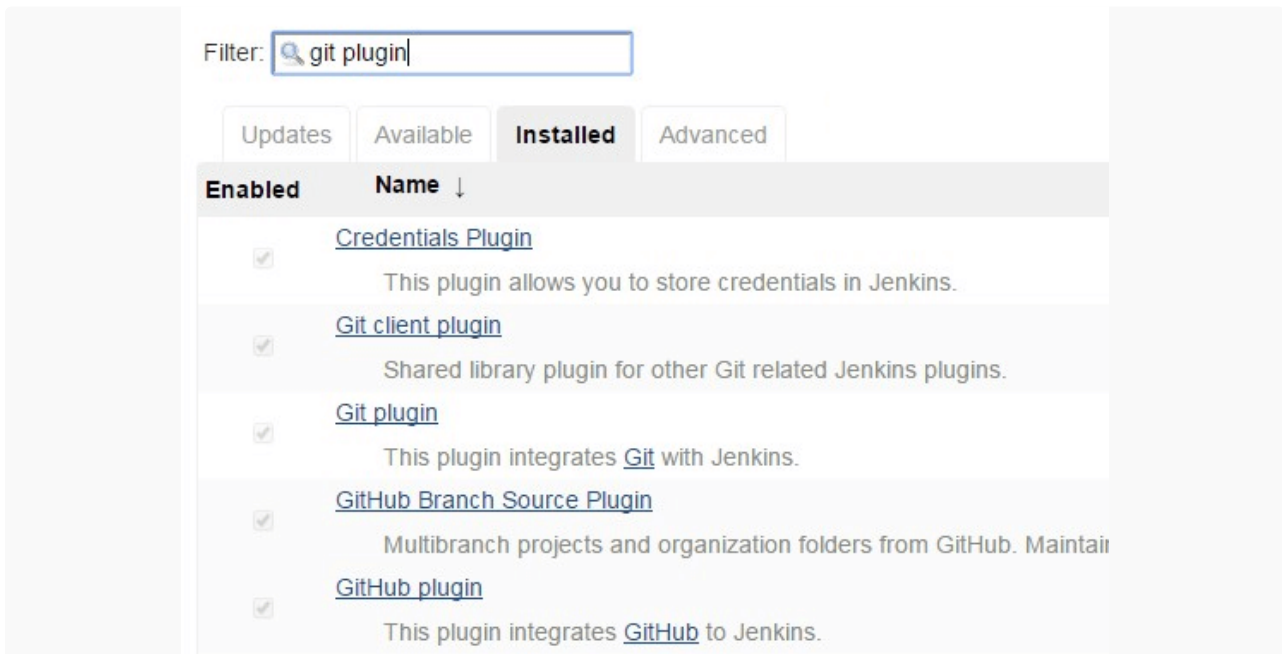


## Install Jenkins Git Plugin

For this, you should have "Git Plugin" installed in your Jenkins.

Login to Jenkins UI using your admin account, and go to "Manage Jenkins" -> Manage Plugins -> Click on "Installed" Tab -> From here, search for "git plugin" in the filter.

If the "Git plugin" is already installed, it will display it here as shown below.



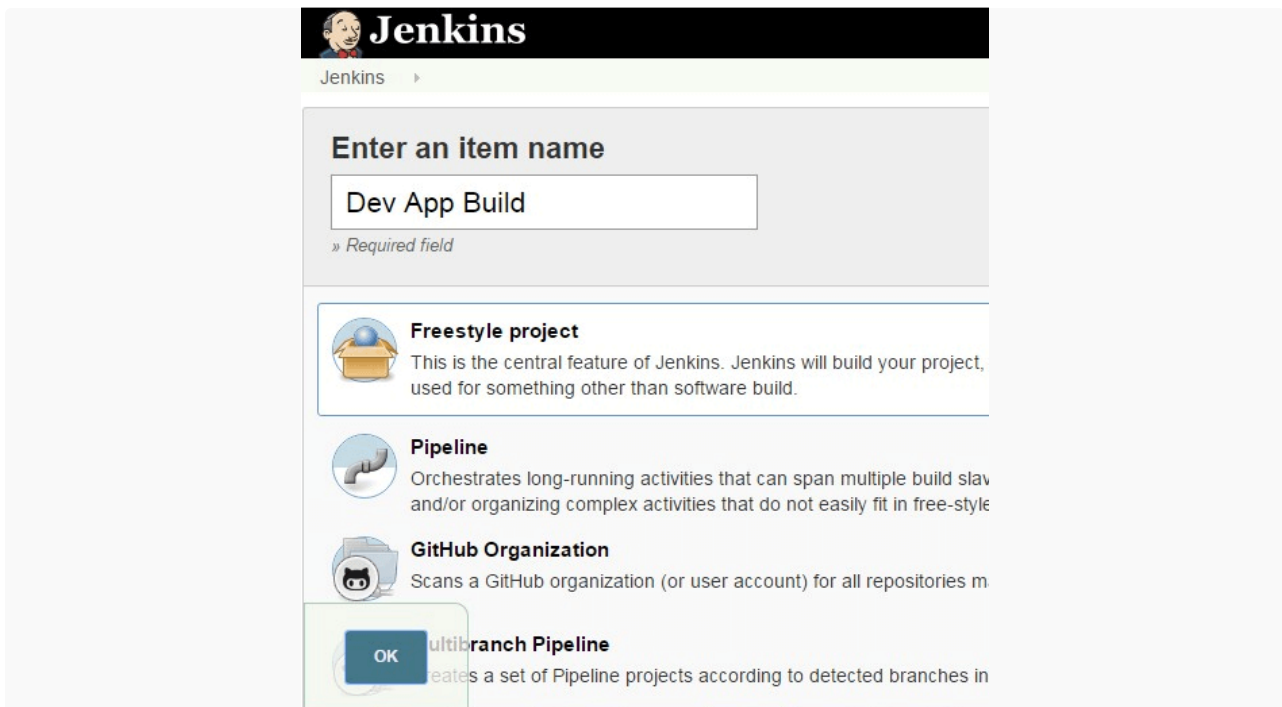
During Jenkins installation, if you've chosen to install the "recommended plugin", you'll already have "git plugin" as shown above.

