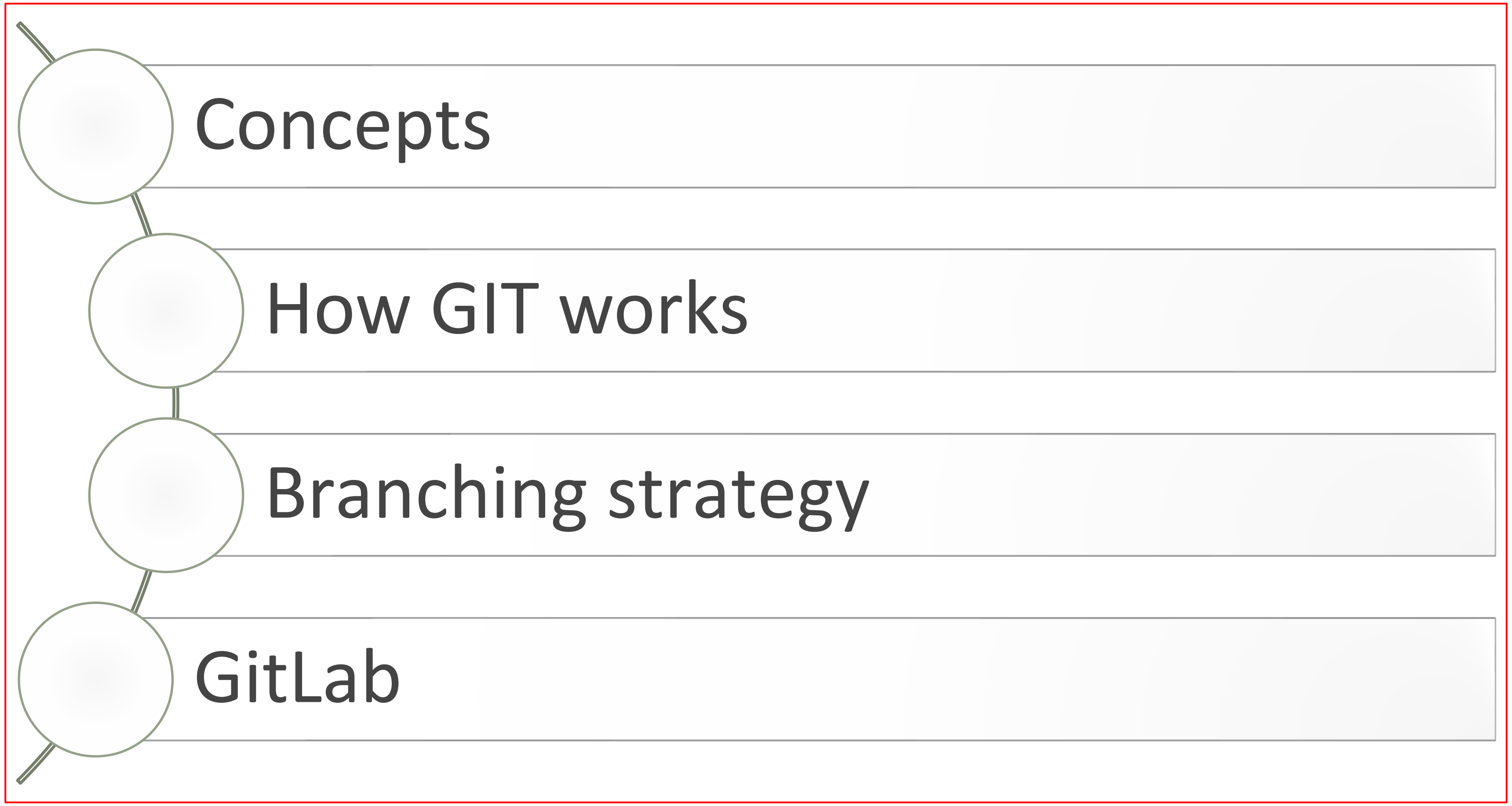
DEVSECOPS COURSE

SOURCE CODE MANAGEMENT

TRAINER: TRAN HUU HOA

AGENDA



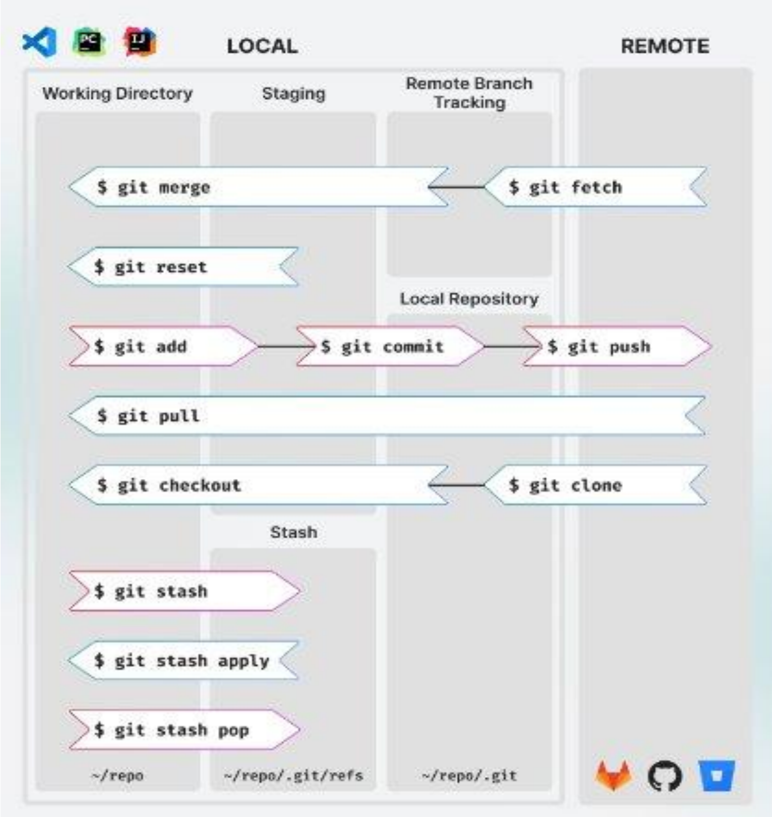
- Concepts
- How GIT works
- Branching strategy
- GitLab

CONCEPTS

Source code management (SCM), also known as version control, is the practice of tracking modifications to source code. It allows developers to maintain a running history of changes made to a codebase, ensuring that they work with accurate and up-to-date code. SCM helps resolve conflicts when merging code from multiple contributors

