Git and BitBucket

Medhat Dawoud

/med7atdawoud
/med7atdawoud

9 February 2013

This workshop will include

- Source Code Management Systems
- Web-Based Code Hosting
- Popular open source projects using SCM
- Intro to Git
- Let’s try Git Commands
- Bitbucket as a web code server and management tool

Source Code Management System

- *It’s a way that we use to store projects’ source code files in a tree of versions.*
- *With no deal, all those SCMs creators build them on the same rules and structure.*
- *A lot of popular open source projects source codes are collected together all over the world using SCM systems.*

SCM Systems Examples

Lots of this type of software was created like:

CVS

BitKeeper
SOURCE MANAGEMENT

mercurial

monotone

FOSSIL

Bazaar

Web based Code Hosting

- *We can use SCMS on our machine locally that manage the code for you and create versions.*
- *but you can also host your code on web, this solution make it very secure and no lose of data under any circumstances.*

Web based Code Hosting Examples

A lot of Web-Based Code Servers provide Free and paid services Like:

CloudForge by CollabNet

github SOCIAL CODING

CodePlex

assembla Accelerating software development

JavaForge Scalable Project Hosting

Google code

sourceforge

Popular projects that use web-based code Hosting and Source Code Management Systems

9 February 2013

Source1: https://git.wiki.kernel.org/index.php/GitProjects

Source2: https://github.com/popular/starred

Popular projects that use web-based code Hosting and Source Code Management Systems

debian GNU/Linux

fedora

GNOME

android

9 February 2013

Source1: https://git.wiki.kernel.org/index.php/GitProjects

Source2: https://github.com/popular/starred

Popular projects that use web-based code Hosting and Source Code Management Systems

django

Code Igniter

debian GNU/Linux

fedora

GNOME

android

symfony

CakePHP

phpBB creating communities worldwide

Source1: https://git.wiki.kernel.org/index.php/GitProjects

Source2: https://github.com/popular/starred

Popular projects that use web-based code Hosting and Source Code Management Systems

UNDERSCORE.JS

Code Igniter

django

Backbone.js

Scalatra

debian
GNU/Linux

fedora

android

node JS

GNOME

prototype

symfony

HTML 5

RAILS

phpBB
creating communities worldwide

jQuery
write less, do more.

CakePHP

9 February 2013

Source1: https://git.wiki.kernel.org/index.php/GitProjects

Source2: https://github.com/popular/starred

SCMS

- *We approved before when we start that all SCMSs are built on the same structure without any deal between them.*

- *Those all are called Source Code Management Systems (SCMSs).*

- **Main task of any SCMS is to:**
  - Track changes to files.
  - Repository / database of changes
  - Working directory / current state

SCMS Operations

We can sort operations that we can do with any SCMS into 4 main categories:

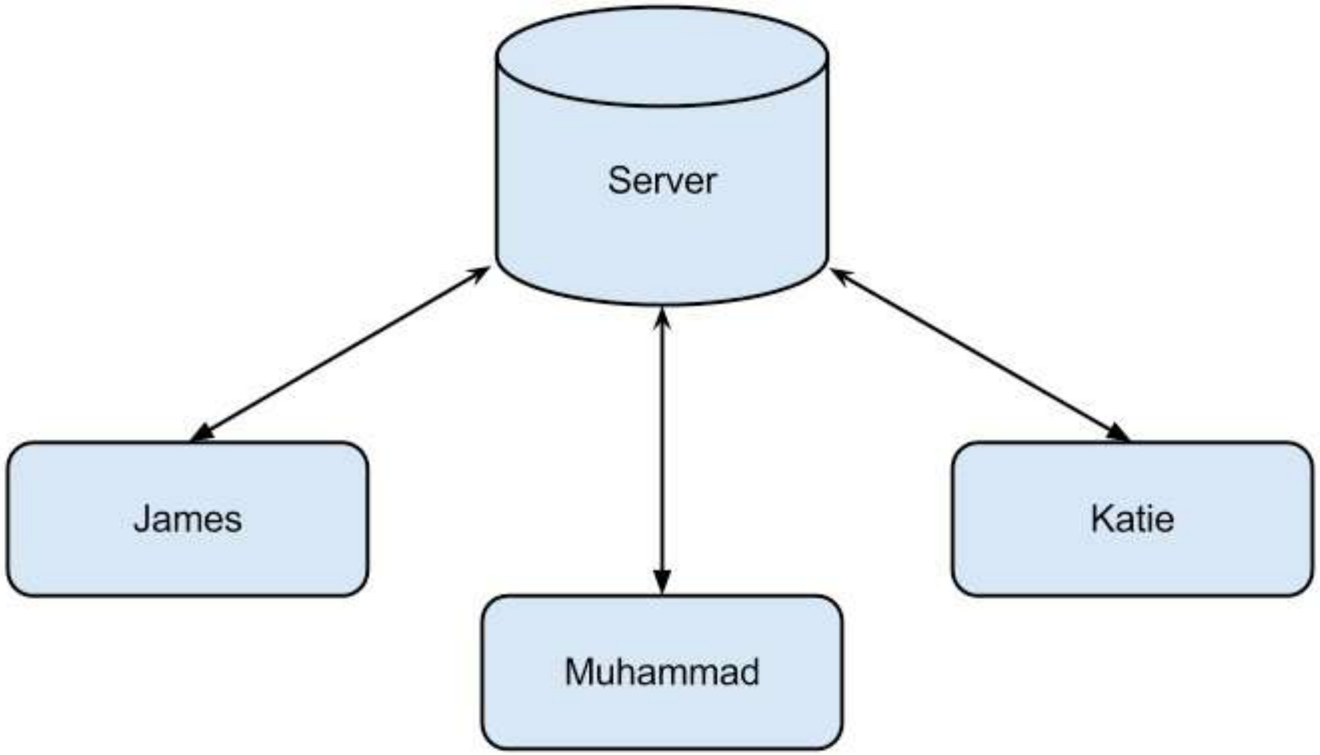
- **Bootstrap** (init, checkout, switch branch)
- **Modify** (add, delete, rename, commit)
- **Information** (status, diff, log)
- **Reference** (tag, branch)

