



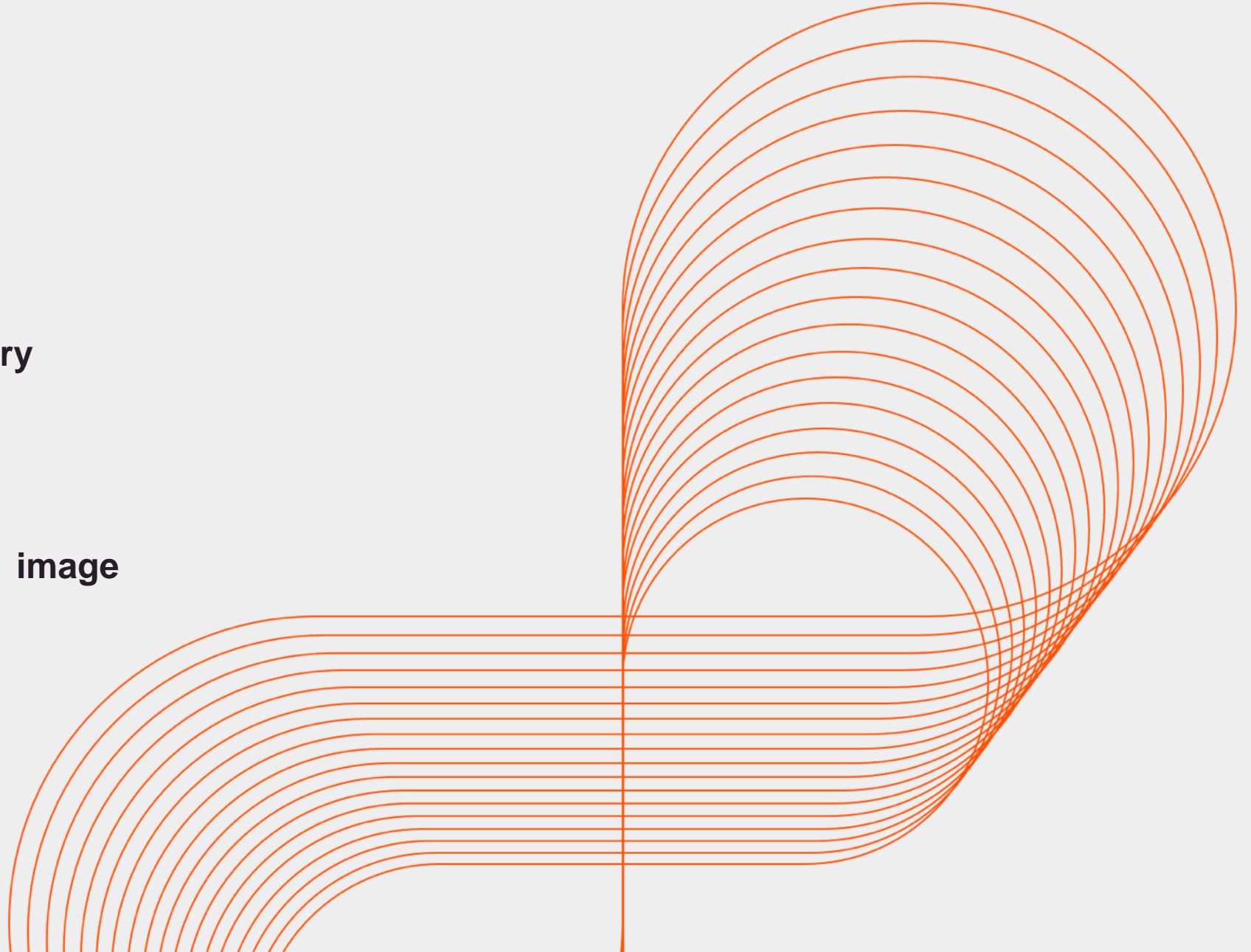
Persistent

Docker Images and registry

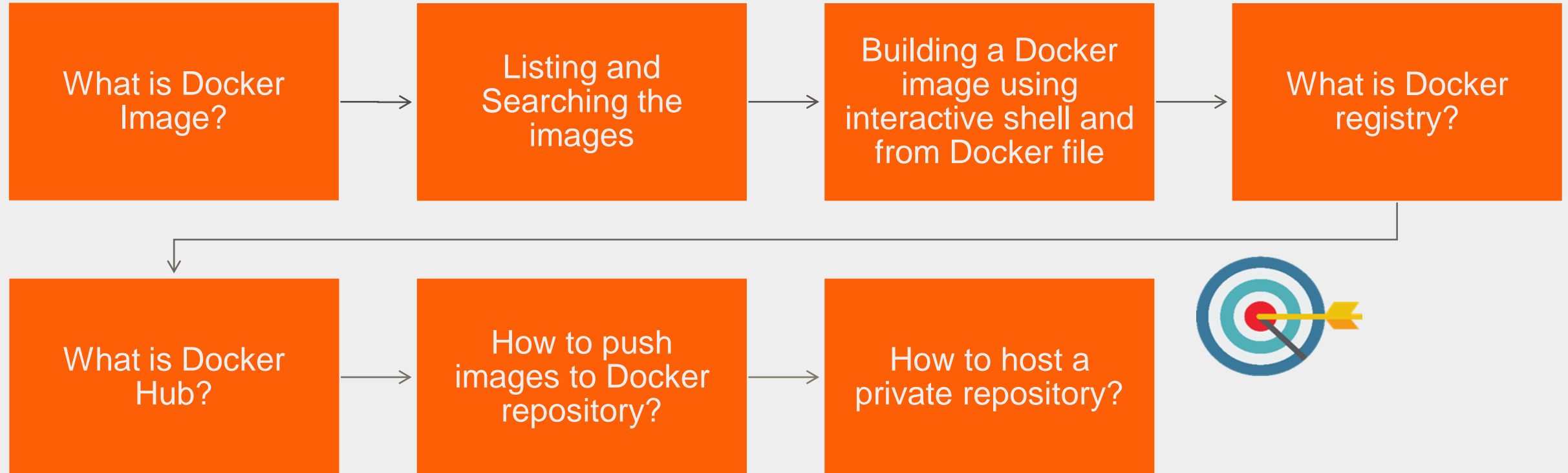


docker

image



Objectives



Objectives

At the end of this module, you will be able to Learn

- What is Docker Image?
- Listing and Searching the images
- Building a Docker image using interactive shell and from Docker file
- What is Docker registry?
- What is Docker Hub?
- How to push images to Docker repository?
- How to host a private repository?

What is Docker image?

Docker image is a read only template, used to create containers.

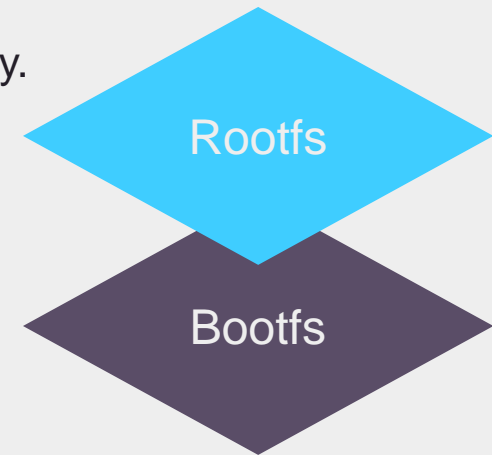
A Docker image is made up of filesystems layered over each other.

At the base is a boot filesystem, bootfs, which resembles the typical Linux/Unix boot filesystem.

