DevOps Introduction

By:
Bipin Sinhaa

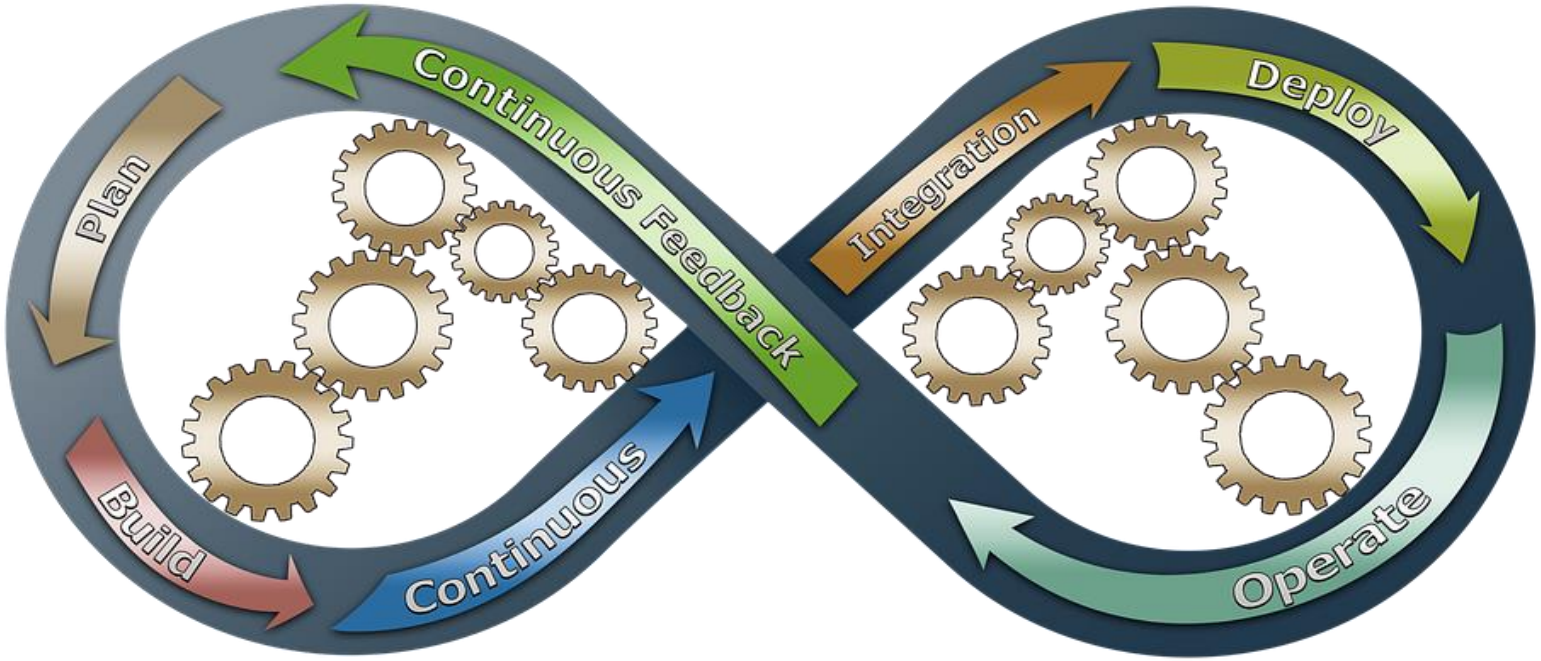
Index

What is Cloud ?

Example:

Build, Skill and Enable

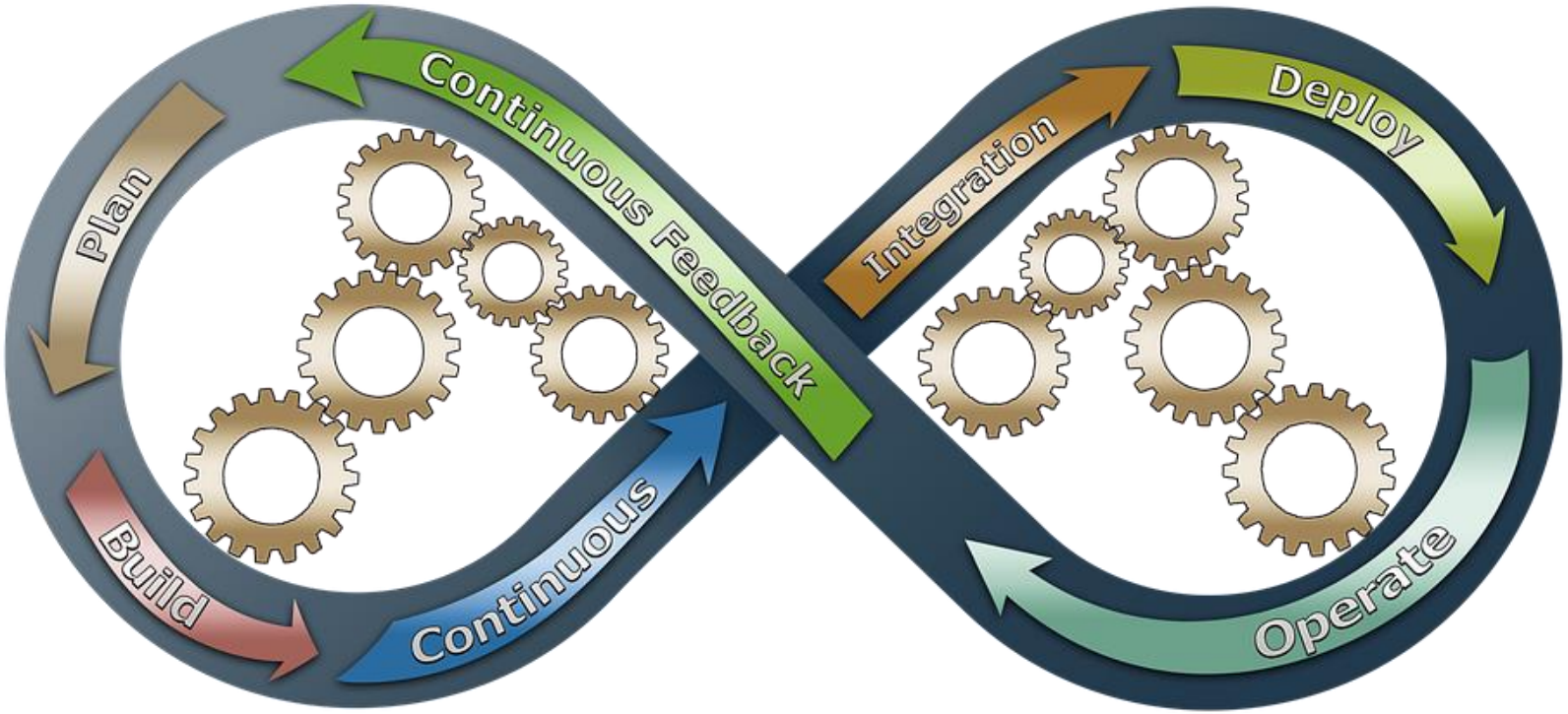
DevOps : Introduction



Build, Skill and Enable

What is DevOps

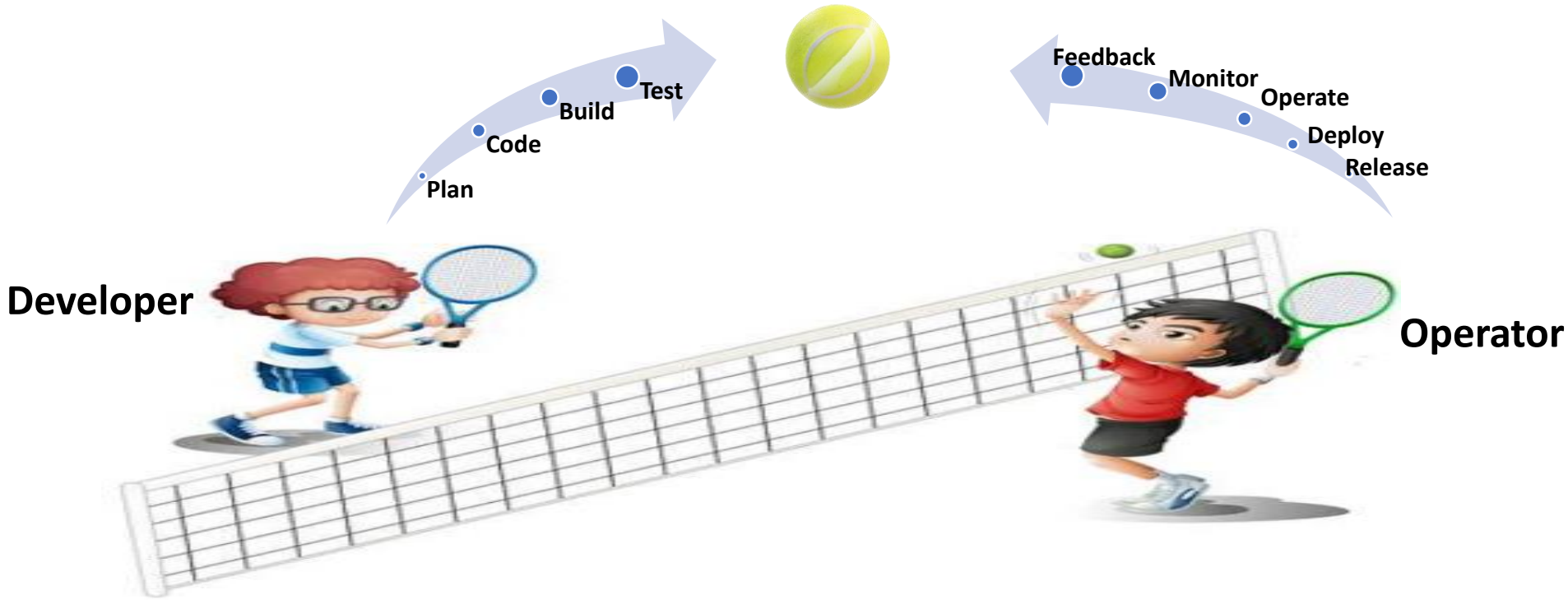
Set of practices which is intended to reduce the time between committing a change to a system and the change being placed into normal production, while ensuring high quality