Git is a distributed version control system that tracks versions of files. It's commonly used by programmers to manage source code collaboratively

HOW GIT WORKS



BRANCHING STRATEGY

Concept:

The strategy a software development team employs when writing, merging, and shipping code in the context of a version control system

Use cases:

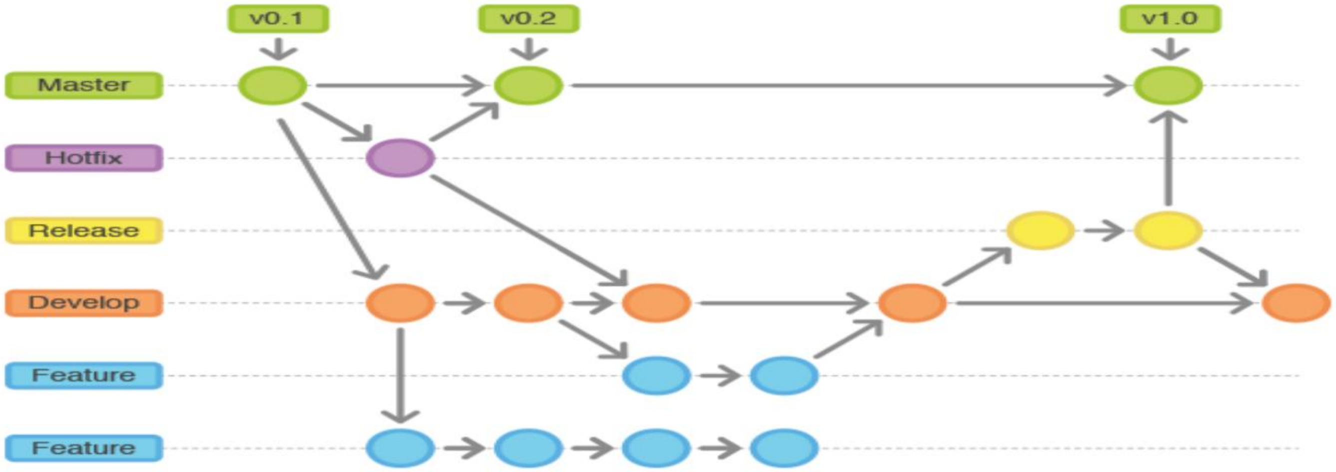
- Typical development workflow
- Emergency hotfixes
- Small vs. large changes
- Standard vs. experimental changes

Popular flows:

- GitFlow Workflow
- Trunk-based development
- Etc.

