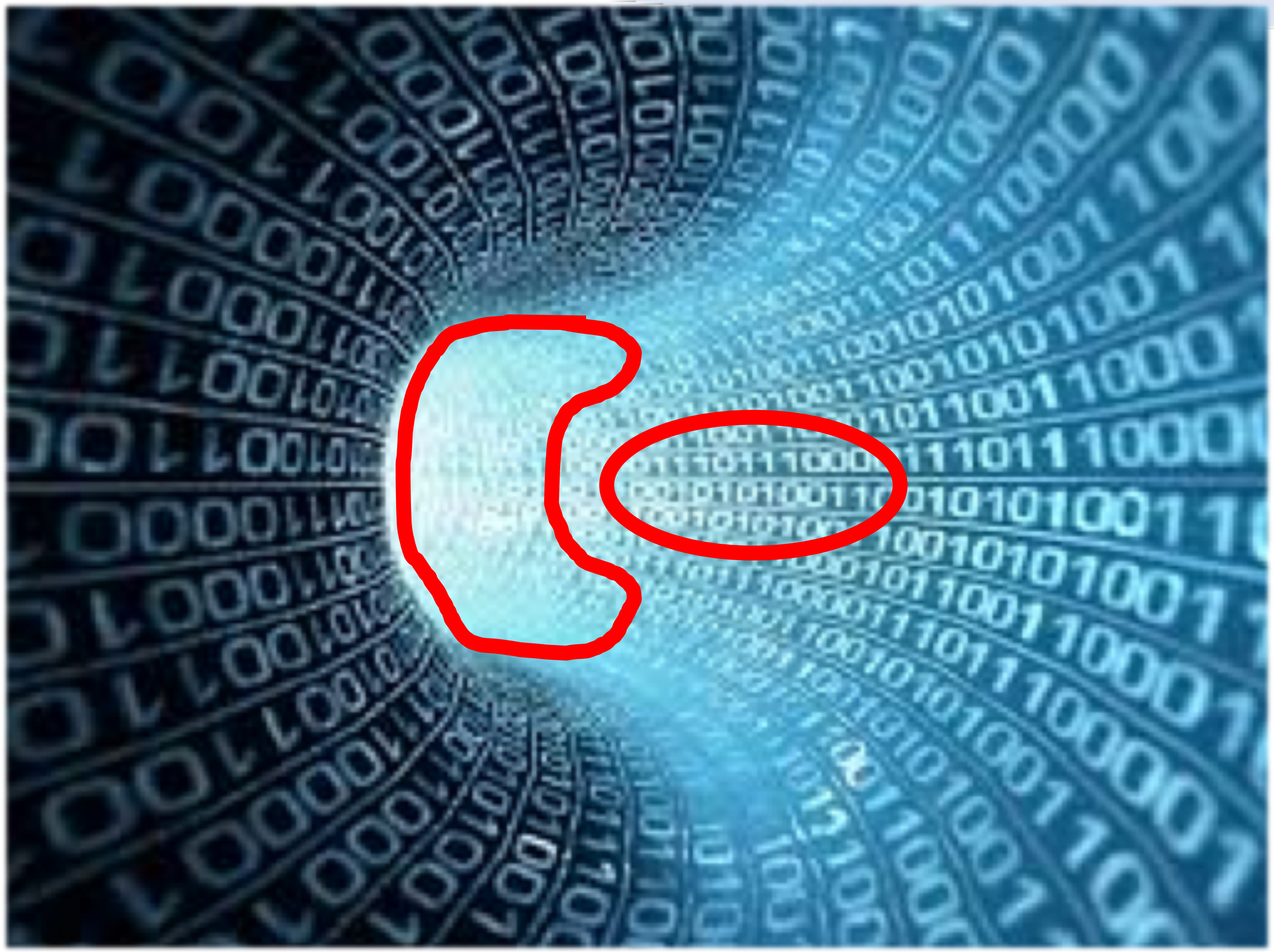
250K Hotels X 200 rooms X 365 days X ….

