

# IT Fundamentals

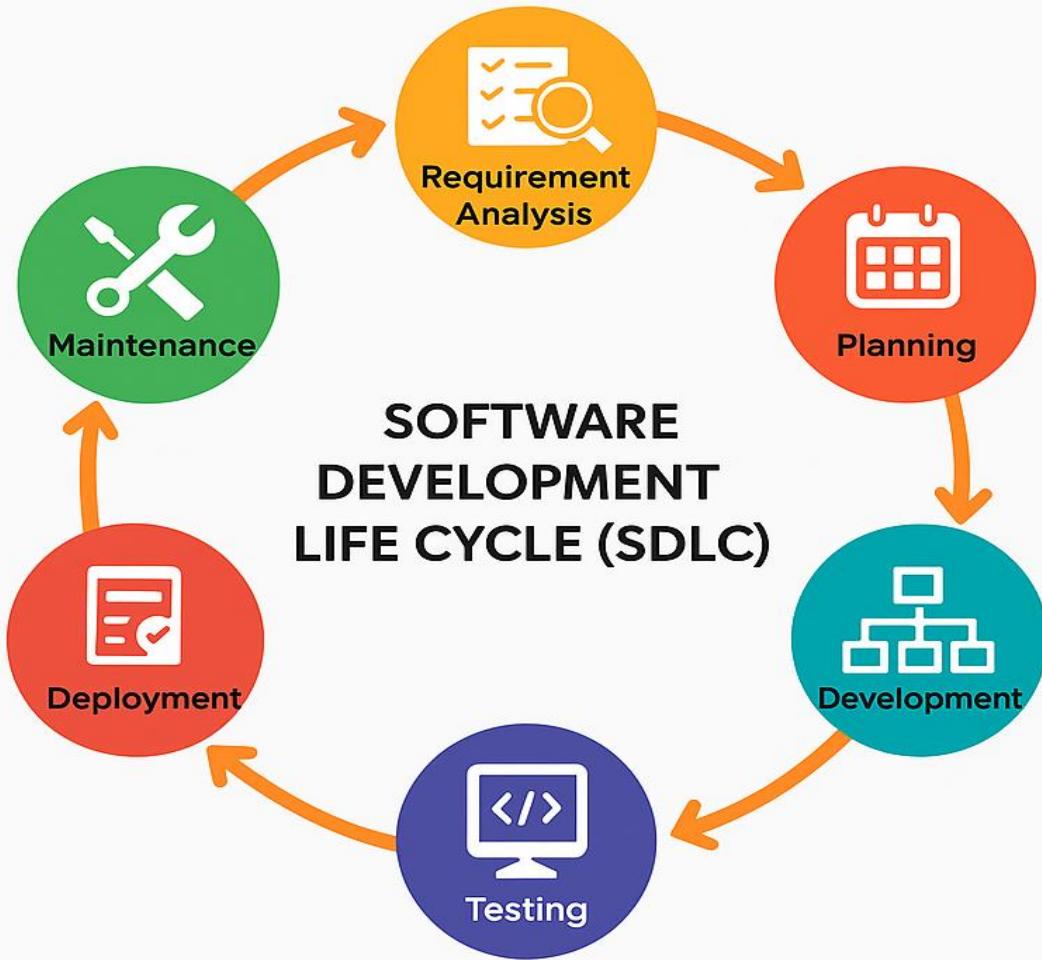
SDLC , CLOUD ARCHITECTURE , GIT/GITHUB

By  
Anirudh  
Yashwanth  
Bhargav

# Introduction

- SDLC stands for Software Development Life Cycle
- It is a set of planned steps used to develop any software product.
- SDLC divides the entire software development process into different phases, which:
  - 1.Improves accuracy
  - 2.Ensures all requirements are satisfied
  - 3.Makes maintenance easier
- Without SDLC, it becomes difficult for project managers to:
  - 1.Track progress
  - 2.Monitor project status

# SOFTWARE DEVELOPMENT LIFE CYCLE (SDLC)



# Requirements

- This is the first and most critical phase of the SDLC
- Business Analyst (BA) interacts with the client
- Client needs and expectations are carefully collected
- Requirements are analyzed and clarified
- After analyzing, a document is prepared called BRS (Business requirement specification)

## **From Requirement Analysis to Software Requirement Specification**

After requirement analysis, an SRS document is prepared.

