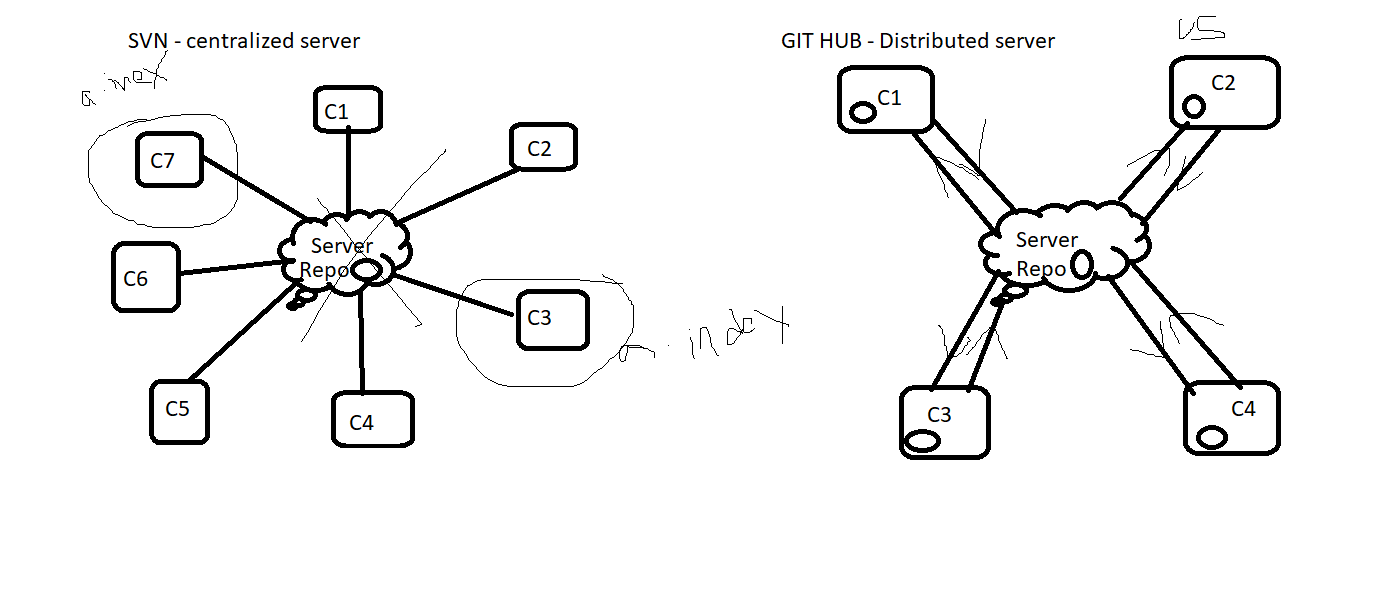
What is difference between centralized server and distributed server in github

In the context of GitHub, understanding the difference between **centralized servers** and **distributed servers** involves contrasting two distinct approaches to version control and collaboration:

****

**1. Centralized Server**

* **Definition**: In a centralized system, there is a single central server that hosts the repository, and all developers directly interact with this central repository.
* **Workflow**:
  + Developers check out (pull) code from the central server.
  + All changes are committed back to this single server.
* **Pros**:
  + Simpler to understand and manage for small teams.
  + Central point of truth.
* **Cons**:
  + If the central server goes down, development halts.
  + Requires continuous network access to the server for operations like commit, update, or merge.
* **Example**: Systems like Subversion (SVN) or Perforce follow this model.

**2. Distributed Server**

* **Definition**: In a distributed system, every developer has a complete copy of the repository (including the full history) on their local machine. This is the model Git (and thus GitHub) uses.
* **Workflow**:
  + Developers clone the repository, creating a local copy of the entire codebase.
  + Changes can be made and committed locally without needing a central server.
  + Changes are shared between developers by pushing to or pulling from a shared repository (e.g., GitHub, GitLab).
* **Pros**:
  + Fully offline work: Developers can commit and view history without a network connection.
  + Redundancy: No single point of failure since every developer has the full repository.
  + Easier branching and merging.
* **Cons**:
  + Requires more disk space because each user has a complete copy of the repository.
  + More complex workflows can be overwhelming for beginners.
* **Example**: Git, Mercurial.

**GitHub Context**

* GitHub uses the **distributed model** of Git. It acts as a remote, shared repository where developers push and pull changes. However, every collaborator has their own complete copy of the repository on their local machine.

**Key Differences:**

|  |  |  |
| --- | --- | --- |
| **Feature** | **Centralized Server** | **Distributed Server** |
| Repository Location | Single central server | Every user has a full copy |
| Offline Capability | Limited or none | Full offline capabilities |
| Fault Tolerance | Server downtime halts work | Work continues locally |
| Collaboration | Requires constant connection | Push and pull changes as needed |
| Performance | Slower due to network reliance | Faster local operations |