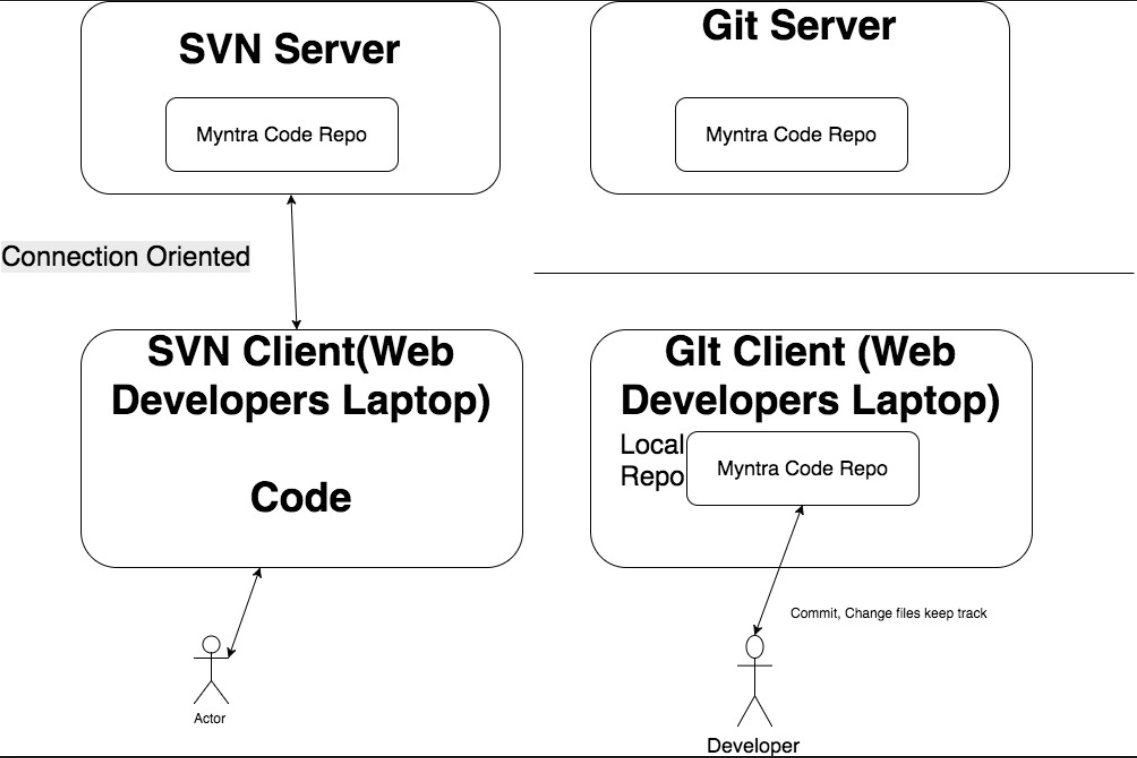
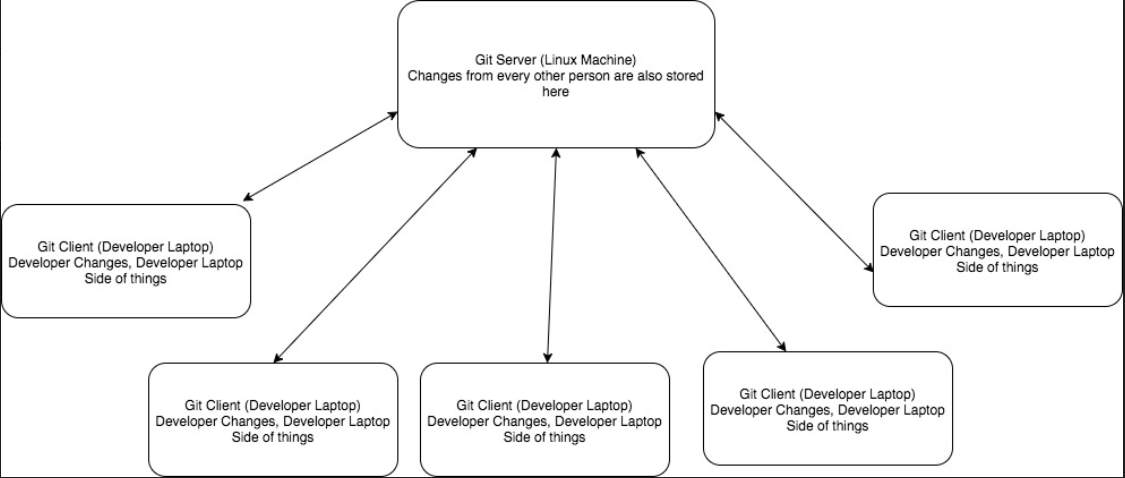
GIT





