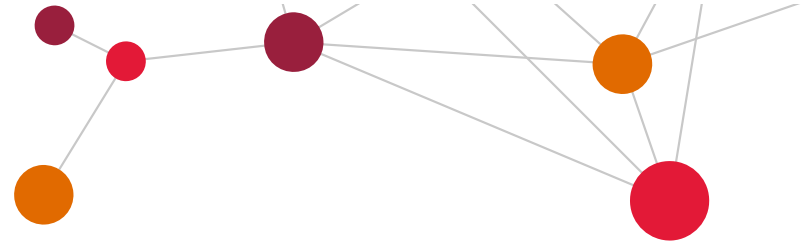


# From Zero to Docker

Mario Dagot – Software Architect  
Hugo Calado – Software Architect  
Jorge Dias – Software Architect

# About the Speakers

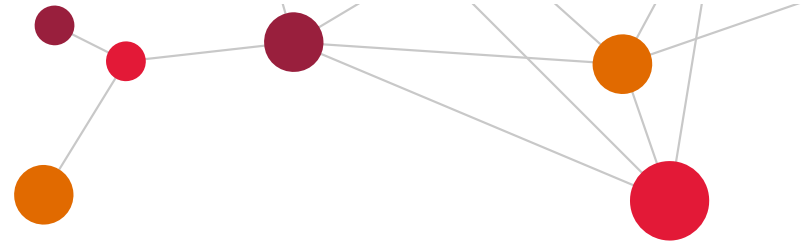


**Mário Dagot** is a software architect at CGI Portugal with over 15 years of experience in the IT industry.

Has worked as a software and system engineer, backend and web developer. Most recently has been part of a DevOps team focusing on improving processes and breaking down silos between development and operation teams.

His passions are, hanging out with family and friends, squash, gym, photography and all that is IT related.

# About the Speakers

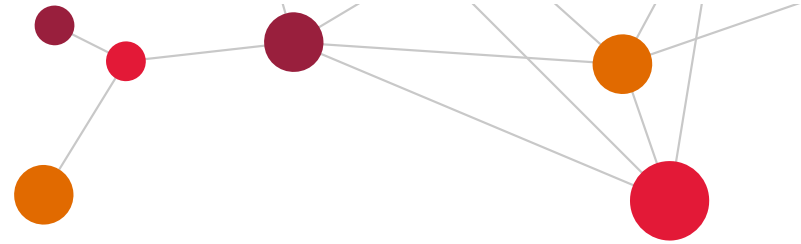


**Jorge Dias** is an IT Consultant with more than 15 years of experience, with a technical background, acting with architect functions, as technical and functional analyst and programmer.

In last 4 years has worked in a nearshore cooperation with Finland CGI, what give him the opportunity to improve his experience on the area of the Continuous Integration and Systems Management.

His passion is sharing some time with family and friends. IT enthusiastic that always try to improve and expand his knowledge.

# About the Speakers



**Hugo Calado** is an IT Consultant with more than 15 years of experience.

In last years have been involved in the development of CGI Sm@rtering product. Have joined CGI Global Delivery Center a year ago where he has been working in Mobilog product along with CGI Finnish team.

His passion is spending time with family and friends, travel, mountain biking, swimming and relax at the beach.

# What is a container

A container is a software package that contains everything it needs to run. This includes the executable program as well as **system tools**, **libraries**, and **settings**.

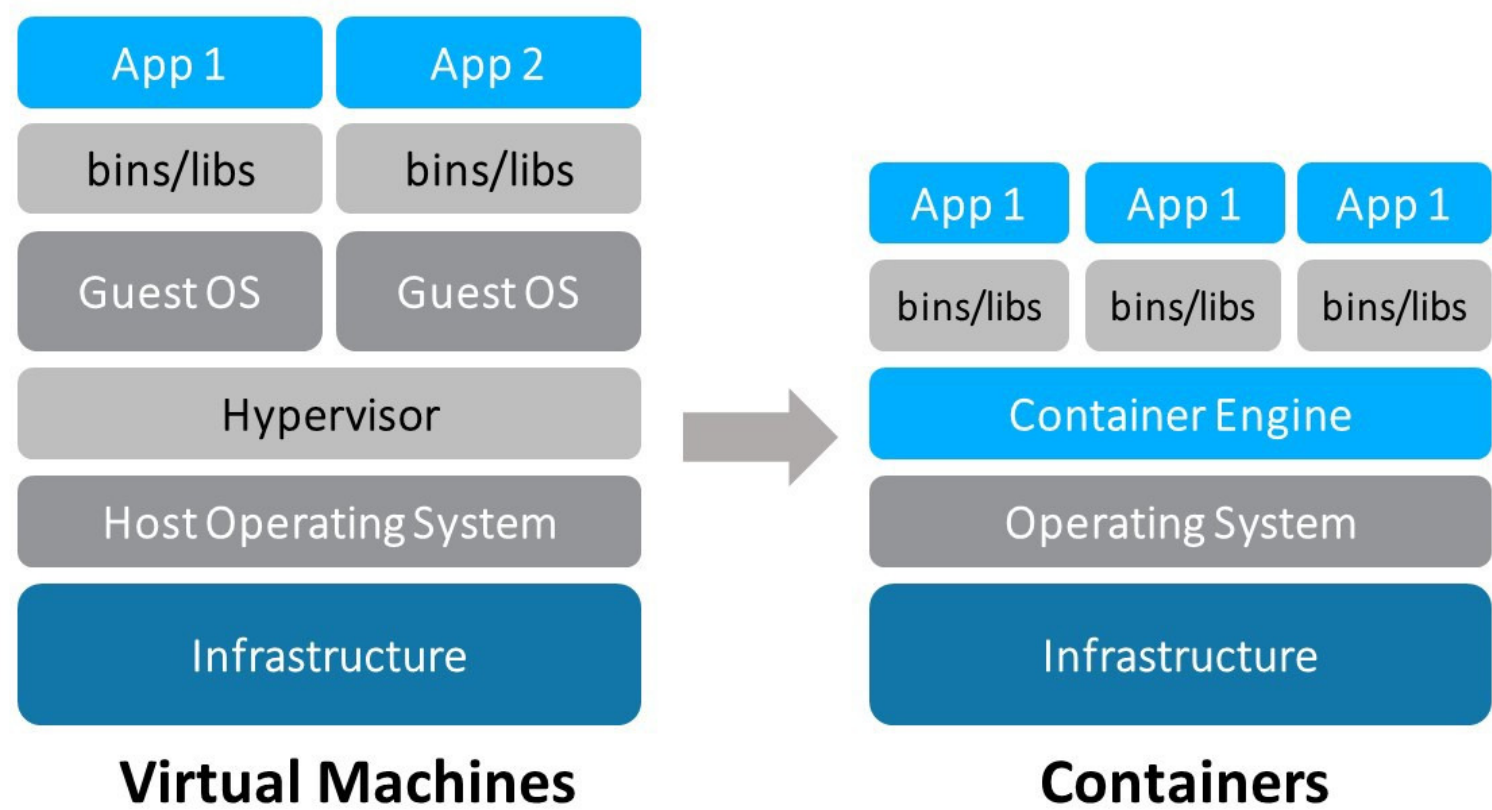
## Strong Points:

- Agile environment
- Enhanced productivity
- Version control
- Computing environment portability
- Standardization
- Secure

## Weak Points:

- Increased complexity
- Private data inside container images
- Persistent storage
- Native Linux support

# Comparing Containers and Virtual Machines



# What is Docker

**Docker** is a computer program that performs operating-system-level virtualization, also known as "containerization".

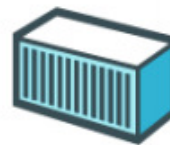
**Docker** allows to package an application into a standardized unit for software development: – The Docker Container.

**Docker promise:** Build, Ship, Run!



Build

Develop an app using Docker containers with any language and any toolchain.



Ship

Ship the "Dockerized" app and dependencies anywhere - to QA, teammates, or the cloud - without breaking anything.



Run

Scale to 1000s of nodes, move between data centers and clouds, update with zero downtime and more.

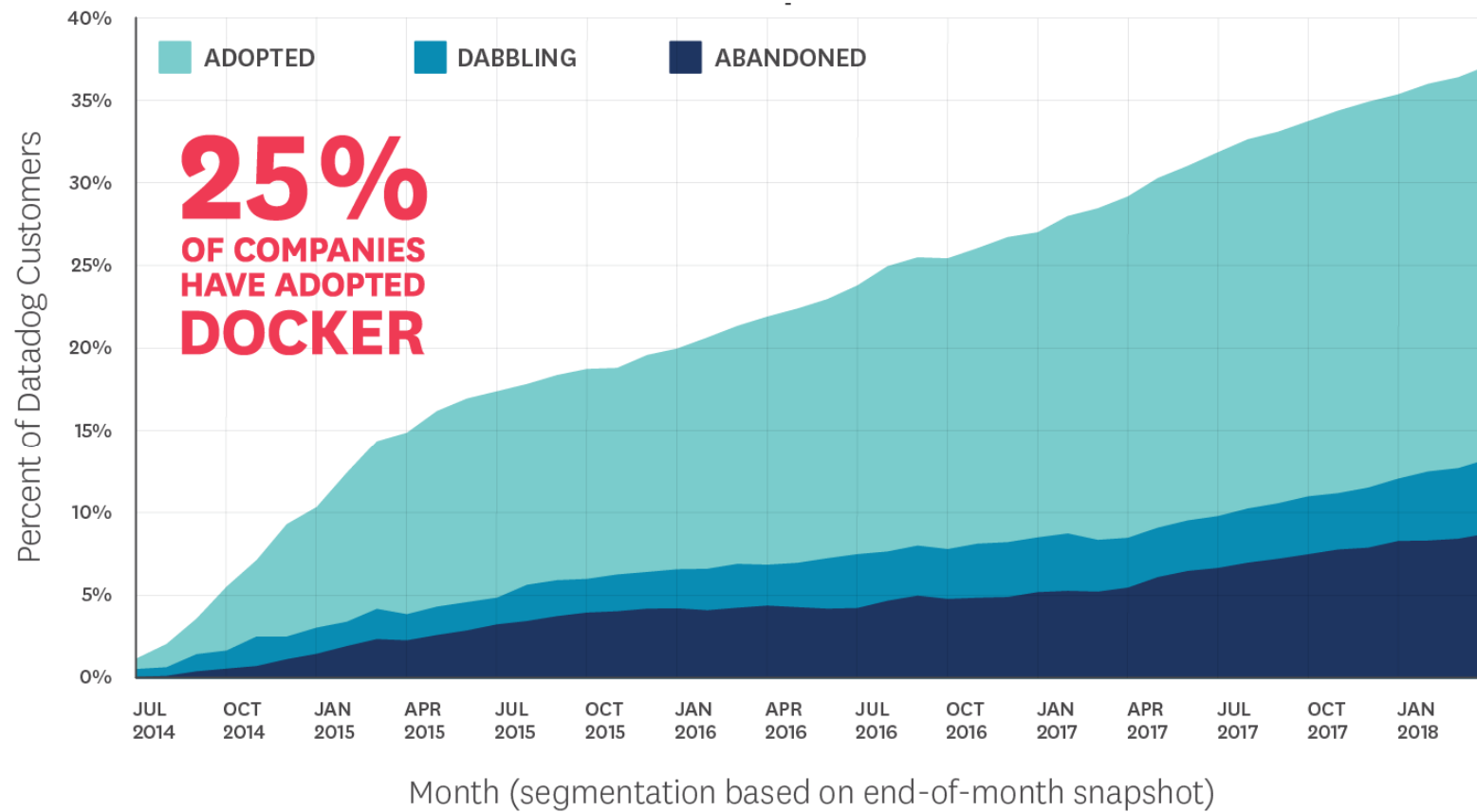
# Docker Facts



<u>Original author(s)</u>	<u>Solomon Hykes</u>
<u>Developer(s)</u>	<u>Docker, Inc.</u>
Initial release	March 13, 2013; 5 years ago
<u>Stable release</u>	18.06.1-ce <sup>[1]</sup> / August 22, 2018; 42 days ago
<u>Repository</u>	<u>github.com/docker/docker-ce</u>
Written in	<u>Go</u> <sup>[2]</sup>
<u>Operating system</u>	<u>Linux</u> , <sup>[a]</sup> <u>Windows</u> , <u>macOS</u>
<u>Platform</u>	<u>x86-64</u> , <u>ARM</u>
<u>Type</u>	<u>Operating-system-level virtualization</u>
<u>License</u>	<ul style="list-style-type: none"> <li>• <b>Binaries:</b> <u>Freemium software as a service</u><sup>[5]</sup></li> <li>• <b>Source code:</b> <u>Apache License 2.0</u></li> </ul>
Website	<u>docker.com</u>