Types of SCMS

There are 2 types of SCM systems:

- ❖ Centralized SCM
- ❖ Distributed SCM

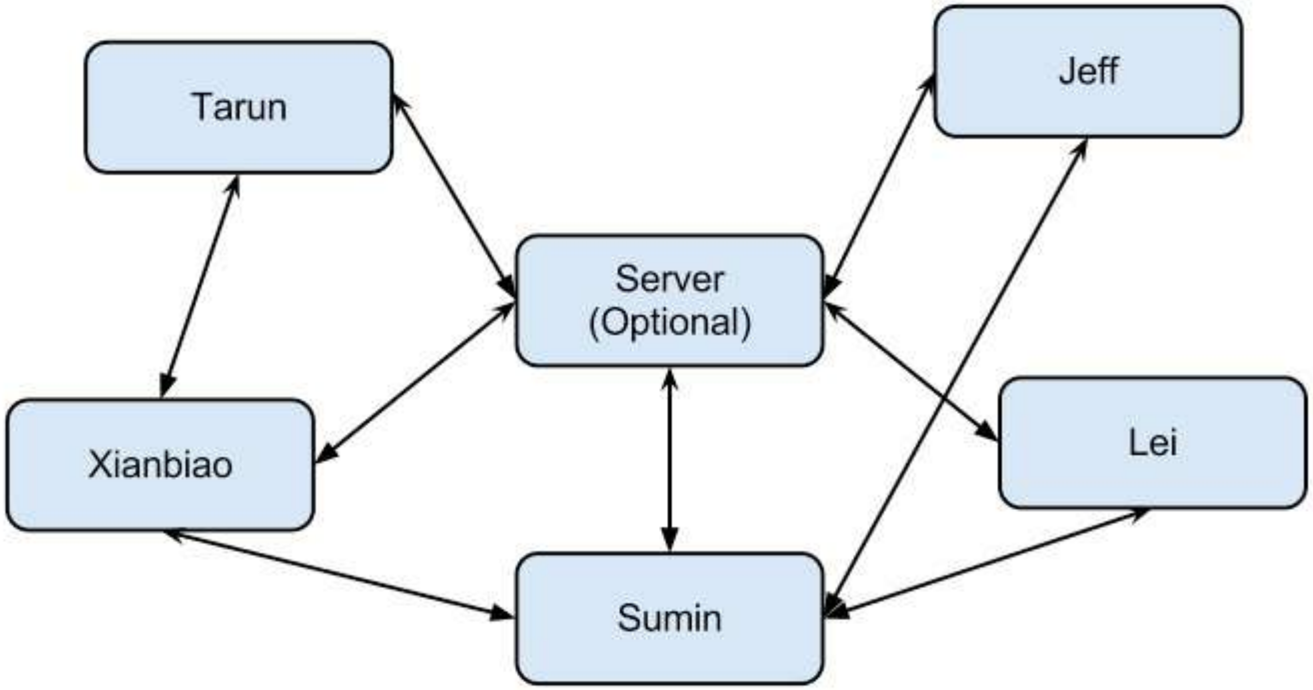
Centralized SCM



Centralized SCM

- *Examples: Subversion, CVS, etc.*
- *Everything goes to the server, commit changes to the sever, checkout the latest revision from the server.*
- *No direct exchange between developers*
- **Operations require server, there are some drawbacks:**
  - Single point of failure
  - Bottleneck

