NOTES

Stephen Jeffs

# TABLE OF CONTENT

[1 TABLE OF CONTENT 2](#_Toc110106179)

[2 Installing git on a Raspberry Pi 3](#_Toc110106180)

[2.1 Creating a git User 3](#_Toc110106181)

[2.2 Configuring OpenSSH on Linux 4](#_Toc110106182)

[2.3 Configuring OpenSSH on Windows 10 4](#_Toc110106183)

[2.3.1 Installing OpenSSH Client 4](#_Toc110106184)

[2.3.2 Installing OpenSSH Server 4](#_Toc110106185)

[2.3.3 Set to run Services automatically 5](#_Toc110106186)

[2.3.4 Start Services 5](#_Toc110106187)

[2.3.5 Create Public/ Private keypair 5](#_Toc110106188)

[2.3.6 Load Private key into ssh-agent 6](#_Toc110106189)

[2.3.7 Copy Public key to the Raspberry Pi 6](#_Toc110106190)

[2.3.8 Testing it 7](#_Toc110106191)

[2.4 Installing and Configuring git on the Pi 8](#_Toc110106192)

[2.4.1 Installing git 8](#_Toc110106193)

[2.4.2 Creating a Bare Repository 8](#_Toc110106194)

[2.4.3 Copying an existing local repository to a Remote respository 8](#_Toc110106195)

[2.4.4 Using a Centralized git user account 9](#_Toc110106196)

[3 APPENDIX A - Troubleshooting git over SSH 20](#_Toc110106197)

[4 APPENDIX A - SSH on Raspberry Pi 21](#_Toc110106198)

[5 APPENDIX A – ICONS USED 22](#_Toc110106199)

[6 APPENDIX B - PYCHARM SYMBOLS 23](#_Toc110106200)

# Installing git on a Raspberry Pi

## Creating a git User

Use sudo adduser *<username>*

|  |
| --- |
| sudo adduser git |

fill out the information

|  |
| --- |
| Adding user `git' ...  Adding new group `git' (1005) ...  Adding new user `git' (1003) with group `git' ...  Creating home directory `/home/git' ...  Copying files from `/etc/skel' ...  New password:  Retype new password:  passwd: password updated successfully  Changing the user information for git  Enter the new value, or press ENTER for the default  Full Name []: git  Room Number []:  Work Phone []:  Home Phone []:  Other []:  Is the information correct? [Y/n] y |

Create the directories required to hold the keys:

|  |
| --- |
| cd ~  mkdir ~/.ssh |

we are going to create a file names "authorized\_keys" in .ssh hold the public key

Note the rights should be aset as follows:

.ssh 700

authorized\_keys 644

|  |
| --- |
|  |

Note: to delete a user account and files:

sudo userdel -r *<username>*

|  |
| --- |
|  |

## Configuring OpenSSH on Linux

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

## Configuring OpenSSH on Windows 10

### Installing OpenSSH Client

Start >> Settings >> Apps >> Apps & Features >> Optional Features

Look for "OpenSSH Client".

If abscent:

+ Add feature

### Installing OpenSSH Server

Start >> Settings >> Apps >> Apps & Features >> Optional Features

Look for "OpenSSH Server".

If abscent:

+ Add feature

### Set to run Services automatically

**Open Powershell as administrator**

Set-Service ssh-agent -StartupType Automatic

|  |
| --- |
| Set-Service ssh-agent -StartupType Automatic |

|  |
| --- |
| Set-Service sshd -StartupType Automatic |

### Start Services

**Open Powershell as administrator**

|  |
| --- |
| Get-Service ssh-agent  Status Name DisplayName  ------ ---- -----------  Stopped ssh-agent OpenSSH Authentication Agent |

|  |
| --- |
| Start-Service ssh-agent |

|  |
| --- |
| Get-Service ssh-agent  Status Name DisplayName  ------ ---- -----------  Running ssh-agent OpenSSH Authentication Agent |

same for ssh-server

|  |
| --- |
| Get-Service sshd  Status Name DisplayName  ------ ---- -----------  Running sshd OpenSSH SSH Server |

### Create Public/ Private keypair

**Open Powershell as administrator**

On window PC navigate to ~\.ssh

|  |
| --- |
| **cd ~\.ssh**  PS C:\Users\conta\.ssh> |

Run the ssh-keygen tool

|  |
| --- |
| **ssh-keygen -t ed25519 -C "your\_email@example.com"**  or  **ssh-keygen -t rsa -b 4096 -C "your\_email@example.com"**  Generating public/private rsa key pair.  Enter file in which to save the key (C:\Users\conta/.ssh/id\_rsa): id\_pi  Enter passphrase (empty for no passphrase): LEAVE\_EMPTY  Enter same passphrase again: LEAVE\_EMPTY  Your identification has been saved in id\_pi.  Your public key has been saved in id\_pi.pub.  The key fingerprint is:  SHA256:2xvmh91YEN9Gp/GyXdEAc6cP8EF4n21NMrnVYhAj8DY steve@MSI-GHOST  The key's randomart image is:  +---[RSA 3072]----+  | ... B\*=o+|  | . .o\*O=B|  | E =+#\*|  | . .. =+X|  | S . \*o|  | o o .|  | . +o + |  | o.o+ . |  | o. |  +----[SHA256]-----+ |

### Load Private key into ssh-agent

use ssh-add to add the key.