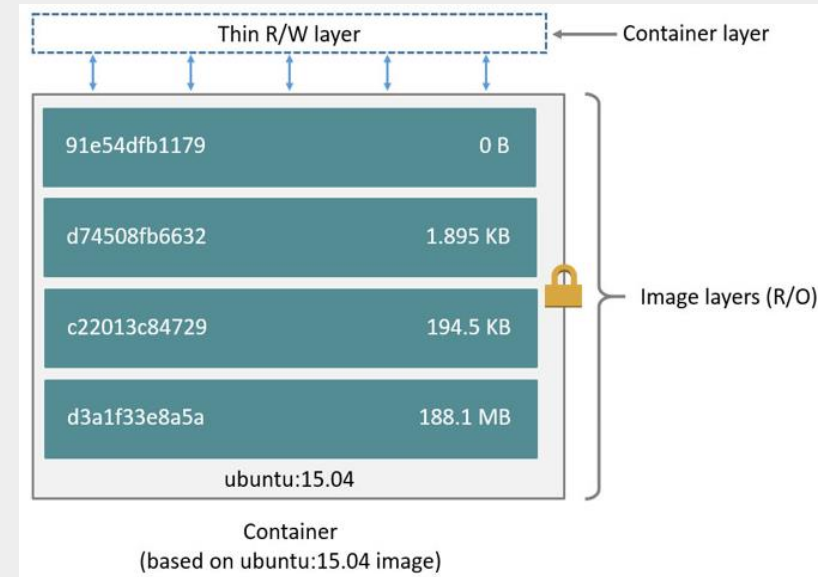
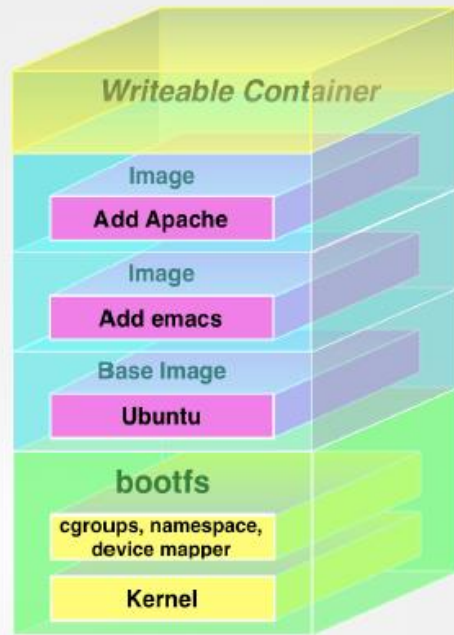
Docker next layers a root filesystem, rootfs, on top of the boot filesystem.

The rootfs can be one or more operating systems (e.g., a Debian or Ubuntu filesystem, the root filesystem stays in read-only mode

Docker images are stored in Docker Hub (i.e. public image repository) or your local repository.



Docker Images



Docker images and Containers

- When a new container is created, a thin writable layer is added on top of underlying stack.
- This layer is called container layer.
- All changes made to the running container writing new files, modifying existing files, and deleting files are written to this thin writable container layer.



Listing Docker images

- The below command lists the Docker images currently available on the host.
- `$ sudo docker images`

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
ubuntu	latest	2fa927b5cdd3	2 weeks ago	122 MB
hello-world	latest	94df4f0ce8a4	6 weeks ago	967 B

- These local images live on our local Docker host in `/var/lib/docker` directory


Docker repositories

- Images live inside repositories, repositories live on the registries.
- The default registry is the public registry managed by Docker – Docker Hub
 - <https://hub.docker.com/>
- You will need to create your own free Docker ID to get access to Docker hub.
- As the Docker registry code is open source, you can run your own registries.



Docker repositories

- Two types of repositories - User repositories (images contributed by users) and top level repositories controlled by Docker.



Dashboard

Explore

Organizations




Q Search

Create



dattatra

Explore Official Repositories

<div></div> <div><div>nginx</div><div>official</div></div>	<div>3.2K</div> <div>STARS</div>	<div>10M+</div> <div>PULLS</div>
<div></div> <div><div>busybox</div><div>official</div></div>	<div>695</div> <div>STARS</div>	<div>10M+</div> <div>PULLS</div>
<div></div> <div><div>ubuntu</div><div>official</div></div>	<div>4.1K</div> <div>STARS</div>	<div>10M+</div> <div>PULLS</div>

Pulling Docker images

- To run a container from the images using docker run command, which downloads the images.
- Other option is to run docker pull command, this saves some time in launching the container from the images.

```
$sudo docker pull Ubuntu
```

```
$sudo docker images ubuntu
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
ubuntu	latest	2fa927b5cdd3	2 weeks ago	122 MB

- Searching the images

```
$ sudo docker search mysql
```

Docker login

- First step is to create a Docker account and then login from the command line.
 - `$sudo docker login`
 - Username: <yourusername>
 - Password:
 - Login Succeeded
- Let's try creating a new Docker image using the base Ubuntu image.

Creating Docker images using Docker commit

- Let's use the Ubuntu container.

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

- Let's install Apache into the container

```
$ apt-get -yqq update
```

```
$ apt-get -y install apache2
```

- We have launched a container and installed Apache in it.
- We will save this container in the current state, first exit from the container.