Decentralized SCM



Decentralized SCM

- *Examples: Git, Mercurial, Bazaar, etc.*
- *Each copy of repository is identical and self-sufficient*
- *No need for a central server, but one may choose to have one*
- *Developers may directly exchange change sets over Wi-Fi at a local coffee shop :D*
- **Workflow :**
  - Clone
  - Pull / fetch
  - push

Ok, Let's Start that workshop now

- Yes, it was just an intro to the Source Code Management Mechanism and the Web-Based Code Servers.
- Now to start talking in our session, I packed Git as our SCM or VCS and BitBucket as a Web-based Code Server and management tool will be discussed later.

Are you Ready?

What is Git ?

- ✓ Decentralized or Destributed Source Code Management(SCMS).
- ✓ Superior branching and merging mechanism.
- ✓ Support various protection devices against corruption.
- ✓ Supported by various code servers.

Git History

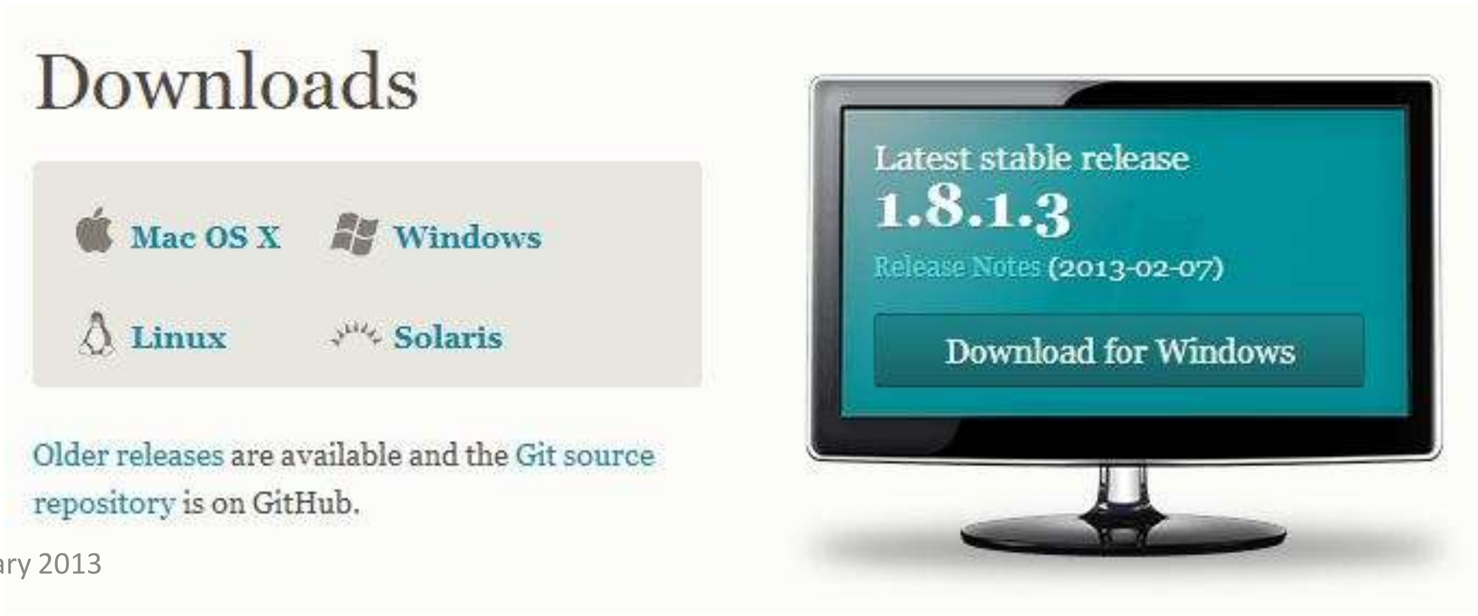
- **2002**
  - Linus uses BitKeeper to track Linux.
  - And BK gets Better, and Linux scale better.
- **April 6, 2005**
  - BitKeeper drops free license.
  - Linus write his own SCM, Git.
- **April 18, 2005**
  - Git can merge.
- **June 16, 2006**
  - Git is officially used to track Linux.
- **Feb 14, 2007**
  - Git 1.5.0 is released.
  - Major usability efforts.

