

Module 6: Git Server Administration

Demo 2:

Problem Statement: Demonstrate git clone with different protocols.

This section will guide you to:

- Different protocols supported by git clone

This lab has four subsections, namely:

1. Introduction to git remote repository
2. Protocols supported by git for data transfer
3. Use case of each protocol and their pros n cons.

Note: All the commands with the black background can be copied and edited. Those are.

Screenshots but these screenshots can be editable so it will be easy for students during execution.

Solution:

The advantage of using git or any version control system is lies as it provides collaborators distributed framework to place copy of the codes remotely and anyone on any location can get access to that remote place even if any other collaborator is offline.

Git provides us functionality to clone the Remote git repository. Remote repository can be considered as intermediate repository having access to all working groups.

To start working with remote git repository first it is required to select the protocol that your server will support. Git supports four protocols to support data transfer to remote repository – Local, ssh, https and git.

Local Protocol:

As name signifies local protocol is used when repository is being created on local server or host and all collaborators provide with the access to the path. It includes shared filesystem like NFS or sharing a local directory to everyone to maintain the code. To clone a repository like this below commands can be used:

- Cloning a git repository from local file system
git clone /home/git/project.git
- Cloning a git repository from NFS like shared filesystem
git clone file:///home/git/project.git

Source Code Management using GIT

Similarly local repository can be added to existing project by running below command.

```
git remote add origin /home/git/project.git
```

```
git remote add origin file:///home/git/project.git
```

Steps required to build Remote Repository using local protocol:

- i) **Initialize the bare remote repository in any of the local directory of the server.**

```
cd /home/git/
```

```
git init --bare project.git
```

```
root@ip-172-31-31-59:/home/git# cd /home/git/
root@ip-172-31-31-59:/home/git#
root@ip-172-31-31-59:/home/git# git init --bare project.git
Initialized empty Git repository in /home/git/project.git/
root@ip-172-31-31-59:/home/git#
```

The initialized directory will consist of all the details of branches and references.

```
root@ip-172-31-31-59:/home/ubuntu# cd /home/git/project.git/
root@ip-172-31-31-59:/home/git/project.git# ls -lrth
total 32K
-rw-r--r-- 1 root root  73 May 21 08:20 description
drwxr-xr-x 2 root root 4.0K May 21 08:20 branches
drwxr-xr-x 4 root root 4.0K May 21 08:20 refs
drwxr-xr-x 4 root root 4.0K May 21 08:20 objects
drwxr-xr-x 2 root root 4.0K May 21 08:20 info
drwxr-xr-x 2 root root 4.0K May 21 08:20 hooks
-rw-r--r-- 1 root root  66 May 21 08:20 config
-rw-r--r-- 1 root root  23 May 21 08:20 HEAD
root@ip-172-31-31-59:/home/git/project.git#
```

- ii) **The initialized repository can be clone as any other remote repository in any project as shown below.**

```
root@ip-172-31-31-59:/home/git/project.git# cd /home/demo/
root@ip-172-31-31-59:/home/demo#
root@ip-172-31-31-59:/home/demo#
root@ip-172-31-31-59:/home/demo# git clone /home/git/project.git
Cloning into 'project'...
warning: You appear to have cloned an empty repository.
done.
```

- iii) To verify if link has been created with remote repository we can use below command:

`git remote -v`

```
root@ip-172-31-31-59:/home/demo/project# git remote -v
origin /home/git/project.git (fetch)
origin /home/git/project.git (push)
root@ip-172-31-31-59:/home/demo/project#
```

- iv) Developers can write code in working directory and push the commit changes to remote repository. Below command is used to push the changed to remote repository (in our case it is on local server)

git push origin master

```
root@ip-172-31-31-59:/home/demo/project# touch sample.txt
root@ip-172-31-31-59:/home/demo/project# git add .
root@ip-172-31-31-59:/home/demo/project# git commit -m "first modification"
[master (root-commit) 8a08db3] first modification
Committer: root <root@ip-172-31-31-59.us-east-2.compute.internal>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 sample.txt
root@ip-172-31-31-59:/home/demo/project#
root@ip-172-31-31-59:/home/demo/project# git push origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 230 bytes | 230.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To /home/git/project.git
 * [new branch] master -> master
root@ip-172-31-31-59:/home/demo/project#
```

Pros:

Source Code Management using GIT

- Simple to configure and can use existing network permission for team to get the access.
- Simple to pull code from another collaborator working directory as all will be working locally.