```
$ sudo docker ps -l -q
```

```
$ sudo docker commit e8e46aae3660 dattatrayhkulkarni/apache2
```

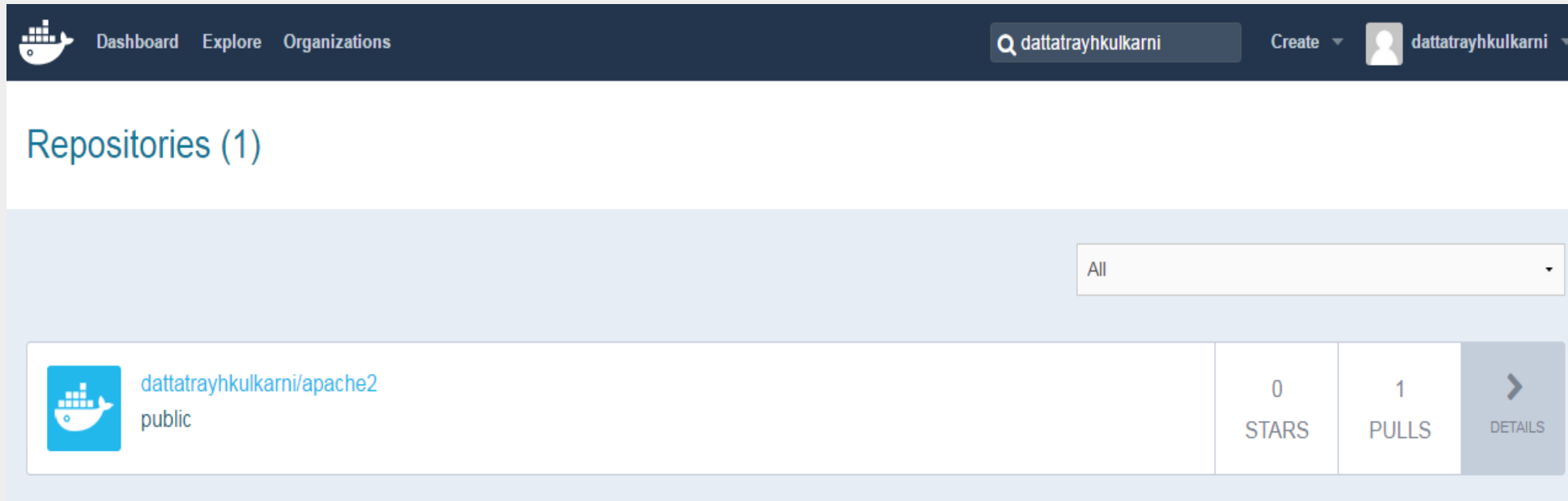
- Output - sha256:d1e510e2c83d1db58409cd01bd44f88cea0f22623a9282a720eef2cef15ee07e

Pushing Docker images to repository

- You can push your images to your repository on the Docker Hub.