SRS is created based on the Business Requirement Specification document.

It contains complete requirements of the software to be designed and developed.

## **2. Planning**

- The Planning Stage also critical step of the Software Development Life Cycle (SDLC).
- It's the phase where we figure out exactly what software we are going to build, why we are building it, and also understand even it is possible or not.
- It also include the factor like time, cost, and technology.
- By this we can understand that how the software is built.

## **3. System Design**

In this we have:

1. High Level Design (HLD)
2. Low Level Design (LLD)

### **1. High Level Design (HLD)**

HLD explains WHAT modules are there and HOW they are connected. It does not contain any coding.

## **HLD Contains**

- 1.System architecture
- 2.Major modules
- 3.Data flow between modules

## **2. Low Level Design (LLD)**

- LLD explains HOW each module will work internally.
- It provides detailed design information that helps developers start coding easily.

## **Development**

- In this stage, coding of the software is done based on the design document(DDS).
- DDS helps developers write code easily and correctly.
- Developers follow coding standards and use tools like compilers and debuggers.
- Programming languages such as C, C++, Java are used depending on the software.

## **Testing**

This stage refers to the testing of the software where software's defects are reported, tracked, and retested until the product reaches the quality standards defined in the SRS (Software Requirement Specification) Testing can be done manually or by using automation tools

## **Deployment**

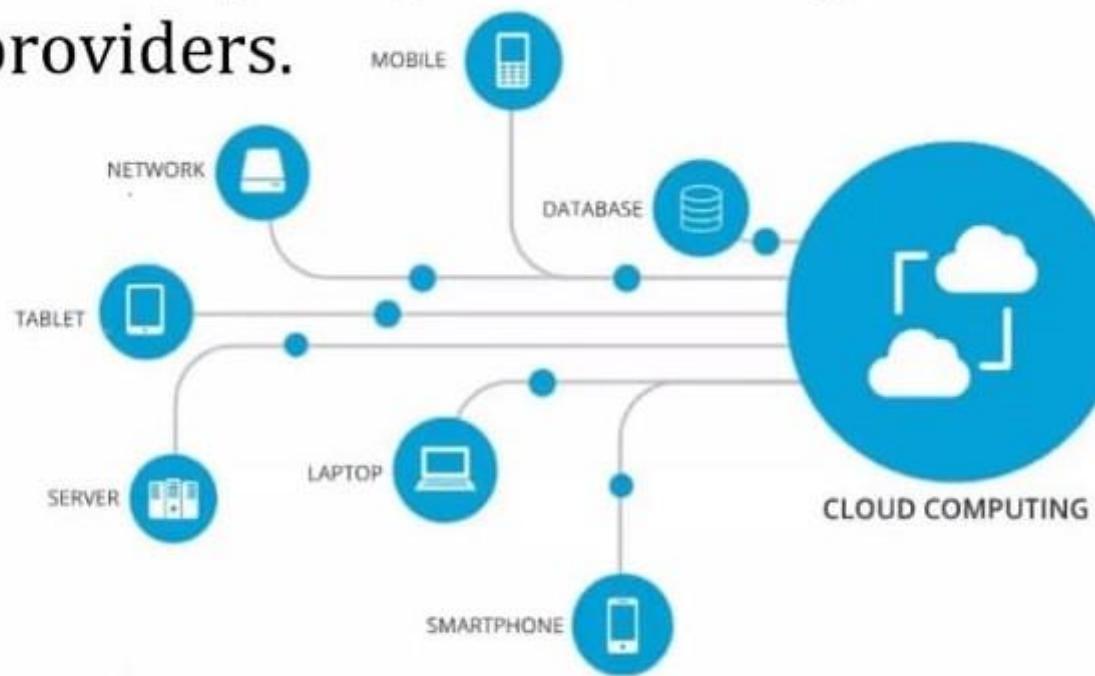
Once the software product is tested and ready to be deployed, it is released in the market.