*“ Nothing is perfect. Git is just *closer* to perfect than any other SCM out there ”*

- Linus

Git First use

- *If you are using Git for the first time, you will need to download the server into your machine according to your operation system.*
- *Just go to: http://git-scm.com/downloads and choose the suitable download and install it.*

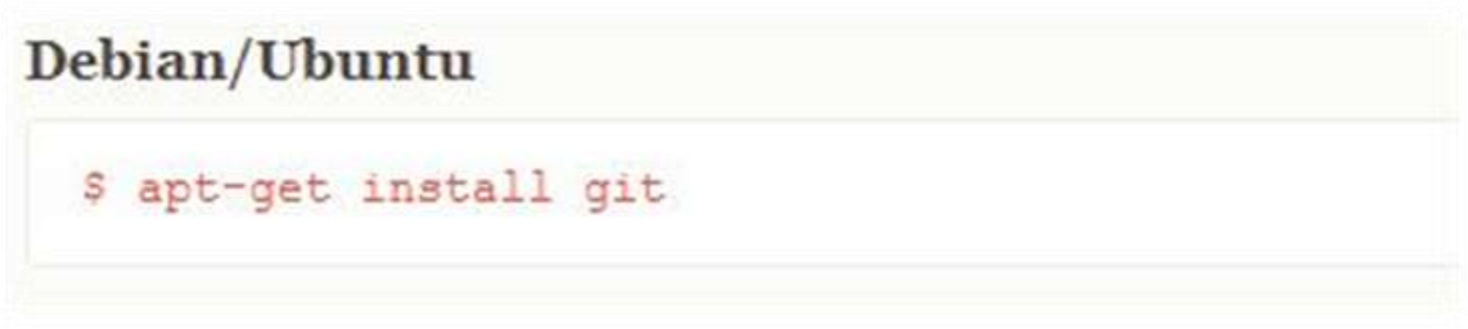


Older releases are available and the Git source repository is on GitHub.

Latest stable release
1.8.1.3
Release Notes (2013-02-07)

Git First use continue ..

- *If you are using linux you will find a command that will install Git from terminal.*
- *For example if you are using Ubuntu you will write this command in you terminal:*



Debian/Ubuntu

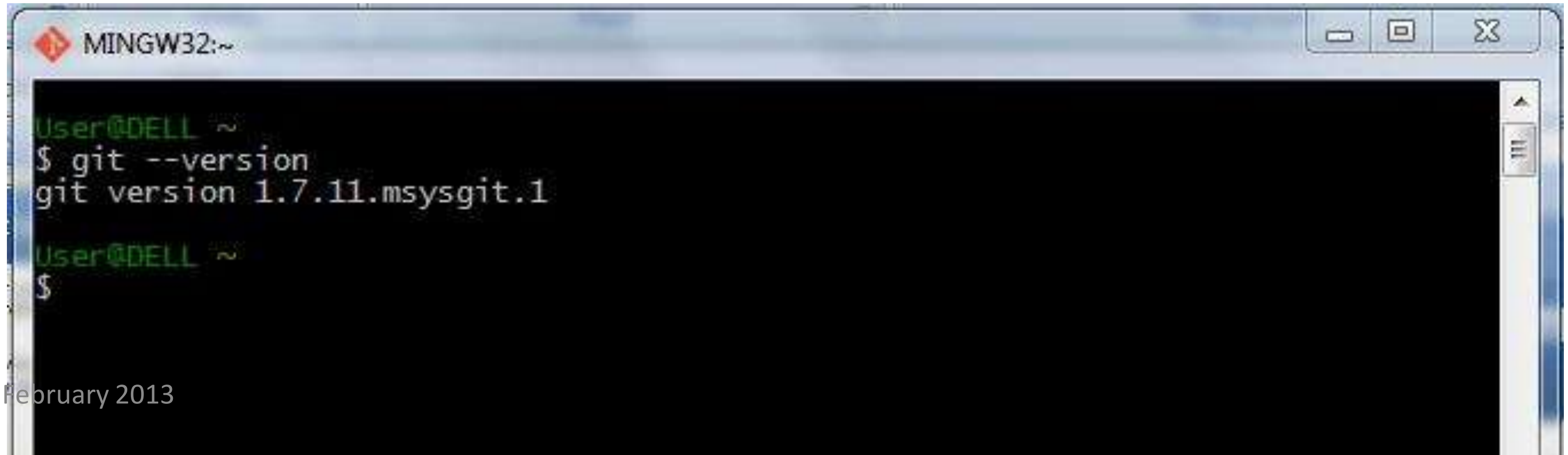
```
$ apt-get install git
```

Git Command line

- Now you have installed Git on your machine and you can go to terminal or CMD according to you OS and write any command, for example:

  ```
  $ git --version
  ```

  this command returned the version of Git you have installed, you should find the result as follow:



MINGW32:~

```
User@DELL ~
$ git --version
git version 1.7.11.msysgit.1

User@DELL ~
$
```