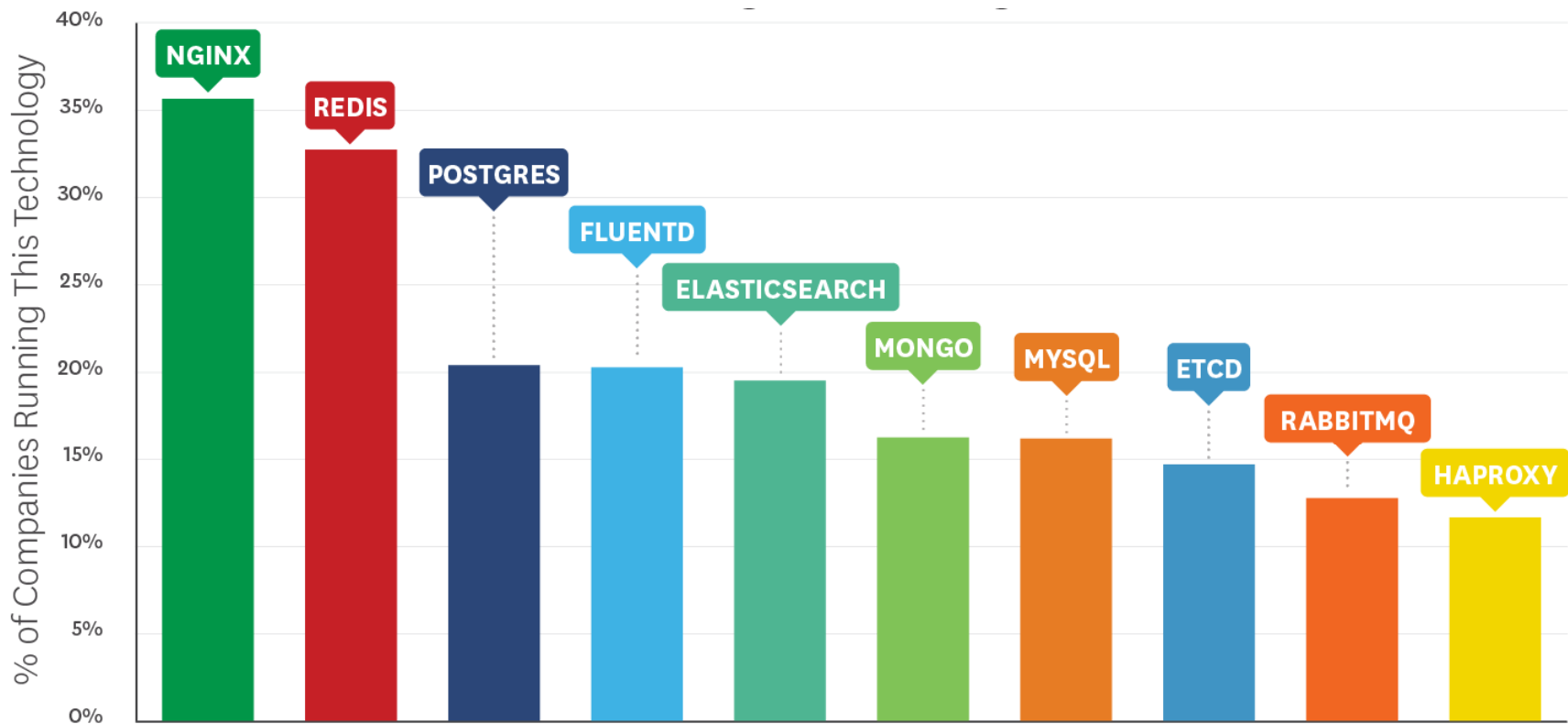


# Docker Adoption Behavior



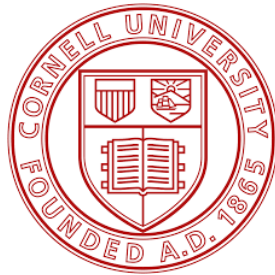
Source: Datadog

# Top Technologies Running on Docker



Source: Datadog

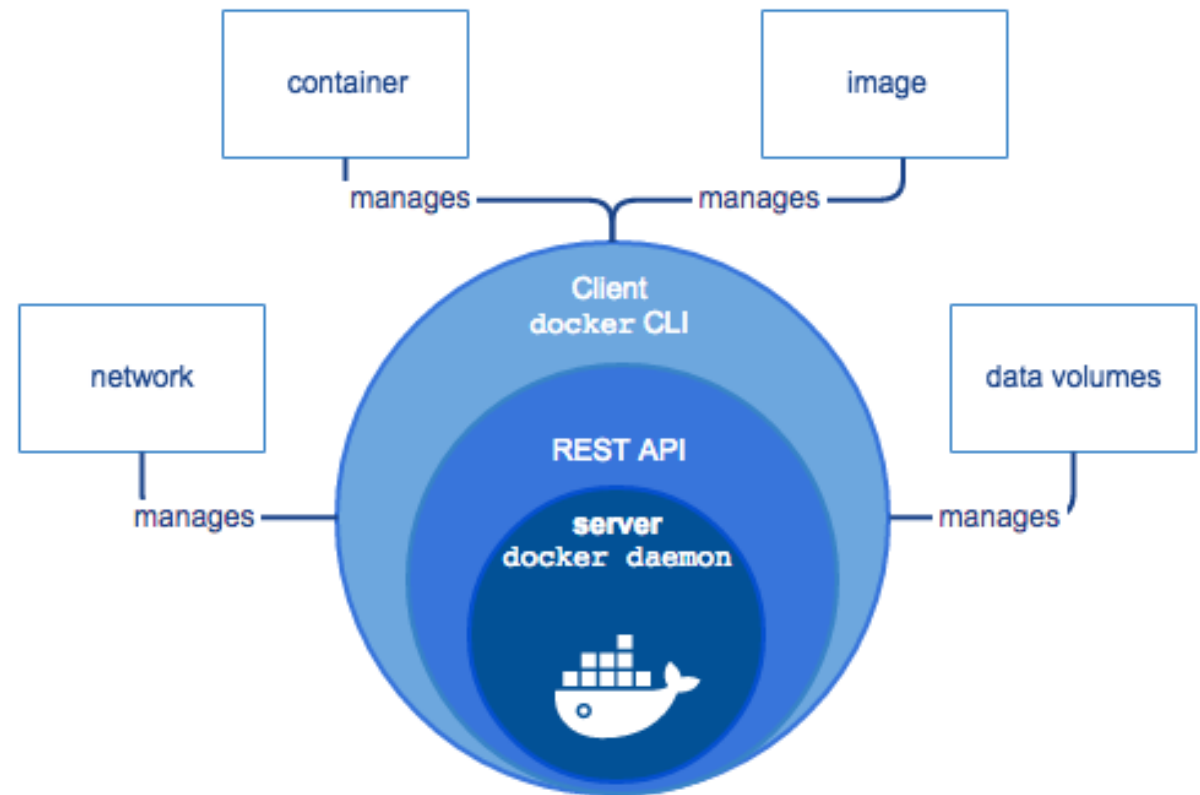
# Who is using Docker?



# Docker Architecture

Docker Engine is a client-server application with three major components:

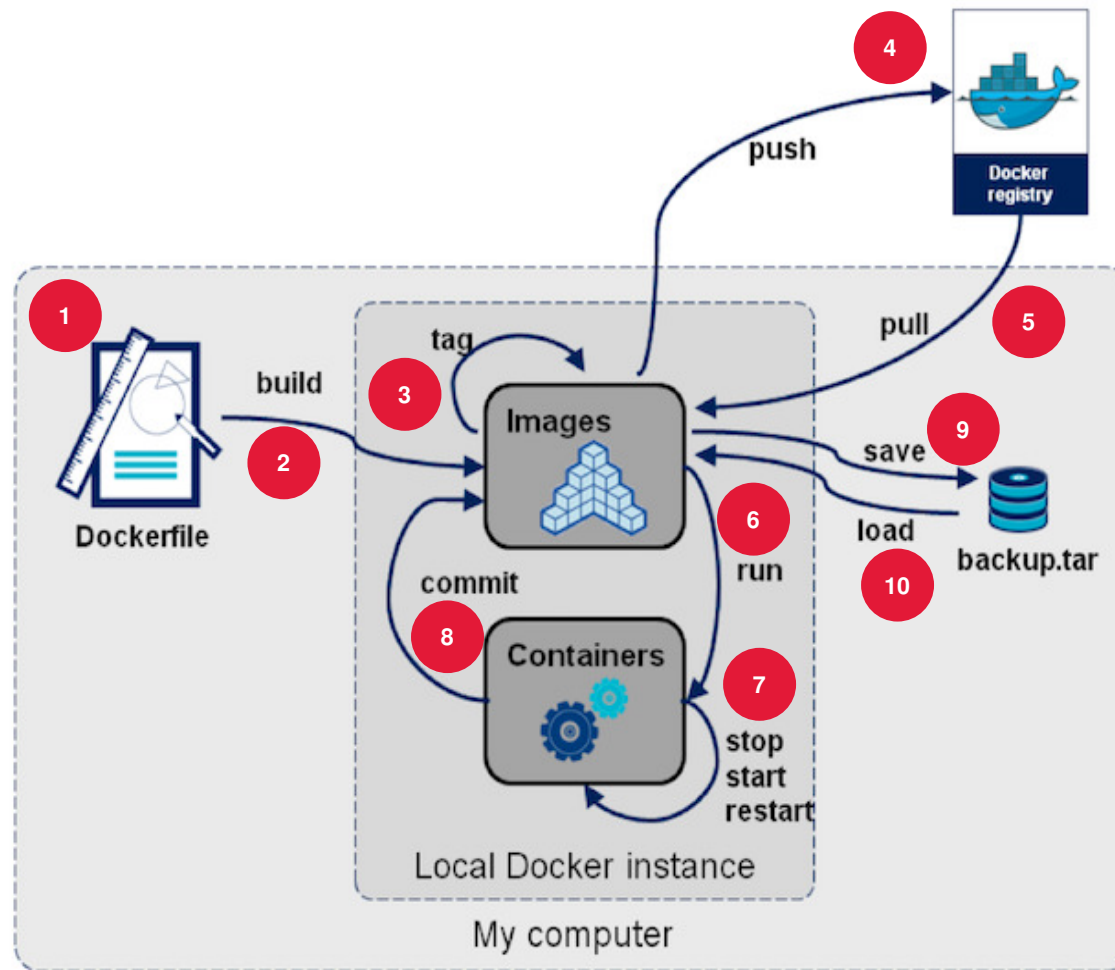
- A server which is a type of long-running program called a daemon process (the **dockerd** command).
- A REST API which specifies interfaces that programs can use to talk to the daemon and instruct it what to do.
- A command line interface (CLI) client (the **docker** command).