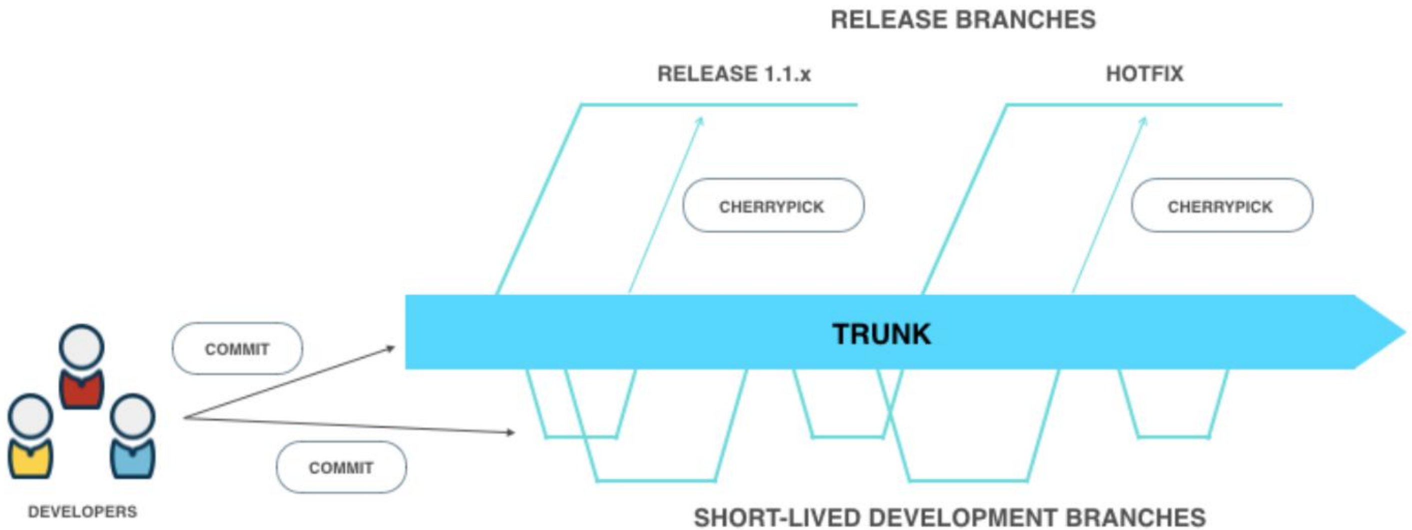
BRANCHING STRATEGY

Sample Gitflow - workflow:



BRANCHING STRATEGY

Sample trunk-based development



GITLAB

- ➢ Introduction
- ➢ Benefit
- ➢ Architecture
- ➢ How-to provision
- ➢ How-to operate

GITLAB

Introduction

GitLab helps your development team collaborate and maximize productivity, sparking faster delivery and increased visibility. With its Git-based repository, GitLab enables clear code reviews, asset version control, feedback loops, and powerful branching patterns to help your developers solve problems and ship value.

- Scale your SDLC for cloud native adoption
- Git-based repository enables developers to work from a local copy
- Automatically scan for code quality and security with every commit
- Built-in Continuous Integration and Continuous Delivery

