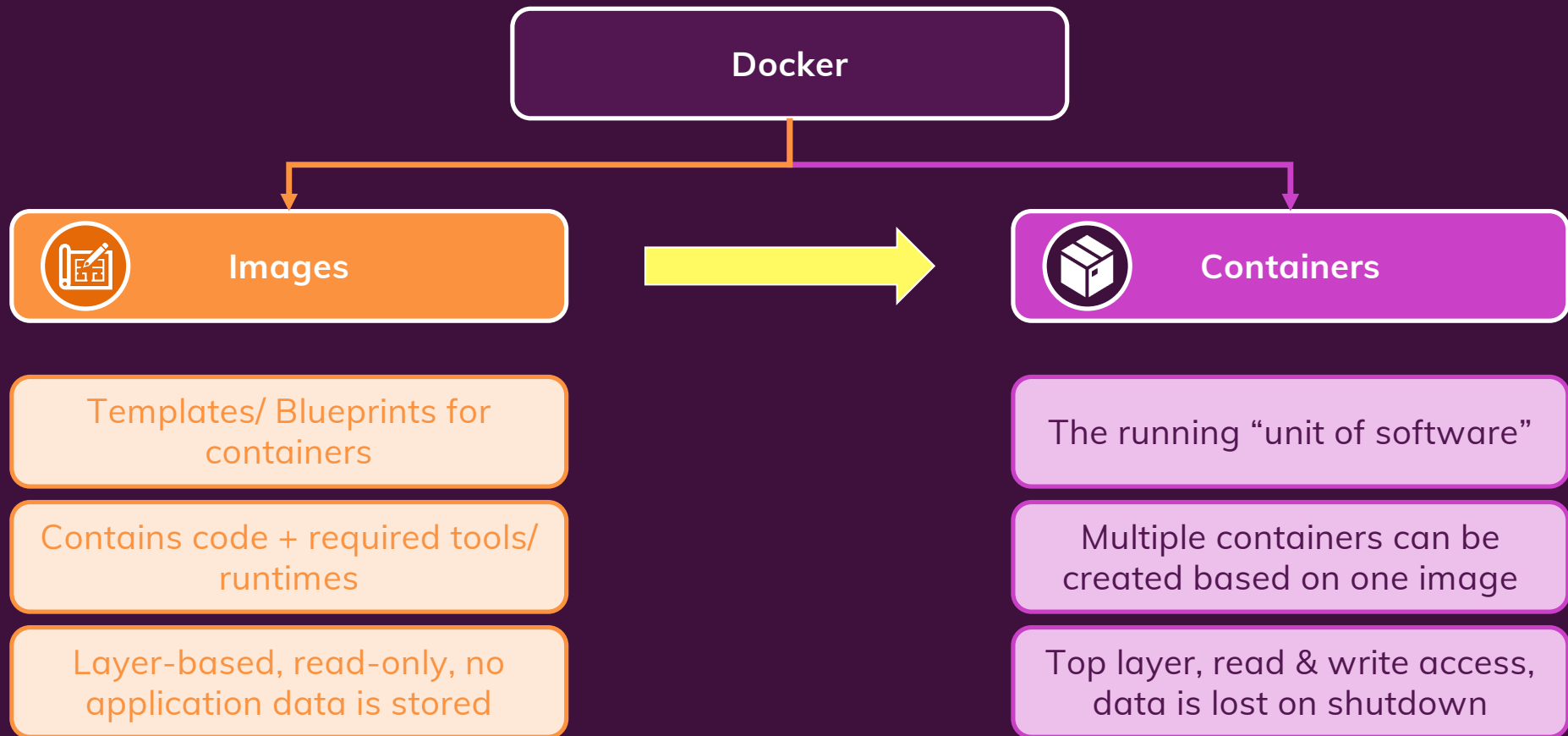
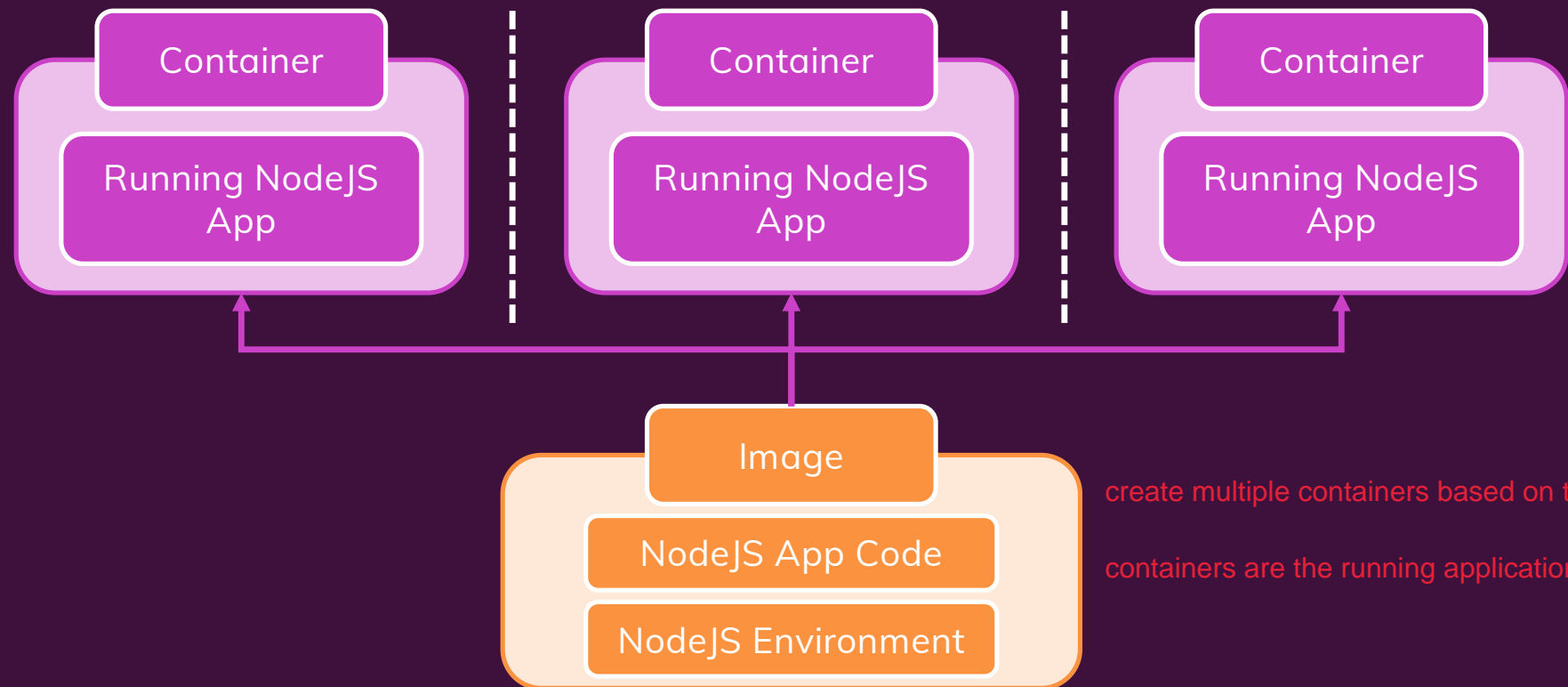