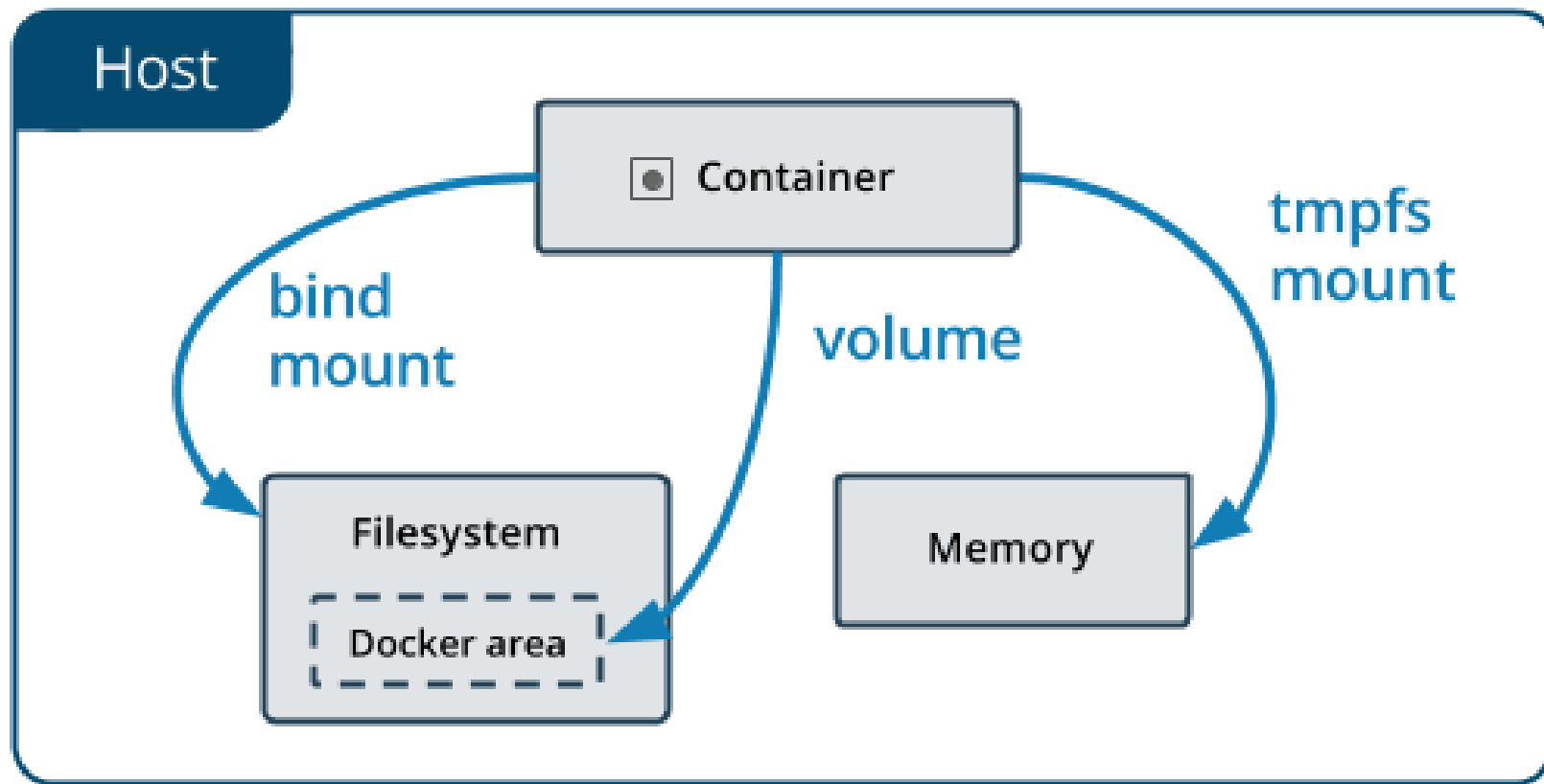
# Docker Concepts

Image	A read-only template with instructions for creating a Docker container
Container	A runnable instance of an image.
Volume	Volumes are the preferred mechanism for persisting data generated by and used by Docker containers.
Service	In a distributed application, different pieces of the app are called "Service". A service only runs one image, but it codifies the way that image runs - what ports it should use, how many replicas of the container should run so the service has the capacity it needs, and so on.
Network	One of the reasons Docker containers and services are so powerful is that you can connect them together, or connect them to non-Docker workloads
Stack	A group of interrelated services that share dependencies, and can be orchestrated and scaled together.
Swarm	A group of machines that are running Docker and joined into a cluster.
Registry	The Registry is a stateless, highly scalable server side application that stores and lets you distribute Docker images. The Registry is open-source, under the permissive <a href="#">Apache license</a> .

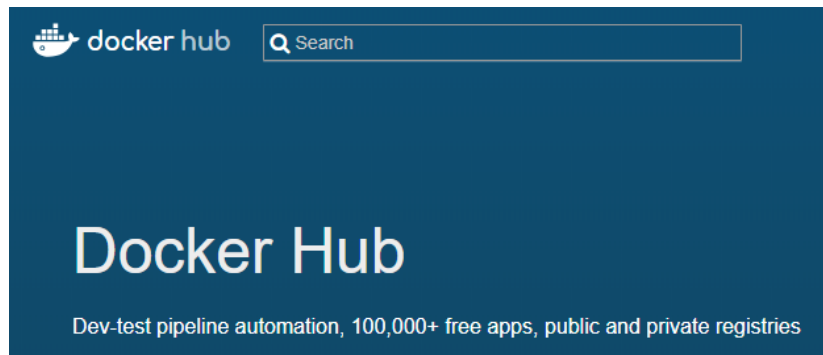
# Docker workflow





# Manage data in Docker




# Docker Registry – Docker Hub



## Repositories (16159)

All			
	tomcat official	2.1K STARS	10M+ PULLS
	bitnami/tomcat public   automated build	25 STARS	100K+ PULLS

 tomcat

Explore Help Sign up Sign in

OFFICIAL REPOSITORY

tomcat ☆

Last pushed: 2 days ago

Repo Info Tags

Short Description

Apache Tomcat is an open source implementation of the Java Servlet and JavaServer Pages technologies

Full Description