If you don't see it, then install it. Go to "Available" tab, search for "git plugin", and click on "Install without restart".

From your plugins list, if you see that "Git Plugin" is an older version, then upgrade it to the latest version as explained here: [How to Upgrade Jenkins to New Version and Update Plugins](#)

## Create New Jenkins Job

Click on "New" to create a new jenkins job. Select "Free Style Project" as type as shown below. Name: Dev App Build. Type: Free Style Project



By default this will go into editing the job after creating it. In the edit mode, you'll see multiple tabs on the top, and "General" tab will be selected.

## Select "Git" for Source Code Management

For this particular job, click on "Source Code Management" tab. This will have the following options:

- None (This option will be selected by default).
- CVS
- CVS Projectset
- Git
- Subversion

Click on "Git" to select it. This will expand the section, and show all git related parameters as shown below.

☐ None  
☒ Git

Repositories

Repository URL

Please enter Git repository.

Credentials

Jenkins

Advanced...

Add Repository

Branches to build

Branch Specifier (blank for 'any')

Under the Git section, in the Repositories section, you'll see the following:

**Repository URL:** This will be the URL of your remote repository. The syntax is same as what you'll specify in the git command line. For example: `username@git-server-ip:/git/repository/full/path`

The following are some possible syntax for the repository URL:

- `ssh://git@github.com/github/git.git`
- `git@github.com:github/git.git`
- `ssh://user@other.host.com/~/repos/R.git`
- `https://github.com/github/git.git`
- `git://github.com/github/git.git`

**Credentials:** By default, this will be "None". There will an "Add" button next to it. From here, you can specify the credentials that are required for the Git URL specified above.

**Advanced button:** When you click on the "Advanced" button, you'll see two more text boxes. For most cases, you don't need to enter anything here.

- Name – You can leave this empty, and Jenkins will automatically generate unique name. This "name" is exactly the same as the one used in the git remote command.
- Refspec – The default refspec is `"+refs/heads:refs/remotes/REPOSITORYNAME/"`. This is Ok for most of you, and you can leave this empty. By default will be retrieve all the branch heads as remotes/repositoryname/branchname. Instead, if you always want to retrieve just one branch, you may want to modify this.

**Add repository button:** Click on "Add repository" to create this as the default git repository to be used by this particular Jenkins Job.

**Branches to build section:** In this section under "Branch Specifier (blank for 'any'):", the default value is \*/master. For most cases, you can leave this to default. This will track only the master branch. But, if you leave this blank, all branches will be examined for changes and built. The syntax for this field is: refs/heads/

The following are various possible examples for the "Branch Specifier" field:

- refs/heads//origin/master
- remotes/origin/master
- refs/remotes/origin/master
- tag-name
- refs/tags/tag-name
- commit-id
- \${ENV\_VARIABLE}
- \${TREEISH}
- refs/tags/\${TAGNAME},...
- Wildcards: e,g: \*, \*\*
- :regular-expression

**Add Branch button:** Click on this button to add the branch specified in the above "Branch Specifier" field to be used.

**Repository browser:** By default, this drop down list will be set to "Auto". In most cases, you can just leave it to default. This will create the links for the "changes" view inside the Jenkins to an external system for browsing. The "Auto" option will automatically choose the best possible repository browser for the selected SCM.