# Images vs Containers



small packages that contains website, server, node modules...

## One Image, Multiple Containers



create multiple containers based on that image.

containers are the running applications

## Finding / Creating Images

Containers are based on images.

We need an Image!

Use an existing, pre-built  
Image

community image  
e.g. via Docker Hub

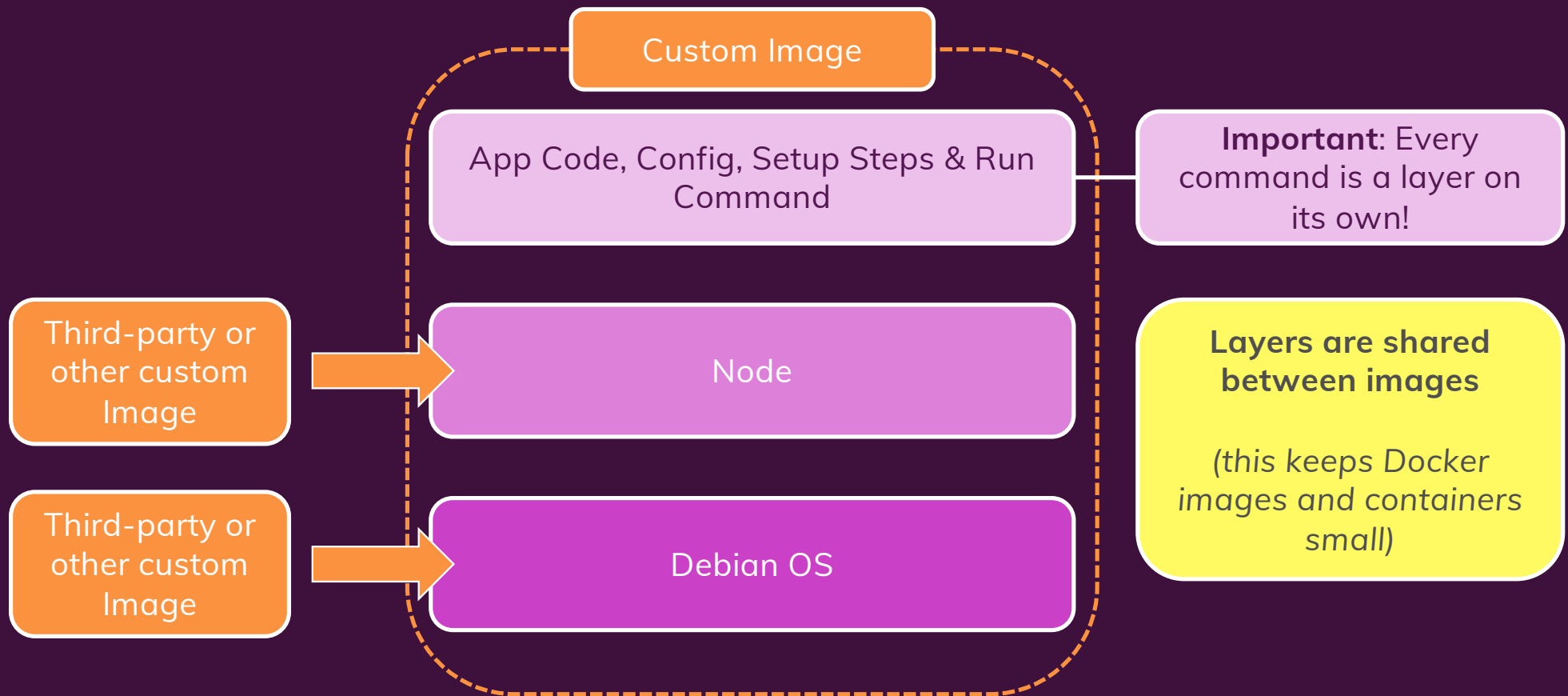
for e.g., colleague already built this image