Git Commands

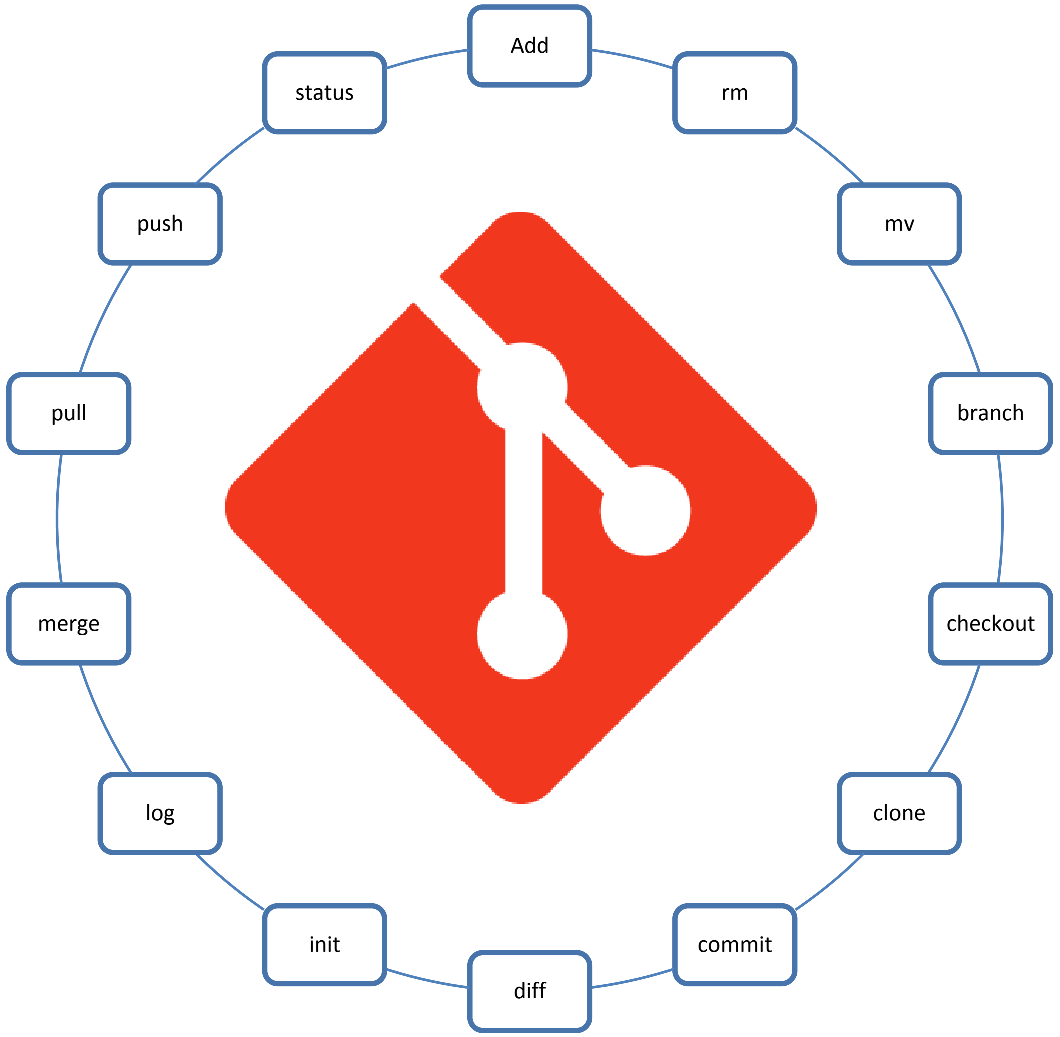
- *Ok, there is a general structure of the git command that all commands use as follow:*

```
$ git <options> command <options>
```

- *Git includes approximately 137 commands.*

- *Actually we don't use all of them every day, so I'll review here the every day use commands and let you check others.*

Git Common commands



Git Help

- *For any new command for you and need a brief documentation for it use this:*

```
$ git <command> -h
```

- *When you need a complete help you should write the following command and it will open a web page locally with the full docs for this command:*

```
$ git <command> --help
```

or

```
$ git help <command>
```

Git Bootstrap

- *Open the project work space (directory) and run this command inside it:*

  ```
  $ git init
  ```

  this will create .git *directory*

- *This directory* (.git) *include all meta data about versions and commits, working trees, changes, all configurations, … .*

Git Staging

- *Staging means specifying files that you will commit to the server.*

- Additions:

```
$ git add file        #This add a specific file
$ git add .           #This add all changed files
```

- Removal:

```
$ git rm file         #This removes a specific file
```

- Renames:

```
$ git mv old new      #This renames a specific file
```

.gitignore file

- You can create a .gitignore file in your project directory and add files or directories that you need not to add to the server, examples for unwanted files:
  - *Automatically generated code (e.g. R.java for Android)*
  - *Settings folder of editors that is created automatically.*
  - *If you are using any dependences on other libraries like in PHP you can add them to composer.json and ignore them.*
  - *Or any other unwanted files.*

- So when you use add all files, git will automatically ignore the list of files you have written in this file.

