**Git Repository Guide for Linux Server User**

# Section 1: LINUX offline installation of Git

Check if git is installed

|  |
| --- |
| git --version |

git version 1.8.3.1 on 66 servers

This article is about offline installation of git under lunix system. If you have installed git before, please delete git related files first.

Centos, Unbantu and other platforms have slightly different installation methods.

Download Git from Git official website

<https://mirrors.edge.kernel.org/pub/software/scm/git/>

Unzip the file

|  |
| --- |
| tar -vxf /usr/local/git/git-2.9.5.tar.gz |

Compile & Install

|  |
| --- |
| ./configure --prefix=/usr/local/git  make && make install |

Environment variable configuration

|  |
| --- |
| # Edit profile file vim /etc/profile # Add to the last line of porfile export PATH=$PATH:/usr/local/git/bin # Effective configuration source /etc/profile |

Verify successful installation

|  |
| --- |
| git --version |

Create git user and manage git service

|  |
| --- |
| useradd git passwd git #Create warehouse mkdir -p data/git/gittest.git git heat --bare data/git/gittest.git cd data/git/  # Change the owner of hangl.git to the git user under the git group chown -R git:git gittest.git/ |

client clone

|  |
| --- |
| git clone git@10.221.5.128:/opt/app/gitdata/gittest.git |

Problem summary

1. bash: git-receive-pack: command not found fatal: Could not read from remote repository.

Solution:

|  |
| --- |
| ln -s /usr/local/git/bin/git-receive-pack /usr/bin/git-receive-pack |

# Section 2: Install GitKraken on LINUX

Unzip the file to the /home/git/software/gitkraken path

|  |
| --- |
| sudo tar -xvzf gitkraken-amd64.tar.gz -C /home/git/software/gitkraken |

exist/home/git/gitkrakenCreate a pointer in/home/git/software/gitkraken/gitkraken/resources/bin/gitkraken.sh of[Soft connection](https://so.csdn.net/so/search?q=%E8%BD%AF%E8%BF%9E%E6%8E%A5&spm=1001.2101.3001.7020)(The following instructions are on the same line, and the line breaks are spaces)

|  |
| --- |
| sudo ln -s /home/git/software/gitcracking/gitkraken/resources/bin/gitkraken.sh /home/git/gitcracking |

Start gitkraken

|  |
| --- |
| cd /home/git/gitkraken  ./gitkraken.sh |

# Section 3: Server warehouse construction (LINUX, offline)

**Step 1: Install Linux, install Git**

Specific steps are omitted here.

**Step 2: Create git user**

Execute the following command with administrator privileges (root) in the Linux system to create a Linux user:

|  |
| --- |
| adduser git |

During the execution of this command, you will be prompted to enter the new user's login password, automatically create the user's home directory /home/git, and perform other necessary tasks.

The user does not have to be called git, any legal name will do, such as liigogit, etc. Of course, subsequent operation commands must be modified accordingly, and the final warehouse address will also be different (the warehouse address starts with this user name).

There are methods on the Internet to prohibit git users from logging into SSH. I think this step is optional.

**Step 3: Create a git repository**

Log in to the Linux system as the git user and execute the following command in the user's home directory /home/git to create a git warehouse:

|  |
| --- |
| it's hot --bare myrepo.git |

It is a convention that the name of the bare warehouse (-bare) directory ends with the suffix .git and is not required.

The local directory of the repository is /home/git/myrepo.git, or ~/myrepo.git, there is no doubt about this.

The repository remote directory can be any of the following:

git@192.168.0.2:/home/git/myrepo.git

git@192.168.0.2:~/myrepo.git

git@192.168.0.2:myrepo.git

We usually use the last one because it is the most concise.

Note that it contains user name, IP address, local directory and other information; note that the symbols @ and : are essential.

**Step 4: Clone the git repository on the client side**

The Git client can clone the warehouse through the following command:

|  |
| --- |
| git clone git@192.168.0.2:myrepo.git |

The username git is already included in the remote directory of the warehouse, and the user password is the password set when creating the git user in step 2. A password is required to clone the repository.

**Step 5: Push the client project to the git repository**

for the first timeWhen pushing a local project to the server repository, you need to change the local project folder into a git repository.