Gitlab setup and installation

GitLab Installation strategy overview

1. Download repository from git lab site
2. Install gitlab
3. Configure gitlab

**Details**

Prerequisite Once the new ubantu machine is setup

1. Perform the below command

*apt-get update && apt-get dist-upgrade -y*

1. change the machine name in /etc/hostnames

Once the above is done, follow the steps given in below URL

<https://about.gitlab.com/installation/#ubuntu>

1. Install and configure the dependency with the below command

apt-get install curl openssh-server ca-certificates postfix -y

1. Add the git lab package repository

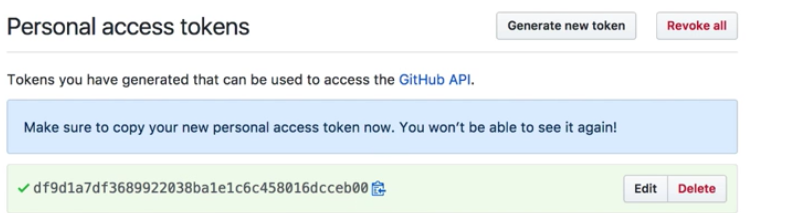
curl -sS https://packages.gitlab.com/install/repositories/gitlab/gitlab-ce/script.deb.sh | sudo bash

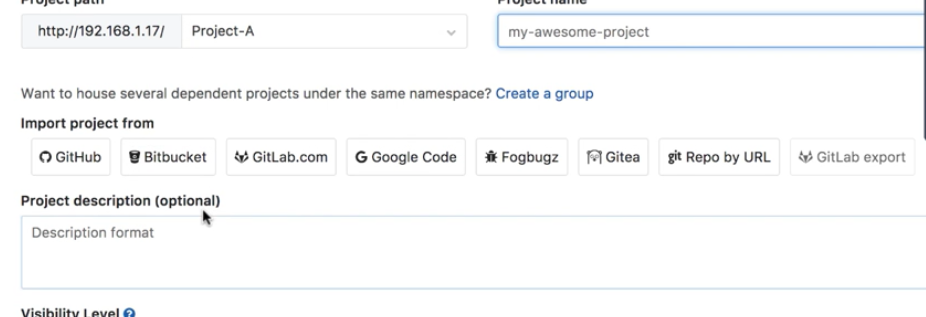
1. Once the repository is setup by above command. Run the below command to update it
2. apt-get update –y
3. Now we can search the gitlab in the apt cache with the below command (optional)
4. Apt-cache search gitlab-ce( with this search command you can search any software in apt cache)
5. Now install the gitlab with the below command
6. apt-get install gitlab-ce
7. The above command just install the basic gitlab with the below message “It looks like git lab has not been configured yet”
8. The above basic installed can be confimed by going into /etc/gitlab
9. Now before doing the remaining installation of gitlab, first do the reconfiguration of gitlab like below
10. Check the IP of the machine (example 192.168.169.5)
11. Now go to the below location and change the **external\_url and replace the site url to ip address like external\_url ‘http://192.168.169.5’**
12. Vim /etc/gitlab/gitlab.rb
13. Now re-configure the gitlab with the below command
14. **gitlab-ctl reconfigure**
15. Once gitlab is reconfigured, check the status with the below command
16. /opt/gitlab/bin/gitlab-ctl status ( the same can be used to start, stop gitlab)
17. Now Gitlab is up and running, go to the above ip address in IE and configure the new password
18. Here in this case the username for github is **root** and password is **redhat123**

GitLab Roles and responsibilities

1. Can setup/install gitlab server on any host machine like ubantu/centOS or on cloud EC2 instances of ubantu/centOS from scratch single hand idly.
2. After install , can set up all the configuration in GIT lab like creating group/user defining permission levels to user etc.
3. Can import/migrate the code repository from any existing version control system like SVN, github to Gitlab and get the business going forward from there.

**Import/Migrate code repository from github to gitlab**

1. Login Go to github 🡪 setting 🡪 personal access tokens🡪 generate a new personal access token(this token is like password)🡪 select all scopes
2. 
3. Now go to gitlab and groups 🡪 new project 🡪 import from git hub-🡪 give access token generated🡪 and import



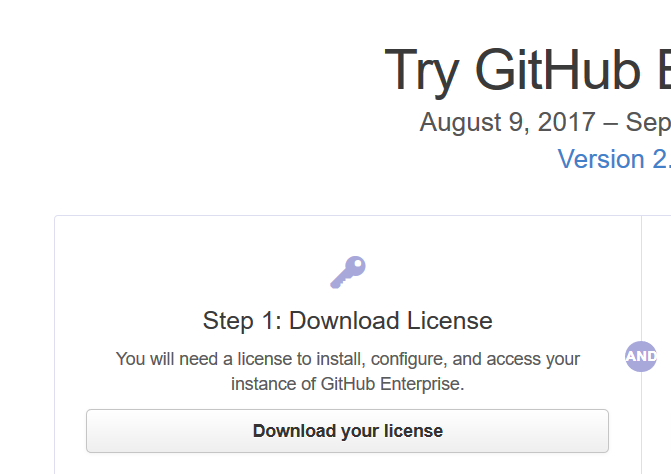
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