## **Maintenance**

Once the product is deployed in the market the client may want to update or want to do some changes to the product then the maintenance part does this task .Also,when product users use the software they may give feedback which should be resolved from time to time

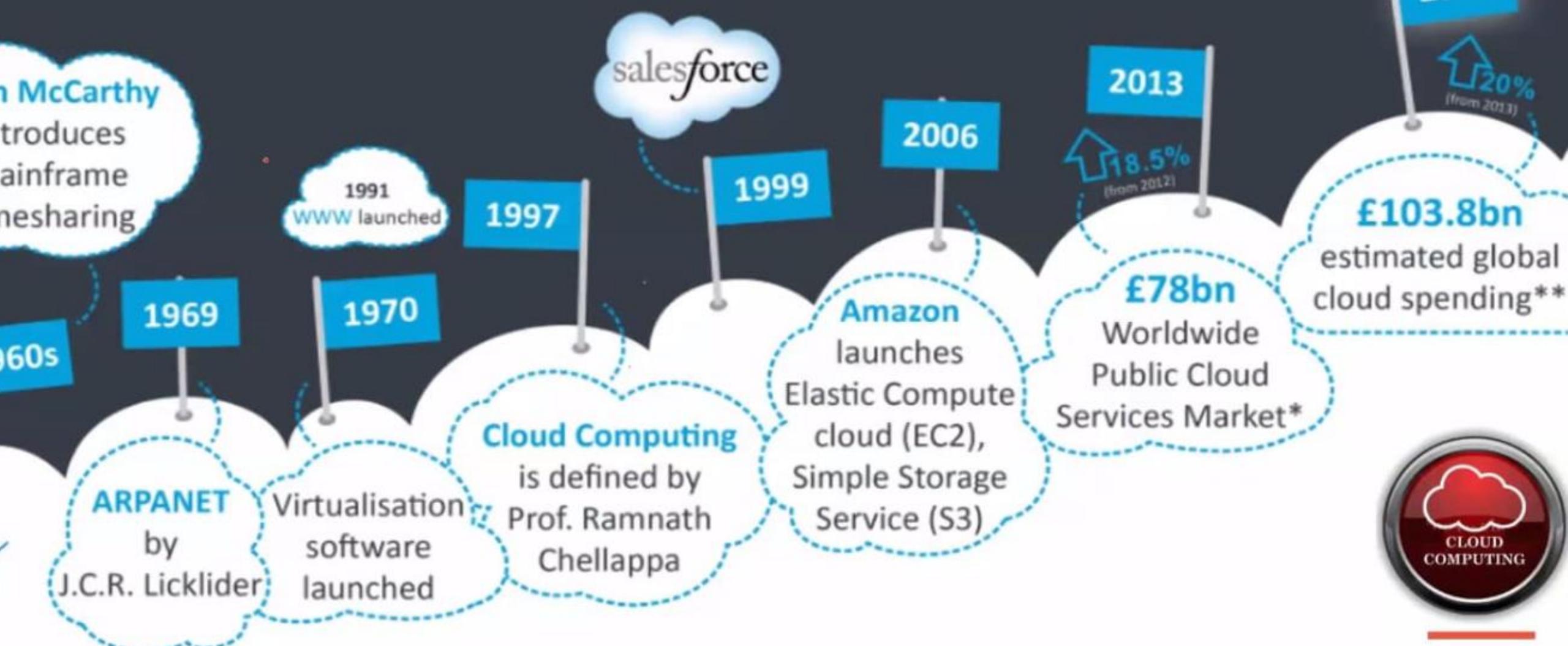
# Cloud Computing - Definition

Cloud computing is the on-demand **delivery of IT assets over the Internet** with **pay-as-you-go** pricing. Instead of owning, buying, and maintaining physical data centres and servers, you can access technology services, such as compute power, storage and database as per the need from a cloud providers.