Cons:

- Providing shared network access is much difficult rather than providing basic network at different places.
- Repository over NFS works slower than repository over ssh.
- More vulnerable to accidental corruption and anyone can easily change.

HTTP Protocol:

In git version 1.6.6 a new smart HTTP protocol has been introduced which allows git to intelligently perform data transfer like SSH. Before version 1.6.6 git supports simple http protocol which is also referred as dumb HTTP and serves read-only access.

Smart HTTP Protocol:

It provides similar operations like SSH, but it is easier for the user. It works on standard HTTP ports and supports various authentication modes. For accessing code from Remote repository, it asks for username and password to authenticate the request which is simpler than configuring ssh keys.

The standard git URL like “*<https://github.com/abc/project.git>*” works over smart HTTP protocol and can be clone with below git command.

```
git clone https://github.com/abc/project.git
```

Similarly, it can be added to existing project.

```
git remote add origin https://github.com/abc/project.git
```

dumb HTTP Protocol:

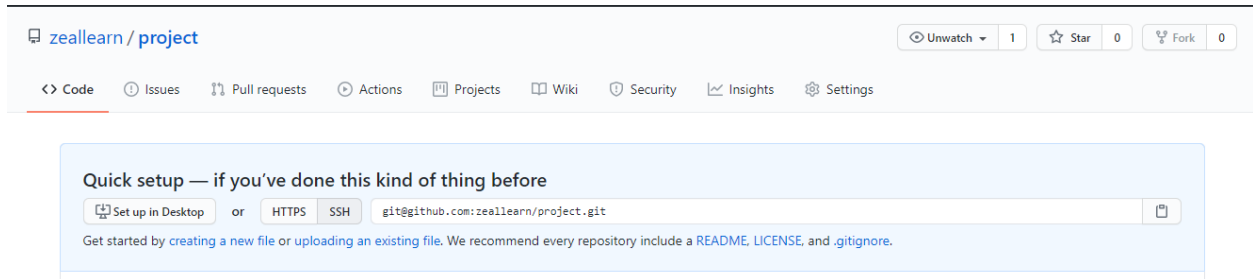
git client uses dumb HTTP protocol if smart HTTP service is unreachable and git server unable to respond to user request. The dumb HTTP treat remote repository as files present in any webserver.

For setting up git repository with dumb HTTP, place the bare repository at webserver path and setup post-update hook.

It makes users having access to the webserver path to clone the repository.

Steps required to build Remote Repository using HTTPS protocol:

- i) Create bare remote repository by login to [GitHub](#) account or organization GitLab account.
- ii) Go to the remote repository required to clone and copy the repo path by clicking on copy icon/box.



- iii) Use copied path on git client and run below command.
git clone <https://github.com/zeallearn/project.git>

```
root@ip-172-31-31-59:/home# git clone https://github.com/zeallearn/project.git
Cloning into 'project'...
warning: You appear to have cloned an empty repository.
```

- iv) Now developers can work with code and push the changes to remote repository over https.

Source Code Management using GIT

```
root@ip-172-31-31-59:/home/project# touch file1
root@ip-172-31-31-59:/home/project# git add .
root@ip-172-31-31-59:/home/project# git commit -m "first commit over http"
[master (root-commit) 0478a3e] first commit over http
  Committer: root <root@ip-172-31-31-59.us-east-2.compute.internal>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 file1
root@ip-172-31-31-59:/home/project# git push origin master
Username for 'https://github.com': zeallearn
Password for 'https://zeallearn@github.com':
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 226 bytes | 226.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/zeallearn/project.git
 * [new branch]      master -> master
root@ip-172-31-31-59:/home/project#
```

