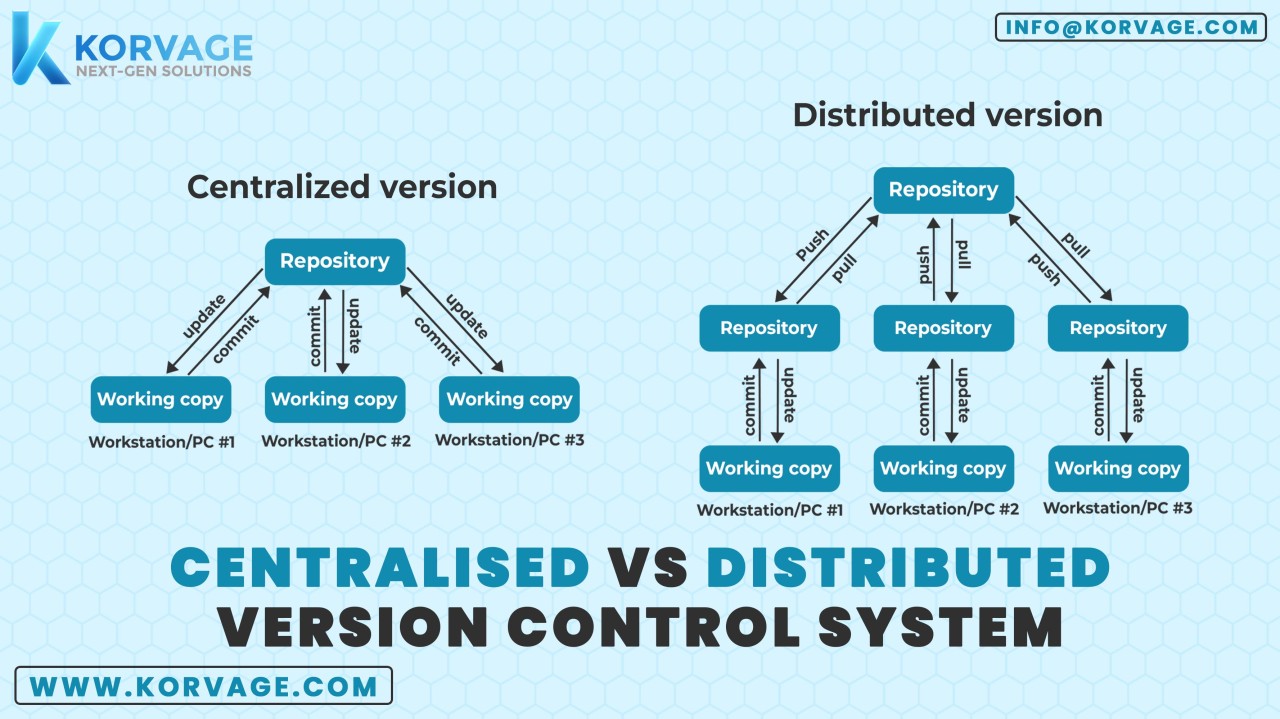
**Explained about VCS/SCM?**

   VCS could refer to Version Control System, which is a software that tracks changes to files over time.

   This allows developers to collaborate and go back to previous versions of files without having to manage them manually.

   VCS can be used for many types of files, including software source code, images, and layouts.



**Distributed Version Control Systems (Distributed VCS)**

Distributed VCS (DVCS) is a type of version control where the entire repository, including its full history, is mirrored on every user's local machine.

**Key Features:**

    Full Repository on Every Client: Each developer has a full copy of the project, including all its history. This allows for operations like commits, diffs, and branches to be done locally without needing to connect to a central server.

    Decentralization: There's no need for a central server, although one can be used for coordination. This enhances redundancy and resilience since the repository exists in multiple locations.

    Branching and Merging: Distributed VCS typically makes branching and merging easier and more efficient, allowing for more flexible development workflows.

    Difference Between Git and GitHub

**Git:**

What It Is: Git is a distributed version control system that helps track changes in source code during software development.

**Functions:**

        Manages code history.

        Tracks changes and versions.

        Supports branching, merging, and collaboration.

**Usage**: Installed locally on a developer’s machine, Git is used via the command line or a GUI. It’s independent and doesn’t require an internet connection for most operations.

**Key Features:** Speed, simplicity, strong support for non-linear development (thousands of parallel branches), and distributed nature

**GitHub:**

    What It Is: GitHub is a web-based platform built on top of Git that provides hosting for Git repositories.

**Functions:**

        Centralized hosting for Git repositories.

        Collaboration tools (pull requests, code reviews, discussions).

        Integrations with CI/CD pipelines, project management tools, and more.

**Usage:** Developers push their Git repositories to GitHub to collaborate with others, share code, and manage open-source projects.

**Key Features:** Social features like following users, starring repositories, and forking projects, as well as project management tools like issue tracking and wikis.

Create GITHUB account

Download git bash

create repository in github [remote repository]

git cmds

---------

 Getting Started - First-Time Git Setup

# ------path]/etc/gitconfig file

**# add global configurations**

    git config --global user.email "vignan.kandelab@gmail"

    git config --global user.name "john Doe"

# To show branch name in "git log" outputs

    git config --global log.decorate auto

**# starting local repository**

        git init   --> Create repository

        git status     # u may get untracked files and floders

        git add .      # move files to staging

        git commit -m " give some message related to the changes" --> moves files to local repo

**# Rename the branch as per remote repo**

    git branch -M main/master   # depends on remote project branch

**#  Git that you want to add a new remote repository.**

        git remote add origin <git repository url >

        git push -u origin master/main            # depends on branch