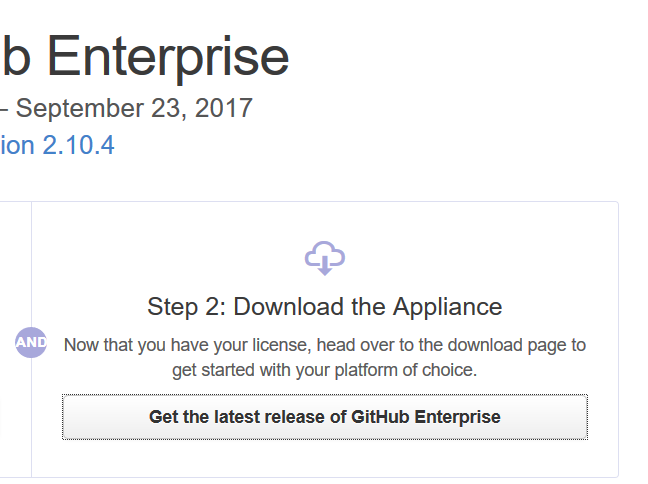
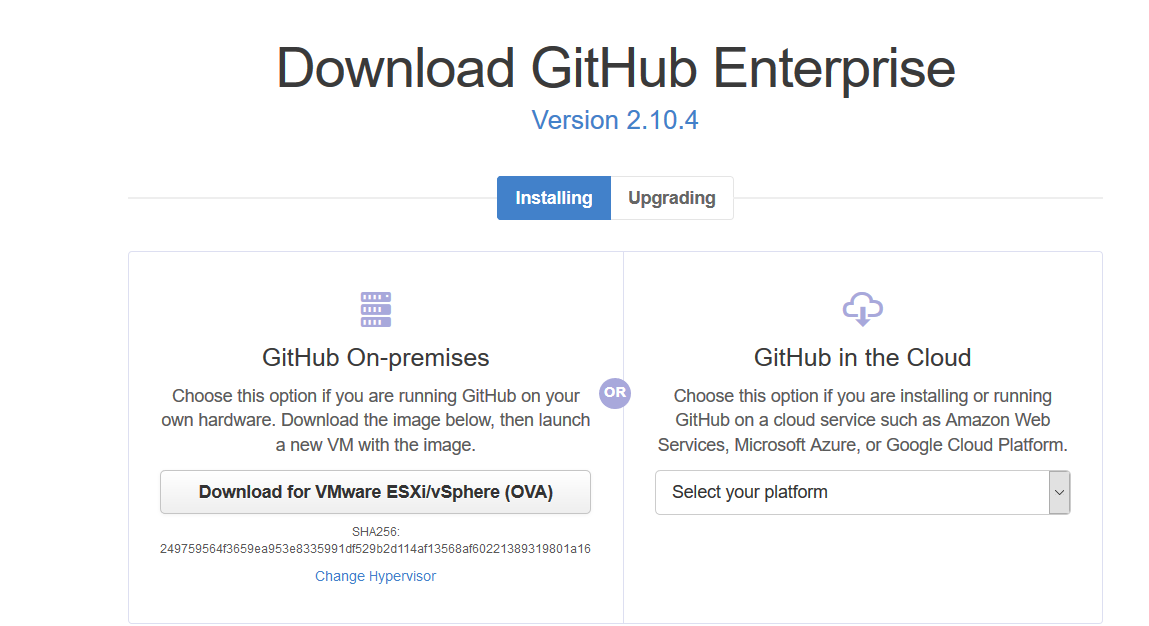
GIT-HUB download and setup

***Prerequisite – download VM ware hyprviser esxi as github is heavy setup and cannot be run in vm hosted on Oracle VM box ( note – vmware esxi is equivalent to Oracle vm box. esxi is used on large servers in data centres)***

***First install vmware workstation( esxi will work only the workstation not directly on windows)***

1. Go to Enterprise github site (<https://enterprise.github.com/>)
2. Download the license



1. Now download the github OVA file
2. 
3. Download the vmware exsi github OVA its 1 gb file( for cloud download the different one)
4. 
5. What is fork project in github.
6. Ans – here you can search any project in github and if you form it then it will be cloned to your hithub account…this basically means downloading to you github account

