Axiom Certified Developer
Cloud Native, Web, and Mobile

Cloud Native Specialist Program

Adil Altaf - Axiom, Panacloud, PIAIC

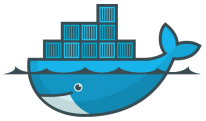





Cloud Native, Web and Mobile Specialist

Module A - 2 Months

Cloud Native Computing



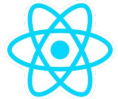
Module B - 3 Months

HTML, CSS, JavaScript

Module C - 3 Months

React & React Native

express



AXIOM

Axiom Leadership

Founder & CEO – *Adil Altaf*

15 Years of Experience in Tech Industry working in multiple sectors across Technology, Management Consulting, and BPO

- Communications, Media & Tech
- Financial Services
- Health & Public Services
- Products
- Resources
- Venture Capital & Startups

Career

- **Consult America, Inc.**
  - Intern 2004-2005
  - Consultant 2006-2010
- **Accenture, LLP**
  - Consultant 2011-2015
- **Crux Solutions, Inc.**
  - Co-Founder & CEO 2014-Present
- **Axiom**
  - Founder & CEO 2016-Present

Axiom Leadership

Founder & CEO – *Adil Altaf*

15 Years of Experience in Tech Industry working in multiple sectors across Technology, Management Consulting, and BPO

- Communications, Media & Tech
- Financial Services
- Health & Public Services
- Products
- Resources
- Venture Capital & Startups

Career

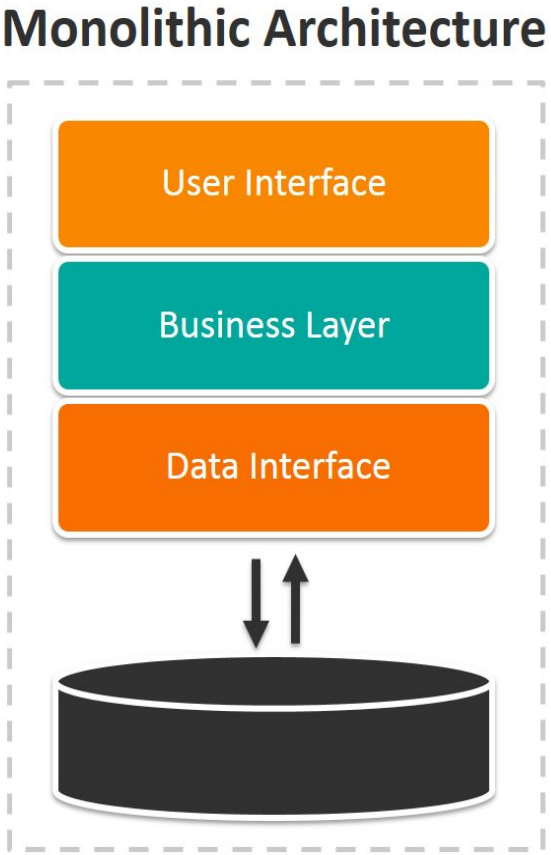
- **Panacloud Pvt. Ltd.**
  - Director North America
    2017 - Present
- **PIAIC**
  - Overall Academics Lead
    2018 - Present
  - E-Learning Lead
    2019 - Present
  - Marketing Lead
    2019 - Present
  - Strategy
    2018 - Present

Traditional Computing Methods

Monolithic Computing

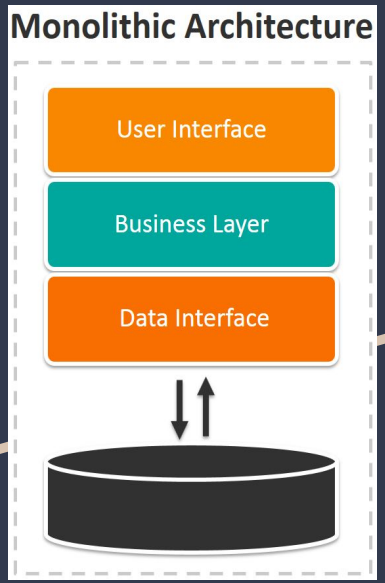
"very large, united, and difficult to change"

Monolithic, in this context, means composed all in one piece



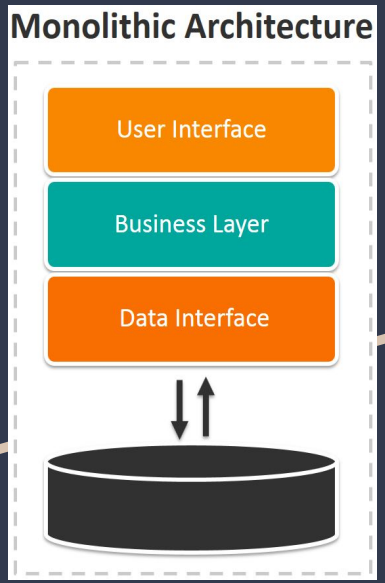
Monolithic software is designed to be self-contained meaning having all that is needed, in itself.

Monolithic Computing



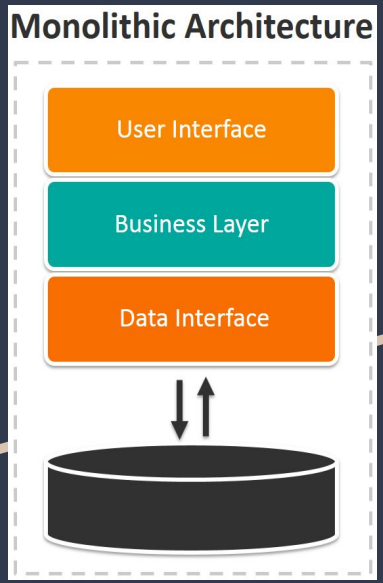
- Components/Layers of the program are interconnected and interdependent
- In a tightly-coupled architecture, each component and its associated components must be present in order for code to be executed or compiled

Monolithic Computing



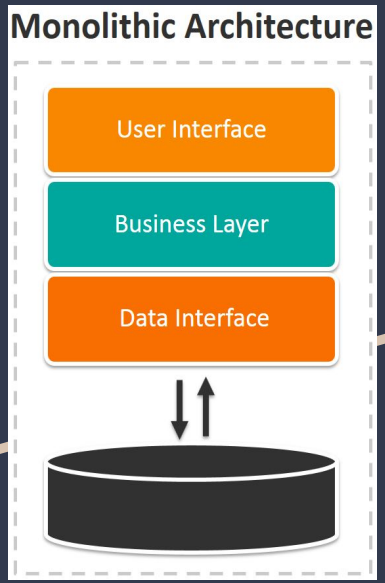
- Application is too large and complex to fully understand and made changes fast and correctly

Monolithic Computing



- The size of the application can slow down the start-up time

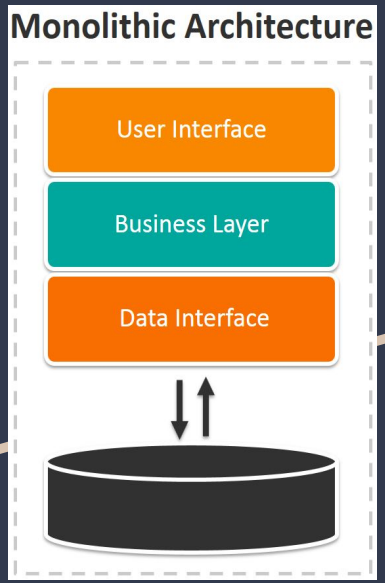
Monolithic Computing



- Another problem with monolithic applications is reliability.