Build, Skill and Enable

What is DevOps

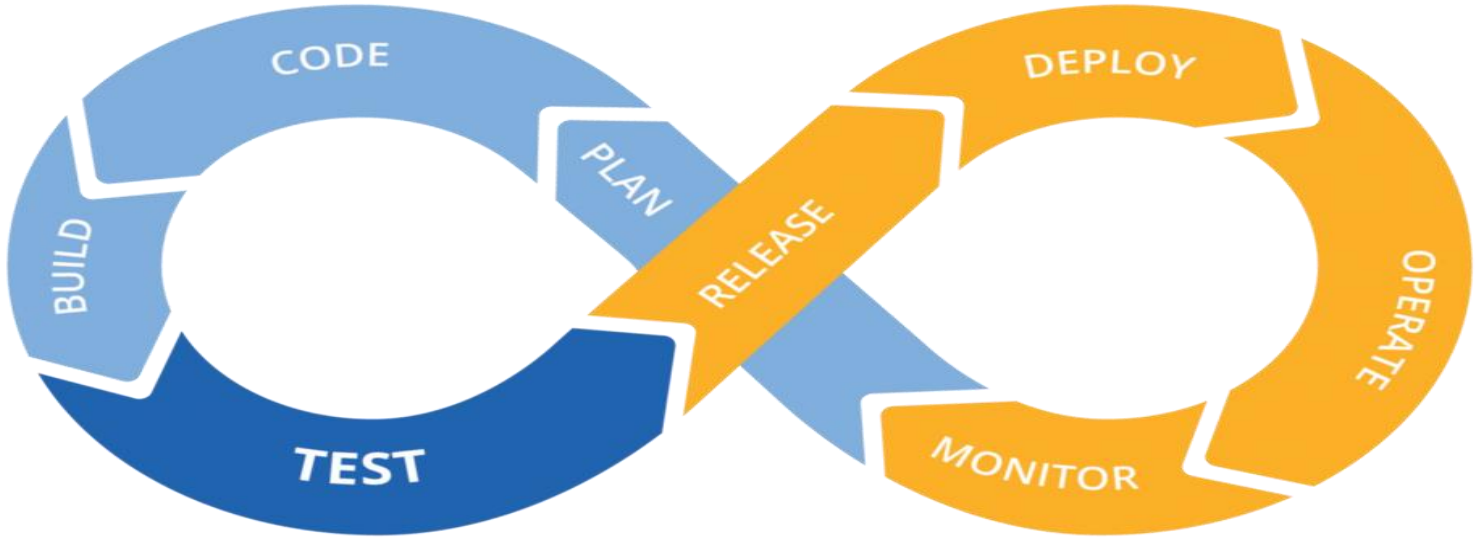
Software development is usually a continues struggle between the developer and the Operator.



Build, Skill and Enable

What is DevOps

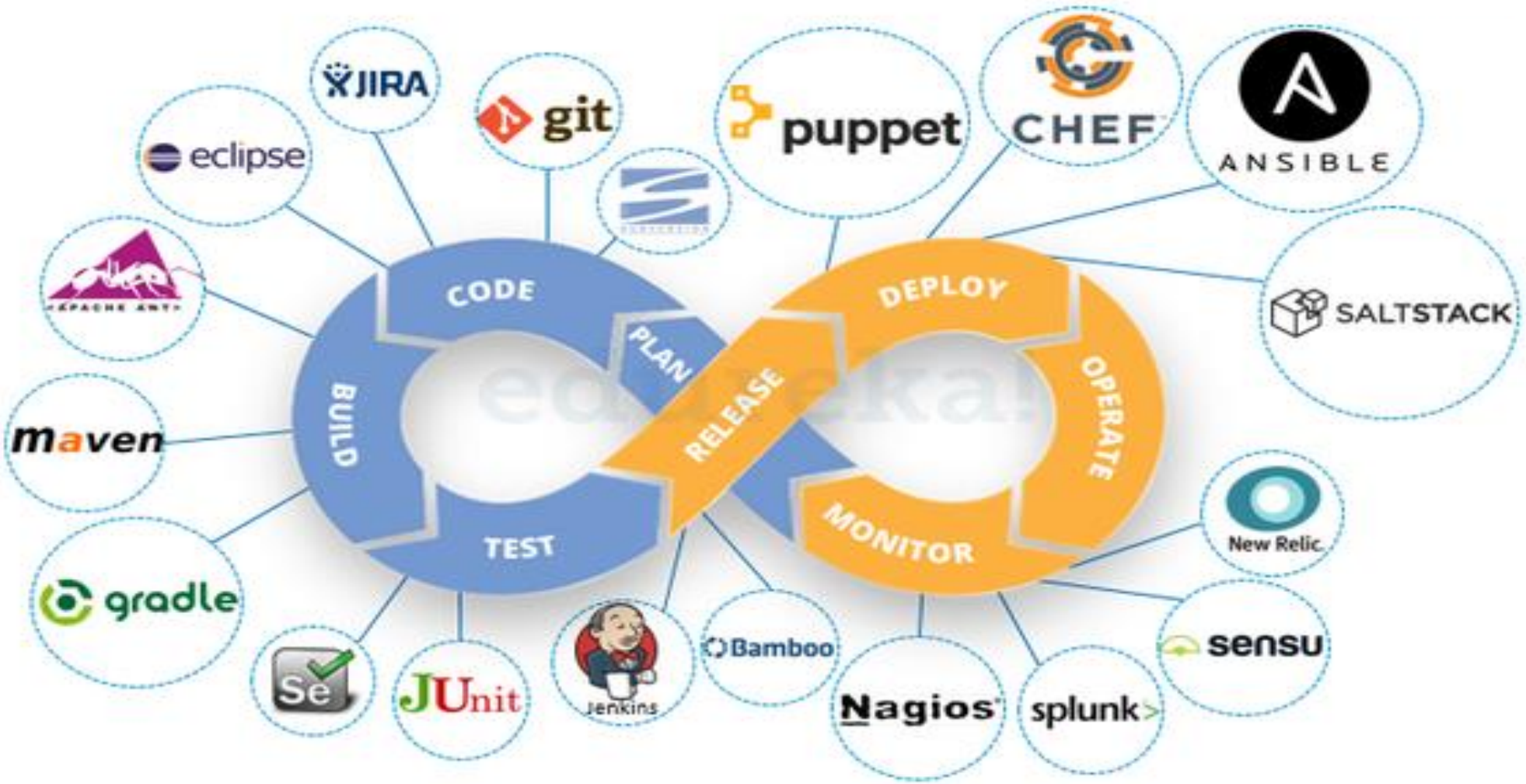
DevOps is a Software Development approach which involves a continuous process of Development, Testing, Integration, Development and Monitoring of the software throughout its development life cycle



DevOps : Tools

Build, Skill and Enable

DevOps Tools



Build, Skill and Enable

DevOps Tools

GitHub is a code hosting platform for collaboration and version control. GitHub lets you (and others) work together on project

Docker is a tool designed to make it easier to create, deploy, and run applications by using containers. Containers allow a developer to package up an application with all of the parts it needs, such as libraries and other dependencies, and ship it all out as one package.