Create your own, custom  
Image

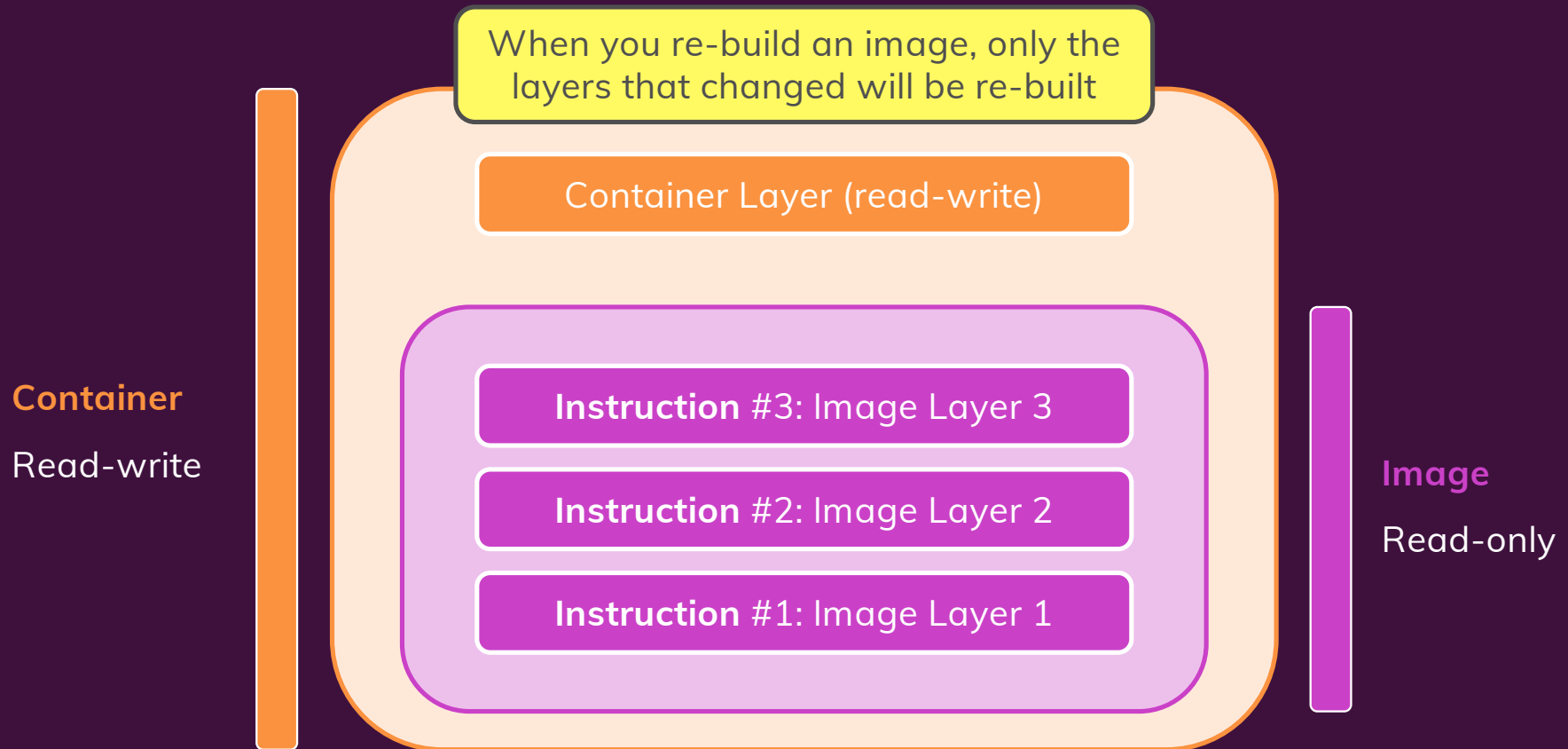
Write your own Dockerfile  
(based on another Image)

building our own image based on the node image.