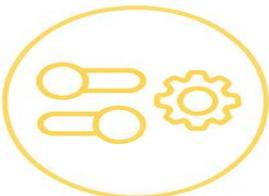


# Cloud Computing - History

## THE HISTORY OF THE CLOUD

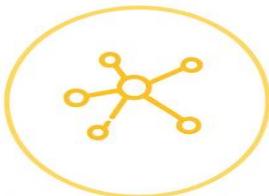


# Cloud Computing - Characteristics



On-demand self-service

No human intervention needed to get resources



Broad network access

Access from anywhere



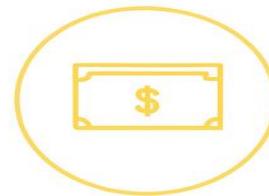
Rapid elasticity

Get more resources quickly as needed



Resource pooling

Provider shares resources to customers

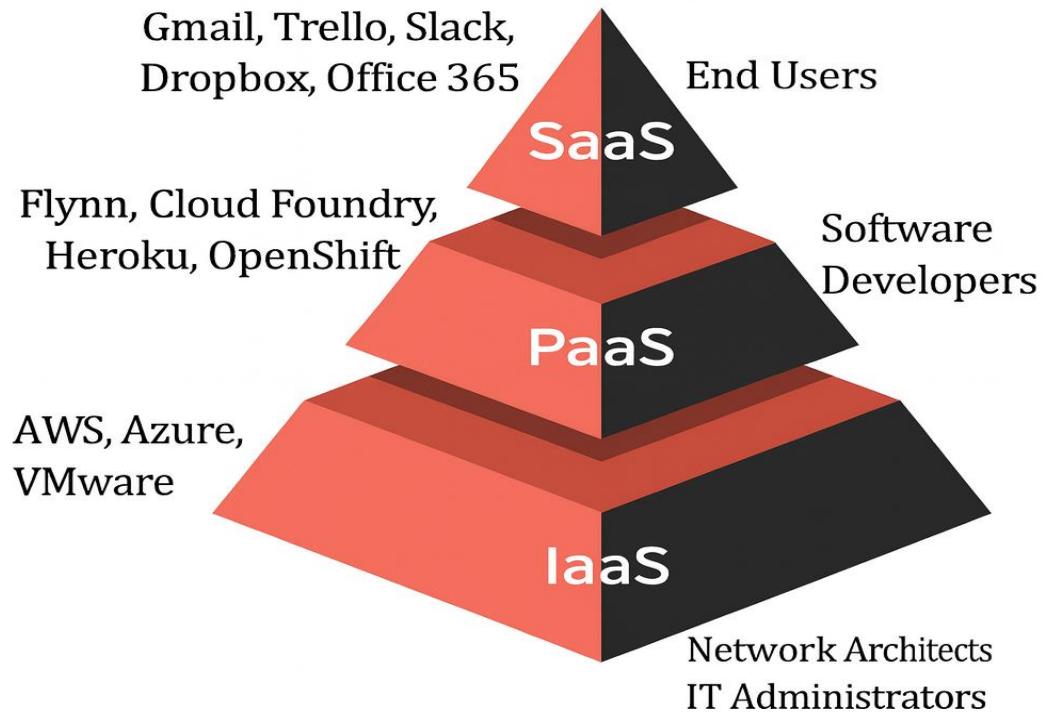


Measured service

Pay only for what you consume

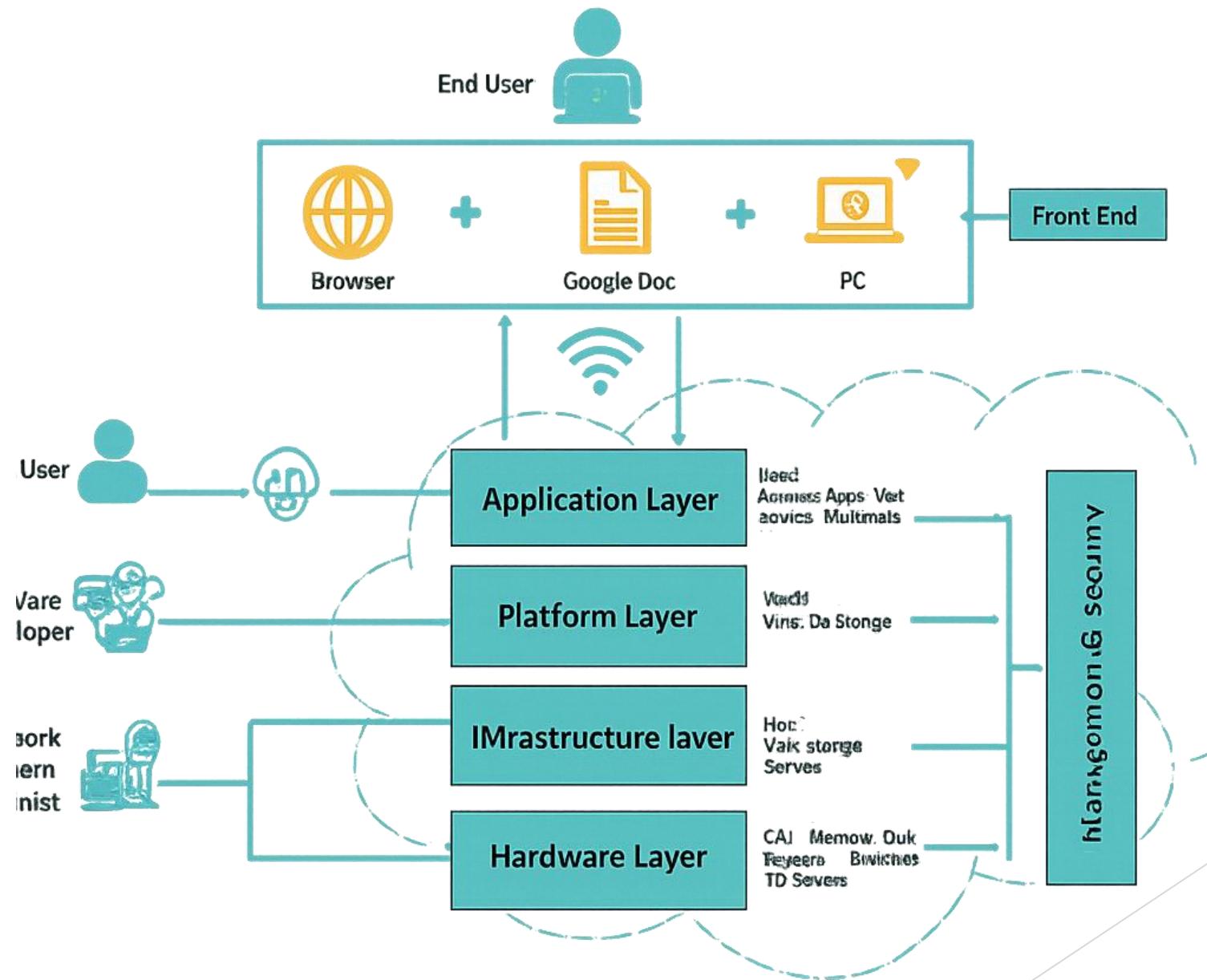


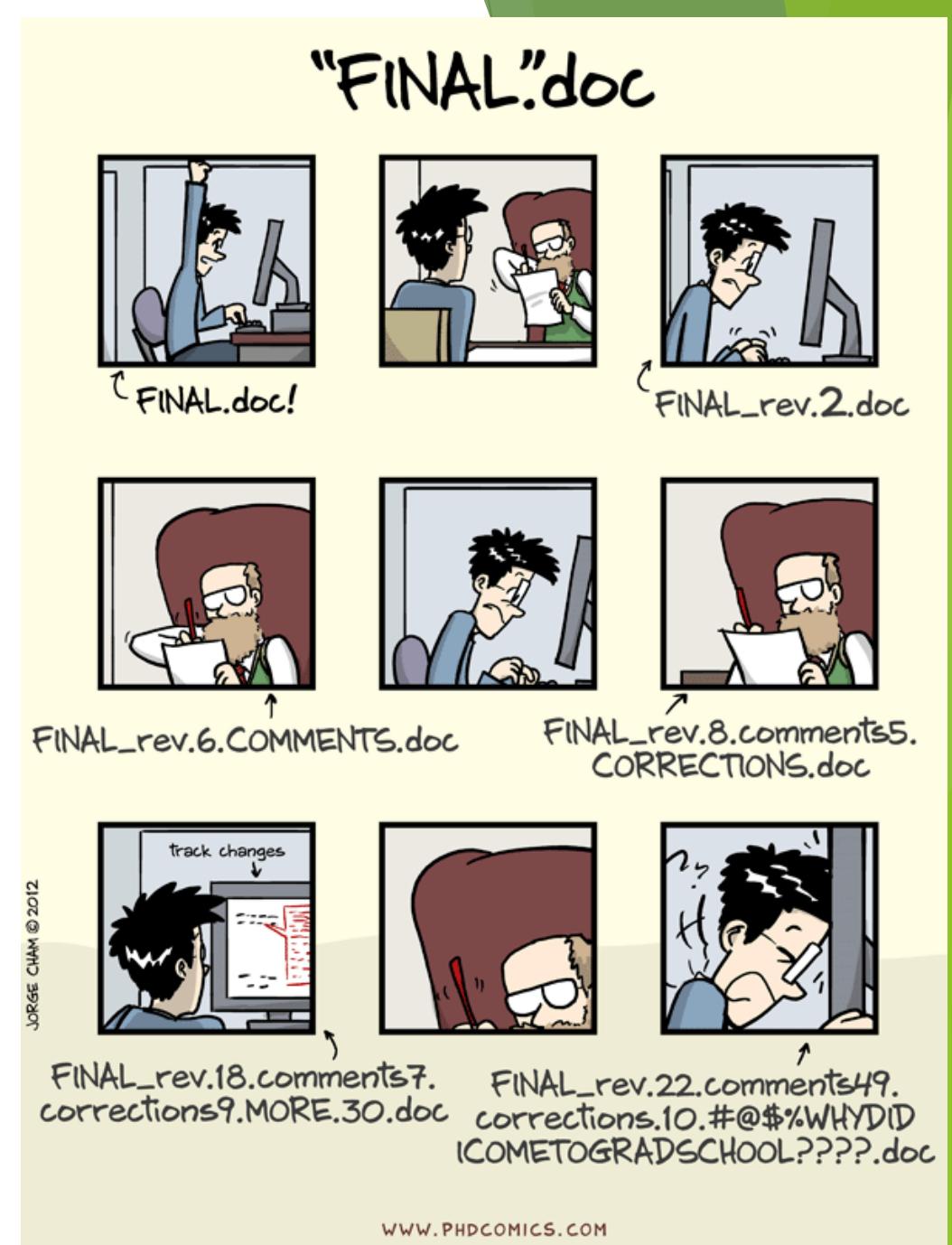
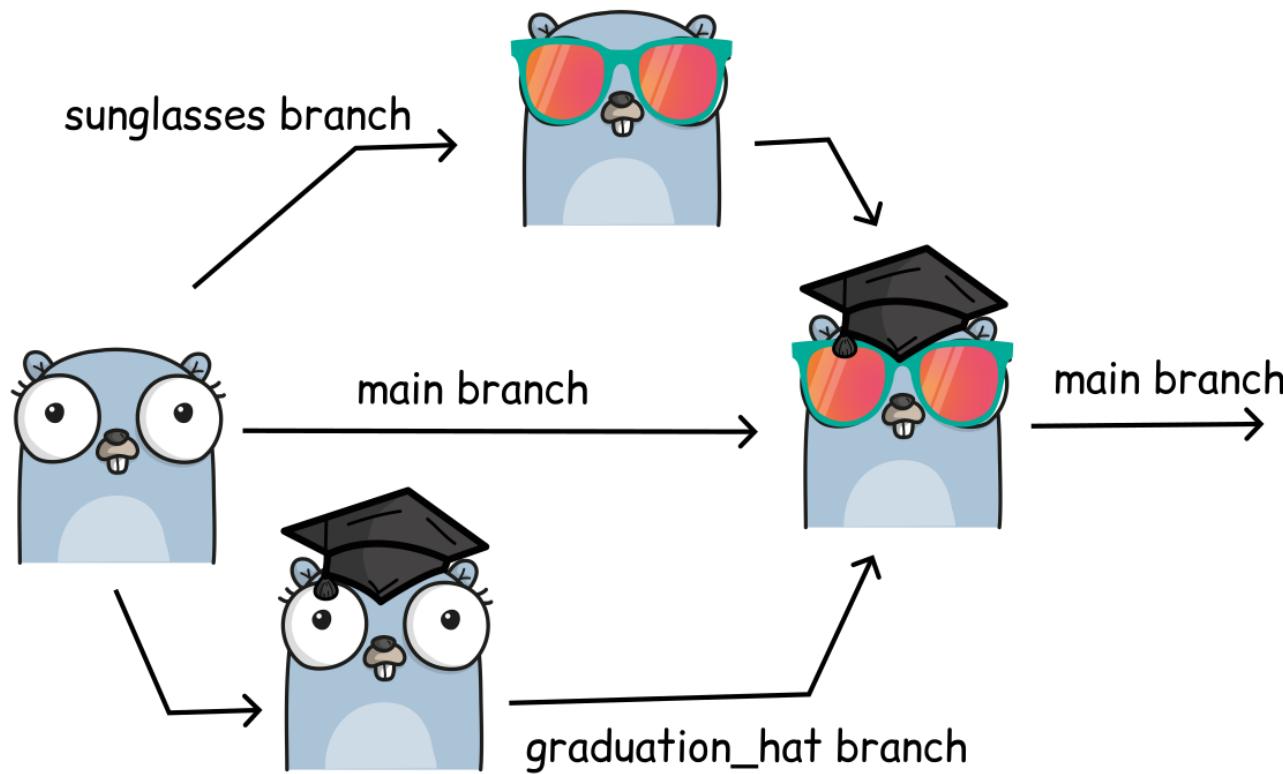
# Cloud Computing - Services



- 1. IaaS – Infrastructure as a Service**
- 2. PaaS – Platform as a Service**
- 3. SaaS – Software as a Service**

# Cloud computing architecture overview





# Git and GitHub Basics for Teams

Commits, Branches, Remotes, Pull Requests, and Permissions



## 1 Commits

Make atomic changes with clear messages; link commits to issue IDs for traceability.