Supported tags and respective Dockerfile links

- 7.0.91-jre7, 7.0-jre7, 7-jre7, 7.0.91, 7.0, 7 ([7/jre7/Dockerfile](#))
- 7.0.91-jre7-slim, 7.0-jre7-slim, 7-jre7-slim, 7.0.91-slim, 7.0-slim, 7-slim ([7/jre7-slim/Dockerfile](#))

Docker Pull Command

docker pull tomcat



# Usefull Commands

Command	Description
docker pull	Pull an image or a repository from a registry
docker run	Run a command in a new container
docker ps	List containers
docker container	Manage containers
docker images	List images
docker start	Start one or more stopped containers
docker stop	Stop one or more running containers
docker inspect	Return low-level information on Docker objects
docker rm	Remove one or more containers
docker rmi	Remove one or more images
docker logs	Fetch the logs of a container
docker build	Build an image from a Dockerfile
docker tag	Create a tag TARGET_IMAGE that refers to SOURCE_IMAGE

# But, it's just the beginning of the journey

## 1. CONTAINERIZATION

- Normally done with Docker containers
- Any size application and dependencies (even PDP-11 code running on an emulator) can be containerized
- Over time, you should aspire towards splitting suitable applications and writing future functionality as microservices

## 3. ORCHESTRATION

- Pick an orchestration solution
- Kubernetes is the market leader and you should select a Certified Kubernetes Platform or Distribution
- <https://www.cncf.io/ck>



## 5. SERVICE MESH

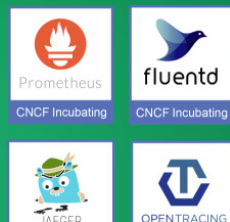
- Connects services together and provides ingress from the Internet
- Service discovery, health checking, routing, load balancing
- Consider Envoy, Linkerd and CoreDNS

