TechLabs

Docker

Intro to Containerization

A part of TechLabs Summer 23 Workshop Weekend

Intro

As useful as it is, Docker can be a difficult topic to get started with for a beginner. For this reason we will attempt to cover this topic in 3 parts:

1. Intro to Containerization (Theory)
2. Docker Tutorial
3. Docker In Application

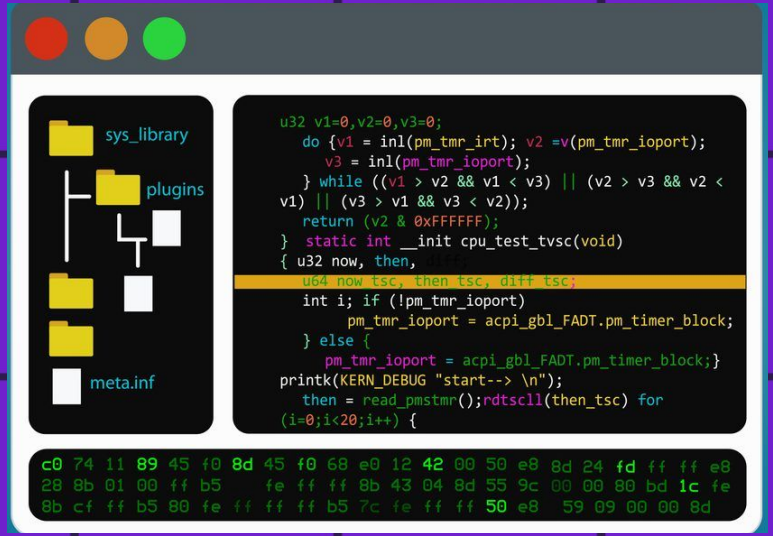
A short trek through history

hello world!

Ever since we've started writing code,we've had the issue of how to ship code!!?

Why? What's the big issue you ask?

What a dev workstation looks like:



Now that you're all familiar with code, you know how messy files, dependencies, packages, assets etc. can get

What the end-user sees:

Ideally, user should never see code. What you want them to see is a single **executable**

01

Packaging!

Process of bundling all code files into a single executable/clickable

Bundling/Packaging

This paradigm worked for a while. For almost 20 years we compiled code and shipped only the end product. This distributable would mostly be in the form of a software floppy/CD

Eventually of course, the world was changed by the advent of ...

WEB 0.0!

The OG disruptor. This changed everything.
Why?! Really??!

Internet Updates

With the internet it was possible to distribute OTW

No connectivity

Software never needed to be connected before

Hybrids

How do offline and online components work together

Online

The new default shifted towards software being up to date/online

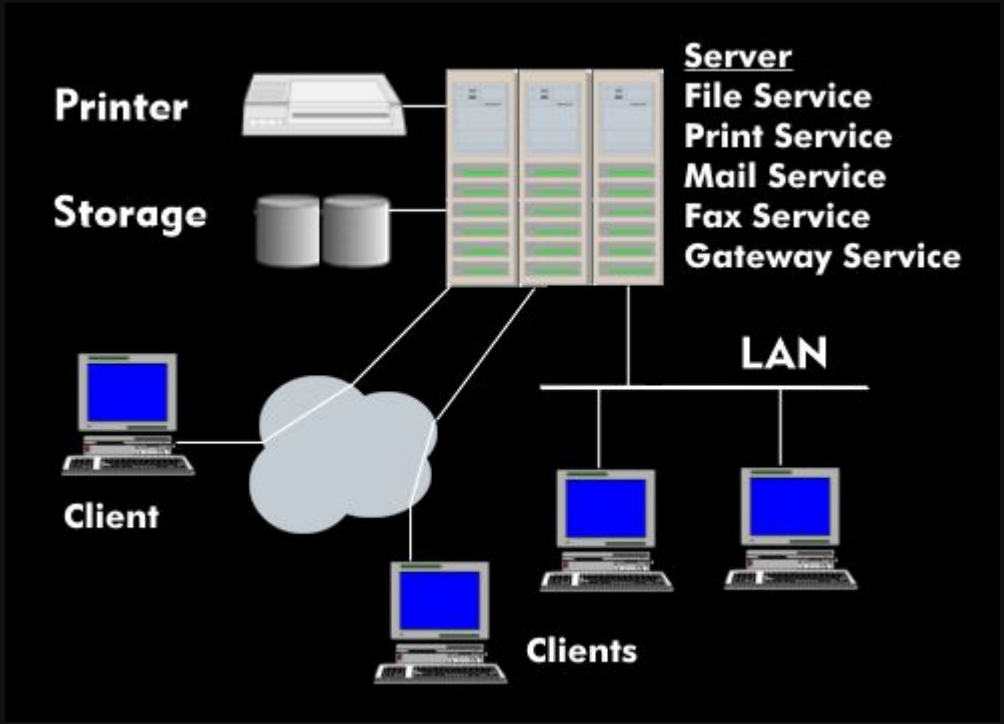
Loading . . .

1990

Tim Berners-Lee sets up the first web server running at CERN

1998

SOAP was released. Chat servers are now popular, early application of the client-server architecture



(REST) 2004

Representational State Transfer is an architecture for well-behaved Web services

With the advent of all of these technologies, frameworks and protocol, we finally had a *blueprint* for reliable web services

Now let’s finally talk about the problem. A slew of advancements led to rapid adoption of web services. But with code being divided between services, microservices, client side applications etc. the modern application stack was once again sharded.

02

Containerization

Packaging for the modern web

The goals remains the same

The challenge is confounded:

- Non-uniform servers
- Different computing power
- Harder to isolate and create clean run environments
- Dependency management
- Distributing configurations

2014 Enter Docker

- Built on Linux Containers (virtualization), extended application by using namespaces to create containers
- Slowly grew into an entire ecosystem for tools that manage containers
- With the growing threat of vulnerabilities, Docker offer secure containers