Lab: Configuring Gitsource on MacOS and Linux

Overview

Gitsource is the Akamai Git repository for enterprise teams. Git allows everyone in your team to easily collaborate within your Git repositories.

Gitsource accounts are referred to as "licenses." Without a license, you can only access public repositories. With a license, you can access repositories you have been granted access, which is controlled by the repositories' owners.

Objectives

Upon successful completion of this lab, you will...

1. Download, install and configure Git Bash
2. Create, commit and push a file to the shared repository

Prerequisites

1. Create and upload valid internal SSH key pair and have the internal key loaded into your session with the `ssh-add` command.
2. Have a valid Gitsource license.
   a. Verify you have a license here: https://git.source.akamai.com/dashboard
      If you do not see your avatar in the upper right corner in the blue bar, then you do not have a license.
   b. If you need a license, go here https://track.akamai.com/jira/servicedesk/customer/portal/13/create/371 then click on the "Create" button.

You can find more information on Gitsource access at the Gitsource FAQ page here: https://collaborate.akamai.com/confluence/display/STASH/git.source.akamai.com+FAQ

Download, Install and Configure Git Bash

Download and Install

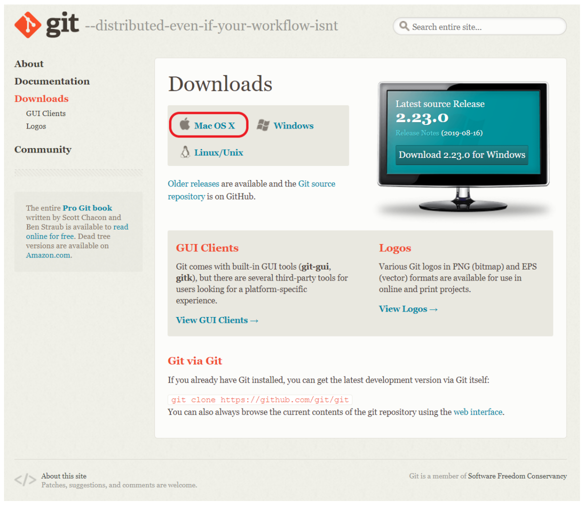
For MacOS

Go to https://git-scm.com/downloads then click on the "Mac OS X" link.

Your download should start automatically; if not, then click the appropriate link.

The downloaded file will be named *git-x.yy.z-intel-universal-mavericks.dmg*, where *x.yy.z* is the current version. When the download is complete, double-click on that file to begin installation.

You should click the "Agree" button to agree to the "Command Line Tools License Agreement."



Downloads

Older releases are available and the Git source repository is on GitHub.

The entire **Pro Git book** written by Scott Chacon and Ben Straub is available to read online for free. Dead tree versions are available on Amazon.com.

GUI Clients

Git comes with built-in GUI tools (**git-gui**, **gitk**), but there are several third-party tools for users looking for a platform-specific experience.

Logos

Various Git logos in PNG (bitmap) and EPS (vector) formats are available for use in online and print projects.

Git via Git

If you already have Git installed, you can get the latest development version via Git itself:

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

You can also always browse the current contents of the git repository using the web interface.

Patches, suggestions, and comments are welcome.

For Linux

For Linux, download Git directly from the Debian / Ubuntu package manager. This command requires super user privileges, therefore in a terminal window type the command where the `sudo apt-get` command means s uper user do `apt-get`. Note that below, the dollar sign ($) represents your system prompt and is not meant to be typed. Enter your system password when prompted. You should see messages similar to the following:

Bash Terminal

```
$ sudo apt-get install git
[sudo] password for <username>:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  git-man liberror-perl
Suggested packages:
  git-daemon-run | git-daemon-sysvinit git-doc git-el git-email git-gui gitk gitweb git-arch git-cvs git-
mediawiki git-svn
The following NEW packages will be installed:
  git git-man liberror-perl
0 upgraded, 3 newly installed, 0 to remove and 5 not upgraded.
Need to get 3,932 kB of archives.
After this operation, 25.6 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

After you type Y you should see something similar to below:

Bash Terminal