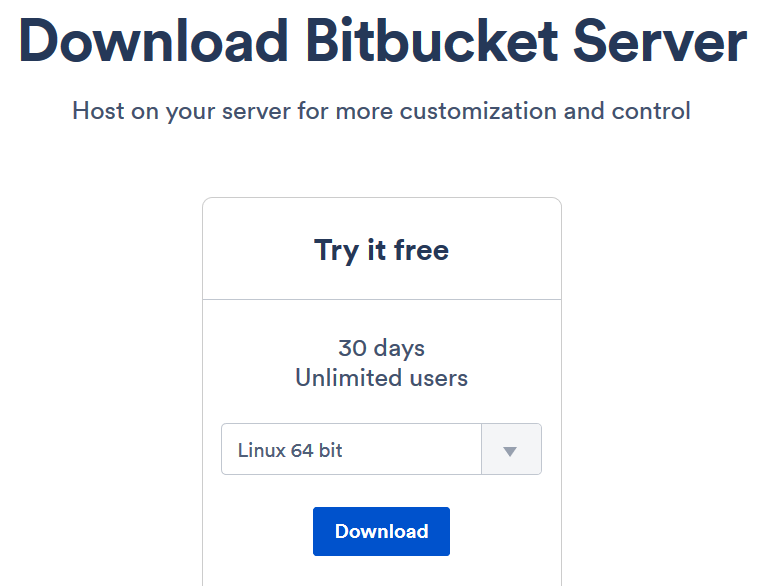
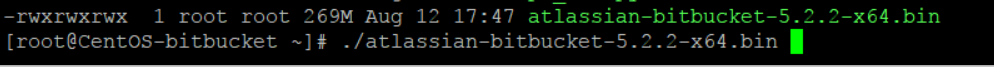
BitBucket Installation

BitBucket Installation strategy overview

1. Download installer from bitbucket site
2. Install bitbucket

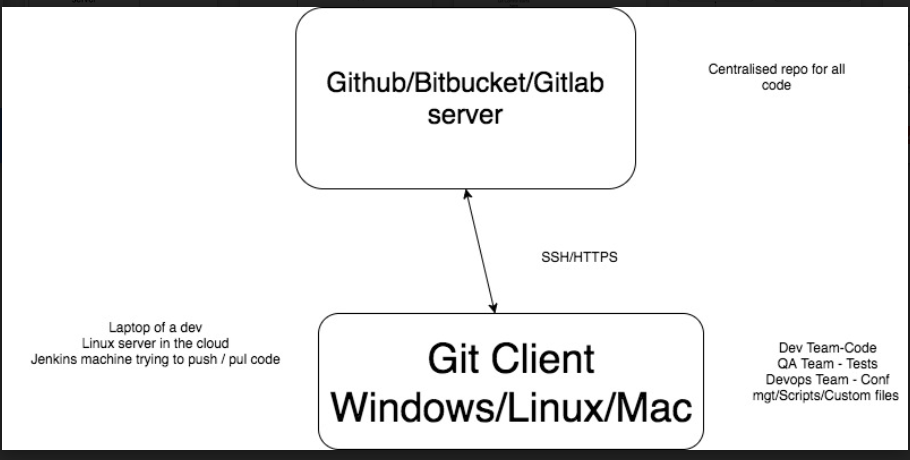
**Details**

**Prerequisite** - Install the git(for centOS only as it’s already present in ubantu package) before installing the bitbucket (Latest version can be found at <https://www.kernel.org/pub/software/scm/git/>)

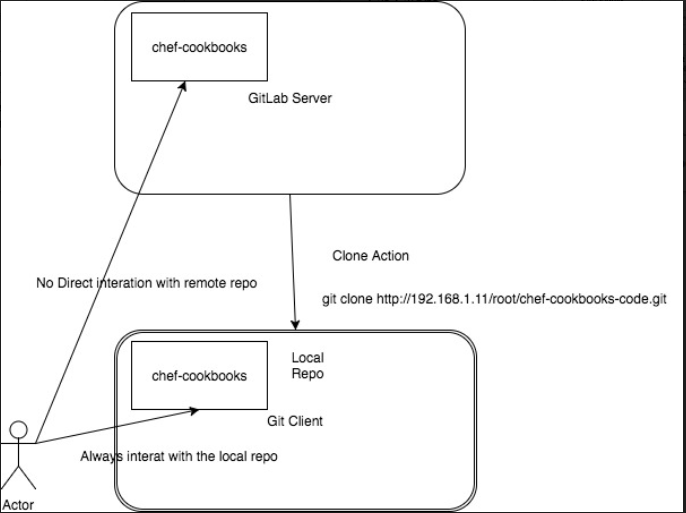
1. Download bitbucket installer file from location <https://www.atlassian.com/software/bitbucket/download>
2. 
3. Give the installer the full access and run it as below
4. 
5. Follow the prompts and install it.
6. Bitbucket is installed at /opt/atlassian/bitbucket/5.2.2
7. Bitbucket data location /var/atlassian/application-data/bitbucket
8. **Additional information** - Bitbucket also requires ports 7992 and 7993 are available to run an embedded Elasticsearch instance that provides search functionality to Bitbucket.HTTP Port Number [7990]
9. **Once installed, launch the URL in explorer (**[**http://192.168.169.6:7990/**](http://192.168.169.6:7990/)**)**