DevOps Tools

Selenium is a portable software-testing framework for web applications. Selenium provides a playback tools for authoring tests without the need to learn a test scripting language.

Jenkins is an open source automation server written in Java, Jenkins helps to automate the non-human part of software development process, with continuous integration and facilitating technical aspects of continuous delivery.

DevOps Tools

Continuous Development

Continuous Deployment

Continuous Integration

Continuous Testing

Continuous Monitoring

Build, Skill and Enable

What is Continuous Integration and Delivery

Step 01

Split the entire chunk of codes into segments

Step 02

Keep small segment of manageable codes

Step 03

Integrate the segmented code, multiples times a day

Step 04

Adopt a continuous integration methodology to coordinate with your team

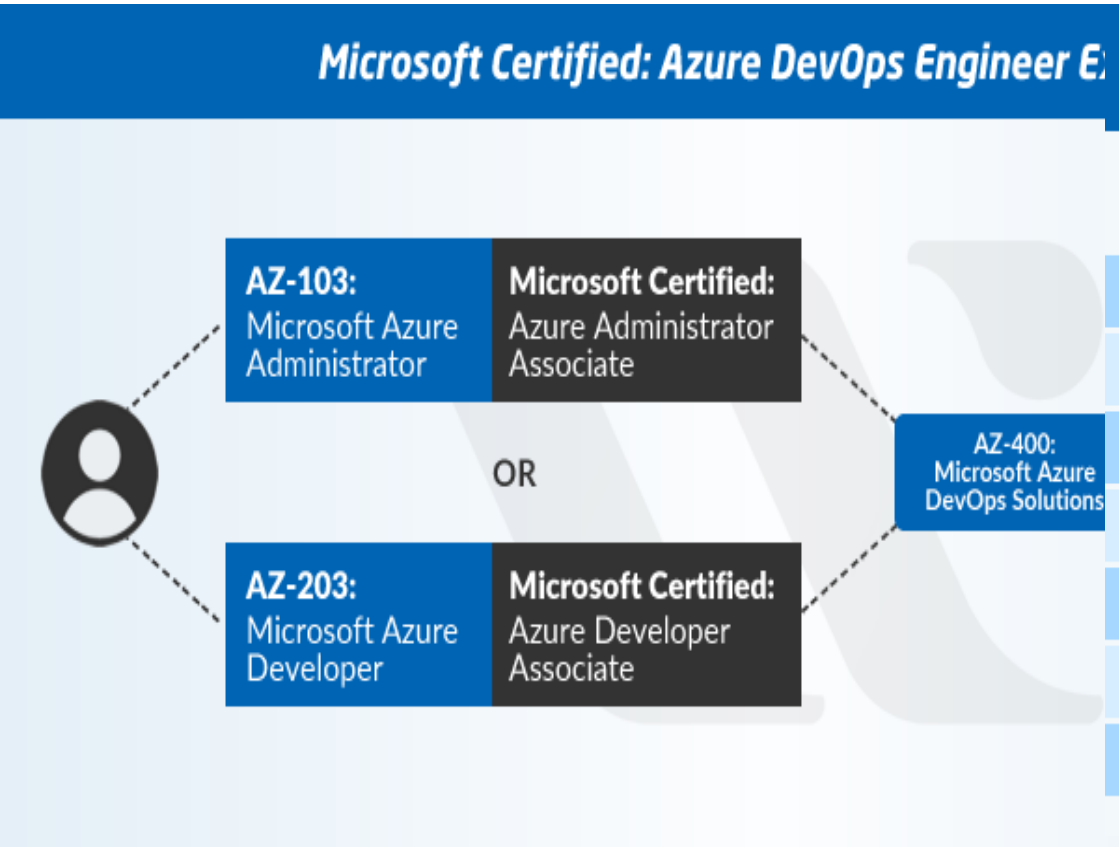
Build, Skill and Enable

Build, Skill and Enable

. AWS Certified DevOps Engineer Professional Exam

Build, Skill and Enable

Azure DevOps Engineer



Domains Covered in AZ-400 Exam

Domain	Weightage of Questions
Design a DevOps Strategy	20-25%
Implement DevOps Development Process	20-25%
Implement Continuous Integration	10-15%
Implement Continuous Delivery	10-15%
Implement Dependency Management	5-10%
Implement Application Infrastructure	15-20%
Implement Continious Feedback	10-15%

Build, Skill and Enable

Docker Certified Associates

Your First DevOps Tool
Git & GitHub

Build, Skill and Enable

Self Introduction – Sample

My Name is Peter . I have 4+ years of experience in IT industry and around 1.5 years of experience In DevOps and AWS.