Note you will get the following error is using a path to the key

|  |
| --- |
| ssh-add ~\.ssh\id\_pi  ~\.ssh\pi\_rsa: No such file or directory |

so navigate to ~\.ssh and supply the keyname only

|  |
| --- |
| ssh-add id\_pi  Identity added: id\_pi (steve@MSI-GHOST) |

### Copy Public key to the Raspberry Pi

logon to the pi and create an .ssh directory for your user

|  |
| --- |
| cd ~  mkdir ~/.ssh |

Now from windows use scp to copy the public key:

|  |
| --- |
| PS C:\Users\conta\.ssh> **scp id\_pi.pub pi@192.168.0.153:~\.ssh\authorized\_keys**  pi@192.168.0.153's password:  id\_pi.pub 100% 570 92.8KB/s 00:00 |

### Testing it

Success!!

|  |
| --- |
| PS C:\Users\conta\.ssh> **ssh pi@192.168.0.153**  Linux pi4a 5.10.103-v7l+ #1529 SMP Tue Mar 8 12:24:00 GMT 2022 armv7l  The programs included with the Debian GNU/Linux system are free software;  the exact distribution terms for each program are described in the  individual files in /usr/share/doc/\*/copyright.  Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  permitted by applicable law.  Last login: Mon Jul 4 18:03:30 2022 from 192.168.0.201  Wi-Fi is currently blocked by rfkill.  Use raspi-config to set the country before use.  pi@pi4a:~ $ |

## Installing and Configuring git on the Pi

### Installing git

Update package index

|  |
| --- |
| sudo apt-get update |

Now install git

|  |
| --- |
| sudo apt-get install git |

### Creating a Bare Repository

**A remote repository is generally a bare repository**—a Git repository that has no working directory. Because the

repository is only used as a collaboration point, there is no reason to have a snapshot checked out on disk; it’s just

Git data. In the simplest terms, a bare repository is the contents of your project’s .git directory and nothing else.

**It is convention to give a bare repository name a .git extension**

NOTE:

you can technically push changes to and pull changes from individuals' repositories, doing so is discouraged because you can fairly easily confuse what they’re working on if you’re not careful

NOTE:

with git there is no server installation - i.e no client / server relationship. It is more peer to peer.

A bare repository however is just a central communication point with no project files just the git objects.

Create an empty repsository area

|  |
| --- |
| sudo mkdir /repo |

Change persimmison so everybody has access

|  |
| --- |
| sudo chmod -R 777 repo |

Initialize a bare repository

|  |
| --- |
| sudo git init --bare --shared *<repo\_name>* |

At this point a user can add the repository as a remote in the usual way, then push, pull and fetch as usual.

### Copying an existing local repository to a Remote respository

To initially set up any Git server, you have to export an existing repository into a new bare local repository

client:

|  |
| --- |
| **git clone --bare my\_project my\_project.git**  Cloning into bare repository 'my\_project.git'...  done. |

Now that you have a bare copy of your repository, all you need to do is put it on a server

client:

|  |
| --- |
| scp -r my\_project.git pi@192.168.0.153:/repo |

Git will automatically add group write permissions to a repository properly if you run the git init command

with the --shared option.

server:

|  |
| --- |
| cd /repo/my\_project.git  git init --bare --shared |

At this point a user can add the repository as a remote in the usual way, then push, pull and fetch as usual.

### Using a Centralized git user account

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

# APPENDIX A - Troubleshooting git over SSH

<https://stackoverflow.com/questions/25388499/how-can-i-run-git-push-pull-commands-with-ssh-verbose-mode>

|  |
| --- |
| git config core.sshCommand "ssh -v" |

# APPENDIX A - SSH on Raspberry Pi

Raspberry Pi OS has the SSH server disabled by default. It can be enabled manually from the desktop:

Launch Raspberry Pi Configuration from the Preferences menu

Navigate to the Interfaces tab

Select Enabled next to SSH

Click OK

Alternatively you can enable it from the terminal using the raspi-config application,

Enter sudo raspi-config in a terminal window

Select Interfacing Options

Navigate to and select SSH

Choose Yes

Select Ok

Choose Finish

Note

For headless setup, SSH can be enabled by placing a file named ssh, without any extension, onto the boot partition of the SD Card. When the Raspberry Pi boots, it looks for the ssh file. If it is found, SSH is enabled and the file is deleted. The content of the file does not matter; it could contain text, or nothing at all.

Also edit the following settings:

/etc/ssh/sshd\_config

|  |
| --- |
| ...  PubkeyAuthentication yes  ...  PasswordAuthentication yes  PermitEmptyPasswords yes  ... |

# APPENDIX A – ICONS USED

|  |  |
| --- | --- |
|  | This icon flags useful information or explains a shortcut to help you understand a concept |

|  |  |
| --- | --- |
|  | This icon explains technical details about the concept being explained |

|  |  |
| --- | --- |
|  | Try not to forget the material marked with this icon |

|  |  |
| --- | --- |
|  | Watch out! This icon flags common mistakes and problems that can be avoided |

  

|  |  |
| --- | --- |
| **RIGHT** | **WRONG** |
|  |  |

|  |
| --- |
| **RIGHT** |
|  |

|  |
| --- |
| **WRONG** |
|  |

**2 Column Table**

|  |  |
| --- | --- |
|  |  |
|  |  |

**3 Column Table**

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |

# APPENDIX B - PYCHARM SYMBOLS

<https://www.jetbrains.com/help/pycharm/symbols.html>

<https://jetbrains.design/intellij/resources/icons_list/>