A bug in any module can potentially bring down the entire process

Monolithic Computing



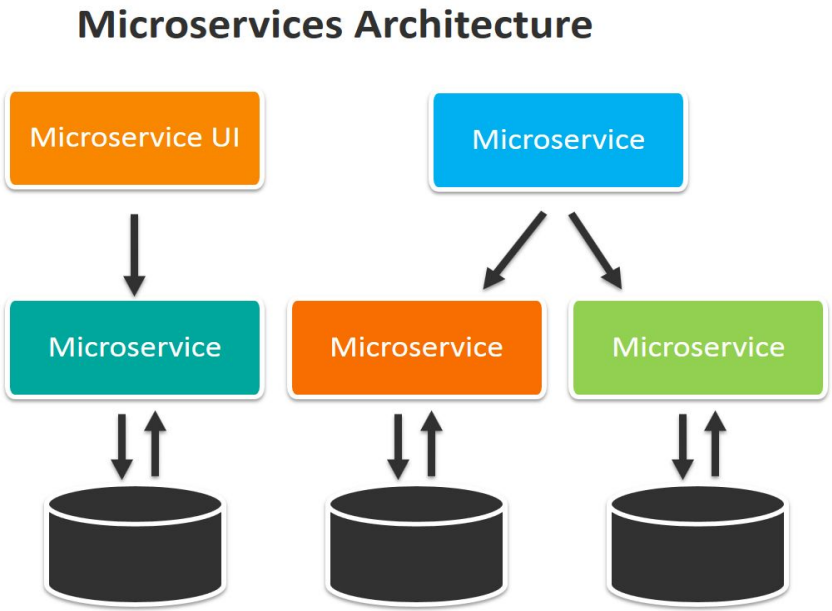
- Monolithic applications have a barrier to adopting new technologies.

  Since changes in frameworks or languages will affect an entire application it is extremely expensive in both time and cost.

Microservice Architecture

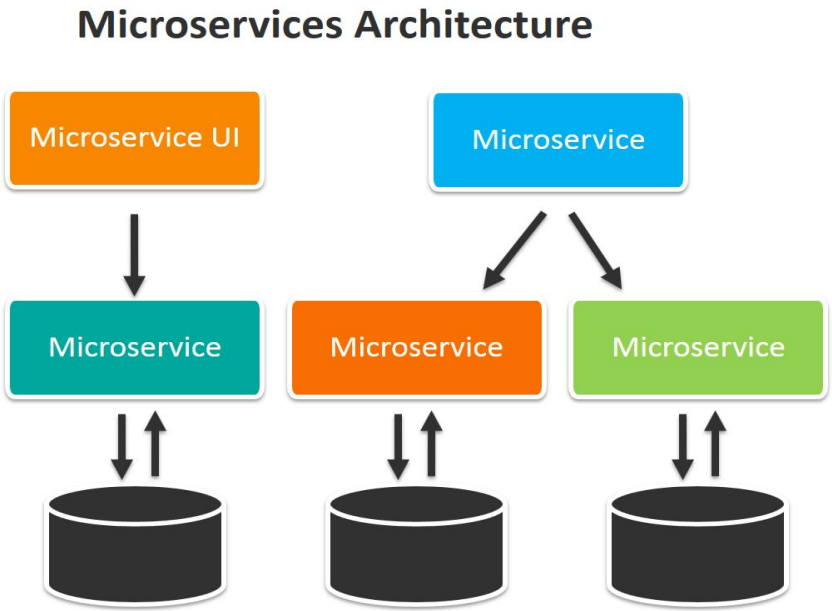
Microservice Architecture

The microservice architectural style is an approach to developing a single application as a suite of small services.



Microservice Architecture

Each runs in its own process and communicates with lightweight mechanisms, often an HTTP resource API.



Distinct Advantages of Microservices Architecture

- Better Organization
- Decoupled
- Performance

Better Organization

Microservice architectures are typically better organized

Each microservice has a very specific job, and it is not concerned with the jobs of other components

1

Decoupled

Decoupled services are also easier to change, update and re-configure to serve the purposes of different apps

They also allow for fast, independent delivery of individual parts within a larger, integrated system

2

Performance

Under the right circumstances, microservices can also have performance advantages depending on how they're organized.

It's possible to isolate hot services and scale them independently of the rest of the app.

3

What is Cloud?

What is Cloud?

A Communications Network

What is Cloud?

The word "cloud" often refers to the Internet, and more precisely to some datacenter full of servers that is connected to the Internet.

What is Cloud?

A cloud can be a wide area network (WAN) like the public Internet or a private, national or global network. The term can also refer to a local area network (LAN) within an organization.

Types of Clouds

- Private Cloud
- Public Cloud
- Hybrid Cloud

Private Clouds

Deploying cloud computing internally.

Private cloud employs cloud computing within a company's own local or wide area networks.

1

Public Cloud

A cloud computing service on the Internet that is available to the general public.

Commercial cloud providers like Amazon, Google cloud, Azure etc.

2

Hybrid Cloud

The use of both private and public clouds to provide an organization's computing needs.

3

What is Cloud Native?

What is Cloud Native?

Cloud Native Computing Foundation (CNCF) which is an open source software foundation dedicated to making cloud native computing universal and sustainable, describe cloud native as,

“Cloud native technologies empower organizations to build and run scalable applications in modern, dynamic environments such as public, private, and hybrid clouds”

What is Cloud Native?

Alternate Definition

“An approach that builds software applications as microservices and runs them on a containerized and dynamically orchestrated platform to utilize the advantages of the cloud computing model”

What is Cloud Native?

Cloud Native Computing is about **HOW** applications are created and deployed, **NOT WHERE**.