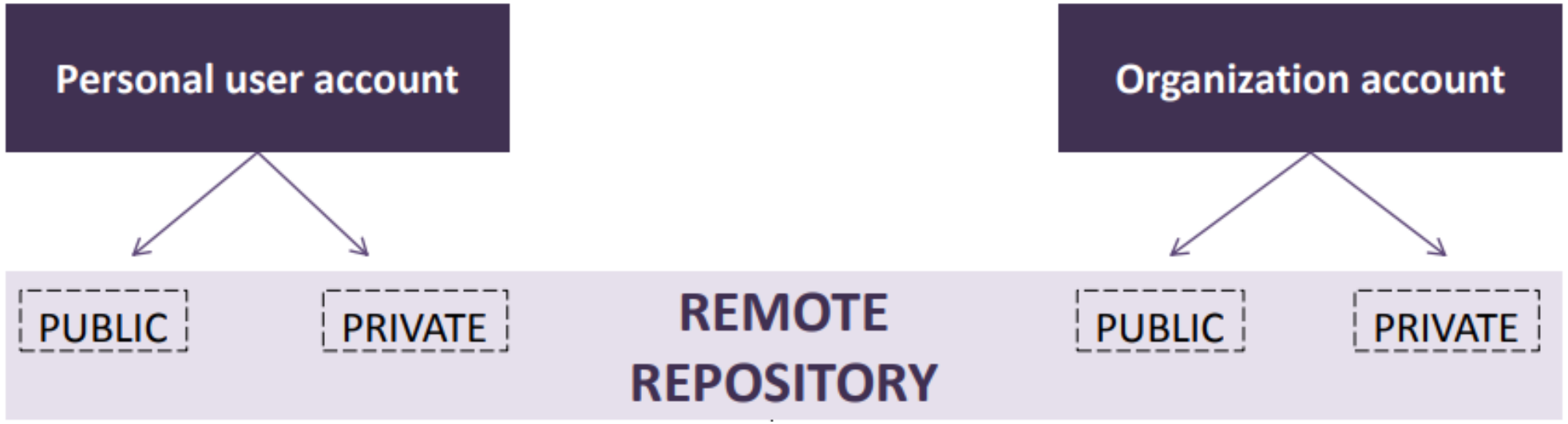
Currently I am working with TechPledge Consulting Pvt Ltd as Freelance DevOps Engineer.

As a part of my role , I am responsible to setup & manage DevOps Ci/CD pipelines.

I have used various DevOps tool like Git (for Version control system),Jenkins (for continues integration),Maven (as a Build tool) , Ansible (for continues Deployment and Configuration Management) and Docker (for Containerization).

My DevOps environment is running on AWS cloud which was setup by me according to our DevOps Architect plan.

GitHub Structure



GitHub –Important Terminology

Creating a repo

Creating a repository for multiple people to work together Master in a repository This is the final version that is considered ready to use by anybody in the team or outside if repository is public.

Creating a Branch

Create a branch in your project, for an environment where you can try out new ideas. Changes you make on a branch don't affect the master unless pull request is accepted.

Adding Commits

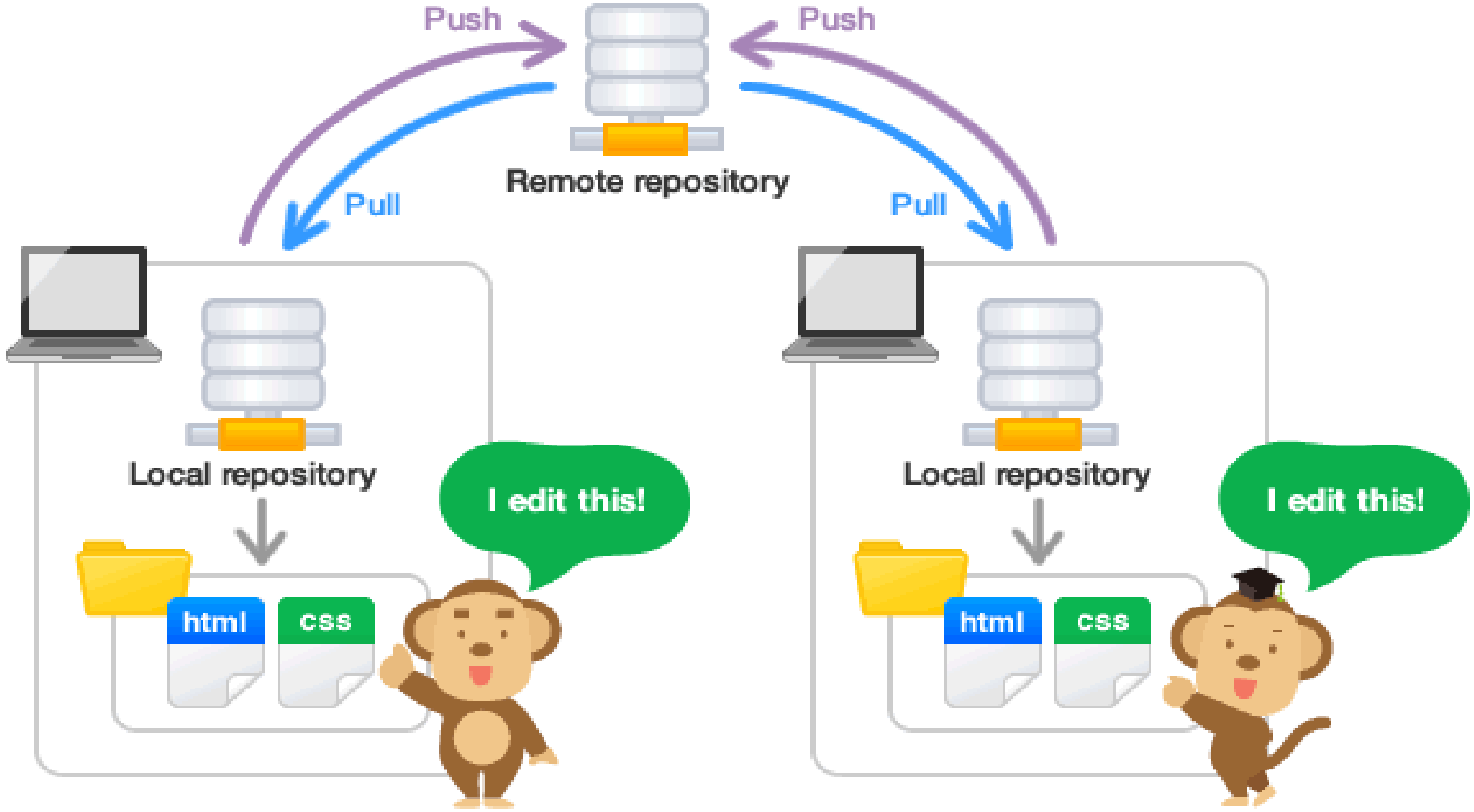
Keeps track of your progress as you work on a branch or master.