Open the command line in the project path and enter

|  |
| --- |
| it's hot |

A .git folder will be generated in this directory

**Step 6: Submit and push files to the remote warehouse**

1. Add all files to staging

|  |
| --- |
| git add . |

1. Submit the contents of the staging area to the warehouse

|  |
| --- |
| git commit -m "Message" |

1. Bind remote warehouse

|  |
| --- |
| git remote add origin "git@192.168.86.66:myrepo.git" |

origin - local name, can be changed

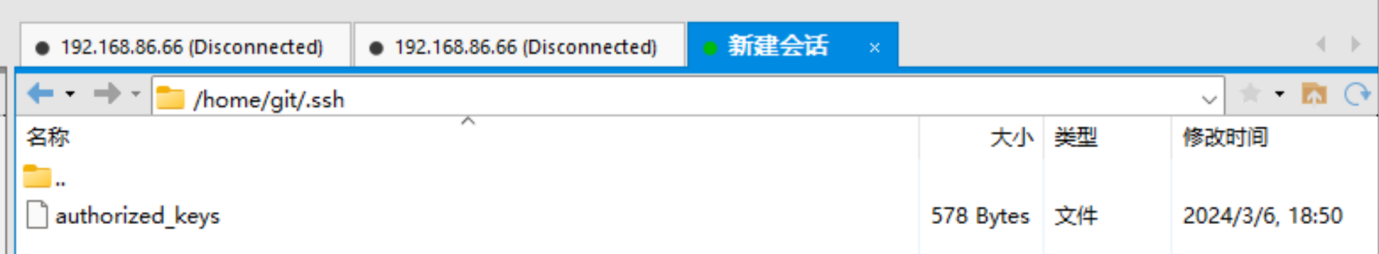
"git@192.168.86.66:myrepo.git" - "Repository link"

1. Push local files to remote repository

|  |
| --- |
| git push -u origin master // Push local files to remote warehouse git push -u origin master -f //Force push local files to remote repository |

## Public key configuration

Send the SSH Key to the administrator and fill it in the Git server configuration, as shown below.



## Problem solving:

**（1）Prompt "fatal: not a git repository":**

At the command line enter it's hot Then just enter

# Section 4: Git user permission settings

## 1. Create user group

First, you can create a user group for managing warehouse permissions. For example, you can create a user group named "gitusers" using the following command:

|  |
| --- |
| $ sudo groupadd gitusers |

## 2. Add users to user groups

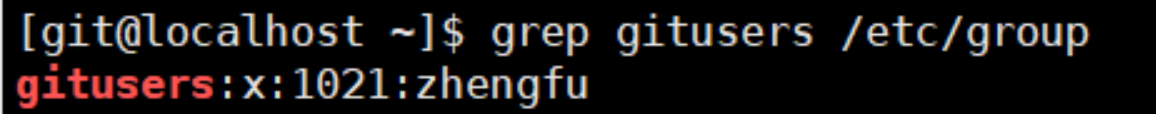
Add users who need permission to manage the warehouse to the user group. For example, suppose there is a user named "tom" who needs to have read and write permissions on the warehouse. You can use the following command to add the user to the user group:

|  |
| --- |
| $ sudo usermod -a -G gitusers zhengfu |

Check whether user groups and users are added successfully

|  |
| --- |
| $ grip gitusers /etc/group |

If added successfully, as shown below:



Modify existing user group

|  |
| --- |
| # Modify the existing gitusers user group name to gitusers\_new  $ groupmod -n gitusers\_new gitusers |

## 3. Create a Git repository

Use Git commands to create a new repository, or use an existing repository. For example, you can use the following command to create a repository named "myrepo.git" in the /home/git directory:

|  |
| --- |
| $ sudo git init -bare /home/git/myrepo.git |

## 4. Set warehouse permissions

Use Linux permission control to set the permissions of the warehouse to ensure that only members of the user group can access the warehouse. For example, you can use the following command to set the owner of the repository to the "gitusers" user group and set the permissions to read-write (rwx):

|  |
| --- |
| $ sudo chown -R :gitusers /home/git/myrepo.git $ sudo chmod -R g+rwX /home/git/myrepo.git |

## 5. Configure SSH key authentication

Allowing users to authenticate using only SSH keys ensures that only authorized users can access the repository. The user needs to generate an SSH key pair and add the public key to the list of authorized keys on the Git server. For example, you can generate an SSH key pair using the following command:

|  |
| --- |
| $ ssh-keygen -t rsa -b 4096 -C "your\_email@example.com" |

Then, add the contents of the public key (usually the `~/.ssh/id\_rsa.pub` file) to the `~/.ssh/authorized\_keys` file on the Git server.

## 6. Configure Git warehouse access permissions

Use Git commands to configure access permissions for the repository to ensure that only authorized users can clone, pull, and push the repository. For example, you can use the following command to create a configuration file named "gitolite.conf" in the root directory of the repository:

|  |
| --- |
| $ sudo nano/home/git/myrepo.git/gitake care.conf |

Then, add the following to the configuration file, specifying the user group and repository name with read and write permissions:

|  |
| --- |
| @gitusers = zhengfu repo myrepo.git  RW+ = @gitusers |

Save and exit the configuration file.

## 7. Submit and push changes

Use Git commands to commit and push the permission settings just made to the warehouse. For example, you can commit and push changes using the following commands:

|  |
| --- |
| $ cd /home/git/myrepo.git $ sudo git add . $ sudo git commit -m "Set repository permissions" $ sudo git push origin master |

Through the above steps, you have successfully set the permissions of the Git warehouse. Now only members of the user group can access, clone, pull and push the repository.