Git Commit

- Commit means to apply changes of staged files or all files to the repository locally.

- Commit changes must provided by a message that you explain in what is the changes in your commit from the last version:

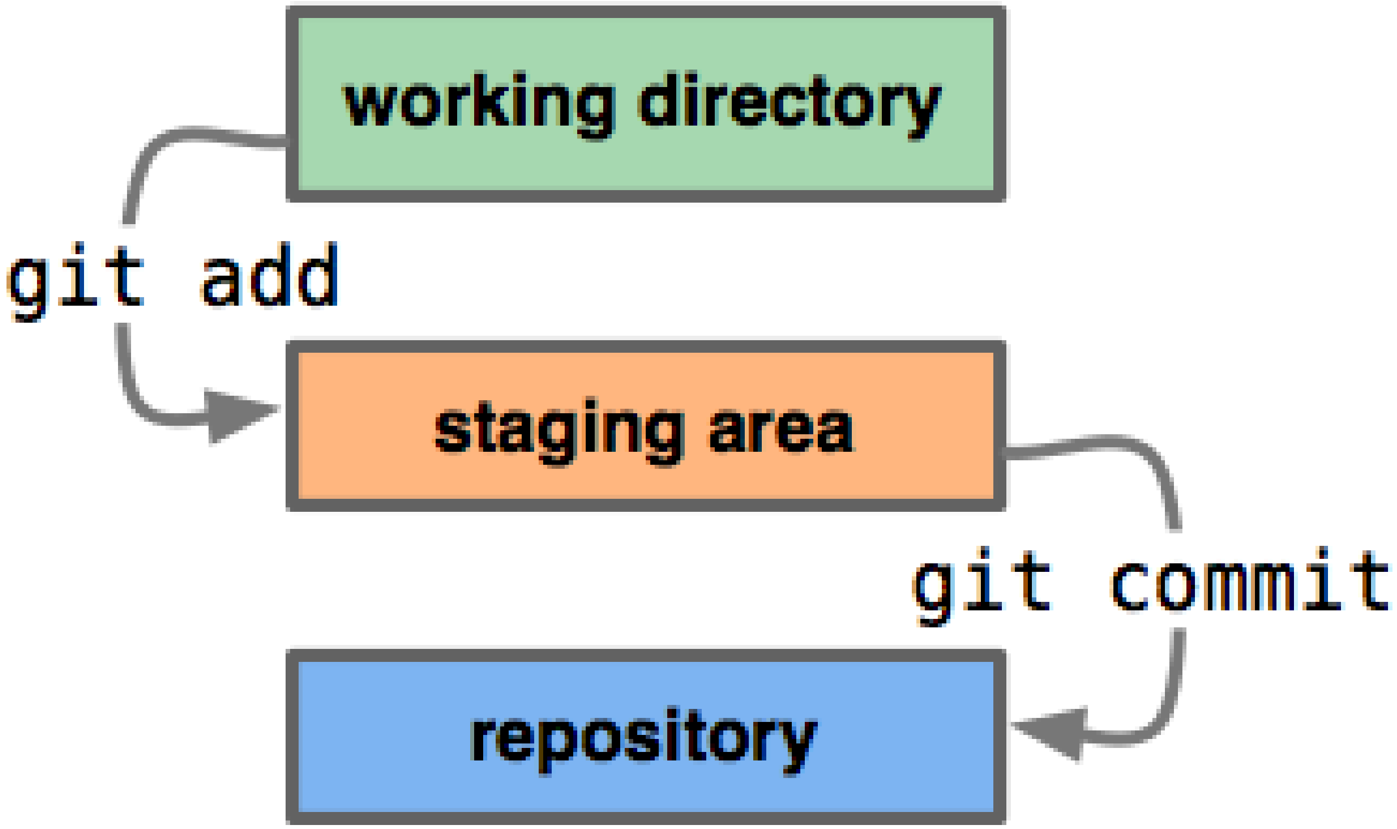
```
$ git commit –m ”this is the message”
```

- The above command commit only the staged files but if you want to commit all files you should use this:

```
$ git commit -a –m ”this is the message”
```

- This means that the commit command here in applied on your machine only ??? …. The answer is yes

Git work flow from work to commit



Git information

- **<u>You can use status:</u>**

```
$ git status
```

  to shows :

  - *Staged*
  - *Unstaged*
  - *Untracked*

- **<u>You can use show:</u>**

```
$ git show
```

  that shows the last commit information, changes, author, and date. It has some more configurations that customize the result.