Creates a transparent history

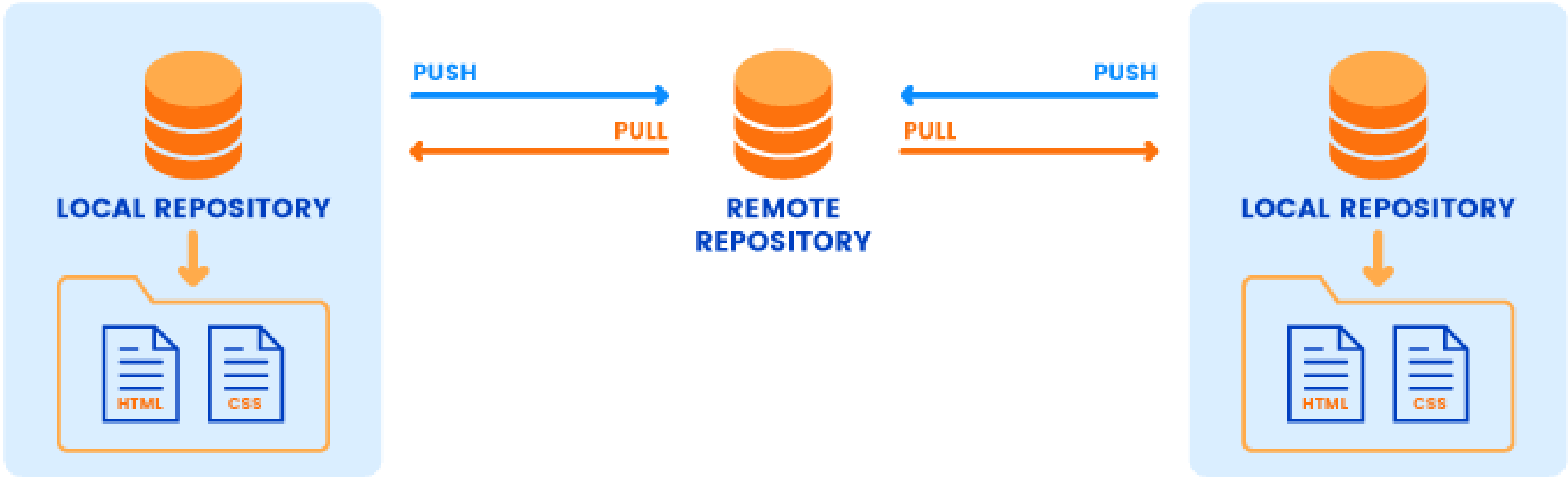
Forking a repository

it creates a copy for you to work on independently without any changes to theirs.

