**Git:**

**Protocols:** Git support ***four******distinct protocols*** to ***transfer data***.

1. Local Protocol
2. HTTP Protocol
3. SSH ( Secure shell) Protocol
4. Git Protocol

**1. Local Protocol**: means the remote repository is just another directory on the same host. Where everyone on the team has access to a shared filesystem such as an NFS mount. Or in the less likely case that everyone logs in to same computer.

If we have shared mounted filesystem then we can clone, push to and pull from a local file bases repository.

To clone

* git clone /srv/git/project.git

or

* git clone file:///srv/git/project.git

To add a local repository to an existing Git project

* git remote add <local\_project> </srv/git/project.git>

To pull

* git pull /home/chakra/project

Local Protocol method is more difficult to set up and reach from multiple locations than basic network access. If we want to push from

Basics:

* Using **Git** over **SSH** instead of **Git** over **HTTP** is a **best practice**.

**2.HTTP Protocol**:

HTTP operates very similarly to the SSH or Git protocols but runs over standard HTTPS ports and can use various HTTP authentication mechanisms, meaning it’s often easier on the user than something like SSH, since you can use things like username/password authentication rather than having to set up SSH keys

To clone a Git repository over HTTP, you can specify an URL like this:

* git clone <https://example.com/gitproject.git>

**3.SSH Protocol:**

A **common transport protocol** for **Git** when self-hosting is over SSH. This is because **SSH access to servers is already set up** in most places — and if it isn’t, it’s easy to do. **SSH is also an authenticated network protocol** and, because it’s ubiquitous, it’s generally easy to set up and use.

To clone a Git repository over SSH, you can specify an ssh:// URL like this:

* git clone ssh://[user@]server/project.git

Or you can use the shorter scp-like syntax for the SSH protocol:

* git clone [user@]server:project.git

In both cases above, if you don’t specify the optional username, Git assumes the user you’re currently logged in as.

**4.Git Protocol:**

This is a special daemon that **comes packaged with Git**; it listens on a dedicated port (9418) that provides a service similar to the SSH protocol, but with **absolutely no authentication**. In order for a repository to be served over the Git protocol, you **must create a git-daemon-export-ok** file — the daemon won’t serve a repository without that file in it — but, other than that, there is no security.

The downside of the Git protocol is the lack of authentication. It’s generally undesirable for the Git protocol to be the only access to your project.