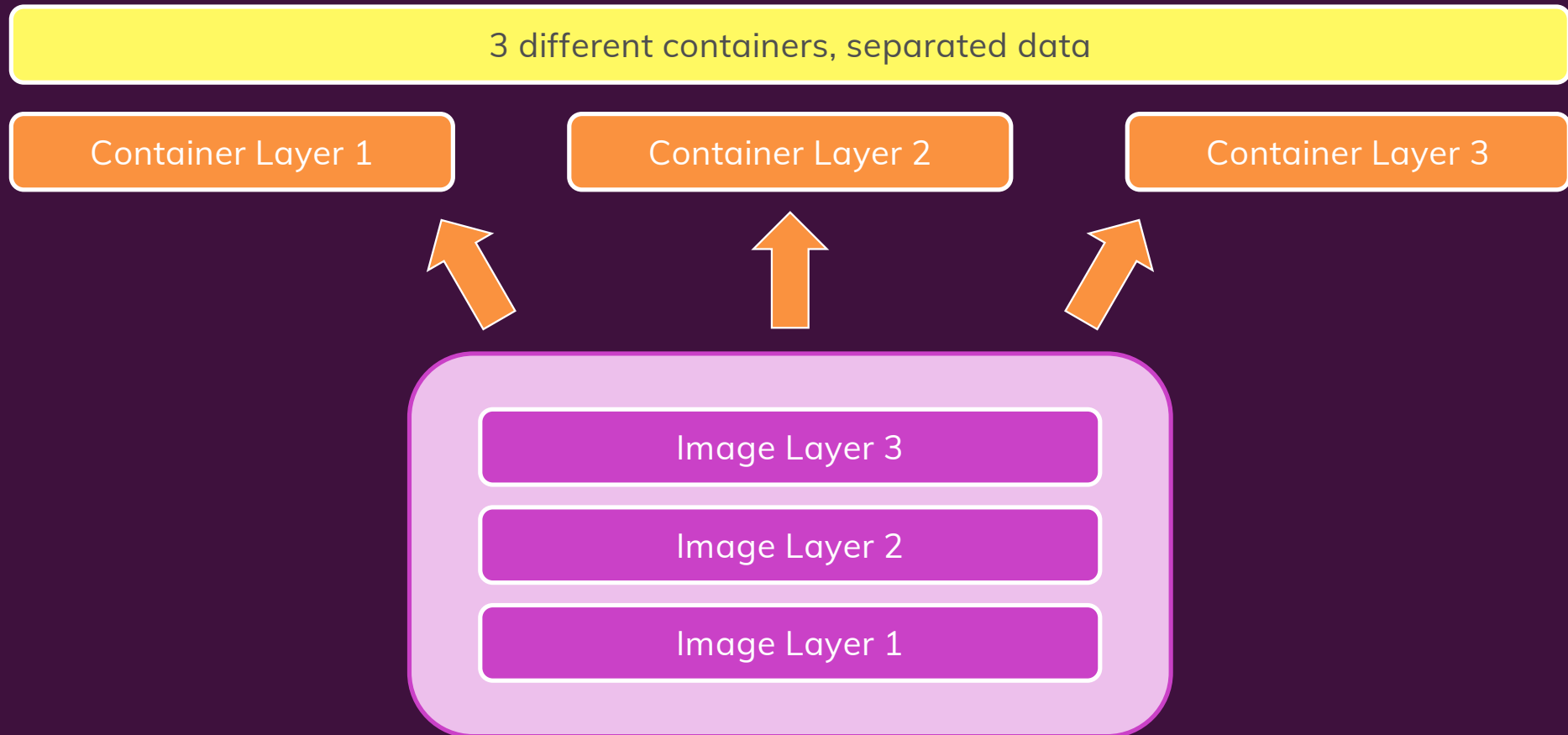
## Images & Image Layers



## A Container Is Based On An Image



## Multiple Containers Can Be Based On The Same Image



## Where To Get Images

Docker Hub

or

Build your Own