Git information

<u>Git show additions:</u>

- *Just shows stats*

```
$ git show --stat
```

- *Just shows name and status*

```
$ git show --name-status
```

Git information

- To review the latest commits or even all commits on some repository we use *log* command as follow:

```
$ git log
```

- But you can also limit it to review commits in a specific period or in some branch or last 5 for example:

```
$ git log -5
```

or

```
$ git log -1 master@{yesterday}
```

or

```
$ git log --author=medhatdawoud
```

- There are lots of additions see in docs or help.

References

Tags
Branches

References

- *References are used to point to commits.*

  - ✓ *To get the local branches we use:*

    ```
    $ git branch -l
    ```

  - ✓ *To get the remote branches we use:*

    ```
    $ git branch -r
    ```

  - ✓ *And to get the local tags we use:*

    ```
    $ git tag -l
    ```

References

- *Creating new branch to HEAD:*

  ```
  $ git branch name
  ```

  new branch “name” on HEAD

- *Creating new branch to a commit:*

  ```
  $ git branch name commit
  ```

  new branch “name” on that commit

References

- *Switching to branch:*

```
$ git checkout name
```

- *We have option of creating and switching in the same command:*

```
$ git checkout -b name
```

- *If you are switching to a branch that has changes, the switching might gives error, then you should merge with switch:*

```
$ git checkout -m name
```

Merging

- *If your HEAD is referring to a branch and want to merge it to other branch, simply use the following command, assume we have A, B branches,*

```
$ git checkout A
$ git merge B
```

Assume that we have A, B, C branches and want to merge them all in one command.

```
$ git checkout A
$ git merge B C
```

Cloning

- *If you have a remote code server on some host and want to get that repository on local, you just want to write this:*

```
$ git clone <remote>
```

- *This will create a directory to the current root, with the same name of the repository you are cloning.*

Let’s Play with Git

9 February 2013

What's Next ?

- *Git is already installed into some editors like eclipse, aptana, … Search for your editor installation.*

- *There is a good free book for git, I recommend it for you to be more efficient in using Git as a Source Code Management System, check it from this link:* http://progit.org

An other Easy and Fast Solution

- *I'll tell you about another easy solution for windows users, it's a git client with great GUI that makes every thing for you, it's TortoiseGit*

- *Simply go to this link, download and install:*

  http://code.google.com/p/tortoisegit/wiki/Download
- *Try it and I will write some more articles about that later on my blog.*
- *There are 4 more clients check them on* git site.

BitBucket

- BitBucket is a web-based code server and also a great management tool for software projects.

- On 29 September 2010, Bitbucket was acquired by VC-funded Atlassian. Initially, Bitbucket only offered hosting support for Mercurial projects. On 3 October 2011, Bitbucket officially announced support for Git hosting.

Why Web-Based Code Hosting?