DevOps

Agile Methodology

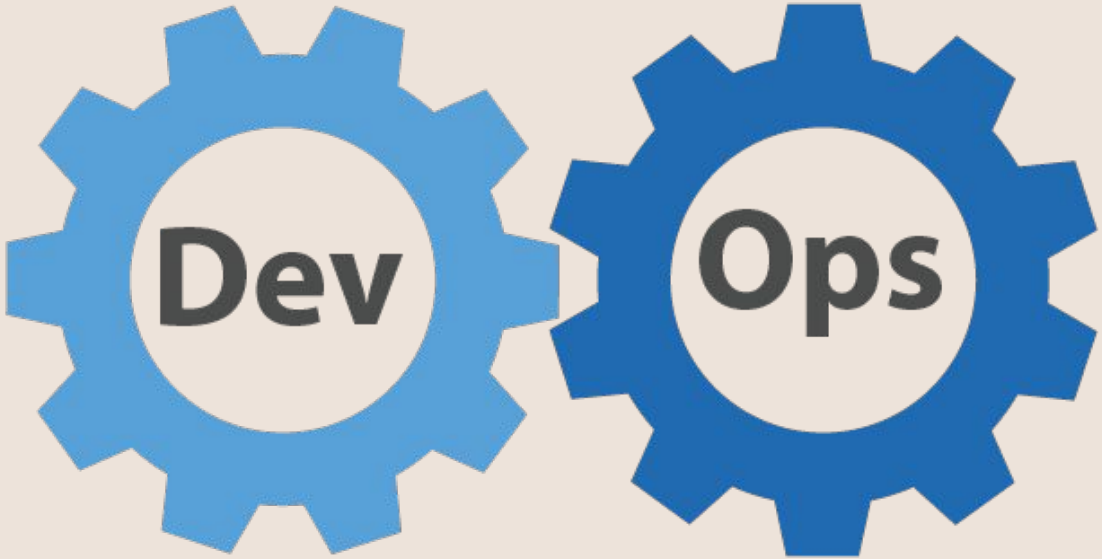
Microservices

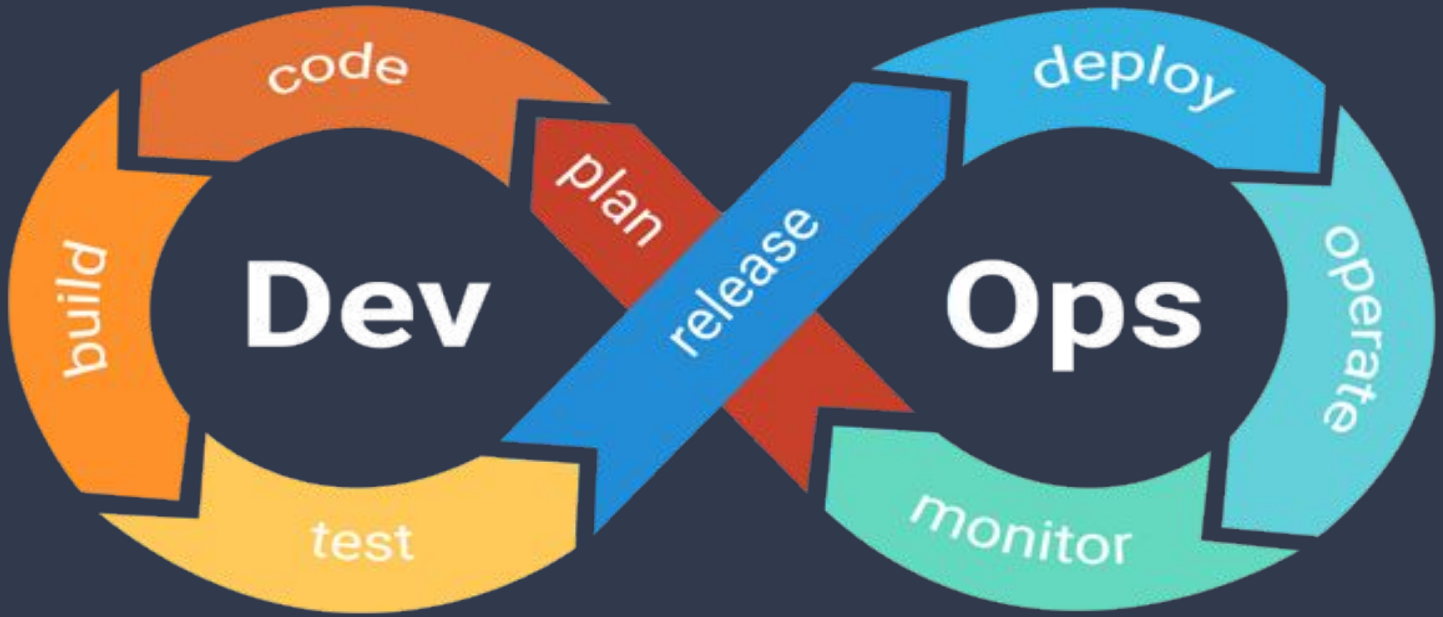
Cloud Computing Platforms

Containerizing Applications

Orchestration Systems

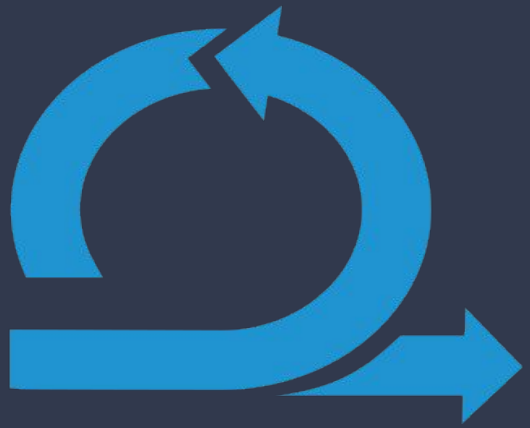
Continuous Delivery


Dev
Ops


code
plan
build
Dev
release
deploy
operate
Ops
test
monitor

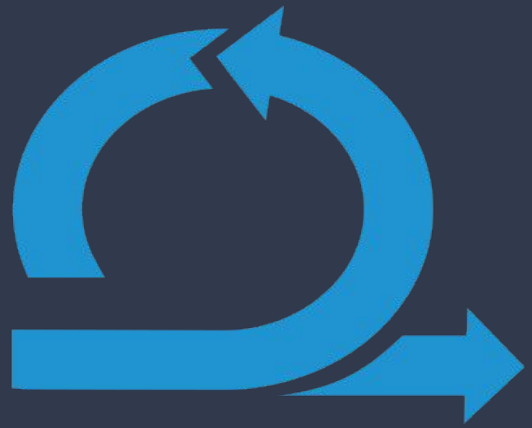
Agile Development

Agile Development



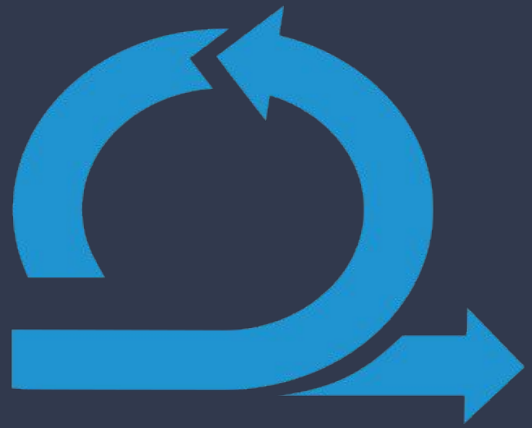
Agile methodology is described as an “iterative” and “incremental” approach.