Demand *Technology* *Supply*

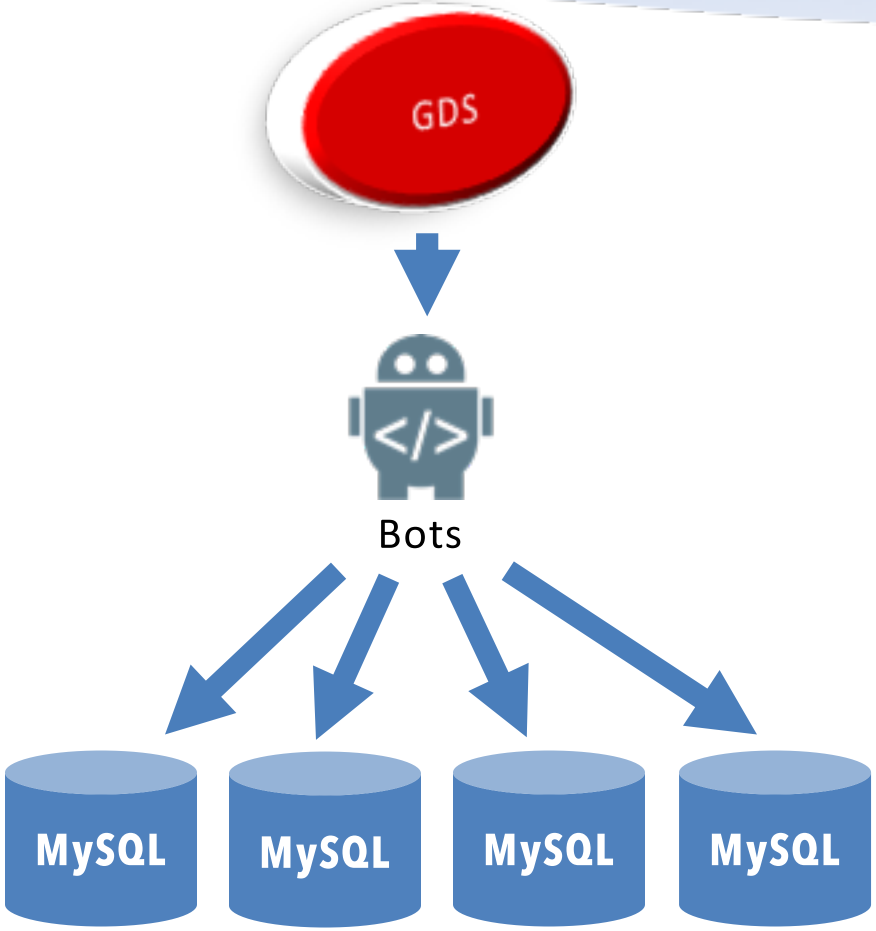



Stadiums of the 2018 FIFA World Cup in Russia, by capacity*
Luzhniki Stadium
80 000
Saint Petersburg Stadium
67 000
Sochi Stadium
48 000
Kazan Arena
45 000
Nizhny Novgorod Stadium
45 000
Volgograd Stadium
45 000
Rostov-on-Don Stadium
45 000
Samara Stadium
45 000
Saransk Stadium
44 000
Ekaterinburg Stadium
35 000
Kaliningrad Stadium
35 000
0
10 000
20 000
30 000
40 000
50 000
60 000
70 000
80 000
90...
Stadium capacity
Source
FIFA
Additional Information:
Russia; FIFA; 2018