```
$ sudo docker push dattatrayhkulkarni/apache2
```

- After pushing the image, it will be available on the public repository on Docker hub.
- You can Search the image based on your user name

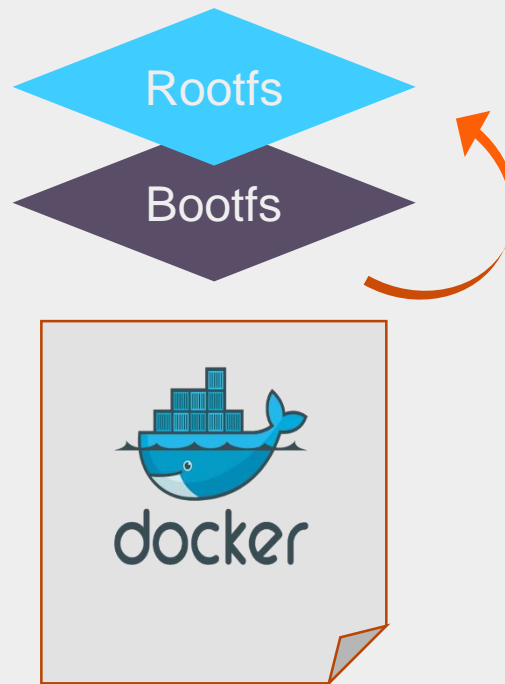


The screenshot shows the Docker Hub interface for the user 'dattatrayhkulkarni'. The top navigation bar includes 'Dashboard', 'Explore', and 'Organizations'. A search bar contains the username 'dattatrayhkulkarni'. The main heading is 'Repositories (1)'. Below this is a filter dropdown set to 'All'. A table lists the repository 'dattatrayhkulkarni/apache2' as 'public'. It shows '0 STARS' and '1 PULLS'. A 'DETAILS' button with a right arrow is next to the pull count.

Repository	Stars	Pulls	Action
dattatrayhkulkarni/apache2 public	0	1	DETAILS

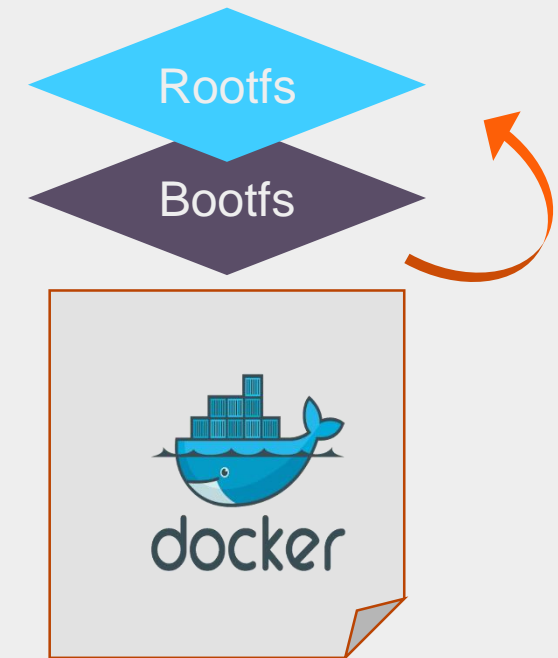
Creating Docker images from the Dockerfile

- While creating an image we will install application and all the necessary libraries into the image
- When you spin up the container with this image, everything required for your application to run is included in the container.



Creating Docker images from the Dockerfile

- Another way to create docker images is using Docker files.
- A Dockerfile is a configuration file that contains instructions for building a docker image.
- This is a more effective way of creating an image than using commit.
- Dockerfile can be used along with continuous integration and deployment process.
- Here is a link to best practices for creating a dockerfile:
- https://docs.docker.com/engine/userguide/eng-image/dockerfile_best-practices/



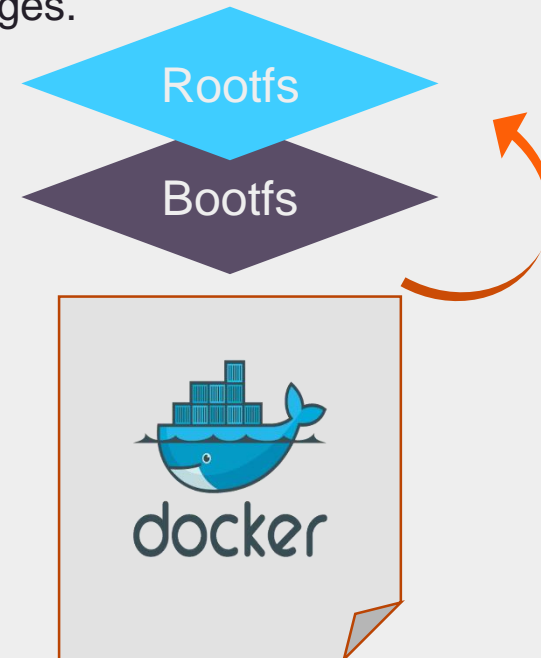
Creating Docker images from the Dockerfile

- Another way to create docker images is using Docker files.
- Dockerfile uses basic DSL with instructions for building docker images.
- Let's try to create a new docker file for a simple static web server.