Agile Development



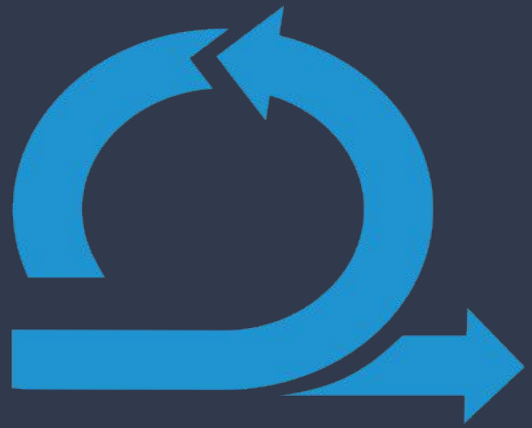
Agile developers visualize the software as a combination of complex parts that interacts with each other rather than a large block of structure.

Agile Development



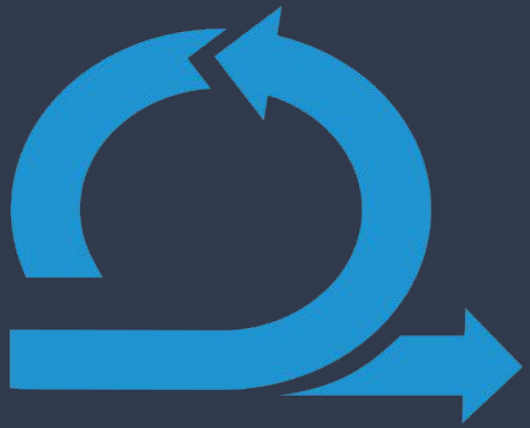
In the waterfall method, the development team will get only single chance to get each phase (like design, development, testing etc) of a project.

Agile Development



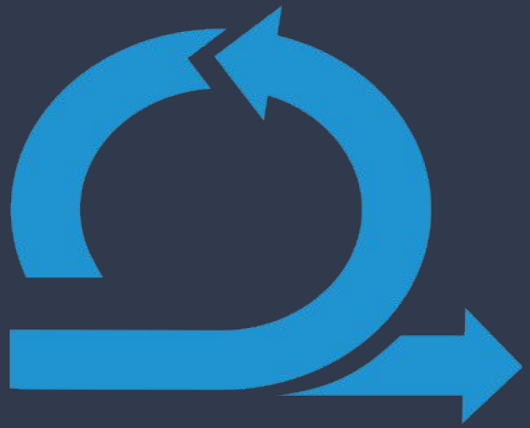
In agile methodology, these phases are continually revisited periodically to identify/understand the project's progress and direction.

Agile Development



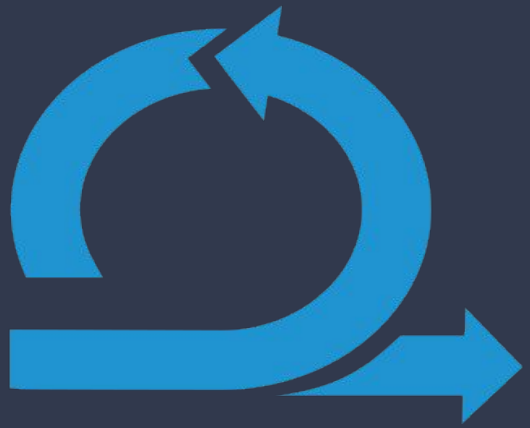
The “inspect-and-adapt” approach from Agile methodology greatly reduces development costs and time to market the product because here teams can develop the software while gathering changes in requirements.

Agile Development

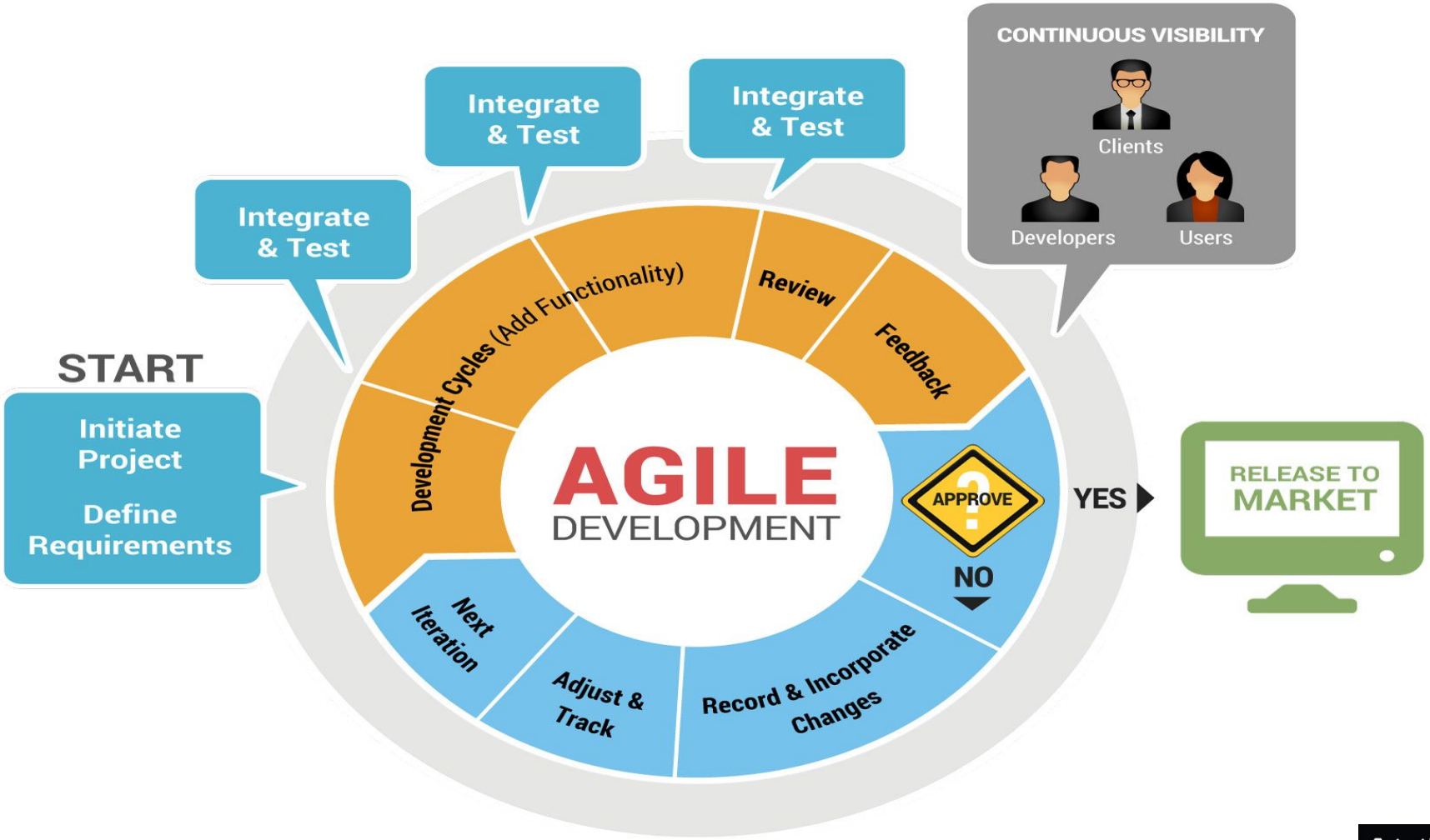


The stakeholders can provide feedback to the development team to improve the quality of the product.