GIT Client

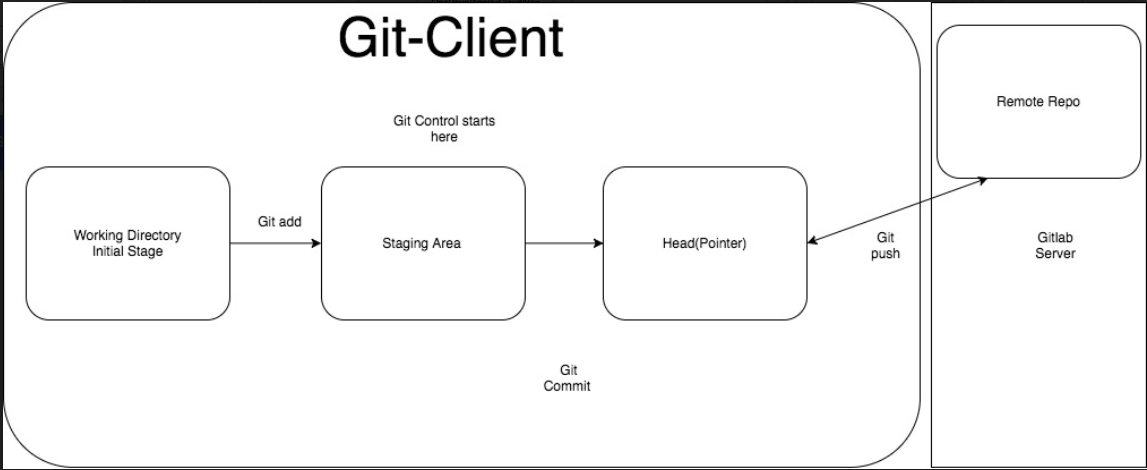
**Summary** – Git client is installed at user end which is used to pull or push the code in these GIT servers (GITLAB, GITHUB and BitBucket)



**GIT cloning process**



**Git Client Request lifecycle**



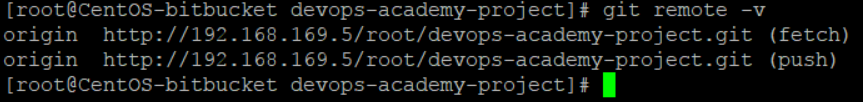
These are the first and basic steps for any git client machine

**For http method cloning**, go to client and run below command( this will clone the project from server to client)

1. git clone <http://192.168.169.5/root/devops-academy-project.git> (***clone command is only for downloading the repository/projects to client, it not for downloading branches***)

2. check status with git status command

3. also run the remote command like git remote –v

4. 

5. Create some file locally in the git clone project and add them to staging area with add command as below

6. git add file\_name

7. to rollback the above staging command use the below command (optional …just for knowledge)

8. *use "git rm --cached <file>..." to unstage*

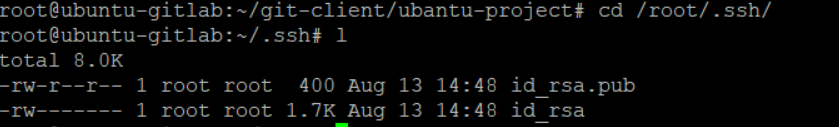
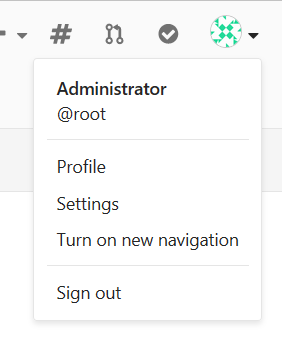
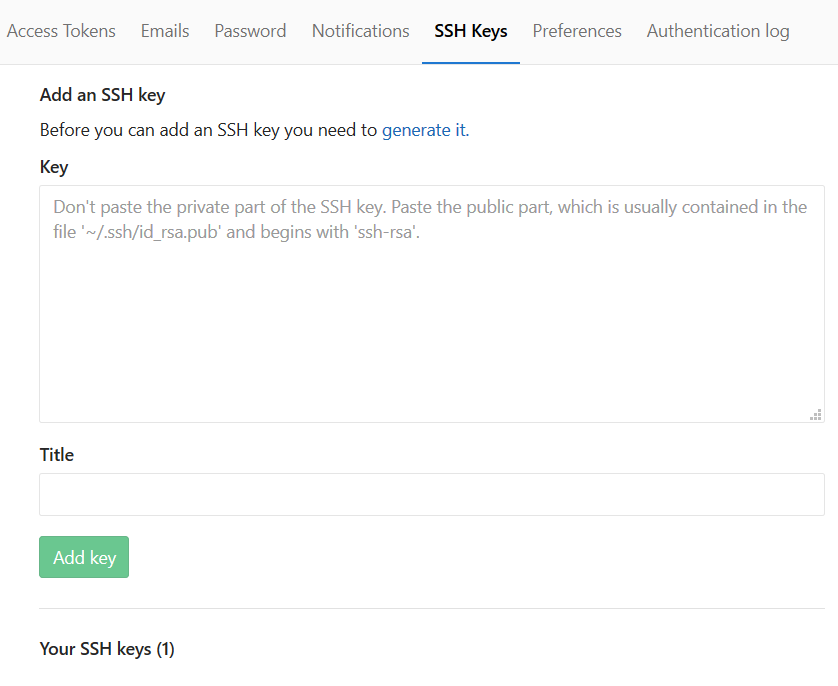
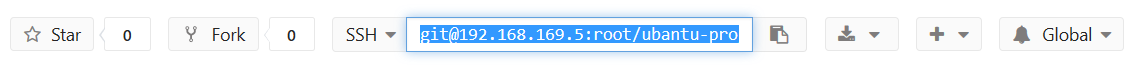
9. Now use the below command to commit them to the header part ( check the above diagram to have clarity on this)

