

# Cloud Computing

## What is Cloud Computing?

No need to buy, own, or maintain your own data centers

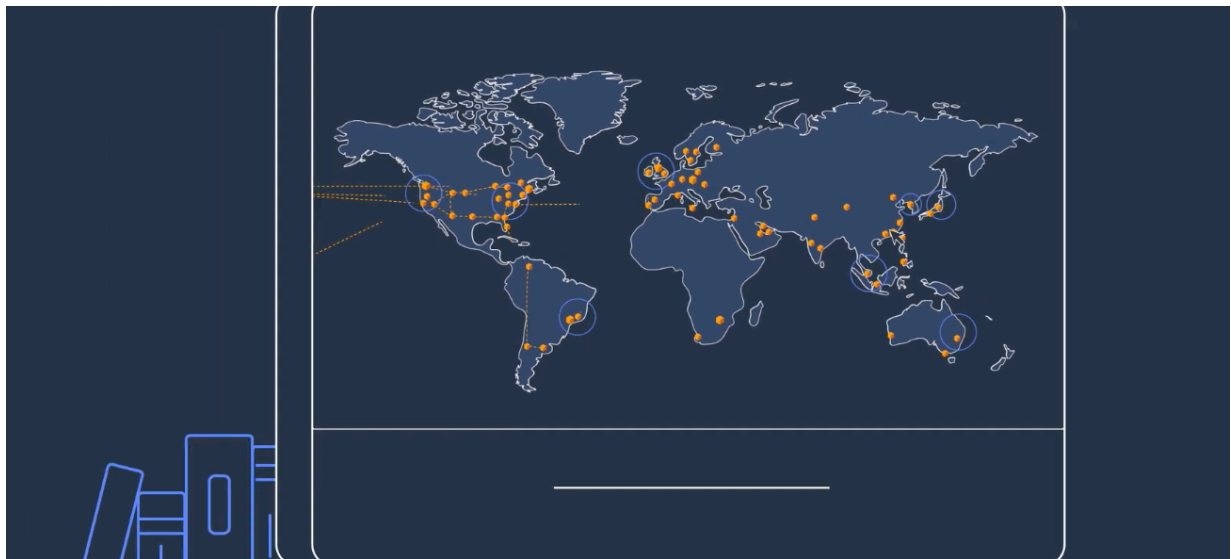
A wide variety of industries are utilizing cloud services, like health care, tech, business, etc.

Instantly scale

Global access in minutes

No need to pay for hardware/ IT you don't use

Access resources as they're needed



Setting cloud servers closer to the user reduces latency and improves user experience

## Cloud Computing in the Year 2020

90% of companies on the cloud

Computing as a service

### Infrastructure as a service

- Storage buckets (iCloud Photos, Keynotes)
- AWS storage

### Software as a service

- Software that acts as a middle man between IaaS and the user

### Platform as a service

- Services that manage database and configurations for projects, requiring developers to only worry about the code

### Backend as a service

- Developers don't have to worry about writing backends much and can focus on user experience



Example of BaaS, easy login code

# realtime app with user auth

two lines of code



```
// User Auth  
firebase.auth().signInWithGoogle(provider);  
  
// Get Data  
firebase.firestore().collection('lunch-🍱').get();|
```



Drawbacks -> Vendor lock in: you may become too reliable on these services for your product

Different types of data centers:

- On-prem : your private data center
- Public cloud: a public data center (ex. AWS)
- Hybrid: a combination of the two

# Hybrid Cloud

On-prem + public cloud



- Multi-Cloud: using multiple clouds to deploy your service, prevent vendor lock-in and optimize pricing

Service Level Agreement: a contract between the cloud provider and the user, guarantees uptime, insurance, quotas, etc.

Identity and Access Management: who has access to your cloud account/data

- You can secure your services using security policies
- Use roles for an umbrella permissions

Service accounts: machine to machine permissions

Region: cloud campus physical location

- Each region has multiple isolated data centers

Cloud services are pay-as-you-go.



Egress -> data sent to the outsider world  
Ingress -> data arriving from outsider world

Scalability:

Vertical -> more cpu cores, ram, and components to increase computation capacity in one virtual machine

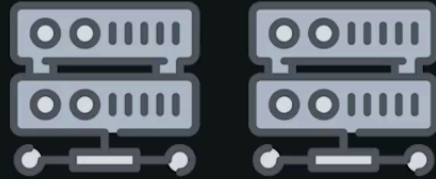
Horizontal -> more virtual machines

Containerization: like virtual machines but they sit on top of operating systems, whereas vm sits directly on top of the hardware

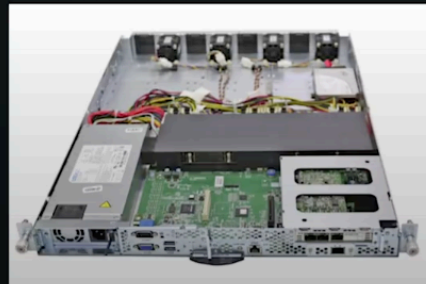
containers



VMs



bare metal



## Container Registry

A place to store containers



Kubernetes -> Organizes containers into groups of pods, then it scales those pods up and down depending on traffic

Docker

Operates at deployment stage

It's a container platform:

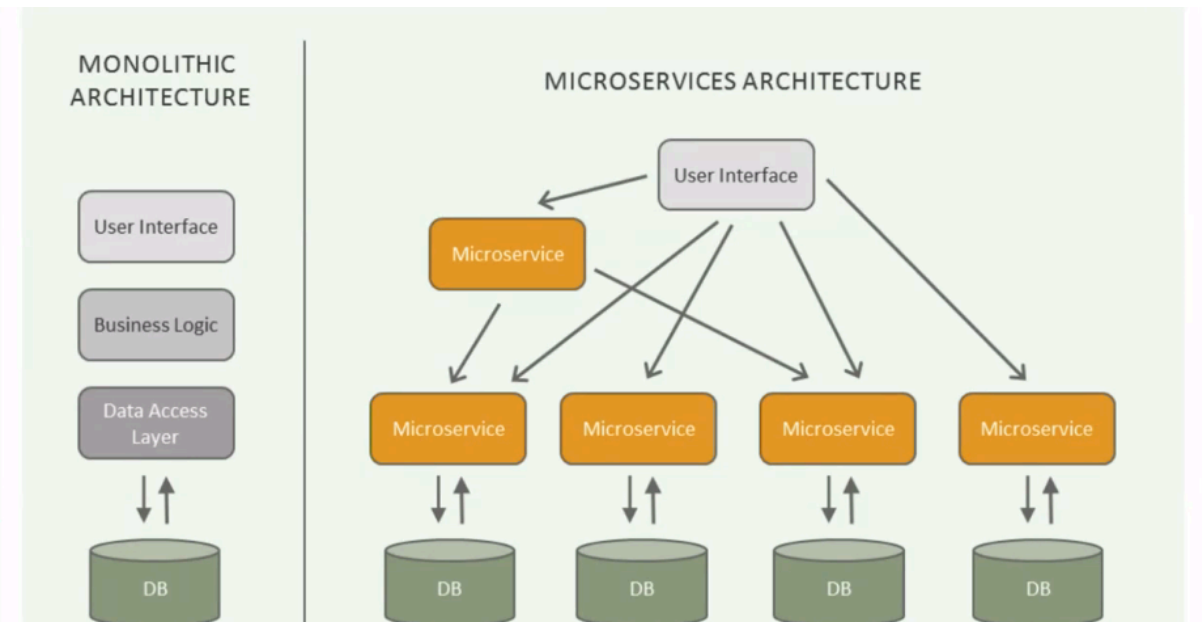
Software isn't guaranteed to work on every platform, so how do we solve that?



Containers made shipping goods easy and efficient and cheap. This new standard revolutionized the shipping industry. We need the same solution for software.

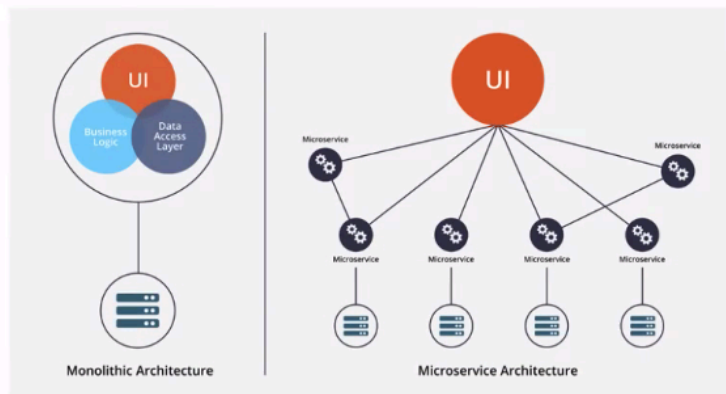
## Microservices

The separation of software into small, specific applications that communicate to transfer data, usually using http and APIs.



## Microservices - Advantages

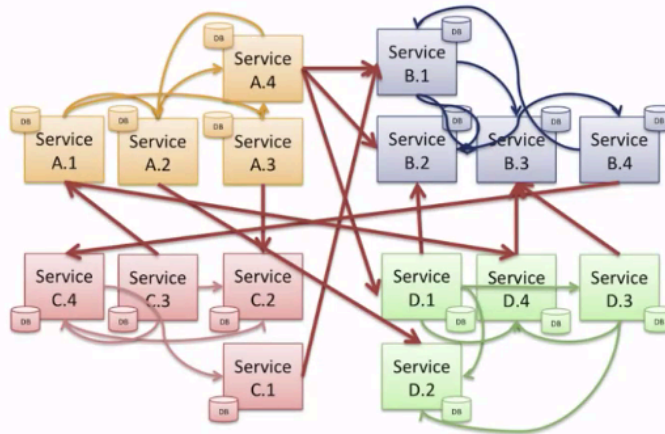
- Language independent
- Fast iterations
- Small teams
- Fault Isolation
- Pair well with containers
- SCALABLE
  - Big plus





## Microservices - Disadvantages

- Complex networking
- Overhead
  - Databases
  - Servers



# Docker

is an open platform for developers and sysadmins to build, ship, and run distributed applications, whether on laptops, data center VMs, or the cloud.

## Docker

“Containers are a way to package software in a format that can run isolated on a shared operating system. Unlike VMs, containers do not bundle a full operating system - only libraries and settings required to make the software work are needed.

This makes for efficient, lightweight, self-contained systems and guarantees that software will always run the same, regardless of where it's deployed.”

Docker files are necessary to build the images in order to run applications, though docker files are only definitions, not runnable code.

Container Orchestrator

# Kubernetes

is an open-source system for automating deployment, scaling, and management of containerized applications.