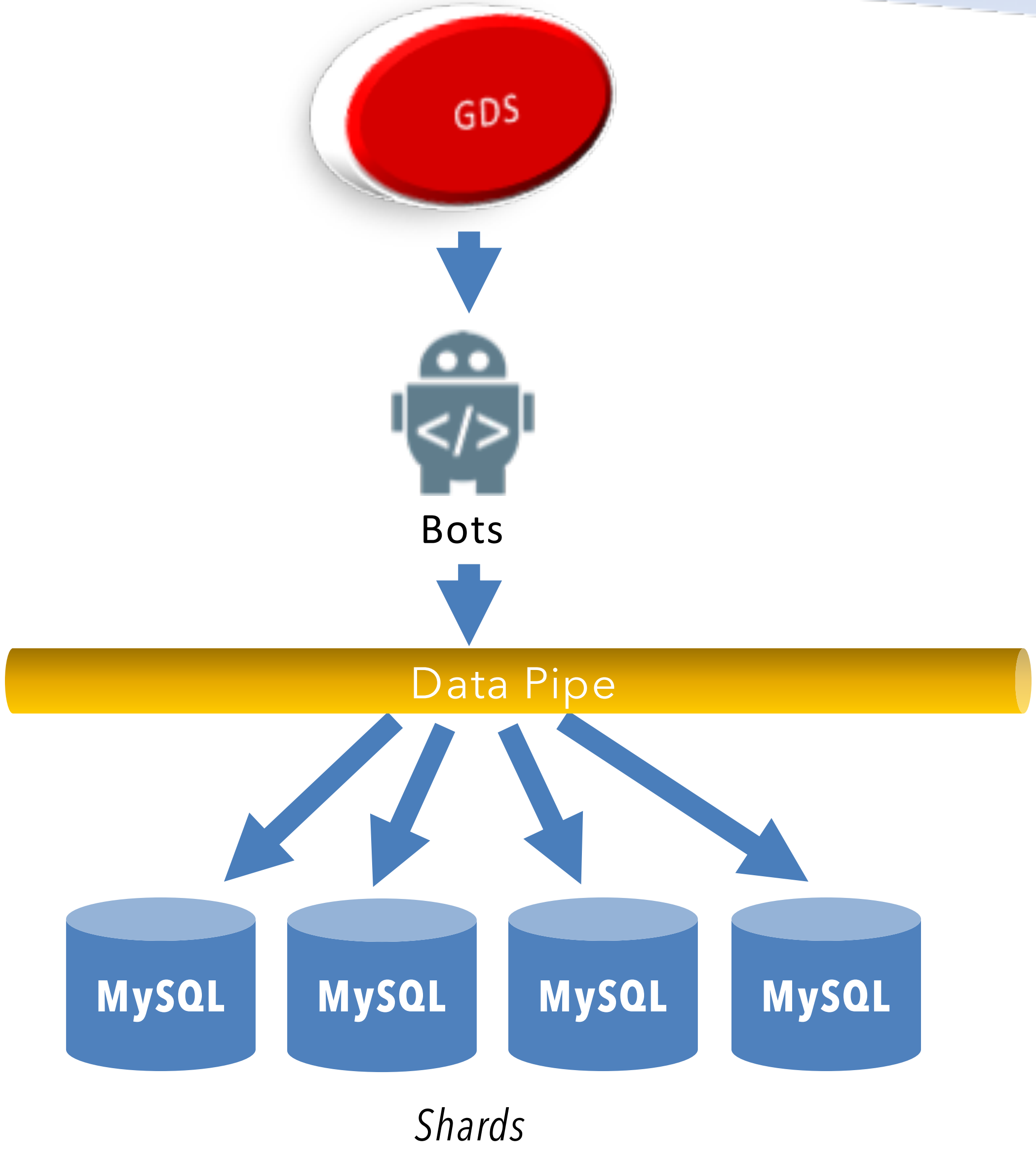




Smart Cache Management



Shards



based on Kafka

named *Missisipi*

has APIs and common libraries

Need:

Automatically spin up MySQL servers, with the right metadata!

Agenda

What’s the problem we are trying to solve?

How Docker solves the problem?

Installing and getting started with Docker

How is it different from Virtual Machines?

Docker in the Cloud—GCP, AWS, Oracle Cloud, etc.

Docker Compose, Swarm, etc..

Basic Command Structure

- The word docker followed by a verb and optionally arguments

```
$ docker ps
$ docker run mycontainer
```

- Help

```
$ docker help
$ docker ps help
```

Coloring Book

- Example of building a Coloring Book
- Docker Concepts
  - **Image**: a set of programs, or building blocks which can be deployed; but *not deployed* yet
  - **Container**: when you deploy an image. It’s the *executable*.

Pull Image

- Pull images from docker hub

```
$ docker pull ubuntu
```

- Pulls the image from the hub to the docker host
- Show the images we have on the docker host

```
$ docker images
```