## 2 Branches

Use short-lived feature branches; name with feature, bugfix, hotfix, release prefixes.



## 3 Remotes

Use origin push and pull; enforce protected branches on remote for safety.



## 4 Pull Requests

PRs are the collaboration unit: include description, linked issues, CI status, reviewers.

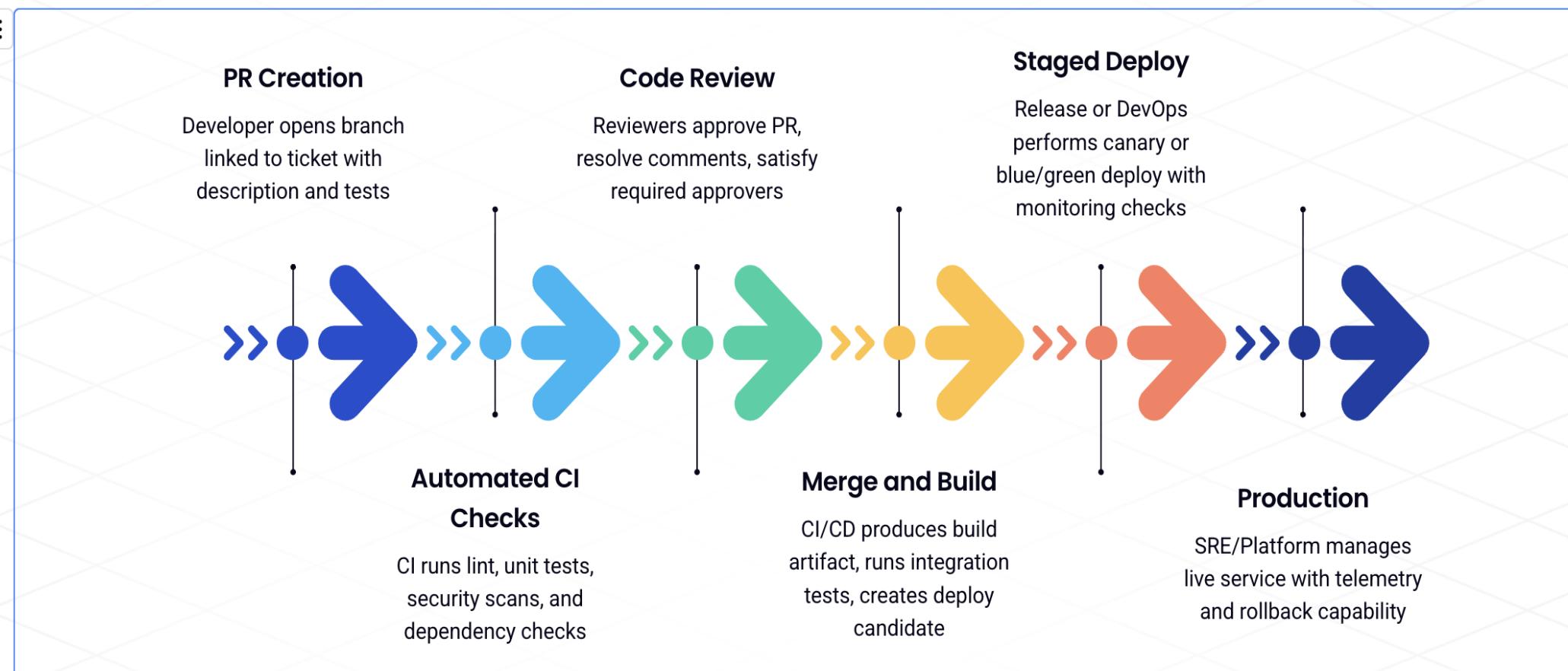


## 5 Permissions model

Roles: Admin, Write, Read; use Teams and fine-grained access for repos and environments.