The following are possible values for this Repository browser:

- Auto
- AssemblaWeb
- FishEye
- Kiln
- Microsoft Team Foundation Server/Visual Studio Team Services
- bitbucketweb
- cgkit
- gitblit
- githubweb
- gitiles
- gitlab
- gitlist
- gitoriousweb
- gitweb
- phabricator

- redmineweb
- rhodecode
- stash
- viewgit

When you create a new Jenkins job, make sure the security is enabled on that particular job as explained here: [How to Setup User Security on Jenkins with Project Matrix Authorization](#)

## Setup Jenkins Credentials for Git

When you create the repository, you might get the following error:

Repository URL error 1: [ramesh@192.168.101.2:/home/git/devapp](#)

Failed to connect to repository : Command "git ls-remote -h [ramesh@192.168.101.2:/home/git/devapp](#) HEAD" returned status code 128:

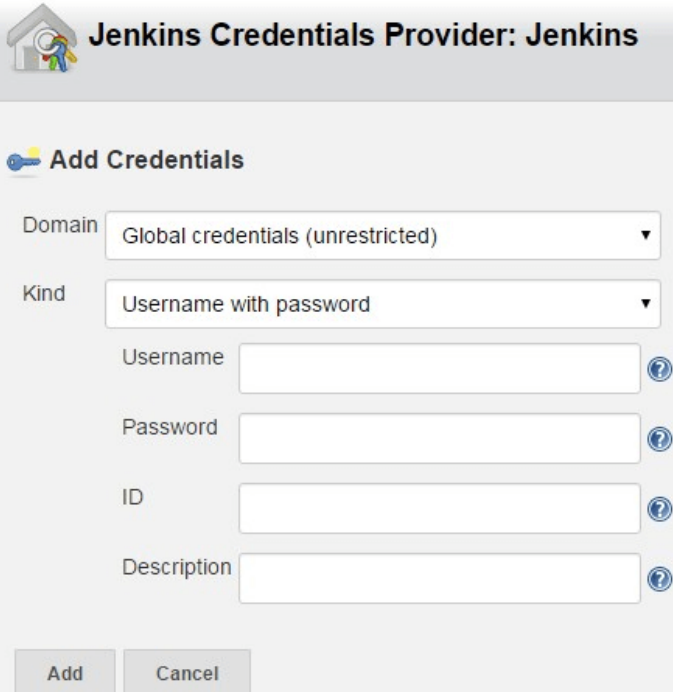
stdout:

stderr: Host key verification failed.

fatal: The remote end hung up unexpectedly

Typically, you'll get the above error message, when your "Credentials" under "Git" is set to "None".

To add a credential, click on "Add" next to "Credentials" -> Select "Jenkins Credential Provider", this will display the following add credentials screen.



- Domain: By default "Global credentials (unrestricted)" is selected. Other option is: "Username and password". Use default.
- Kind: The following are the possible options:

- Username and password (This is shown in the above screenshot).
- SSH Username with private key
- Secret file
- Secret text
- Certificate
- Username – Specify the username that will be used to connect to the remote git repository
- Password – Specify the password for the above username.
- ID – Leave this empty. This is an internal unique ID that are used to identify this credentials in the job. When you leave this empty, Jenkins will auto generate an id.
- Description – Descriptive notes about this credentials

If you choose, the SSH username with private key, you'll see the following screen:

**Jenkins Credentials Provider: Jenkins**

**Add Credentials**

Domain: Global credentials (unrestricted) ▼

Kind: SSH Username with private key ▼

Username:

Private Key: ☐ Enter directly  
☐ From a file on Jenkins master  
☐ From the Jenkins master ~/.ssh

Passphrase:

ID:  ?


Description:  ?

Add Cancel

- Kind: SSH Username with Private Key
- Username – Specify the username that will be used to connect to the remote git repository
- Private Key: The following are three options for this:
  - Enter directly
  - From a file on Jenkins master
  - From the jenkins master ~/.ssh
- Passphrase – The passphrase for your key
- ID – The unique ID (same as what is explained above)
- Description – Notes

Once the credentials are created, complete the remaining sections in the "Git" section under the "Source Code Management" tab for your particular job.

After this, if you want to view all the credentials that you've created so far in Jenkins, go to "Jenkins" -> Click on "Credentials" on the menu, which will list all the credentials that you've created inside Jenkins so far as shown below.

**Jenkins**

Jenkins > Credentials

New Item

People

Build History

Manage Jenkins

My Views

**Credentials**


System





Build Queue

No builds in the queue.

Build Executor Status +



master

**Credentials**

T	P	Store ↓	Domain	ID
		<a href="#">Jenkins</a>	<a href="#">(global)</a>	1c
		<a href="#">Jenkins</a>	<a href="#">(global)</a>	43

Icon: [S](#) [M](#) [L](#)

**Stores scoped to [Jenkins](#)**

P	Store ↓
	<a href="#">Jenkins</a>
	<a href="#">(global)</a>