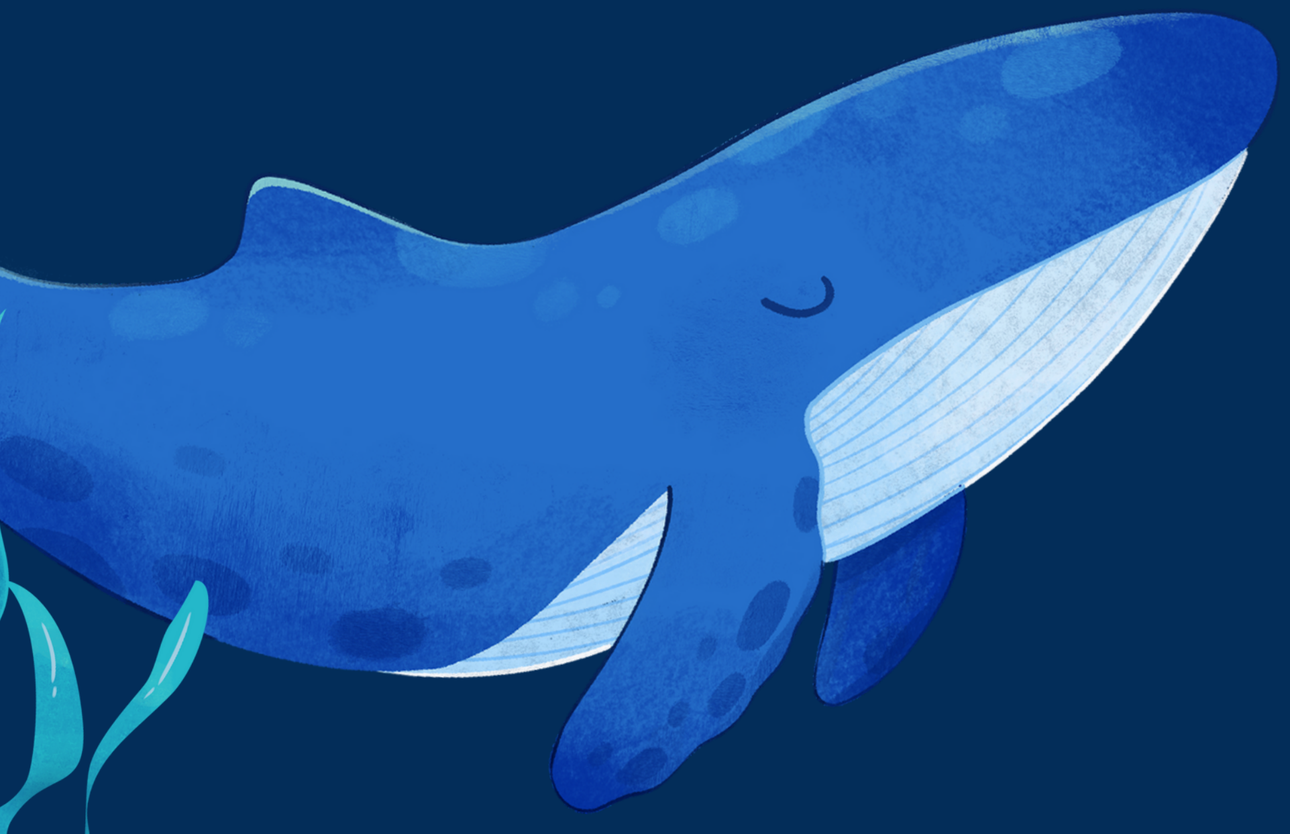
DOCKER CONTAINERS

What is a Container?

> A container is a standard unit of software that packages up code and all its dependencies so the application runs quickly and reliably from one computing environment to another



<u>Standalone Executable Package of Software</u>

- Dictates how the software will run
- Includes everything that it will need to run:
  - Code files
  - Supporting libraries
  - Additional files
  - Settings or configs that can be modified at run time
- Highly Reliable & Available Software Deployment

Virtualizes the operating system, not the hardware like a virtual machine

- Separates the application & its dependencies from its infrastructure
- Requires less space and resources than a VM
- Can be paired with VMs to create large-scale, flexible deployments

Docker Images

The Docker Image is the file which instructs how the Docker Container will be built:

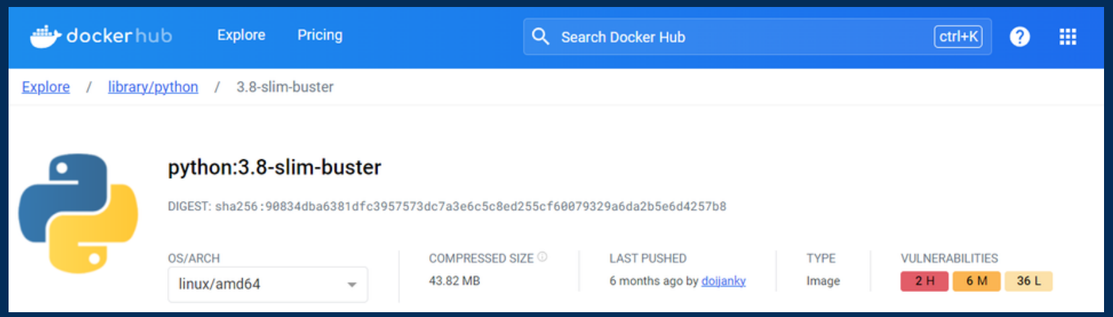
- Built in layers with sequential execution
- Specifies how & what will be running in the container
- Can be built off of existing Docker Images

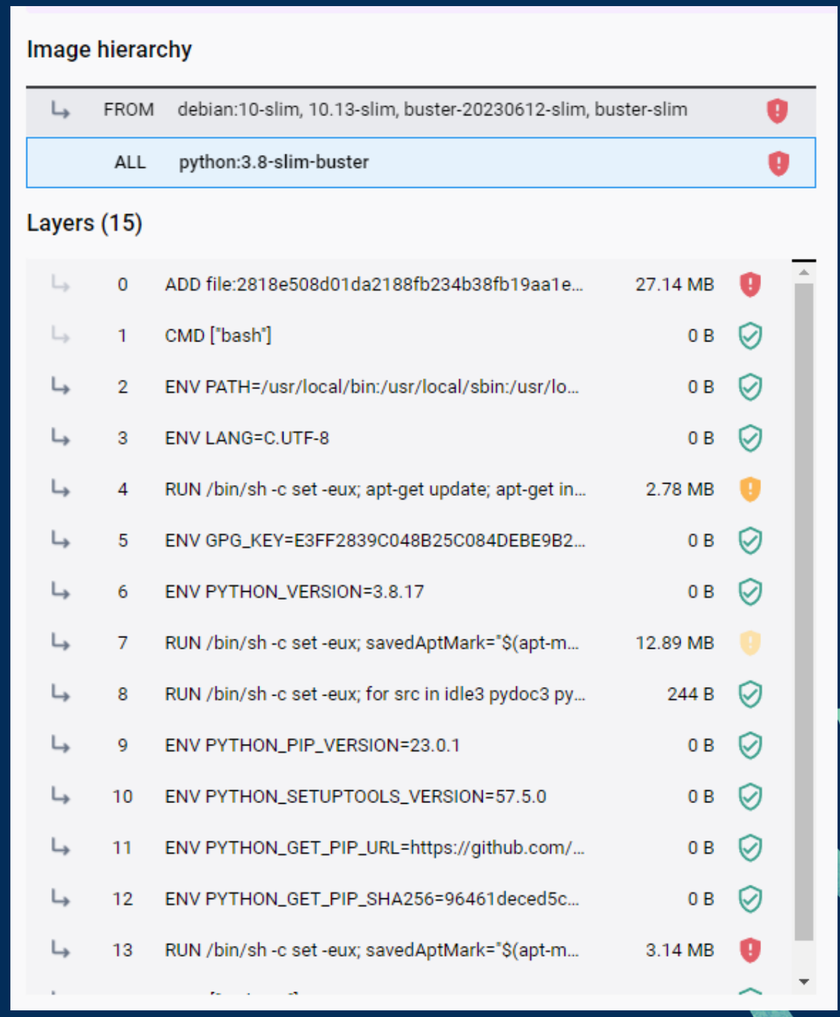
Deploying Python Applications in a Container:

- Requires a base image with an existing OS and Python Environment
- Needs to specify how to install supporting libraries & mount additional files

Base Images can be found on Docker Hub:

- Each image will have a tag or version associated with it
- The layers and heirarchy of the image can be viewed
- Known vulnerabilities will be highlighted
- Lists installed packages (PIP, apt, bash, ...)





Where are Containers Used?

Almost Everywhere:

- On your home computer
- Large Scale ML Datacenters
- Cloud Serives
- Industrial or Manufacturing Facilities

... Anywhere software is being deployed, containers can and are being used to do so!

For large-scale Docker deployments, a ‘Container Orchestrator’ like Kubernetes or Docker Swarm is used to:

- Manage containers
- Configure how they are deployed
- Allow for cluster computing across many machines.