Bring up a container

- Run a container

```
$ docker run ubuntu
```

- See which containers are running

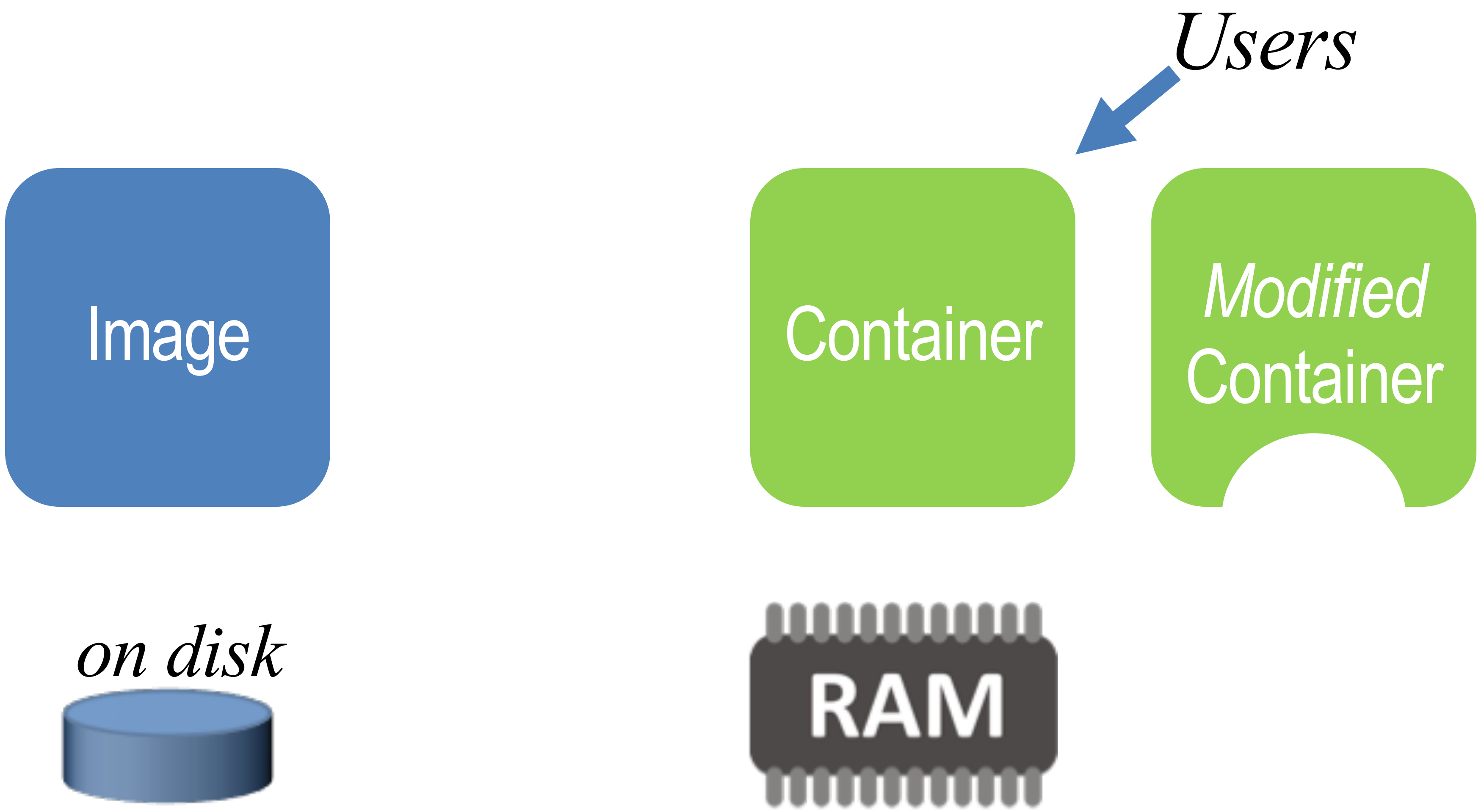
```
$ docker ps
```

- See which containers were running earlier; but not now

```
$ docker ps -a
```

- Containers do not stay up if they have nothing to do

Containers and Images



Users

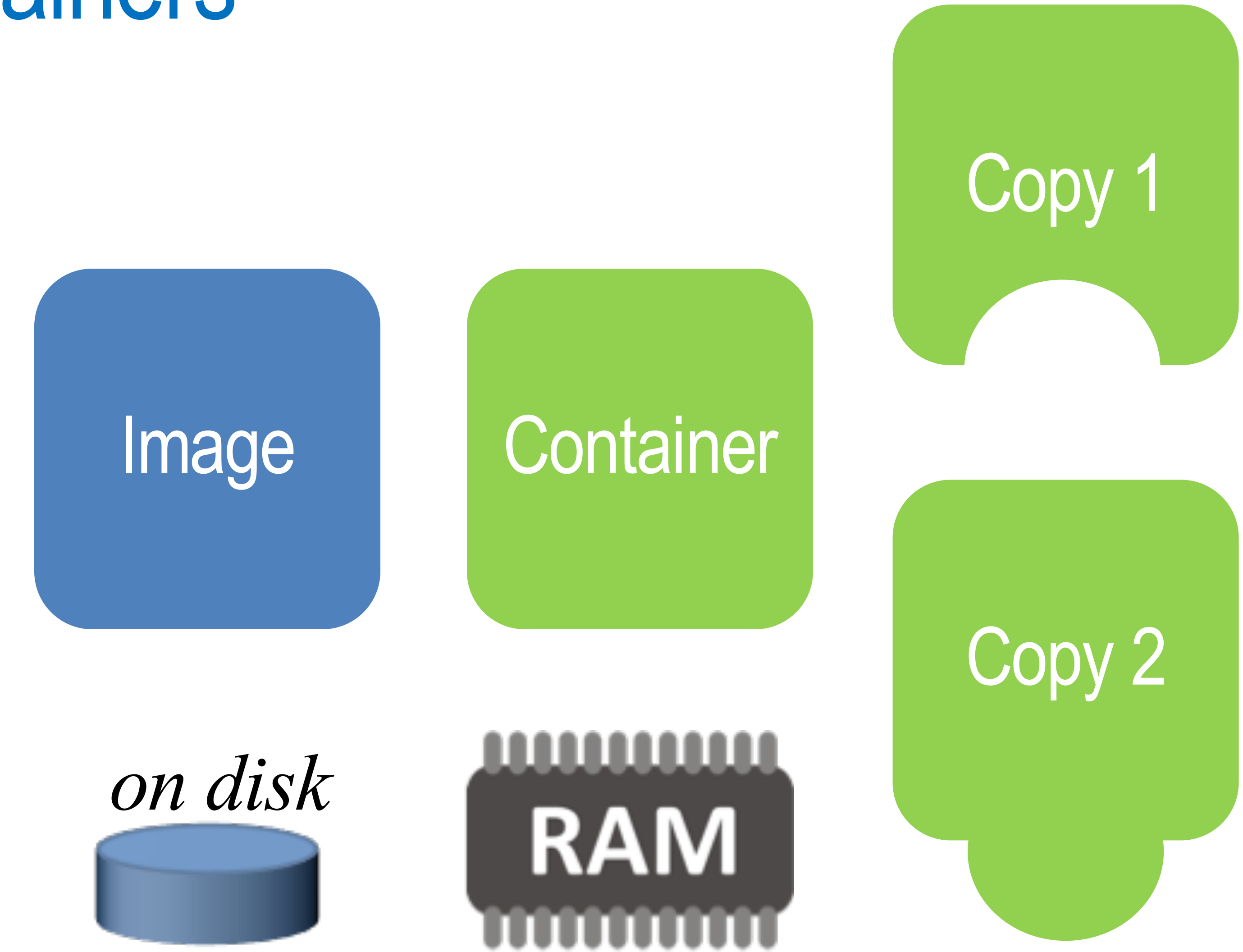
Image

Container

Modified Container

on disk

Containers



Image

Container

Copy 1

Copy 2

on disk

Running Apps in Containers

- Verb run

```
$ docker run --name myhost ubuntu sleep 100
```

- This will not let you come back
- From a different terminal

```
$ docker ps
```

- Stop the container
- `$ docker stop myhost`
- Remove the container

```
$ docker rm myhost
```

Detached mode

- Run the container in detached mode to run in background

```
docker run -d --name myhost ubuntu sleep 100
```

Execute commands

- Execute a command in a running container

```
$ docker exec -it myhost /bin/bash
```

- This gives the OS prompt of the container
- Options:
  - -i gives interactive prompt
  - -t gives TTY prompt (shell)

Execute in a container remotely

- From Docker host

```
$ docker exec ContainerName Program
```

- So to run `docker exec myhost cat /etc/*release*` in container myhost:

```
$ docker exec myhost cat /etc/*release*
```

- A more practical example:
- Shell script to compute bonus: `/usr/bin/bonus.sh`

```
$ docker exec myhost sh /usr/bin/bonus.sh 100
```