## 2. CI/CD

- Setup Continuous Integration/Continuous Delivery (CI/CD) so that changes to your source code automatically result in a new container being built, tested, and deployed to staging and eventually, perhaps, to production
- Setup automated rollouts, roll backs and testing

## 4. OBSERVABILITY & ANALYSIS

- Pick solutions for monitoring, logging and tracing
- Consider CNCF projects Prometheus for monitoring, Fluentd for logging and Jaeger for Tracing
- For tracing, look for an OpenTracing-compatible implementation like Jaeger



## 7. DISTRIBUTED DATABASE

When you need more resiliency and scalability than you can get from a single database, Vitess is a good option for running MySQL at scale through sharding.



## 9. CONTAINER RUNTIME

You can use alternative container runtimes. The most common, all of which are OCI-compliant, are containerd, rkt and CRI-O.



## 6. NETWORKING

To enable more flexible networking, use a CNI-compliant network project like Calico, Flannel, or Weave Net.



## 8. MESSAGING

When you need higher performance than JSON-REST, consider using gRPC. NATS is a publish/subscribe message-oriented middleware.



## 10. SOFTWARE DISTRIBUTION

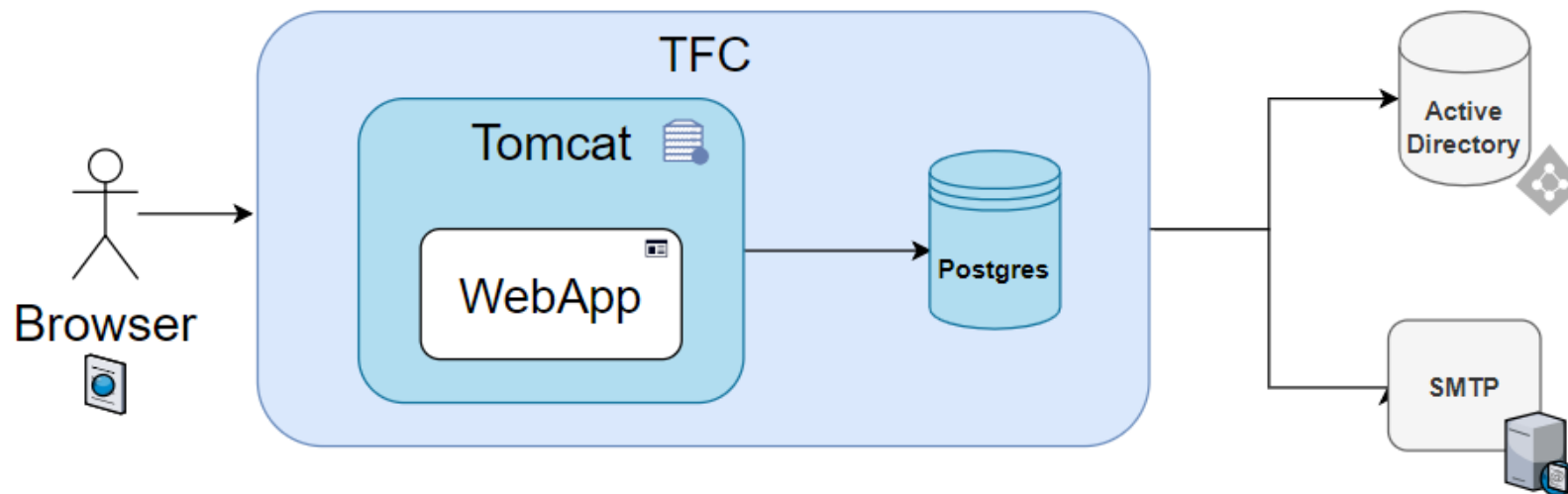
If you need to do secure software distribution, evaluate Notary, an implementation of The Update Framework.