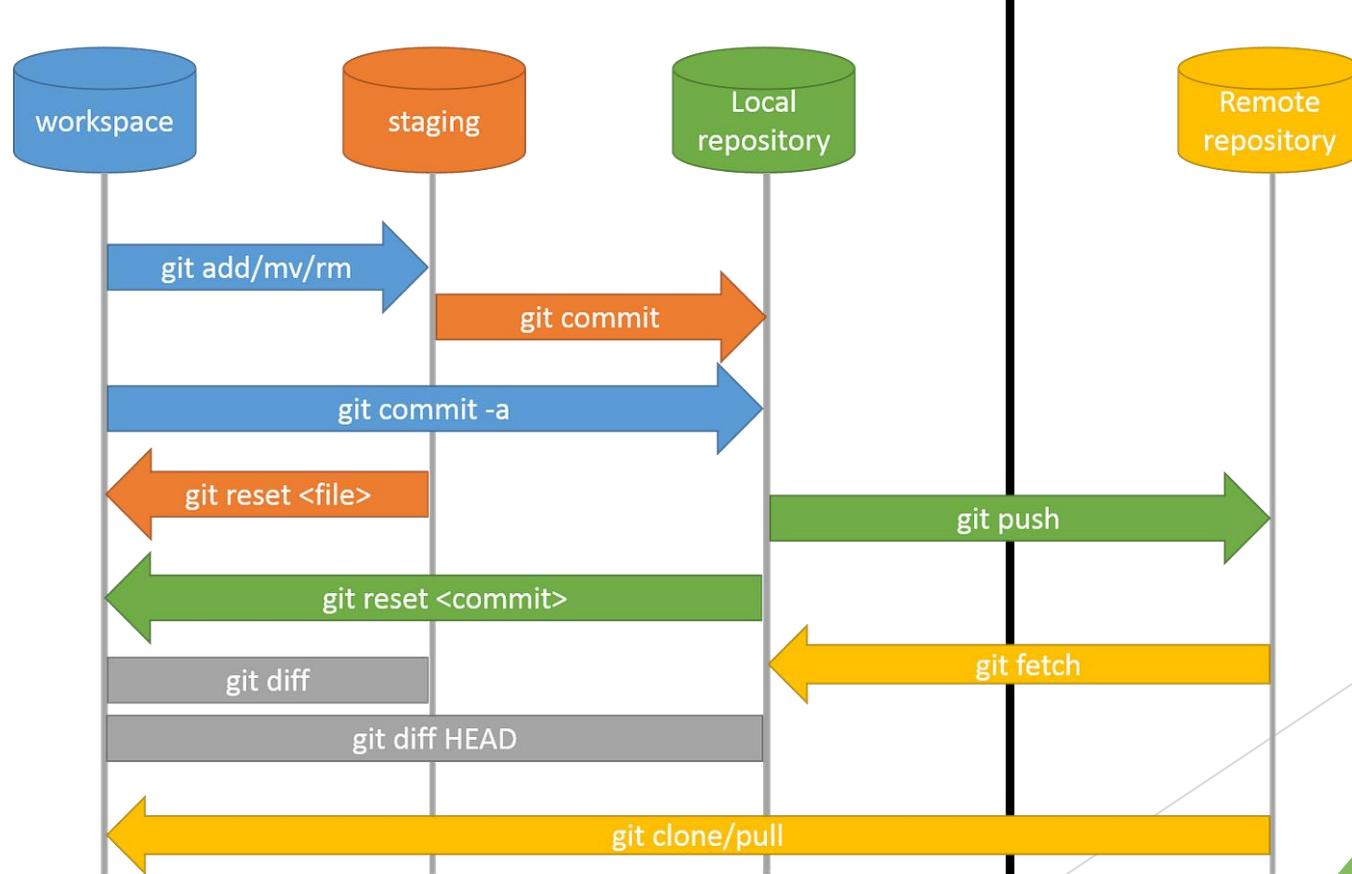
# Code Review, CI/CD, and Automation: From PR to Production

Practical pipeline steps, gating policies, and operational ownership



# GitHub – Cloud Repository Platform

- Hosts Git repositories online
- Enables team collaboration
- Acts as a **central code hub**



# Integrating SDLC, Cloud, and Git for End-to-End Traceability

Clear links from requirement to production with audit-ready evidence