10. git commit -m "Pushing first time from client"

11. **Note- For centos and windows machine**, before running the above commit command…first run the below command (git config --global user.email “[invincible.amits@gmail.com](mailto:invincible.amits@gmail.com)” and git config --global user.name "amitsmca")

12. final commad used to push the code on gitserver repository *git push -u origin master*

**Now to Do the same above stuff through SSH method ( the ssh advantage is this because it is faster than http and does not require username password for every command)**

1. Run the below command on git client server
2. ssh-keygen -t rsa
3. go to cd /root/.ssh/
4. 
5. Copy the .pub file content and paste on the gitlab server side (in setting🡪 keys)
6. 
7. 
8. Now save the key here and go back to client and do the git client normal stuff using ssh URL now (instead of HTTP)
9. 

GIT Commands

1. If git client is pointing to remote github URL, then below command can be used to make it point to new gitlab url (this is usefull when we migrate from hub to lab and want client to point to this new lab url)
2. Git remote -v (tells that where the client is pointing)
3. *root@ubuntu-gitlab:~/git-client/github-project1# git remote -v*
4. *origin https://github.com/amit-gitgub/github-project1.git (fetch)*
5. *origin https://github.com/amit-gitgub/github-project1.git (push)*
6. use **git remote set-url origin lab/hub url like below**
7. *root@ubuntu-gitlab:~/git-client/github-project1# git remote set-url origin git@192.168.169.5:root/ubantu-project.git*
8. *root@ubuntu-gitlab:~/git-client/github-project1# git remote -v*
9. *origin git@192.168.169.5:root/ubantu-project.git (fetch)*
10. *origin git@192.168.169.5:root/ubantu-project.git (push)*

* **To check out only one of specified files or all files , do the following**

1. Make sure that client is pointing to the correct URL (see above commands)
2. Go to the directory or repository and give command **git pull(this command also fetch the newly created branch at server to client )**
3. This pull command will update all the content inside that directory.
4. Note – As you move in different directory, you get the different branches ( this can be checked using git branch)

* To switch between branches use **git checkout branch\_name** ( always remember on which URL you are pushing and pulling by using **git remote –v**
* **How to create a new branch from client(backend)**

1. **CD in to the project**
2. git checkout -b testbranchfromCLIENT
3. Now tell the git server about this new branch on the client side with below command
4. Git add .
5. Git commit – m “New branch”
6. Git push –u origin testbranchfromCLIENT
7. Now go to the server and check if the new branch is created under the project or not.

* **How to get server side newly created branches, to the client side.**

1. Go to project and do git pull
2. Then git checkout branch-name

* **Merging from one branch to other (suppose we want to merge code from dev branch to master branch)**

1. Go to project and check the status of the branch with git status
2. Check out the master branch like git checkout master
3. Now run git merge dev
4. Now git push (or git push –u origin master )(as we are already in the master branch)

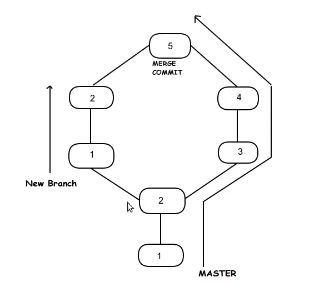
* **Git log command is used for the activity happened inside the git like all commits, check out etc some command are given below(Imp – Release notes are created from this)**