<https://www.cncf.io/>

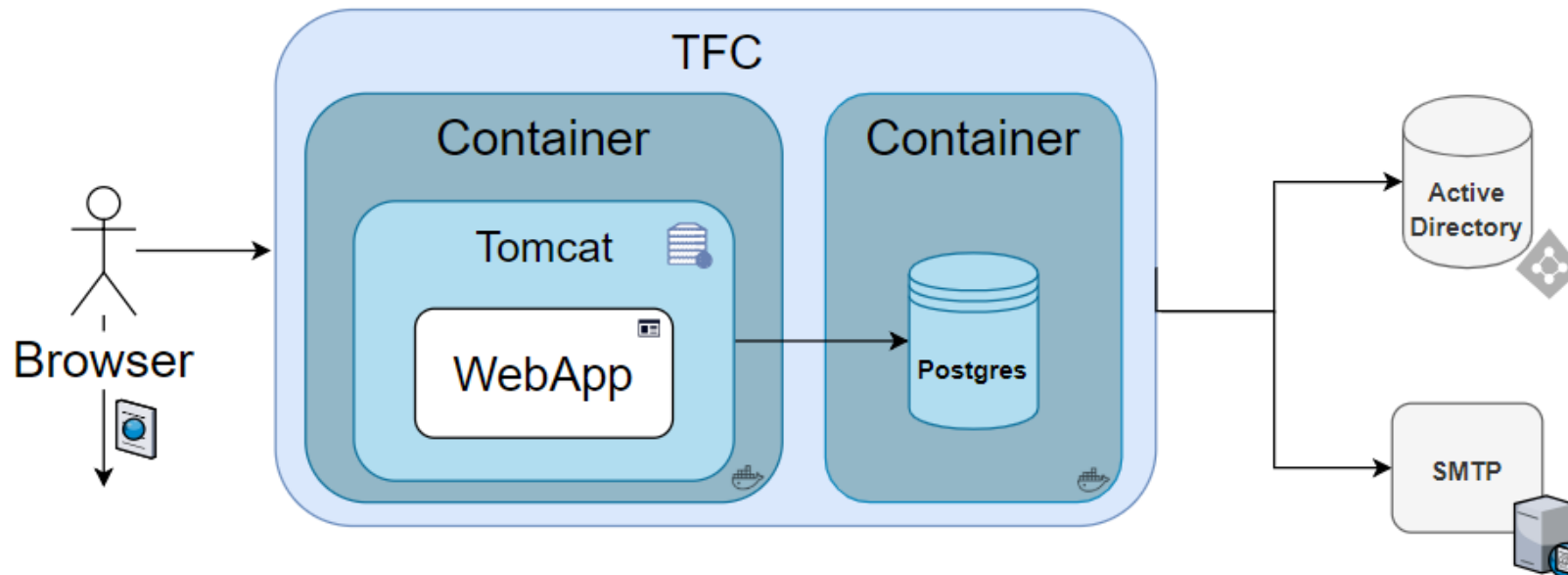
# Aplicação de Gestão de TFC's

- What is it? – Aplicação de Gestão de Trabalhos de Final de Curso
- How it works

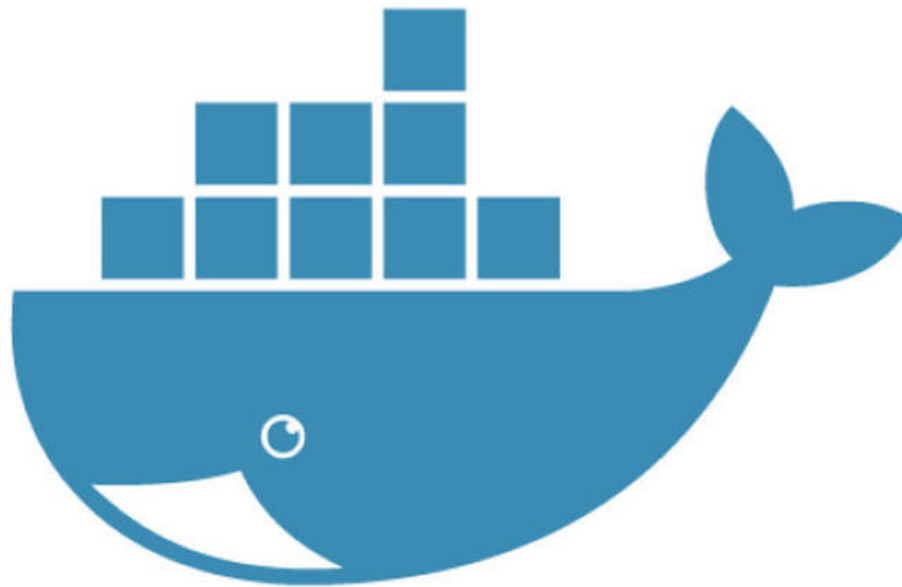


# Aplicação de Gestão de TFC's

- Proposed architecture
- Advantages
- Services/Components envisioned



# Hands On with Docker



# References and ...

From Zero to Docker on GitHub (Presentation and Hands-on material)	<a href="https://github.com/mariodagot/from-zero-to-docker">github.com/mariodagot/from-zero-to-docker</a>
Docker Get Started	<a href="https://docs.docker.com/get-started/">https://docs.docker.com/get-started/</a>
Docker Registry	<a href="https://docs.docker.com/registry/">https://docs.docker.com/registry/</a>
Docker file reference	<a href="https://docs.docker.com/engine/reference/builder/">https://docs.docker.com/engine/reference/builder/</a>
Docker Compose	<a href="https://docs.docker.com/compose/overview/">https://docs.docker.com/compose/overview/</a>
Docker compose file reference	<a href="https://docs.docker.com/compose/compose-file/">https://docs.docker.com/compose/compose-file/</a>
Docker Swarm	<a href="https://docs.docker.com/engine/swarm/">https://docs.docker.com/engine/swarm/</a>
Docker stack file reference	<a href="https://docs.docker.com/docker-cloud/apps/stack-yaml-reference/">https://docs.docker.com/docker-cloud/apps/stack-yaml-reference/</a>
Container Technologies Overview	<a href="https://dzone.com/articles/container-technologies-overview">https://dzone.com/articles/container-technologies-overview</a>
	<a href="https://en.wikipedia.org/wiki/Hype_cycle">https://en.wikipedia.org/wiki/Hype_cycle</a> <a href="https://www.slideshare.net/spnewman/confusion-in-the-land-of-the-serverless">https://www.slideshare.net/spnewman/confusion-in-the-land-of-the-serverless</a>
Docker adoption	<a href="https://www.datadoghq.com/docker-adoption/">https://www.datadoghq.com/docker-adoption/</a>