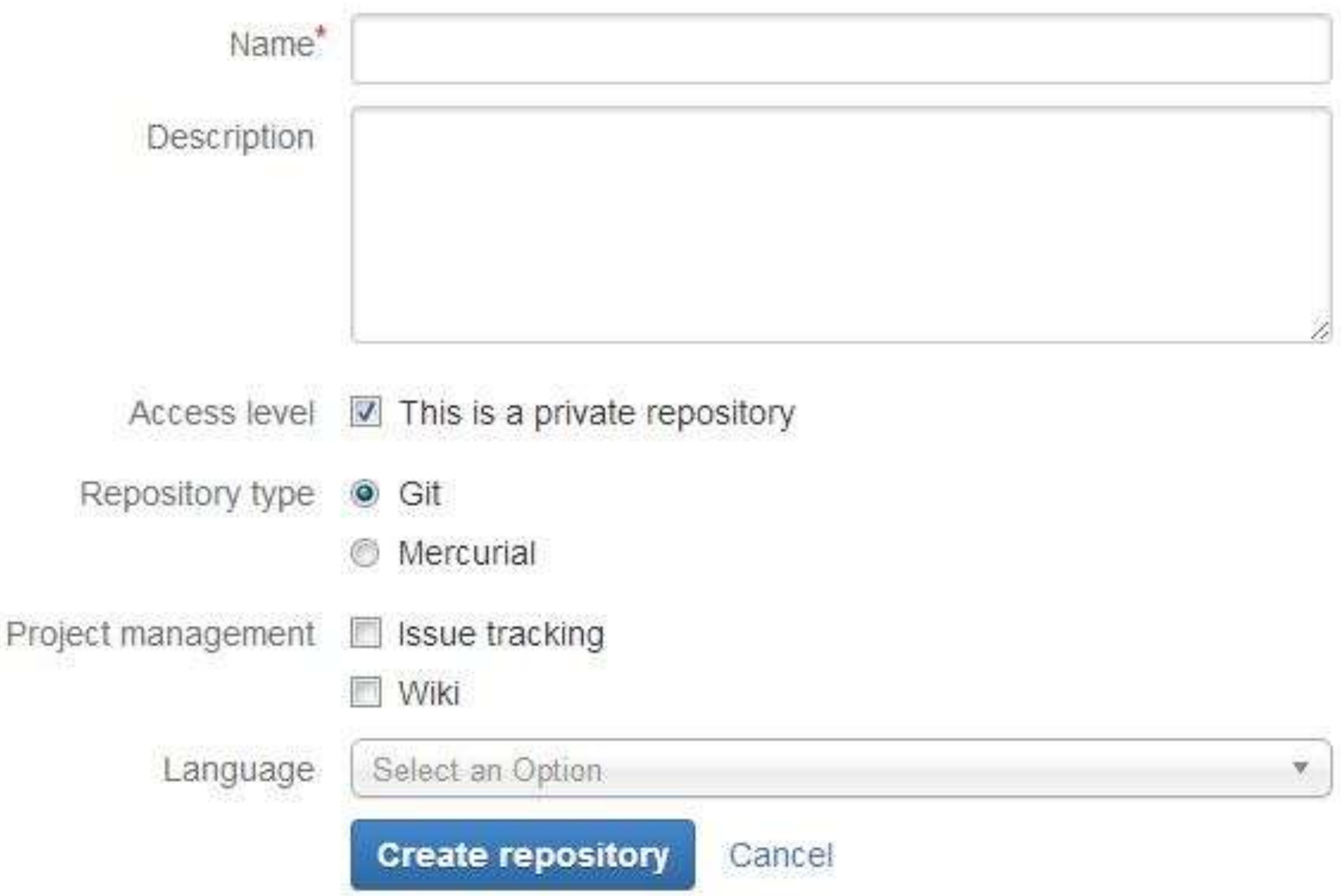
- ✓ Not to be confused with a version control system (or SCM system).
- ✓ Not a necessity, but good to have for more effective collaboration

Why BitBucket?

- Bitbucket is completely free if you have a .edu email address.

- Gives any one any number of repositories, free for only 5 users, otherwise see the payment on their site.
- Site: https://bitbucket.org

Creating Repository


Name*

Description

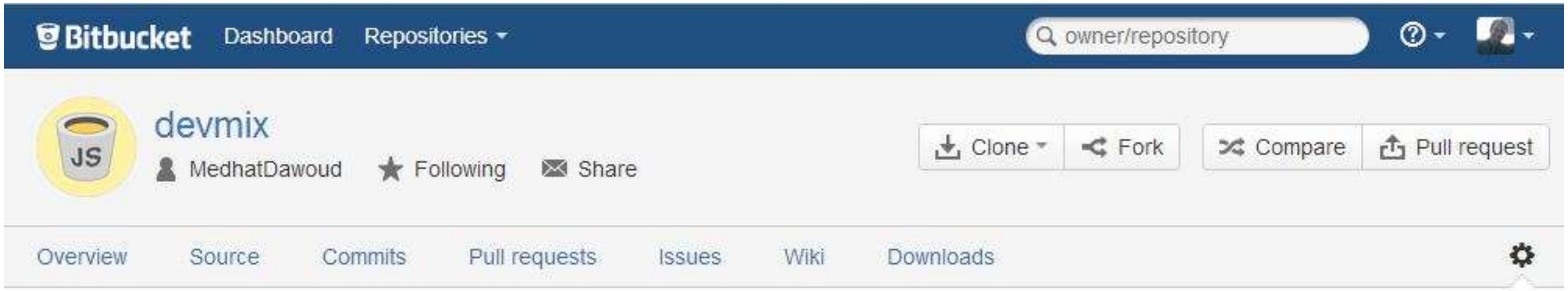
Access level [✓ This is a private repository]

Repository type [✓ Git] Mercurial

Project management Issue tracking Wiki

Language Select an Option

Repository Page



- *Notice these buttons in the right.*
- *Simple design that gives you only what you want from a code hosting, no noisy design.*
- *Notice the menu (overview, source, commits, pull requests, issues, wiki, downloads)*
  - *This menu has the most important functionality that bitbucket provide for us.*
  - *In the right of the menu is the settings of repository*

Let’s get a tour in the site features

Try Demos with Online Repositories

End of the workshop

Thanks

Medhat Dawoud

www.medhatdawoud.com

/med7atdawoud

/med7atdawoud

9 February 2013