### Introduction à Git et son rôle dans DevOps/DevSecOps

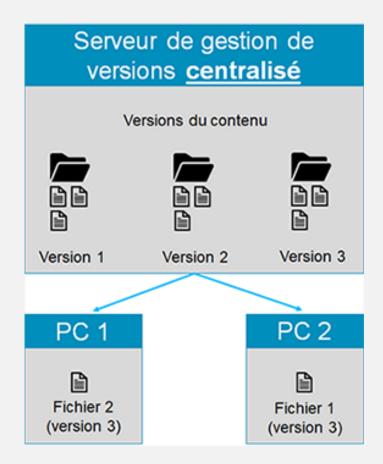
Présentation générale de Git Créé par Linus Torvalds en 2005

Adoption massive via GitHub & GitLab

Microsoft rachète GitHub en 2018

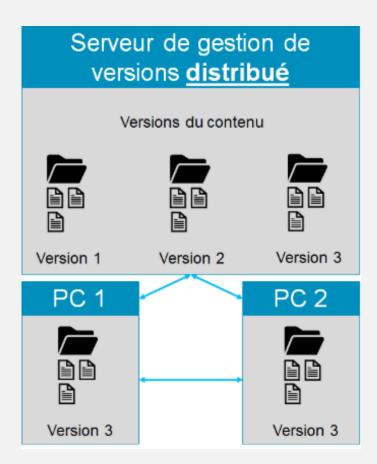
# Systèmes de gestion de versions centralisés

- Serveur central unique
- Exemples : SVN, Perforce
- Dépendance forte au serveur central



# Systèmes de gestion de versions distribués

- Chaque dépôt est dupliqué (sauvegarde intégrale)
- Travail hors ligne possible
- Collaboration flexible



## Comparaison centralisé vs distribué

	Système de gestion de version centralisé	Système de gestion de version distribué
Modèle	Client-Serveur	Distribué en dépôt
Single Point of Failure	Oui	Non
Hors ligne	Limité	Oui
Performance	Liée au serveur central	Élevée grâce aux opérations locales
Complexité	Simple	Complexe
Collaboration	Limité à la connexion au serveur central	Très flexible
Conflits de branche	Moins fréquents	Plus fréquents

## Gestionnaire de versions et SCM

- Version Control System : suivi et retour arrière
- SCM: collaboration autour du code
- Git = VCS + SCM distribué

### VCS

Version Control System

### SCM

Source Control Management

#### **Distributed**

Modifications locales

Repository (dépôt) : contient code + historique

Commit: ensemble de changements (SHA1 unique)

Principaux termes de Git

Branch: timeline de commits

HEAD : pointeur vers le dernier commit

Remote : dépôt distant (GitHub/GitLab)

	Initier un dépôt (git init)	
	Créer fichiers, modifications locales	
Workflow Git - Étapes principales	Indexer (git add)	
	Committer (git commit)	
	Partager (git push)	
	Récupérer (git pull)	