Agile Development

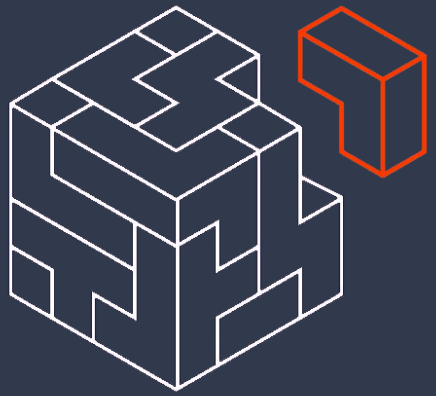


Agile development does save lot of resource which could have spent on something not needed anymore.


CONTINUOUS VISIBILITY
Clients
Developers
Users
Integrate & Test
Integrate & Test
Integrate & Test
START
Initiate Project
Define Requirements
Development Cycles (Add Functionality)
Review
Feedback
AGILE
DEVELOPMENT
APPROVE
YES
NO
RELEASE TO MARKET
Next Iteration
Adjust & Track
Record & Incorporate Changes

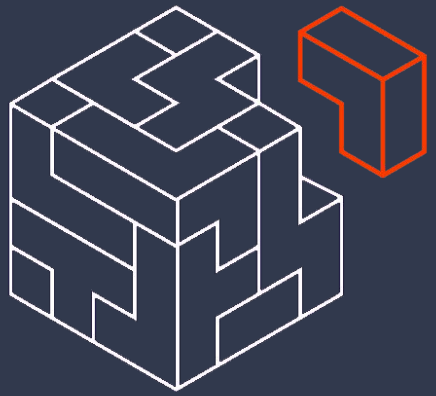
Microservices

Microservices-Oriented



Cloud native applications are built as a system of microservices.

Microservices-Oriented

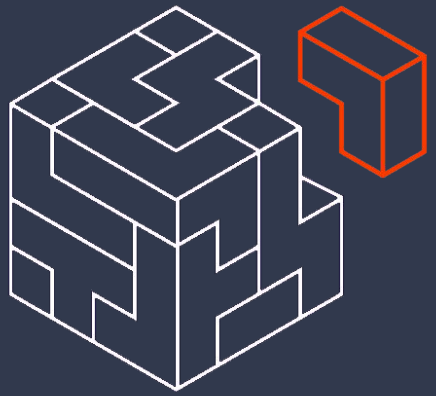


The general idea of this architectural style is to implement a system of multiple, relatively small applications.

These are called microservices.

They work together to provide the overall functionality of your system.

Microservices-Oriented

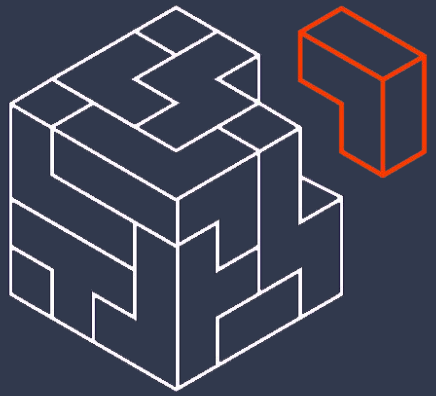


They work together to provide the overall functionality of your system.

- **Exactly one functionality**
- **Has a well-defined boundary and API (Application programming interface - used for communication)**
- **Gets developed and operated by a relatively small team**

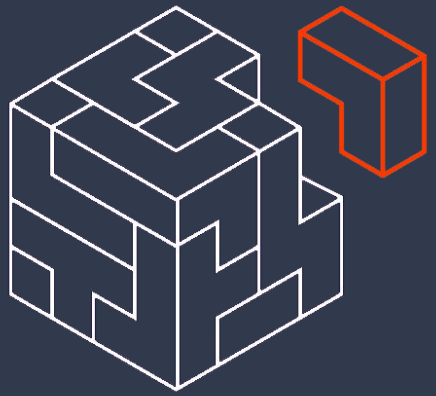
Benefits of Microservices

Microservices-Oriented



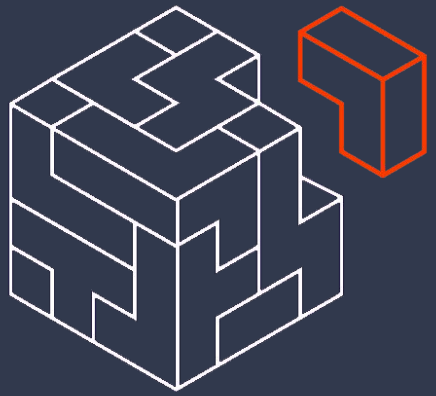
A lot easier to implement and understand a smaller application that provides one functionality, instead of building a large application that does everything.

Microservices-Oriented



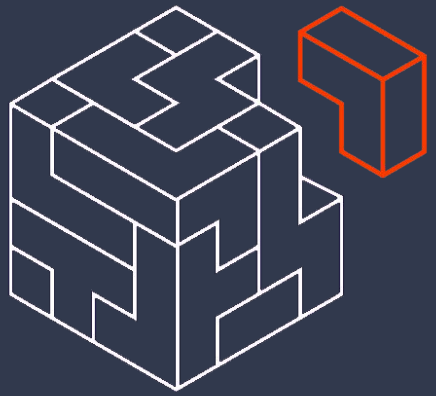
That speeds up development and makes it a lot easier to adapt the service to changed or new requirements.

Microservices-Oriented



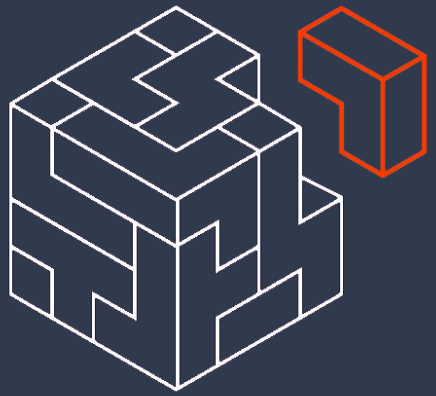
You need to worry a lot less about unexpected side effects of a seemingly small change, and you can focus on the development task at hand.