1. Git log (it will list down all the activity)
2. Git log –author amit
3. Git log –oneliner
4. Git log –before 2.days.ago
5. Git log –after=”2014-07-01” –before “2014-07-31” --oneline > /home/amit/ release\_notes\_july\_2017.txt

* **Git Merge and rebase**

**Git Merge** - In merge we simply merge the two branches in the merge commit.

Advantage – Going to back to the previous state is possible in merge**.**

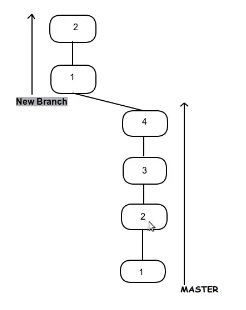


**Git Rebase** – Its means changing the base. Initially when the branch was cut, base was 2. But in rebase, it gets changes to 4 and then the new branch changes merges.

Drawback – It rewrite the commit history.So going bach to earlier state is not possible

Command –git checkout new\_branch

Git rebase master



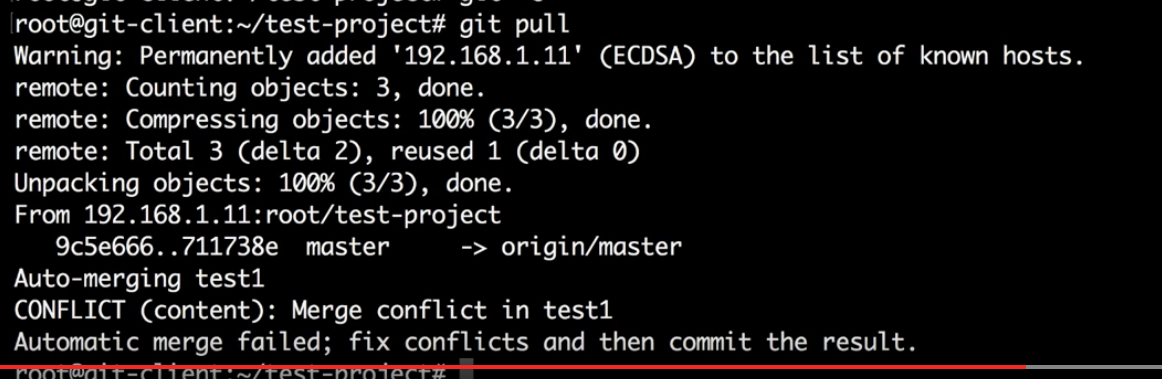
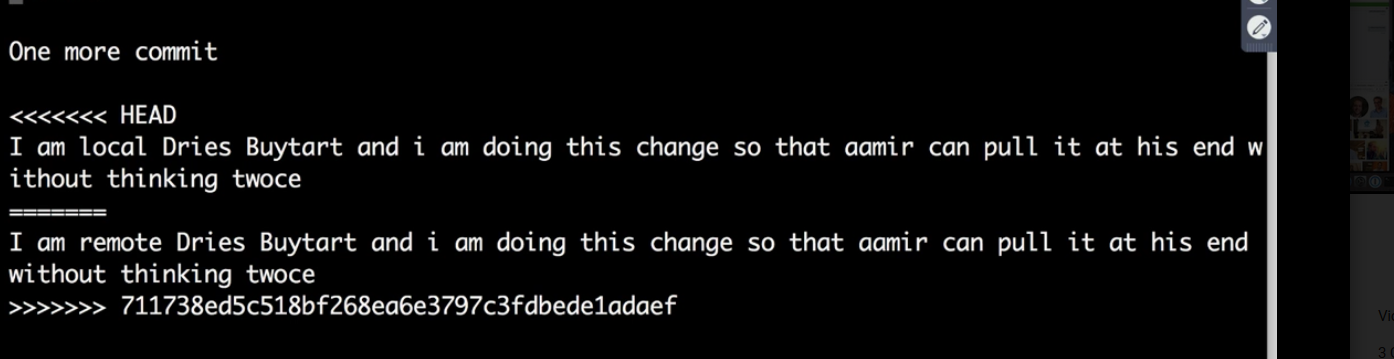
**Git HEAD -** It is always the reference to the latest commit.

It can be used to check the difference between 2 commits.