Source Code Management using GIT

```
root@ip-172-31-31-59:/home/project# touch file1
root@ip-172-31-31-59:/home/project# git add .
root@ip-172-31-31-59:/home/project# git commit -m "first commit over http"
[master (root-commit) 0478a3e] first commit over http
Committer: root <root@ip-172-31-31-59.us-east-2.compute.internal>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

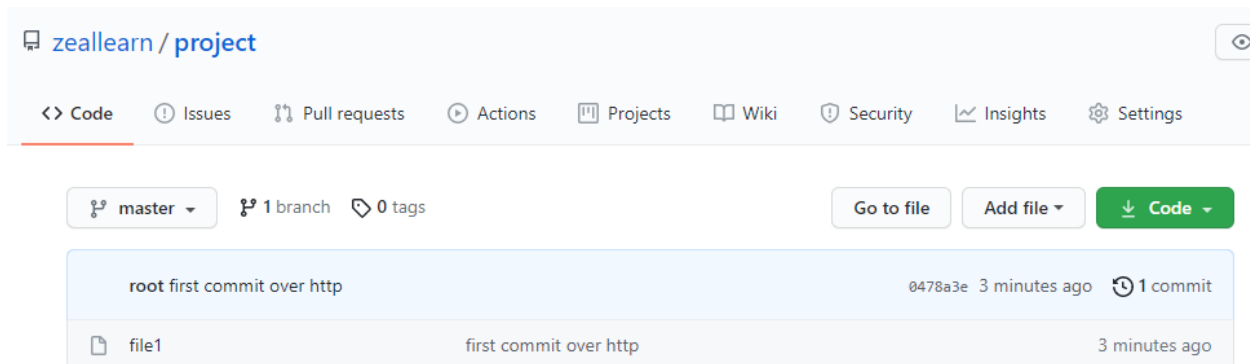
    git config --global --edit

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 file1
root@ip-172-31-31-59:/home/project# git push origin master
Username for 'https://github.com': zeallearn
Password for 'https://zeallearn@github.com':
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 226 bytes | 226.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/zeallearn/project.git
 * [new branch]      master -> master
root@ip-172-31-31-59:/home/project#
```

- v) Changes can be verified by login GitHub account.
Same changes and commit message will be reflected on https account.



Pros:

- Smart HTTP protocol provides secure access by prompting authentication request.
- It is fast and efficient protocol over network.
- HTTPs supports read-only repositories so that content transfer can be encrypted.

Cons:

- Providing username password details over https sometimes become tedious.
- Configuring git over HTTPs could be more complicated if secure or SSL support would be required.

SSH Protocol:

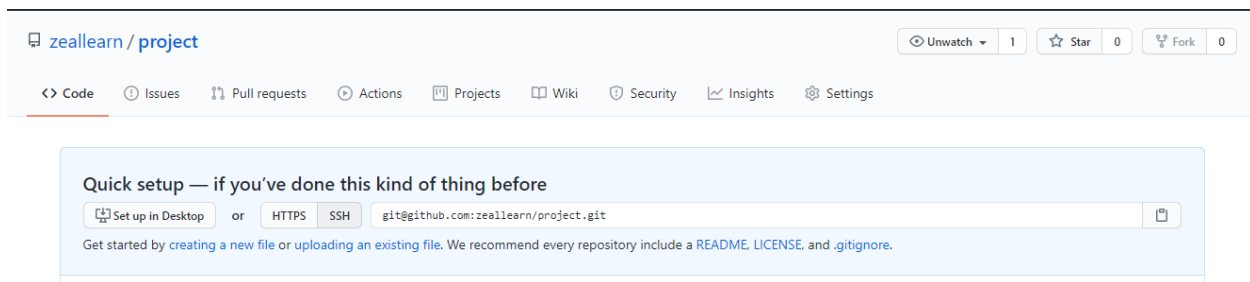
Secure Shell or ssh is global protocol known for its authenticated data transfer. It is easy to use protocol if servers are configured over ssh.