Persisting Changes

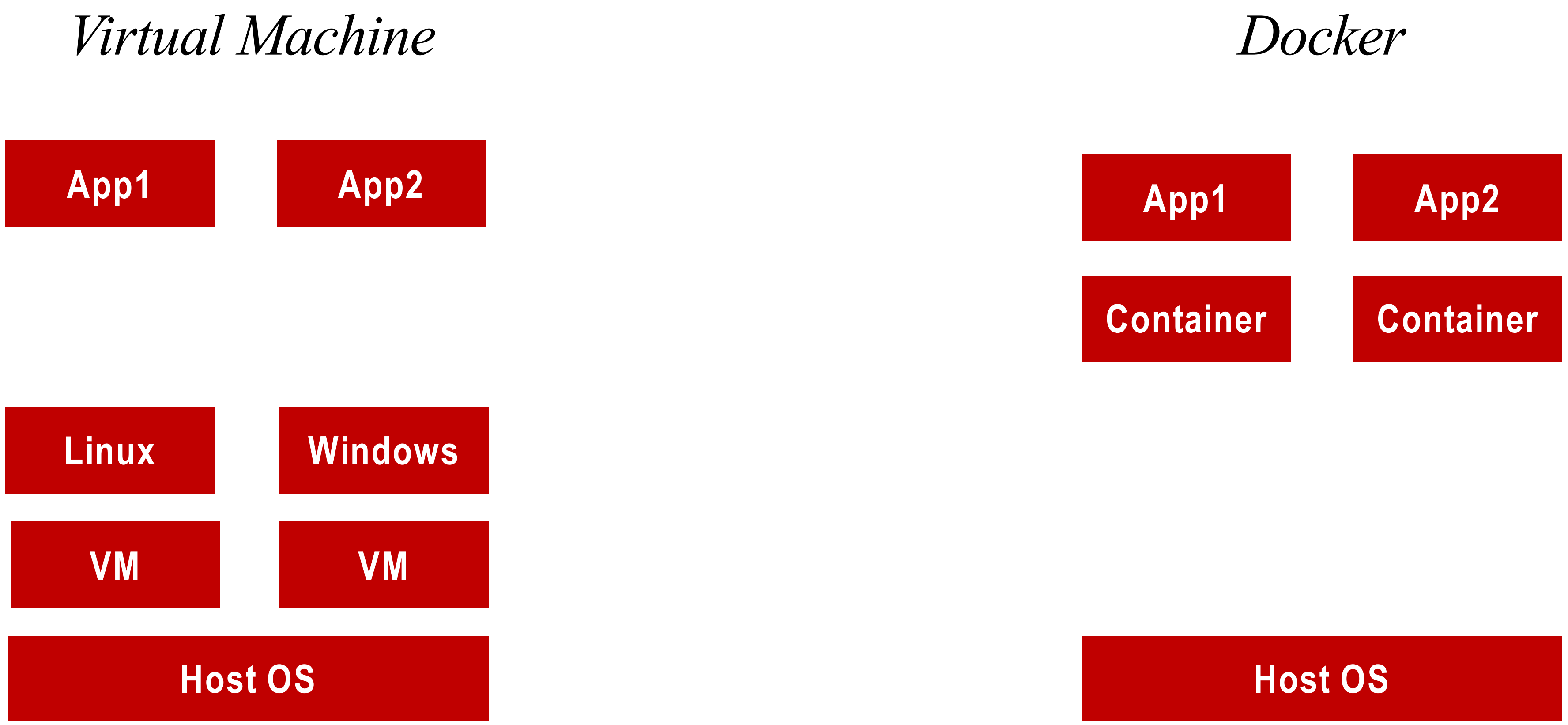
- “Commit” Changes

```
$ docker commit Container Repository
```

- Repository, not Image
- You may add a “tag”

```
$ docker commit  docker commit Container Repository:v1
```

Comparison with VMs



Virtual Machine

App1

App2

Linux

Windows

VM

VM

Host OS

Docker

App1

App2

Container

Container

Host OS

Volume

- Databases need persistent data
- Makes a filesystem persist in the host even after the container is shutdown
- On the docker host, create a directory /dockervol
- Add this to the container as a filesystem named /fs

```
$ docker run --name myhost -d -v /dockervol:/fs
myubuntu1:v1 sleep 1000
```