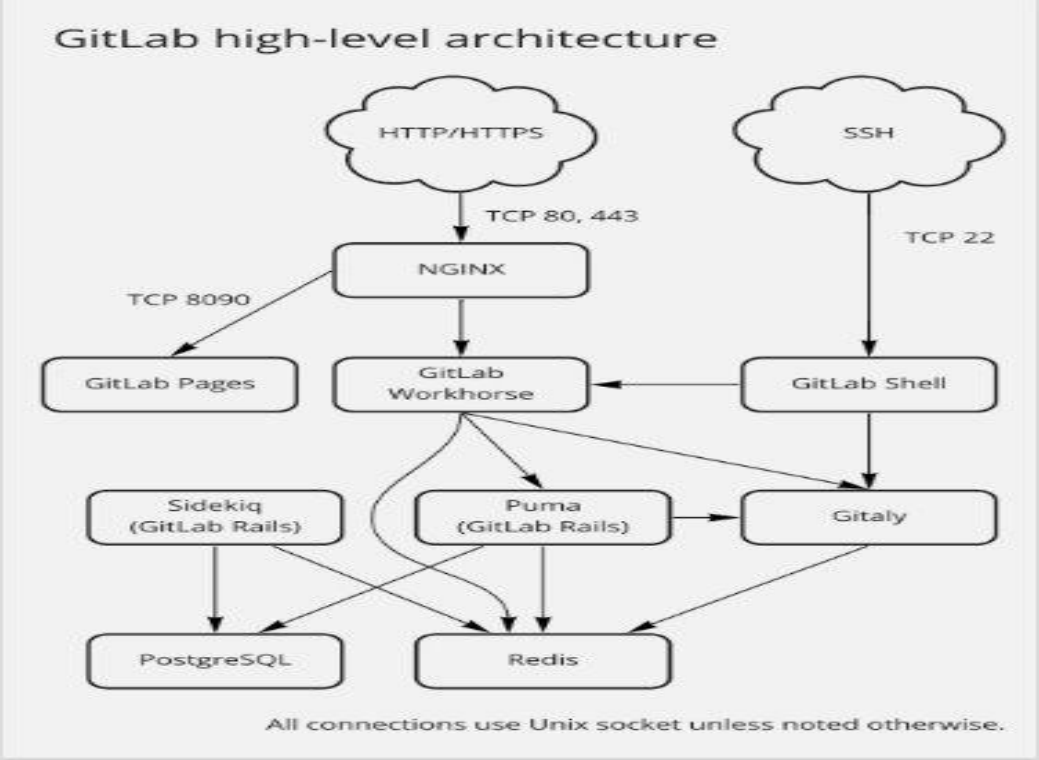
GITLAB

Benefits

- **Collaborate**
  - Review, comment, and improve code
  - Enable re-use and inner sourcing.
  - File locking prevents conflicts.
  - Robust Web IDE accelerates development on any platform
- **Accelerate**
  - Git-based repository enables developers to work from a local copy
  - Branch code, make changes, and merge into the main branch
- **Compliant and Secure**
  - Review, track, and approve code changes with powerful merge requests
  - Automatically scan for code quality and security with every commit
  - Simplify auditing and compliance with granular access controls and reporting

GITLAB

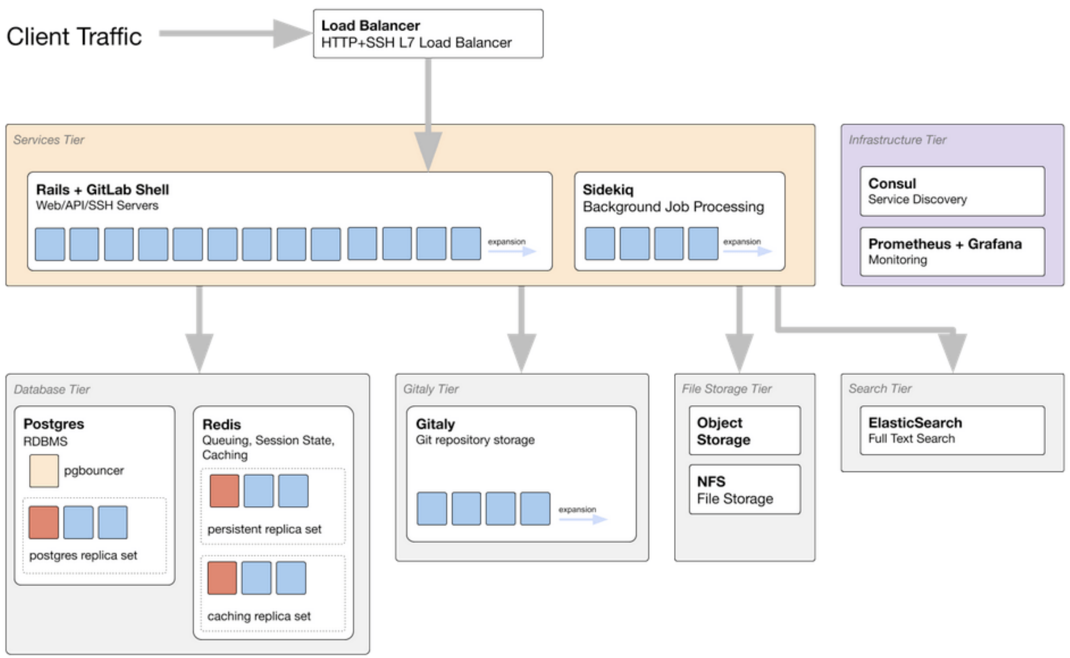
Architecture



GITLAB

Architecture

Scalability and reliability



GITLAB

How-to provision

Requirements(for GitLab current version 13.x):

- Redis version >= 5.0
- Storage for installation: 2.5 GB
- CPU 4 cores, RAM 4GB for 500 users
- CPU 8 cores, RAM 8GB up to 1000 users
- PostgreSQL >= 11.0

GITLAB

How-to provision - methods

- Using Omnibus package tool
- Helm chart (Kubernetes)
- Operator (Kubernetes)
- Docker
- Self-compiled from source code

GITLAB

How-to operate

- Manage users and their permissions
- Setup organization and projects
- Manage GIT repositories