Below command can be used to clone a git repository over SSH

```
git clone ssh://username@hostaddress/project.git
```

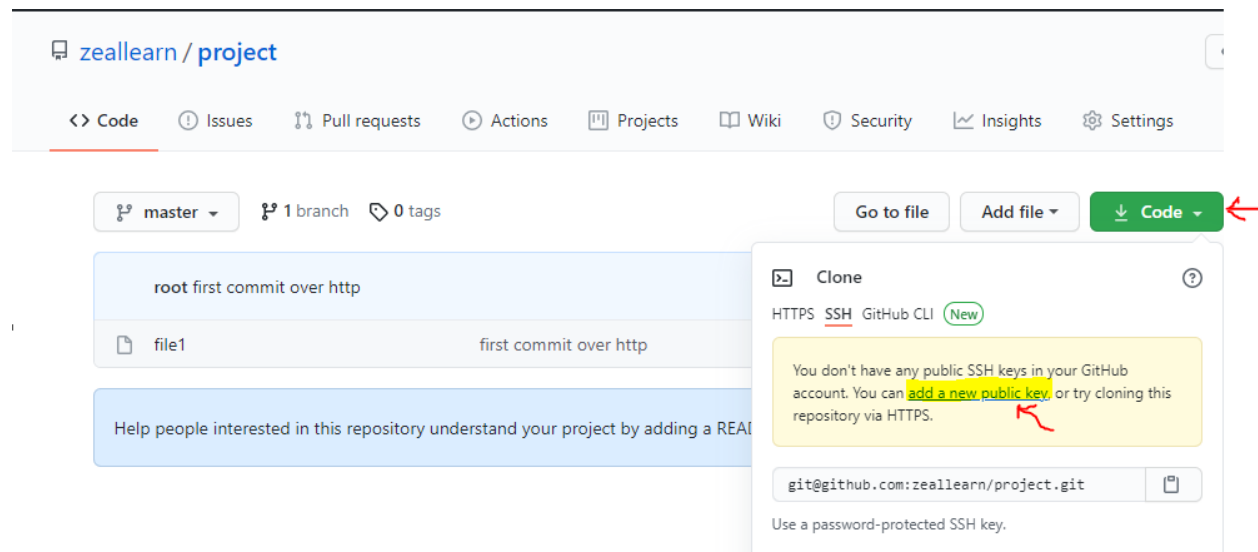
Steps required to build Remote Repository using SSH protocol:

- Create bare remote repository by login to [GitHub](#) account or organization GitLab account.**

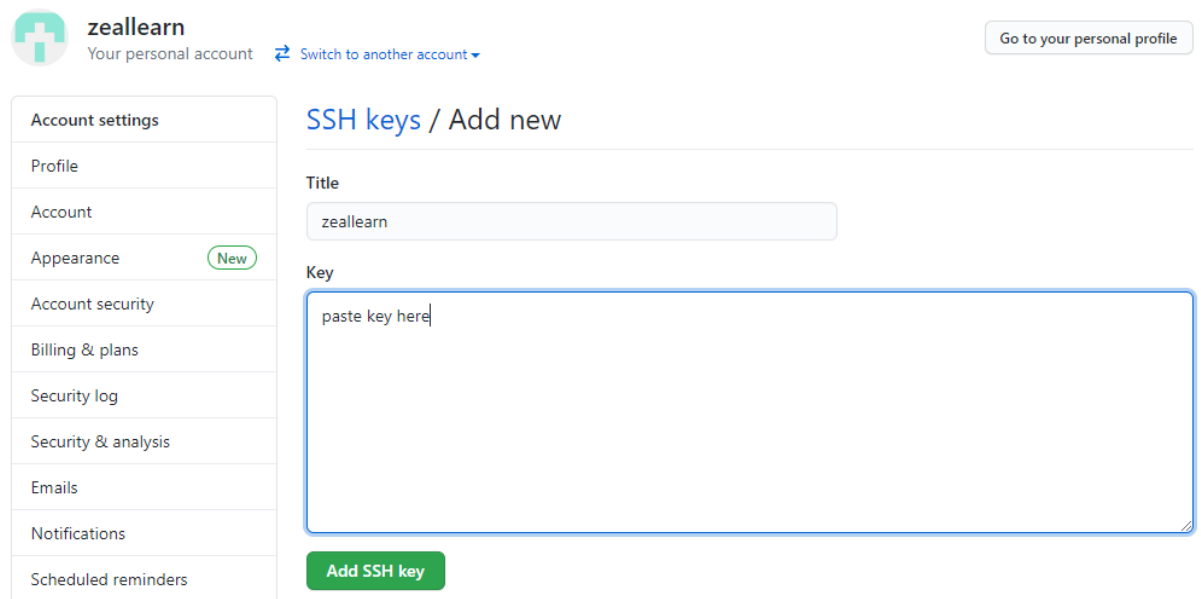


- Add ssh keys to the git remote repository by clicking on code button to give the path of repository to be clone. Option to add ssh keys requires public ssh keys of the git client.**