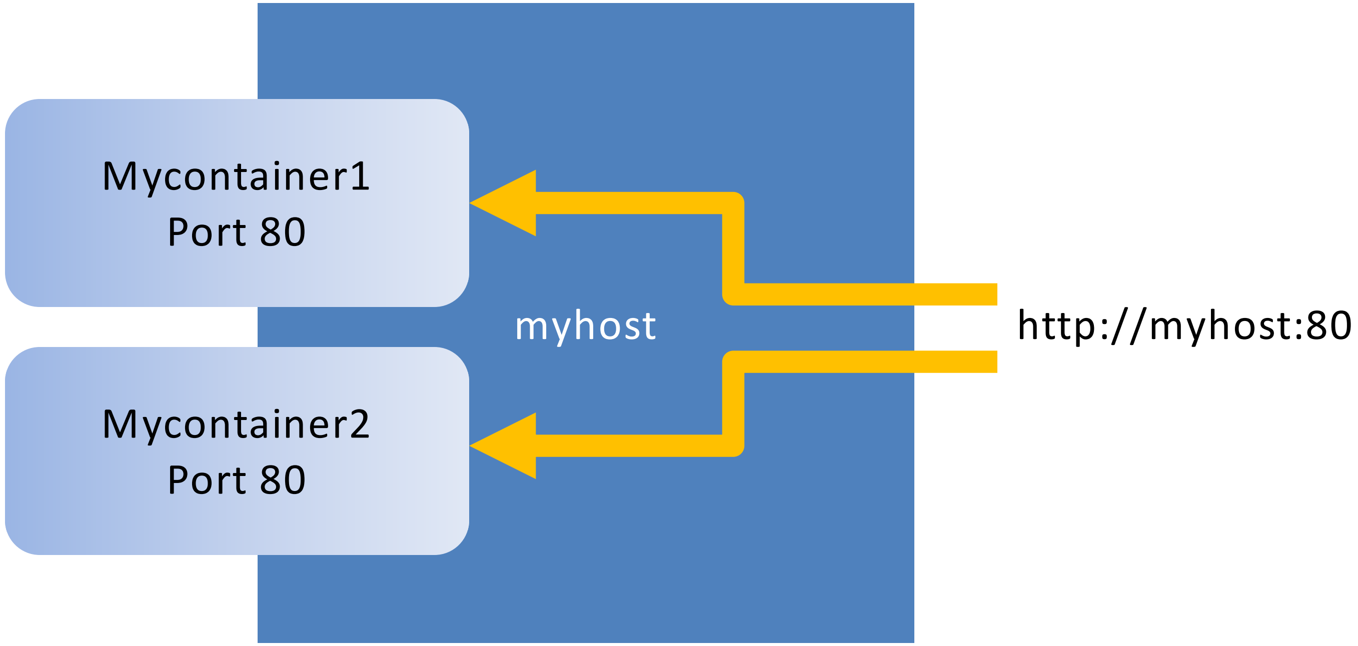
- This will create a filesystem on the container named /fs

Copying files to/from containers

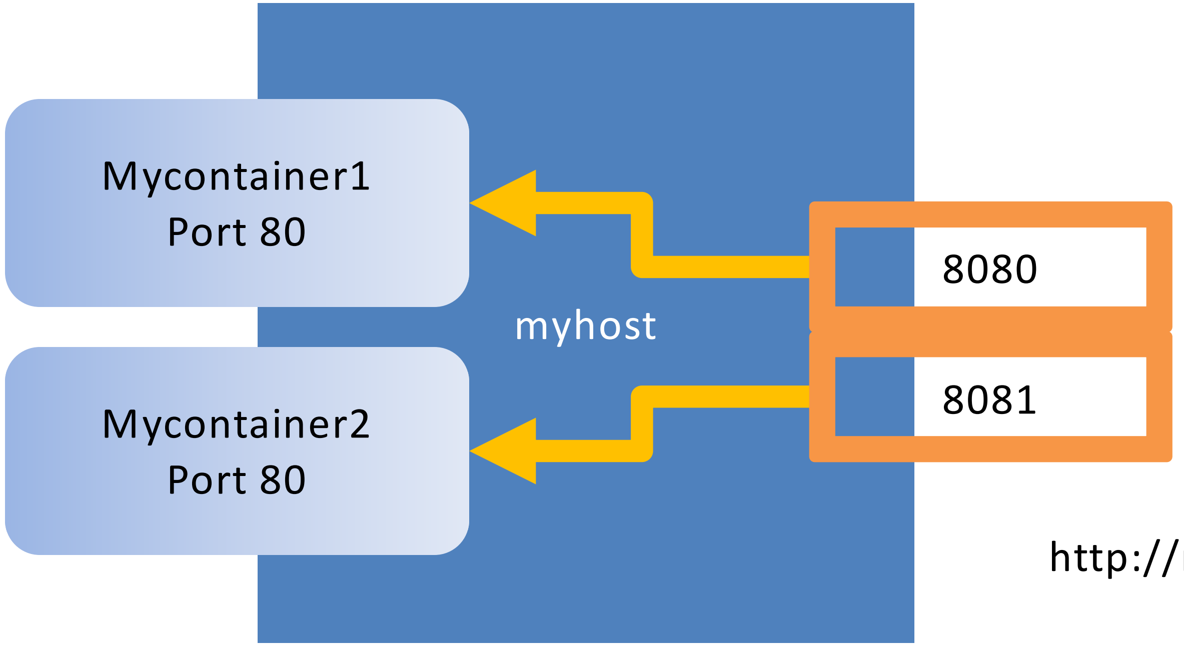
- Copying

```
$ docker cp myhost:/var/www/html/index.html .
```

Port Mapping



Port Mapping



http://myhost:8080

Port Mapping

- Start the container

```
$ docker run -d -p 8080:80 --name myhost ubuntu2 nginx -
g "daemon off;"
```

- Now you can kick off the webpage
  - http://192.168.56.101:8080/
  - where 192.168.56.101 is the address of the Dockerhost.