```
Get:1 http://atlantis.akamai.com/ubuntu xenial/main amd64 liberror-perl all 9.99-9.9 [19.6 kB]
Get:2 http://atlantis.akamai.com/ubuntu xenial-updates/main amd64 git-man all 9:9.9.9-0ubuntu9.9 [736 kB]
Get:3 http://atlantis.akamai.com/ubuntu xenial-updates/main amd64 git amd64 9:9.9.9-0ubuntu9.9 [3,176 kB]
Fetched 3,932 kB in 0s (31.9 MB/s)
Selecting previously unselected package liberror-perl.
(Reading database ... 223360 files and directories currently installed.)
Preparing to unpack .../liberror-perl_9.99-9.9_all.deb ...
Unpacking liberror-perl (9.99-9.9) ...
Selecting previously unselected package git-man.
Preparing to unpack .../git-man_1%3a9.9.9-0ubuntu9.9_all.deb ...
Unpacking git-man (9:9.9.9-0ubuntu9.9) ...
Selecting previously unselected package git.
Preparing to unpack .../git_1%3a9.9.9-0ubuntu9.9_amd64.deb ...
Unpacking git (9:9.9.9-0ubuntu9.9) ...
Processing triggers for man-db (9.9.9-1) ...
Setting up liberror-perl (9.99-9.9) ...
Setting up git-man (9:9.9.9-0ubuntu9.9) ...
Setting up git (9:9.9.9-0ubuntu9.9) ...
$
```

Configuration

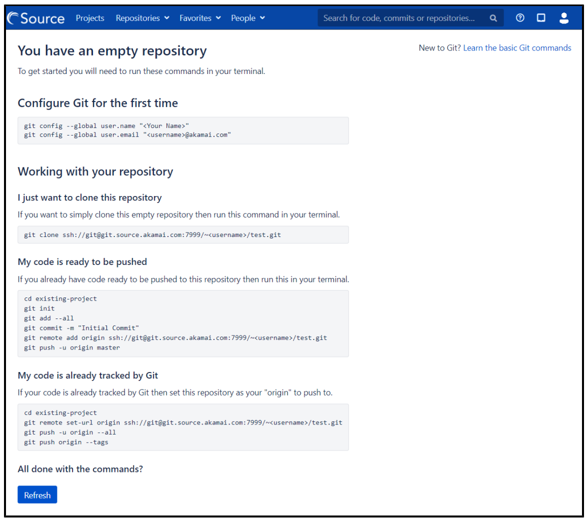
To complete the configuration, navigate to https://git.source.akamai.com. Once there, click on your avatar at the upper right corner and choose “View Profile”. You will be taken to a page with “You have an empty repository” across the top like the example to the right.

This page contains commands you need to execute in a terminal window.

Copy from the webpage the mail setup commands into the terminal window.

Bash Terminal

```
$ git config --global user.name "<Your Name>"
$ git config --global user.email
"<username>@akamai.com"
```



You have an empty repository

New to Git? Learn the basic Git commands

To get started you will need to run these commands in your terminal.

Configure Git for the first time

```
git config --global user.name "<Your Name>"
git config --global user.email "<username>@akamai.com"
```

Working with your repository

I just want to clone this repository

If you want to simply clone this empty repository then run this command in your terminal.

```
git clone ssh://git@git.source.akamai.com:7999/~<username>/test.git
```

My code is ready to be pushed

If you already have code ready to be pushed to this repository then run this in your terminal.

```
cd existing-project
git init
git add --all
git commit -m "Initial Commit"
git remote add origin ssh://git@git.source.akamai.com:7999/~<username>/test.git
git push -u origin master
```

My code is already tracked by Git

If your code is already tracked by Git then set this repository as your "origin" to push to.

```
cd existing-project
git remote set-url origin ssh://git@git.source.akamai.com:7999/~<username>/test.git
git push -u origin --all
git push origin --tags
```

All done with the commands?

Create, Commit and Push a File to the Shared Repository

Create a Repository

Start by creating a new repository. Create a directory called `Git` in your Documents directory, `/home/<username>/Documents/Git`. with the `mkdir` (make directory) command. Remember that you can use the tilde (~) to refer to your home directory, `/home/<username>`.

Bash Terminal

```
$ cd ~/Documents
$ mkdir Git
$ cd Git
```

This directory will serve as the home of all your local Git repositories.

From your Git Bash window, go to your new Git directory with the `cd` (change directory) command. You may need to type the entire path to this directory, but in most cases you can go there from your home directory (represented with the tilde, ~) with just `cd ~/Documents/Git` or from any directory. You may need to adjust the path if you have a custom setup.

Next, copy the following commands from the *I just want to clone this repository* section of your Confluence Git webpage (shown above) then paste them into your Git Bash window.

⚠ If your internal SSH key is uploaded correctly, the command shown will begin with `git clone ssh:`. If it instead begins with `git clone https:` then your key is not uploaded correctly.

Please go back to the SSH Keys lab to complete that setup.

Rarely, you have completed the key generation and upload process correctly, but your key was inadvertently not uploaded to the Git server. You can manually upload your key through the same Git webpage.

You will first need to get the contents of your internal public key into your clipboard. Use the `cat` command to display the contents of your internal public key then copy it to your clipboard (start at `ssh-rsa` and copy through the end of your comment, most likely the date you generated the key.

Bash Terminal