Source Code Management using GIT

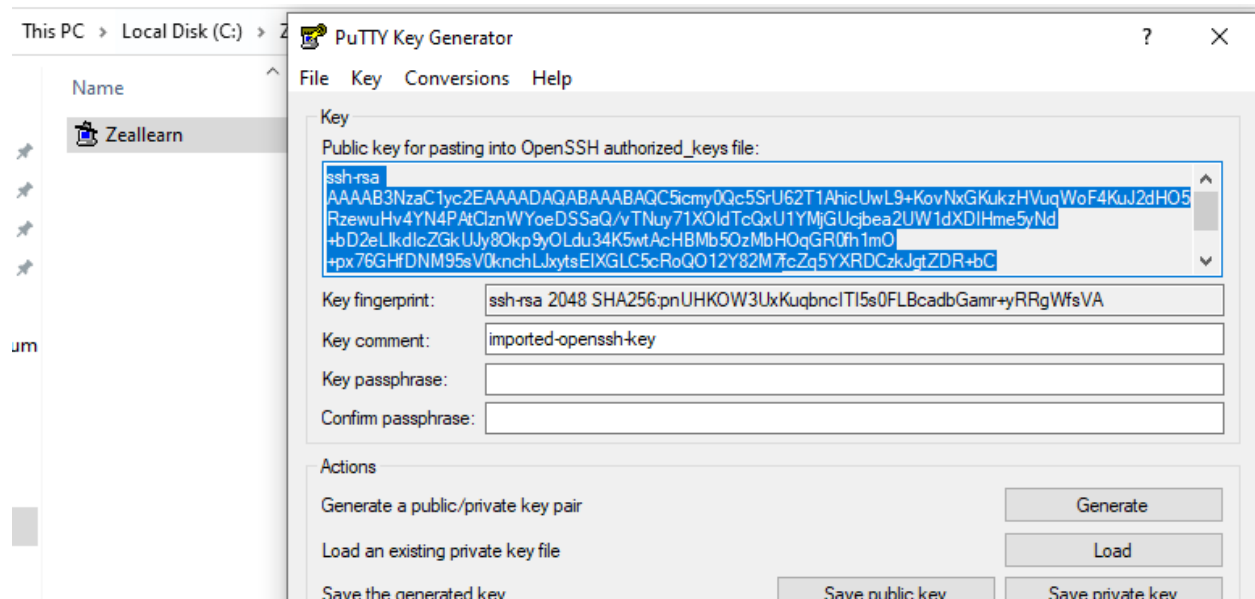


iii) *Once clicked, repository settings option to add ssh key will appear.*

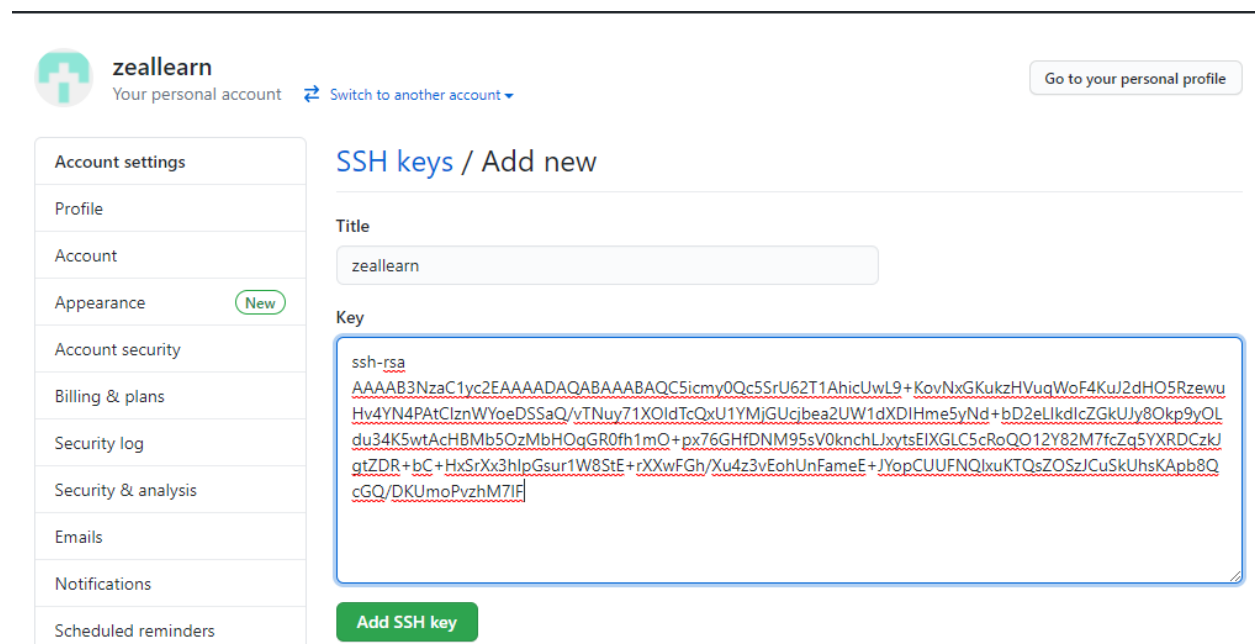


iv) *For generating ssh key, puttygen can be used to get the ssh string.*

Source Code Management using GIT



v) *Copy the string and use the same string in git repo, then add the ssh key.*



vi) *Once added ssh key will be reflected on git server repository.*

Source Code Management using GIT

The screenshot shows the GitHub interface for user 'zeallearn'. On the left is a sidebar with 'Account settings' and its sub-items: Profile, Account, Appearance (marked 'New'), Account security, Billing & plans, and Security log. The main area is titled 'SSH keys' with a 'New SSH key' button. It contains a list of SSH keys for 'zeallearn', showing a key with SHA256: pnUHK0w3UxKuqbncITI5s0FLBcadbGmr+yRRglHfsVA, added on May 21, 2021, and marked 'Never used — Read/write'. A 'Delete' button is next to it. A note at the bottom says 'Check out our guide to generating SSH keys or troubleshoot common SSH problems.'

vii) *To clone copy the ssh key and use on git client.*

The screenshot shows the GitHub repository page for 'zeallearn/project'. The 'Code' tab is selected. A 'Clone' dropdown menu is open, showing options for 'HTTPS', 'SSH' (highlighted with a red arrow), and 'GitHub CLI'. The SSH URL 'git@github.com:zeallearn/project.git' is displayed and marked 'Copied!'. Below the menu, there is a note 'Use a password-protected SSH key.' and a button 'Open with GitHub Desktop'.

viii) *Use below command to clone the remote repository over ssh.*

Git clone git@github.com:zeallearn/project.git

```
root@ip-172-31-31-59:/home/demo# git clone git@github.com:zeallearn/project.git
Cloning into 'project'...
```

Pros:

- SSH is common and global protocol hence users are familiar of its working.
- Easy to setup as many network experts are known to it.

Source Code Management using GIT

- Data transfer is encrypted and compact which makes it more efficient.

Cons:

- Read-only access without ssh access to the machine wont possible.

Git Protocol:

Git provides daemon process which runs on specific port (9418) by default and uses git protocol. Git protocol serves like SSH but there is no authentication.

Any repository can be work with git protocol if git-daemon-export-ok file is present on it, without that file daemon does not treat the file as repository.

Pros:

- It is the fastest network protocol for data transfer from remote repository.
- Easy for giving read access.

Cons:

- No authentication, so projects can be accessible by anyone over the public internet.
- Since it runs on own daemon so xinted configurations are required to be set up.
- 9418 port is required to be open on network firewall.
- Most difficult protocol to set up.