Cloud registry for third-party  
images

You can pull them to run as  
container

You can build your own images  
based on other images



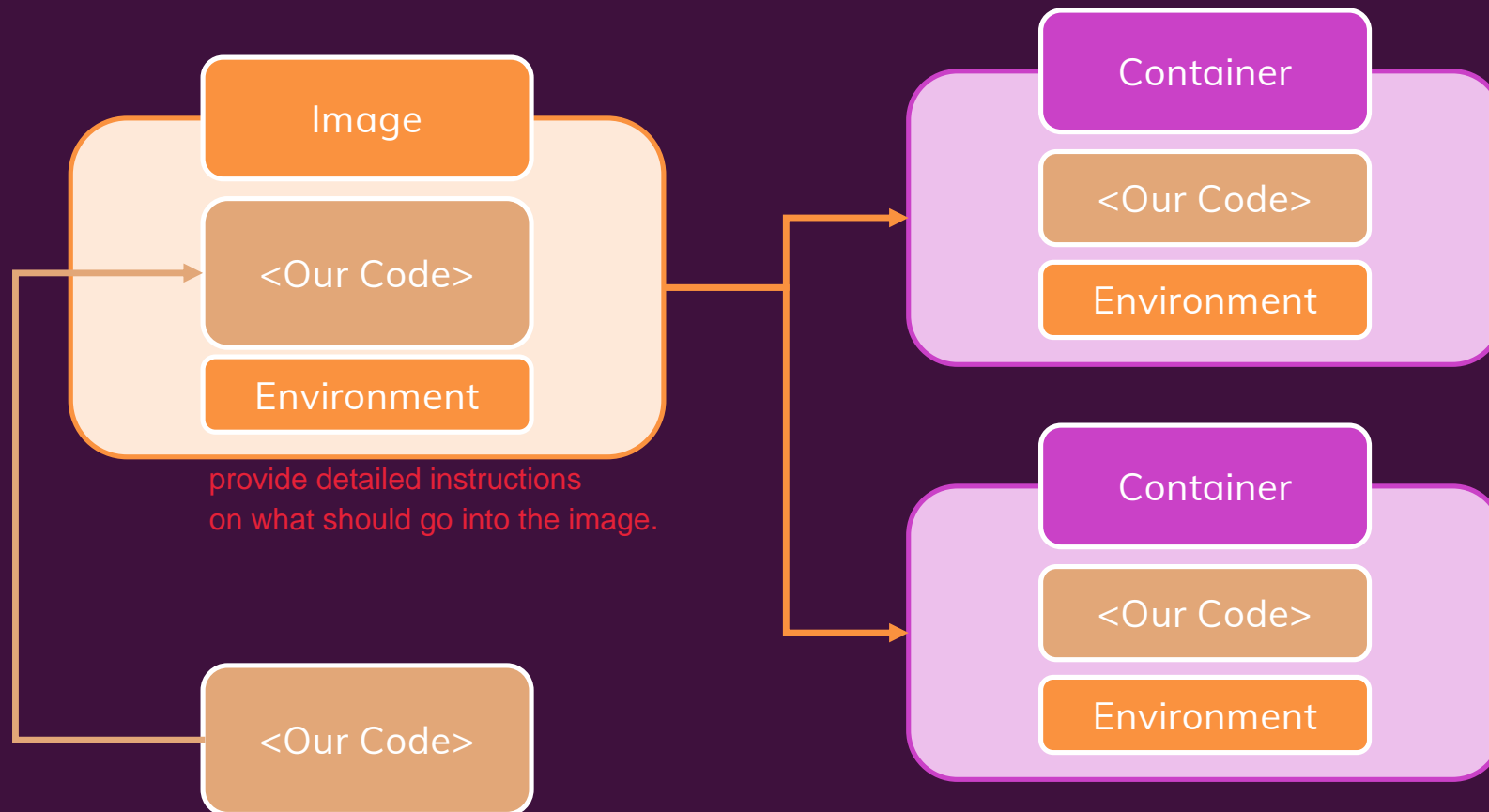
Configure all tools and setup  
steps you need in your containers