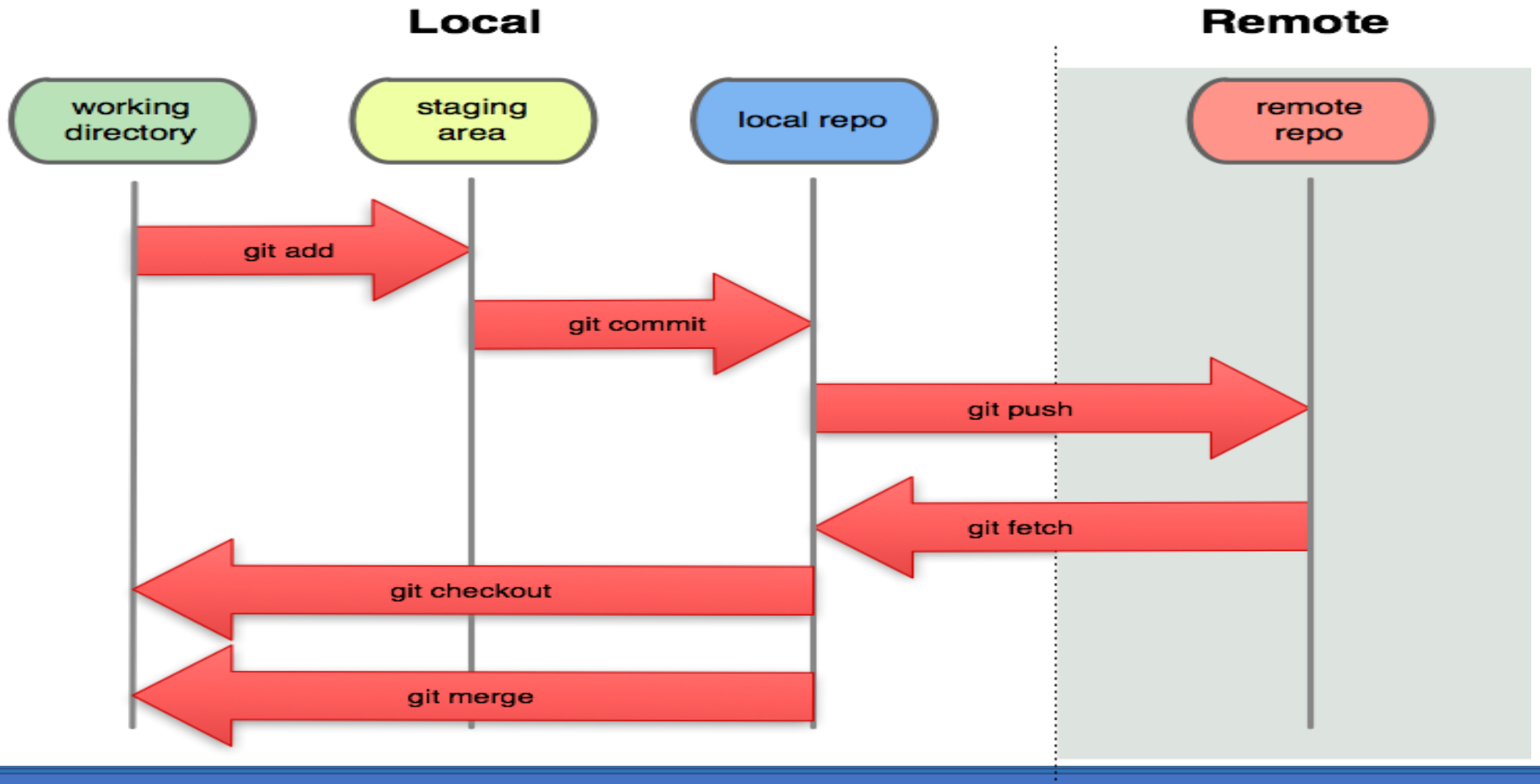
Overview of GIT



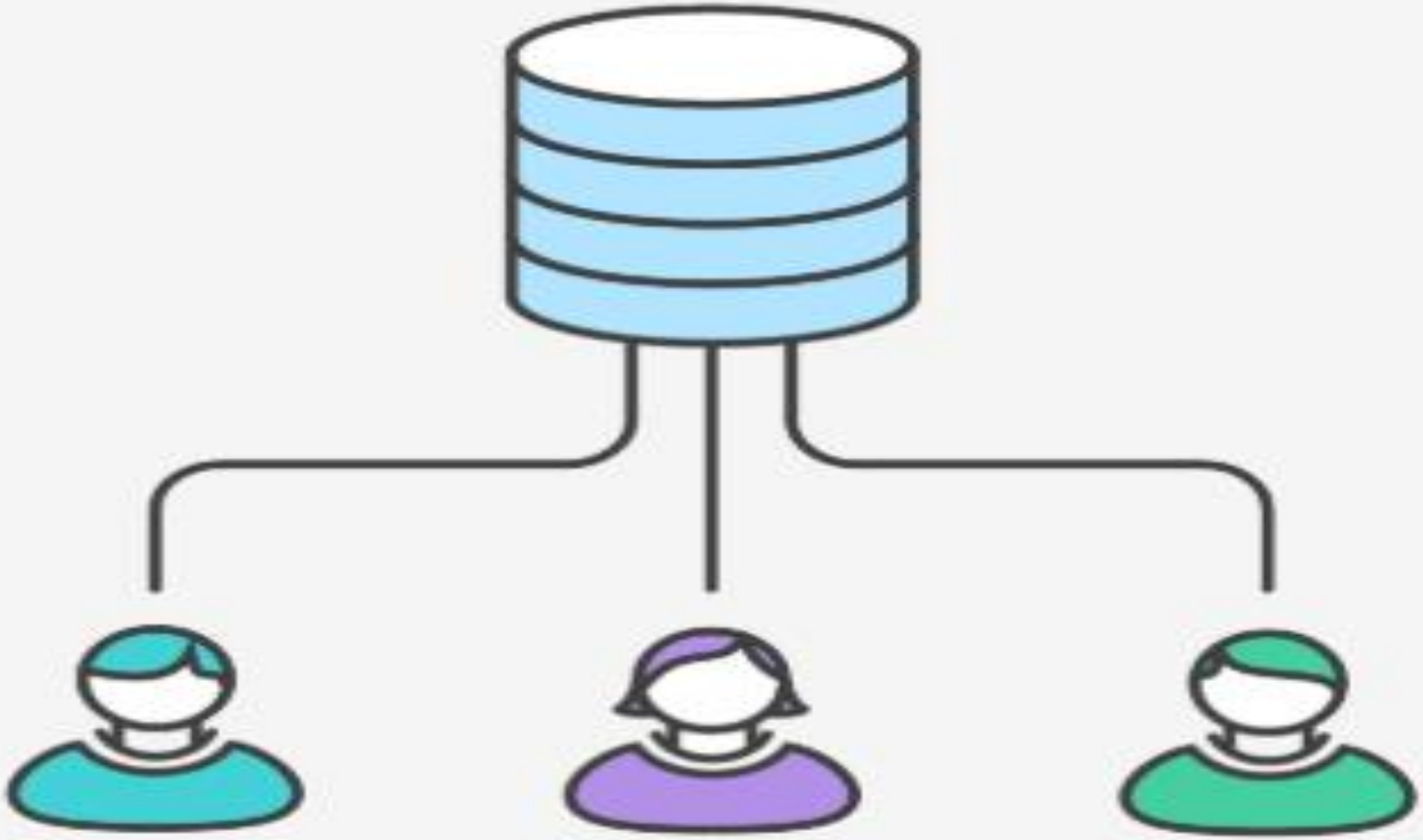
Its all About Code Repository



Working with GIT Client



Everybody Clone the Repository



Git Pull Request

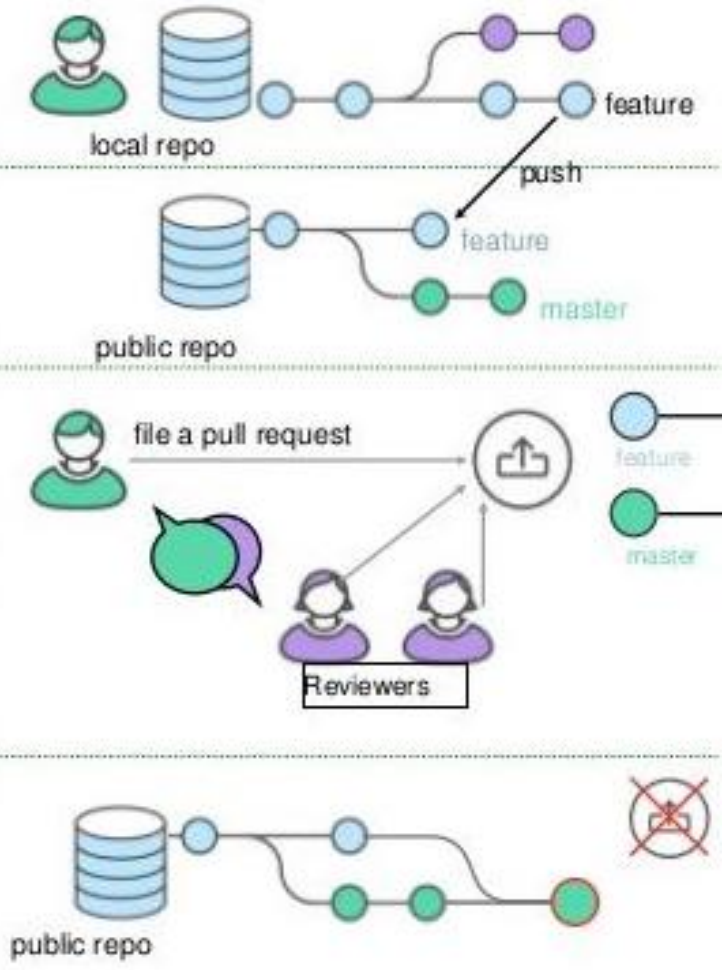
A developer creates the feature in a dedicated branch in their local repo.

The developer pushes the branch to a public Bitbucket repository.

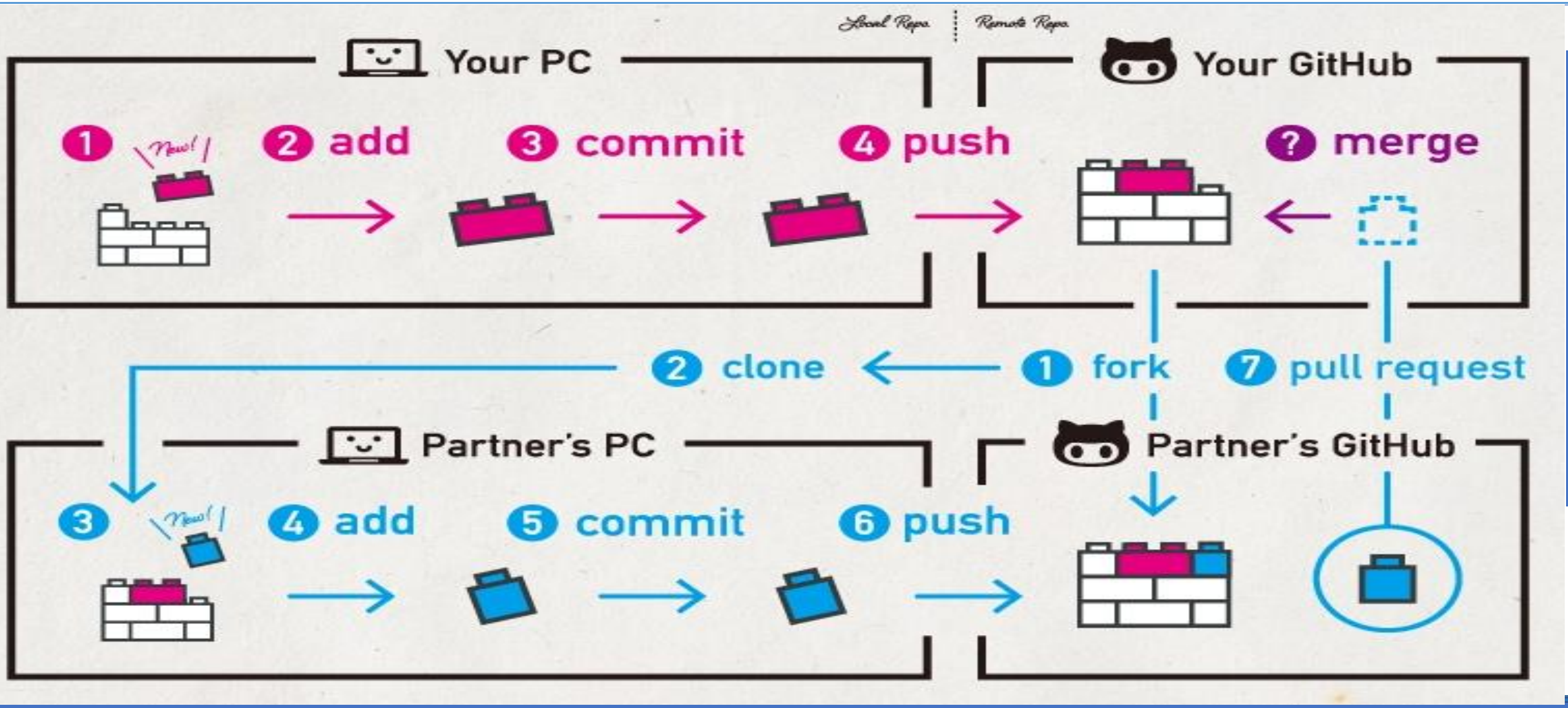
The developer files a pull request via Bitbucket.

The rest of the team reviews the code, discusses it, and updates it.

The project maintainer merges the feature into the official repository and closes the pull request.



Git & GitHub


Local Repo.
Remote Repo.
Your PC
Your GitHub
1 New!
2 add
3 commit
4 push
? merge
2 clone
1 fork
7 pull request
Partner's PC
Partner's GitHub
3 New!
4 add
5 commit
6 push