Dockerfile

- Create a file named `Dockerfile` in the directory /dockerfiles

```
FROM ubuntu:latest
RUN apt-get update
RUN apt-get install -y vim
RUN apt-get install -y nginx
RUN echo "My first ubuntu docker image for vim and nginx" >
/var/www/html/index.html
EXPOSE 80
```

- Create the image from the dockerfile

```
$ docker build -t ubuntu2 /dockerfiles
```

Changes in the file

- To find out the changes in the files since the container was created,

```
$  docker diff myhost
```

- Sample output:

```
$ docker diff myhost
C /root
C /root/.bash_history
C /root/.viminfo
C /tmp
A /tmp/a
C /usr
C /usr/bin
C /usr/bin/bonus.sh
```

Inspect Containers

- Get detailed inner settings of a container

```
$ docker inspect myhost2
```

Shows all the important details

Get Usage

- Stats

```
$ docker stats myhost2
```

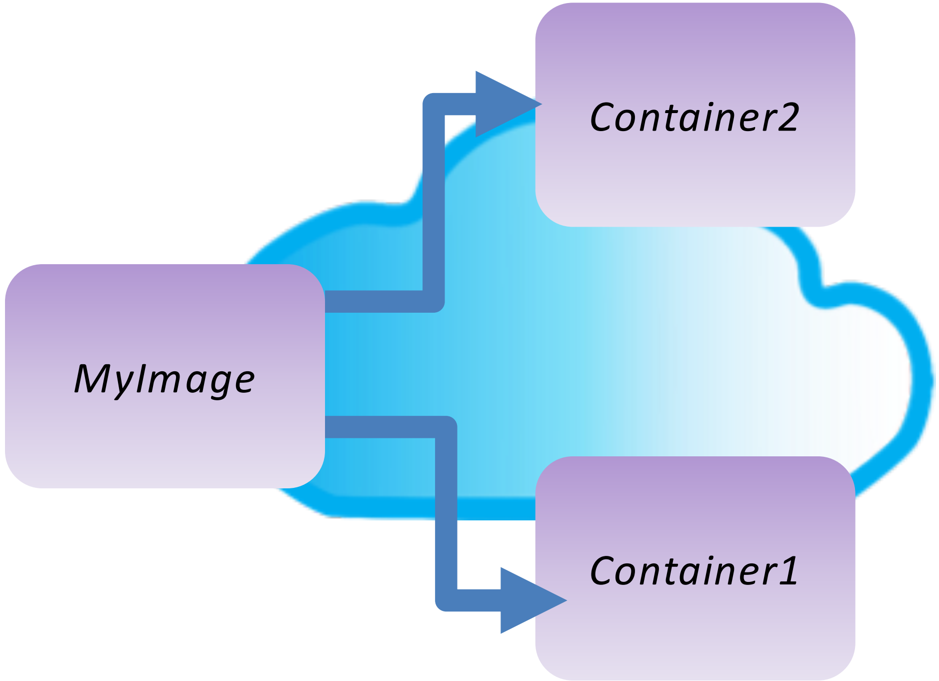
Compose

- Deploy multiple containers on multiple hosts
- Uses a YAML file to define the various components

Swarm

- Bring up multiple containers
- A cluster
- Automatically bring up containers
- Automatically restart after failures

Container Clouds



Amazon Container Cloud
Oracle Container Cloud Service

Summary

- Docker is just a set of processes running inside a host OS; it doesn’t have an OS
- This makes it very quick to bring up and deploy
- Takes very little memory and CPU
- Also makes it less secure and isolated
- Image: the executable programs and processes to be deployed
- Container: the running processes
- Docker Hub: the storage for all docker images
- Dockerfile: a file that instructs what to do when building containers
- Compose: a method to automate deployment of docker containers
- Swarm: the automated way to manage a network of containers

Thank You!

Blog: arup.blogspot.com
Tweeter: @ArupNanda
Facebook.com/ArupKNanda