Possibly share with other  
developers

Combine third-party images with  
custom setup steps and tools

Typically, you'll combine both

## Images & Containers – First Summary





# Managing Images & Containers

Add **--help** to see all options

## Images

Can be **tagged** (named)  
*-t, docker tag ...*

Can be **listed**  
*docker images*

Can be **analyzed**  
*docker image inspect*

Can be **removed**  
*docker rmi, docker prune*

## Containers

Can be **named**  
*--name*

Can be **configured in detail**  
*see --help*

Can be **listed**  
*docker ps*

Can be **removed**  
*docker rm*

## Understanding Image Tags

name : tag

Defines a **group** of,  
possible more  
specialized, images

Example: "node"

Defines a **specialized**  
image within a group of  
images

Example: "14"

Combined: A unique identifier

# Sharing Images & Containers

pull images from docker hub

Everyone who **has an image**, can create **containers** based on the image!

share images

Share a **Dockerfile**

Simply run **docker build .**

**Important:** The Dockerfile instructions **might need surrounding files** / folders (e.g. source code)

Share a **Built Image**

**Download** an image, **run a container** based on it

**No build step** required, **everything is included in the image** already!

## Sharing via Docker Hub or Private Registry

Free Usage Possible!

Docker Hub

Official Docker Image Registry

Public, private and “official”  
Images

Private Registry

Any provider / registry you want to use

Only your own (or team) Images

Share: `docker push` **IMAGE\_NAME**

Use: `docker pull` **IMAGE\_NAME**

Needs to be  
**HOST:NAME** to  
talk to private  
registry

## Key Commands

`docker create`

Create a new container

`docker run`

Create and start a new container based on an image

`docker stop`  
`docker start`

Stop a running container  
Start a stopped container

`docker rm`

Removed a stopped container (delete it)

`docker push / pull`

Share / Download a remote image

`docker build`

Build a new image based on a Dockerfile

`docker rmi`

Remove a local image

## Module Summary

Docker is all about **Images** & **Containers**

**Images** are the **templates / blueprints** for **Containers**, multiple **Containers** can be created based on one **Image**.

**Images** contain **multiple layers** (1 Instruction = 1 Layer) to optimize build speed (caching!) and re-usability

**Containers** can be **listed** (*docker ps*), **removed** (*docker rm*) and **stopped + started** (*docker stop / start*)

**Images** are either downloaded (*docker pull*) or created with a **Dockerfile** and *docker build*.

**Containers** are created with *docker run IMAGE* and can be configured with **various options / flags**

**Images** can also be **listed** (*docker images*), **removed** (*docker rmi*, *docker image prune*) and **shared** (*docker push / pull*)