```
$ cd ~/.ssh
$ cat <username>_internal_20990101.pub
ssh-rsa AARAB3BzaC1yc2EAAAABKQAAAQEKghR925pAfufz9xHDzxXXzjQfOGA2k8yN5iPqHTizZX51gJO5
3EPjHGVdsvBE0Vvzow/NDy7Iu7bfSEuvOYcasn3jUYSjKYY6uFK6WlbeXAdXYAk56c6G1niUbsHCYoyU3n9w
+RuIUtTKkfhlB/ZKo0UrCglzb0R72QzatrVV6zUoFFNhSwb3jJhRFB0Y8BwQZ1YzknmZBkUCAAEZqcG6pgFx
cH2gpCguDcab3aEJ3NgIxM/+Ndcej/+I451/s5OvncmRUpA9/EXpkFKPd4N7TwdoTB1XYZntqLkQaGmbzL3Z
U+RT04HwOhG6IKtJO9QKPLUGNgI4R0+OvpzRFv1IyiQ== <username>-internal-key-20990101
$
```

On the Gitsource web page, click on your avatar in the upper right corner then click on “Manage account”. In the left pane, click on “SSH keys”. Now click on the “Add key” button. You will now see a window titled “Add public key”. Paste your entire public key content into this window then click the “Add key” button.

Clone a new empty repository called “test”. with the `git clone ssh://git...` command. After you type the command, you will see a response with information about the Git server. You need to verify that the server’s MD5 fingerprint returned to you in the terminal window is “d8:a6:ac:c0:c7:0f:ca:36:7f:ab:9b:6e:6e:0e:4d:65” or the SHA256 fingerprint is “m+eRlAxYgdbvGBbpWNJdFVHClrrfkMDnCpjMbNr7Sj0”.

Since this is most likely the first time you are connecting to this server, you will be asked if you want to add this server’s key to your machine’s list of trusted servers. If the MD5 or SHA256 fingerprint matches what is above and below in the example, then type `yes` to complete the cloning.

Bash Terminal

```
$ git clone ssh://git@git.source.akamai.com:7999/~dnekrasz/test.git
Cloning into 'test'...
The authenticity of host '[git.source.akamai.com]:7999 ([172.27.117.25]:7999)' can't be established.
RSA key fingerprint is SHA256:m+eRlAxYgdbvGBbpWNJdFVHClrrfkMDnCpjMbNr7Sj0.
Are you sure you want to continue connecting (yes/no)?
```

After you type `yes` you should see:

Bash Terminal

```
Warning: Permanently added '[git.source.akamai.com]:7999,[172.27.117.25]:7999' (RSA) to the list of known
hosts.
warning: You appear to have cloned an empty repository.
Checking connectivity... done.
$
```

Commit and Push

First, make a file we will use to push to your new test repository. This file can be anything, such as a Word document or a text file created in Notepad or some other text editor. Ensure you save this file in your new test repository, `C:\Users\<username>\Documents\Git\test`.

To create a text file from the Git Bash terminal window, navigate to your new repository folder if not there and use the touch command to create a new empty file called “readme.txt”. You can use a text editor to add text to this file or you can add text from the Git Bash terminal window with the `echo` command. Run the git status command to verify your new file, “readme.txt”, is currently untracked by Git.

Bash Terminal

```
$ cd ~/Documents/Git/test
$ echo "First change to test" > readme.txt
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        readme.txt

nothing added to commit but untracked files present (use "git add" to track)

$
```

Add this file with the `git add` command. Then run the `git status` command to verify the file is now tracked but uncommitted by Git.

Bash Terminal

```
$ git add readme.txt
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   readme.txt

$
```

Commit the file using the `git commit` command with the `-m` option to add a simple message to the commit. Adding a description of the commit in a message is good practice.

Bash Terminal

```
$ git commit -m "my first readme commit"
[master (root-commit) 7ac52f2] my first readme commit
 1 file changed, 6 insertions(+)
 create mode 199999 readme.txt
$
```

The file is now committed but still needs to be pushed to the server. Use the `push` command:

Bash Terminal

```
$ git push origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 106 bytes | 106.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
remote: There is an announcement for you to review. Please visit https://git.source.akamai.com/users
/<username>/repos/test/browse
remote:
remote:
remote:
To ssh://git.source.akamai.com:7999/~<username>/test.git
 * [new branch]      master -> master
Branch master set up to track remote branch master from origin.
$
```

You may be prompted for your internal key passphrase for the `push` command to successfully complete. If so, enter it.

Refresh your Git webpage with the *Refresh* button at the bottom of the page and you should see your file in the repository and the message you added with your commit.

Conclusion

Review Objectives

Please review the lab's objectives and verify you have met them by checking off each one. If you were unable to meet any of the objectives, please describe which one(s) you could not meet and why in the Feedback section.

- ☐ Download, install and configure Git Bash
- ☐ Create, commit and push a file to the shared repository

Feedback

If you have any questions, comments or feedback, please add a comment below. This page is monitored and your comment will be addressed.