Microservices-Oriented



It also allows you to scale more efficiently.

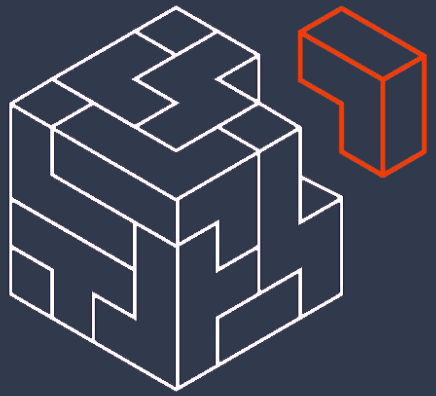
Microservices-Oriented



And even if you only use a small part of the monolith, you still need to acquire additional resources for the other, unused parts but in a cloud environment, you pay for the usage of hardware resources.

Challenges Using Microservices

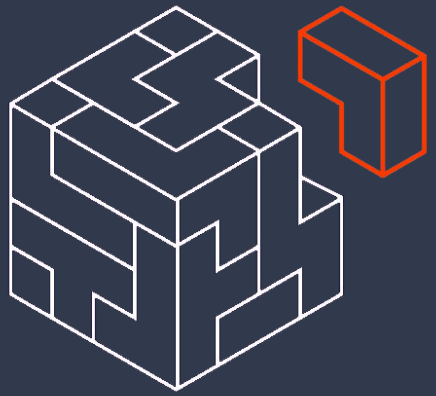
Microservices-Oriented



Microservices remove some complexity from the services themselves and provide better scalability, but you're now building a distributed system.

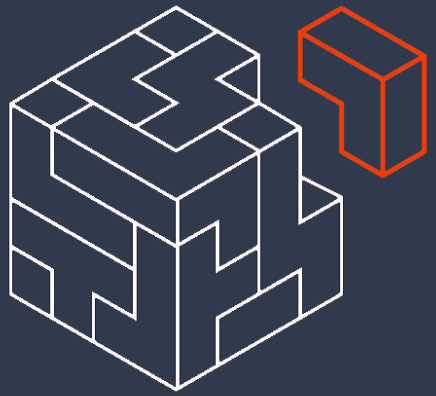
That adds a lot more complexity on the system level.

Microservices-Oriented



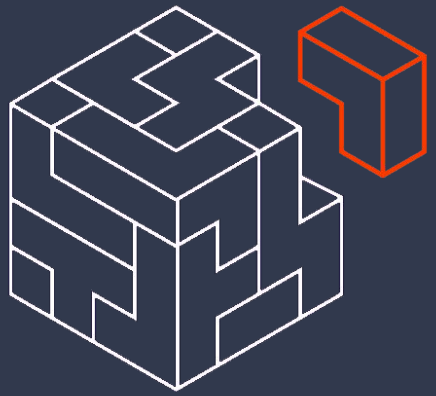
To make sure that dependent services find each other and communicate efficiently its a challenging task when number of microservices are many.

Microservices-Oriented



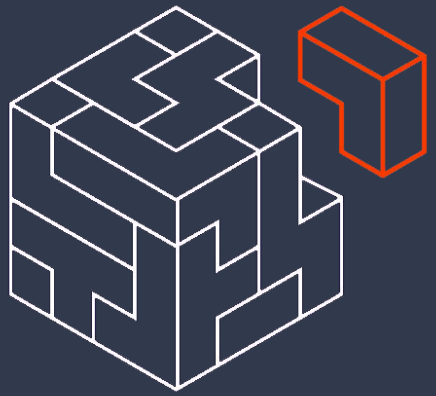
You also need to handle slow or unavailable services so that they don’t affect the complete system.

Microservices-Oriented



The distributed nature of your system also makes it a lot harder to monitor and manage your system in production.

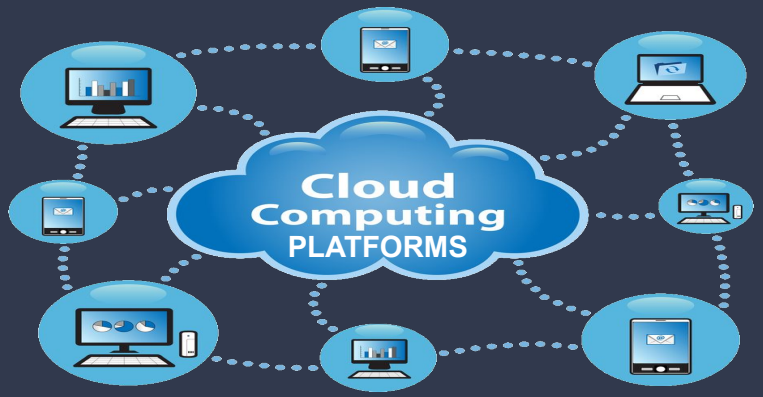
Microservices-Oriented



Instead of a few monoliths, you now need to monitor a system of microservices, and for each service, there might be several instances that run in parallel.

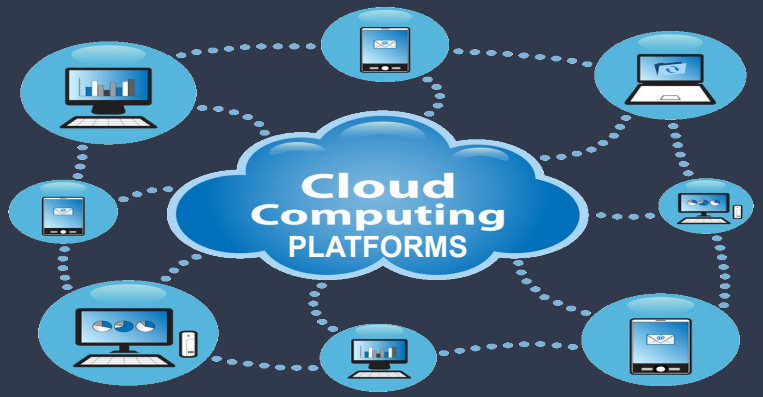
Cloud Computing Platforms

Cloud Computing Platforms



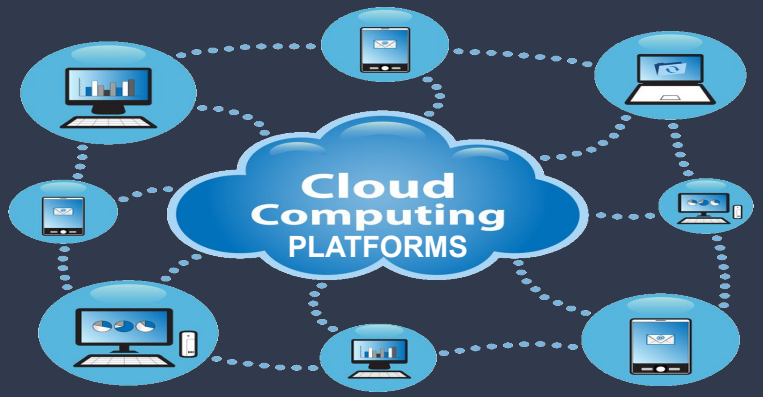
Cloud computing is the On-demand availability of computer system resources, especially data storage and computing power, without direct active management by the user.