```
$ mkdir static_web
```

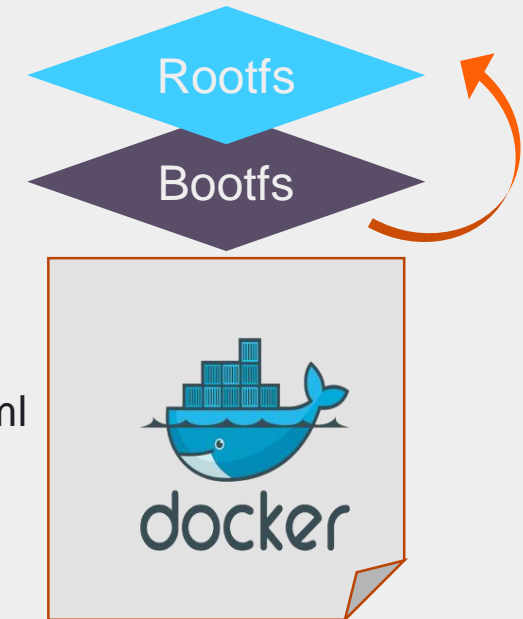
```
$ cd static_web/
```

```
$ touch Dockerfile
```



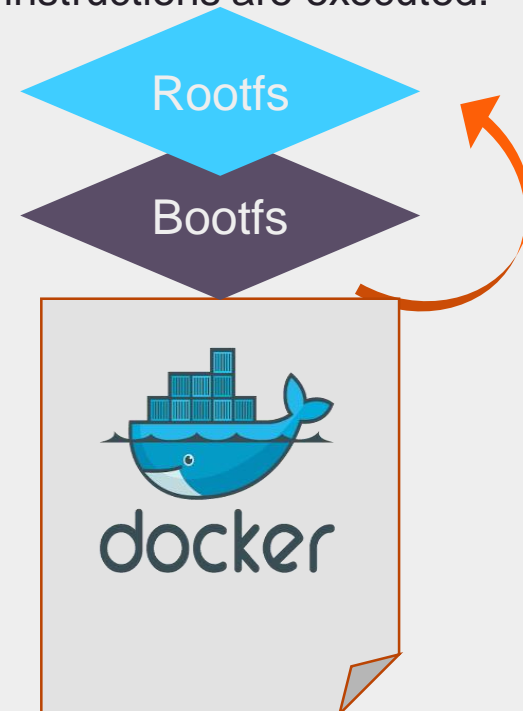
Creating Docker images from the Dockerfile

- # Version: 0.0.1
- FROM ubuntu:14.04
- MAINTAINER Dattatray Kulkarni "dattatrayhkulkarni@gmail.com"
- RUN apt-get update
- RUN apt-get install -y nginx
- RUN echo 'Hi I am in your new container for Nginx' \> /usr/share/nginx/html/index.html
- EXPOSE 80



Creating Docker images from the Dockerfile

- Docker runs a container from the image.
- An instruction executes and makes changes to the container.
- Docker runs the equivalent of `docker commit` to commit a new layer.
- Docker then runs a new container from this new image.
- The next instruction in the file is executed and the process repeats until all the instructions are executed.
- `$ sudo docker build -t="dattatrayhkulkarni/static_web" .`
- `$ sudo docker images`
- `$ sudo docker run -d -p 80 --name static_web dattatrayhkulkarni/static_web`
- `$ sudo docker ps -l`



Running your own Docker registry

- Here are the two possible options for running your own Docker registry –
- Make use of private repositories on the Docker Hub.
- Run your own registry behind the firewall.
- You can run the registry from the Docker container also.

```
sudo docker run -p 5000:5000 registry
```



Reference Material : Websites & Blogs

- <https://www.docker.com/>
- <https://training.docker.com/self-paced-training>
- <https://www.youtube.com/watch?v=Q5POuMHxW-0>
- Docker up and Running by Karl Matthias and Sean kane

Key Contacts

Docker Interactive

Dattatray Kulkarni

dattatray_kulkarni@persistent.co.in

Asif Immanad

asif_immanad@persistent.co.in

Vice President

Shubhangi Kelkar

shubhangi_kelkar@persistent.co.in



Persistent

Thank you!

Persistent University