Cloud Computing Platforms



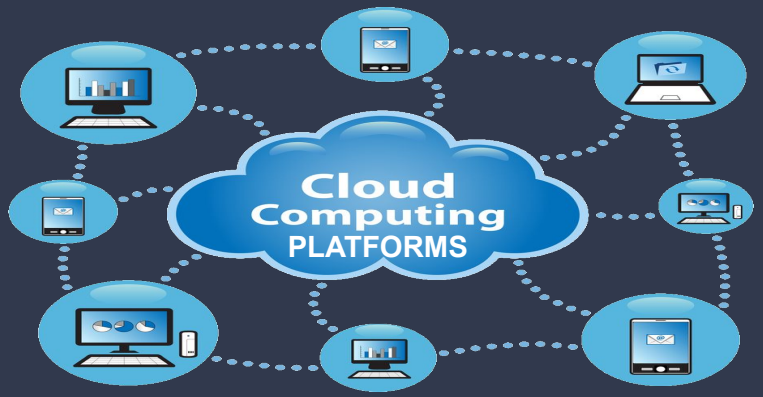
The term is generally used to describe data centers commercially available to many users over the Internet, they are Cloud Computing Platforms.

Cloud Computing Platforms



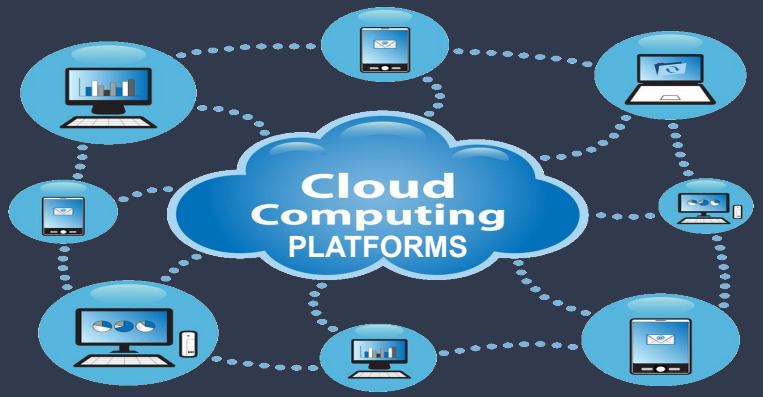
Large clouds, predominant today, often have functions distributed over multiple locations from central servers. If the connection to the user is relatively close, it may be designated an edge server.

Cloud Computing Platforms



An edge server also called content delivery network or content distribution network (CDN) is a geographically distributed network of proxy servers and their data centers. The goal is to provide high availability and high performance by distributing the service spatially relative to end-users.

Cloud Computing Platforms



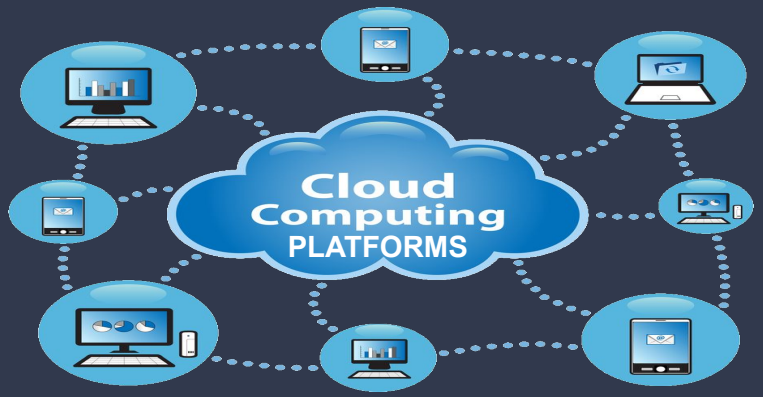
Clouds may be limited to:

A single organization (enterprise clouds)

Be available to many organizations (public cloud)

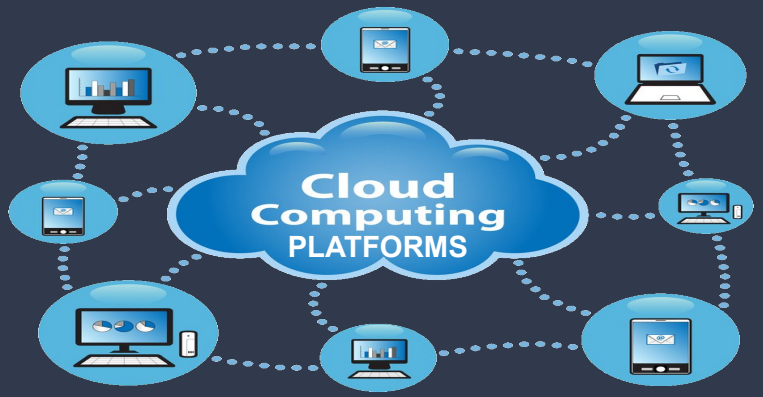
A combination of both (hybrid cloud)

Cloud Computing Platforms



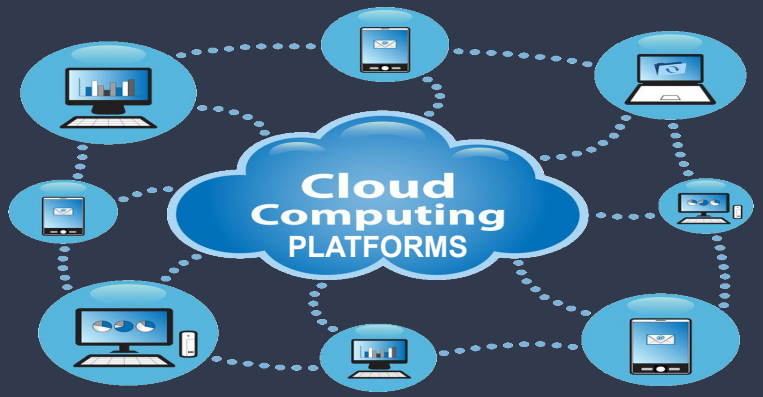
The largest public cloud is Amazon AWS. There are many others like Google cloud, microsoft Azure, Alibaba cloud, IBM cloud etc

Cloud Computing Platforms



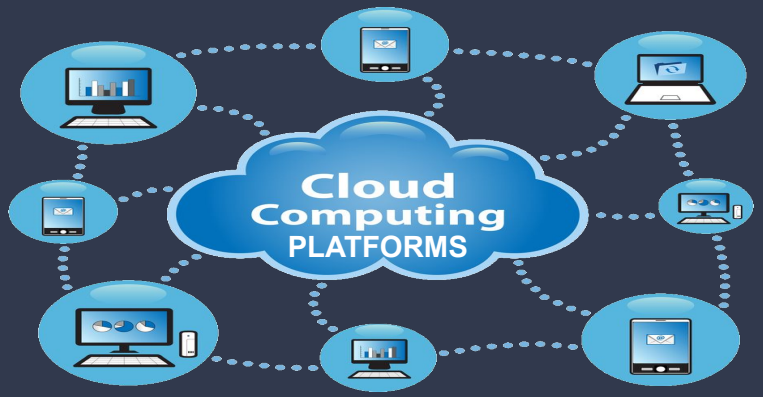
Advocates of public and hybrid clouds note that cloud computing allows companies to avoid or minimize up-front IT infrastructure costs.

Cloud Computing Platforms



Experts also claim that cloud computing allows enterprises to get their applications up and running faster, with improved manageability and less maintenance, and that it enables IT teams to more rapidly adjust resources to meet fluctuating and unpredictable demand.

Cloud Computing Platforms



Cloud providers typically use a "pay-as-you-go" model.

Containers

Containers

A Container is a runtime instance of an image.

Making a container using image for any application is called Containerization.

Docker is widely use to containerize your application.

Containers

An Image is an executable package that includes everything needed to run an application.

- **Code**
- **Runtime**
- **Libraries**
- **Environment variables,**
- **Configuration files**

Containers Are

Containerization is increasingly popular because containers are:

- **Flexible**
- **Lightweight**
- **Interchangeable**
- **Portable**
- **Scalable**
- **Stackable**

Flexible

Even the most complex applications can be containerized.

Lightweight

Containers leverage and share the host kernel.

Interchangeable

You can deploy updates and upgrades on-the-fly.

Portable

You can build locally, deploy to the cloud, and run anywhere.

Scalable

You can increase and automatically distribute container replicas.

Stackable

You can stack services vertically and on-the-fly.

Orchestration

Orchestration

Deploying your application with all dependencies into a container is just the first step.

Scaling apps based on the current load of your system isn't that easy.

NOT SO EASY.

Orchestration

Monitor your System

1

Orchestration

Trigger the Startup or Shutdown of a Container

2

Orchestration

Ensure all required configuration parameters are in place

3

Orchestration

Balance the load between active application instances

4

Orchestration

Share authentication secrets between Containers

5

Orchestration

Doing all of this manually requires a lot of effort and is too slow to react to unexpected changes in system load.

KUBERNETES

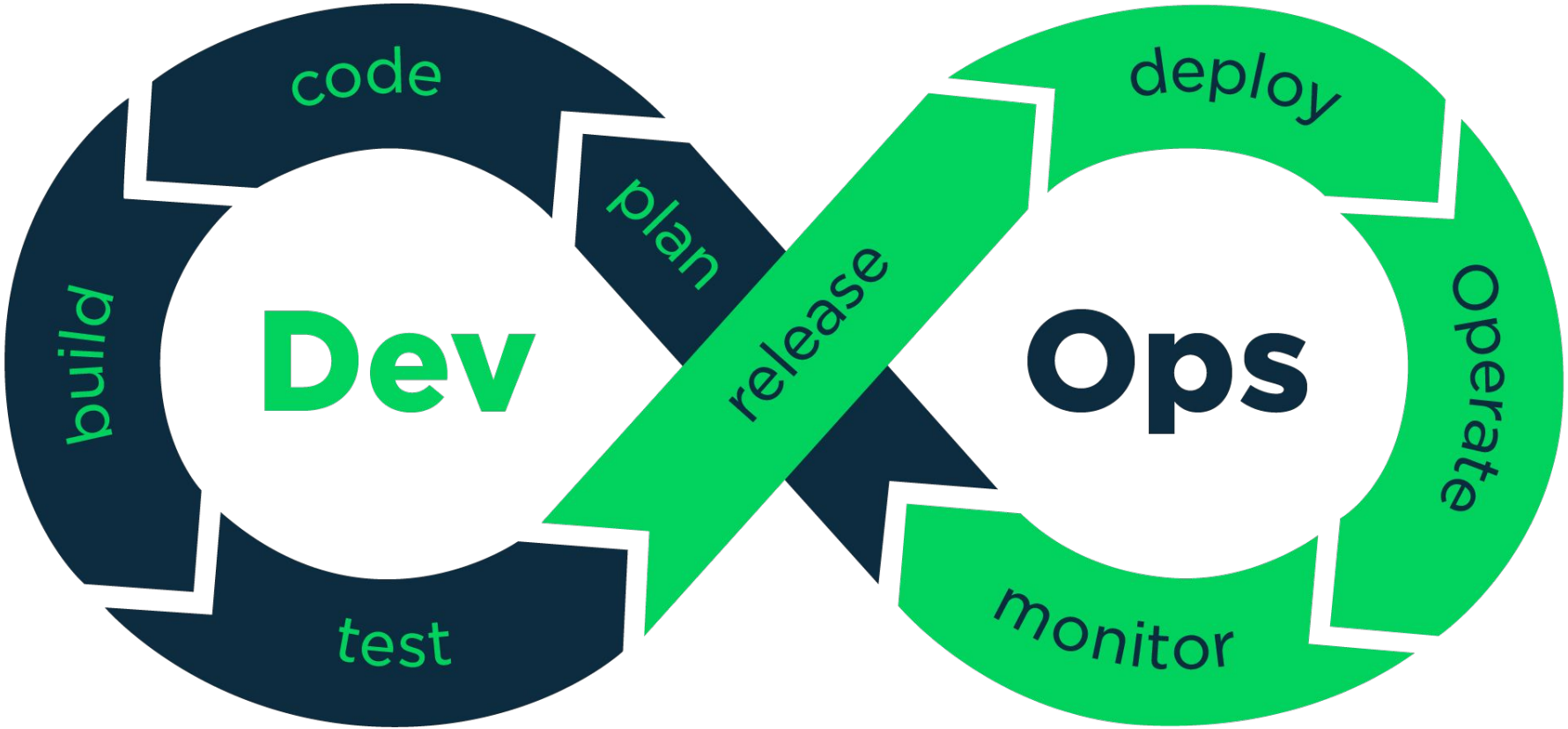
Continuous Integration
Continuous Deployment

CI/CD

A method to frequently deliver apps to customers by introducing automation into the stages of app development.

Ongoing Automation.

Continuous Monitoring.


code
plan
build
Dev
test
release
deploy
operate
Ops
monitor

CI/CD Pipeline

Taken together, these connected practices are often referred to as a “CI/CD pipeline”

Jenkins

Questions?

Next: Linux for Beginners