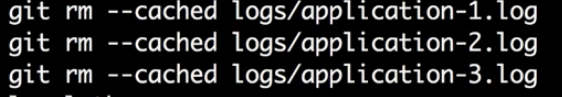
* **Difference Between git pull and git fetch**

1. **Git pull = git fetch + git merge**
2. **Git fetch :- it takes the commit from the remote repo and keep them on the local repo without merging them. While pull command merge them as well.**
3. **Do a git fetch in dev branch then do git diff origin/master (note in this master branch file was updated by other developer)**
4. **Now do the git merge origin/master (now this will sync the changes in your dev branch from master branch)**

* **Merge Conflict**

1. **Merge conflict happen when two developers try to change the same line of the sa me file**
2. **Scenario – change some file at remote and commit, now change the same line for the same file at local and do the following**
3. **Do git add. && git commit –m “local comit” && git pull . this will show the conflict like below.**
4. 
5. **Now open that file in vi and you will see something like below**
6. 
7. **Now remove the stuff and keep what ever you want from <<< head and >>>> and push the changes.**

* **Git rm command**

1. **This rm command is used to delete the some file at remote server side. Ex. Log file or any other file you don’t want.**
2. 
3. **After above command say git commit –m “removing logs” and then git push**

**Stash Command – this is like temporary area where you can keep you changes for some time**

1. **Modify some file and if you don’t want to remove the changes for some time then type git stash.**
2. **Now check your file, your changes would not be there.**
3. **Now to get back the changes git stash apply**

**Init command – this git init command make some normal directory as git directory**

**Git config command – this lets you set the user related things like**



**Gitlab email integration**

1. **Email service provided like amazon SES provide the smtp setting which we have to keep in** /etc/gitlab/gitlab.rb file

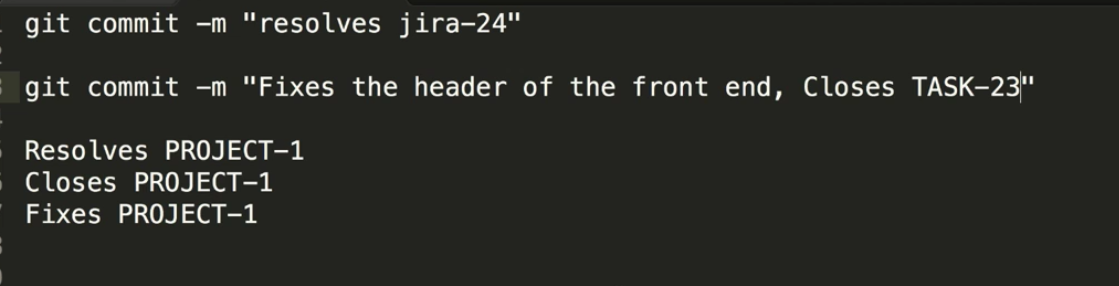


1. **One above is added do the gitlab-ctl reconfigure**

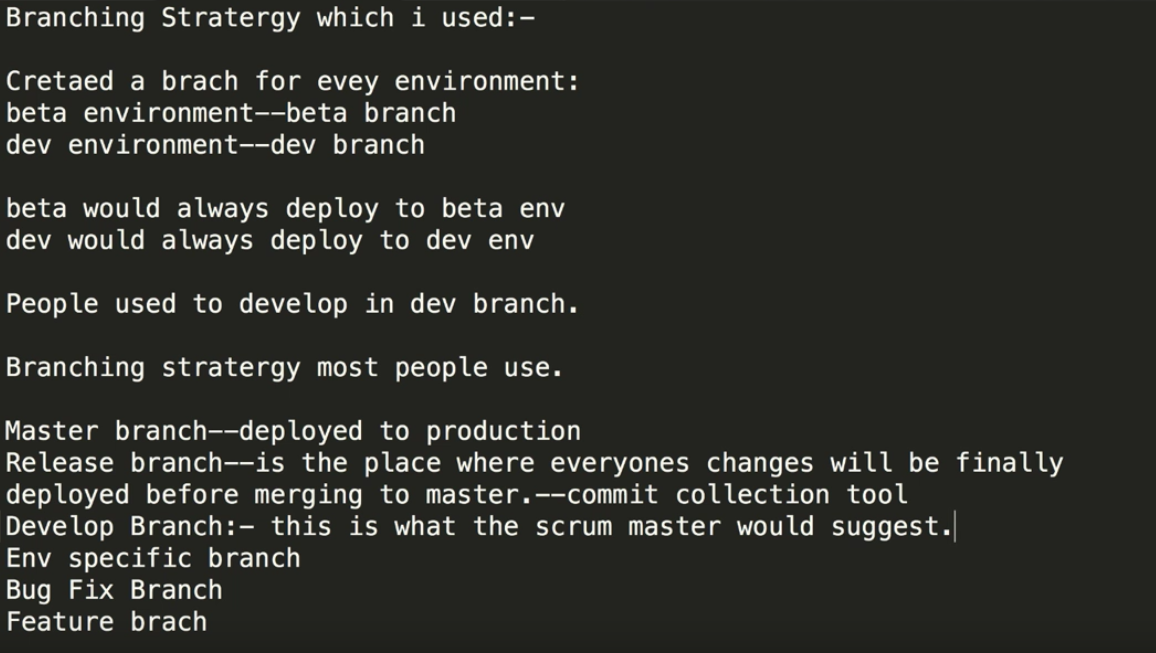
**Gitlab and jira integration(follow below)**

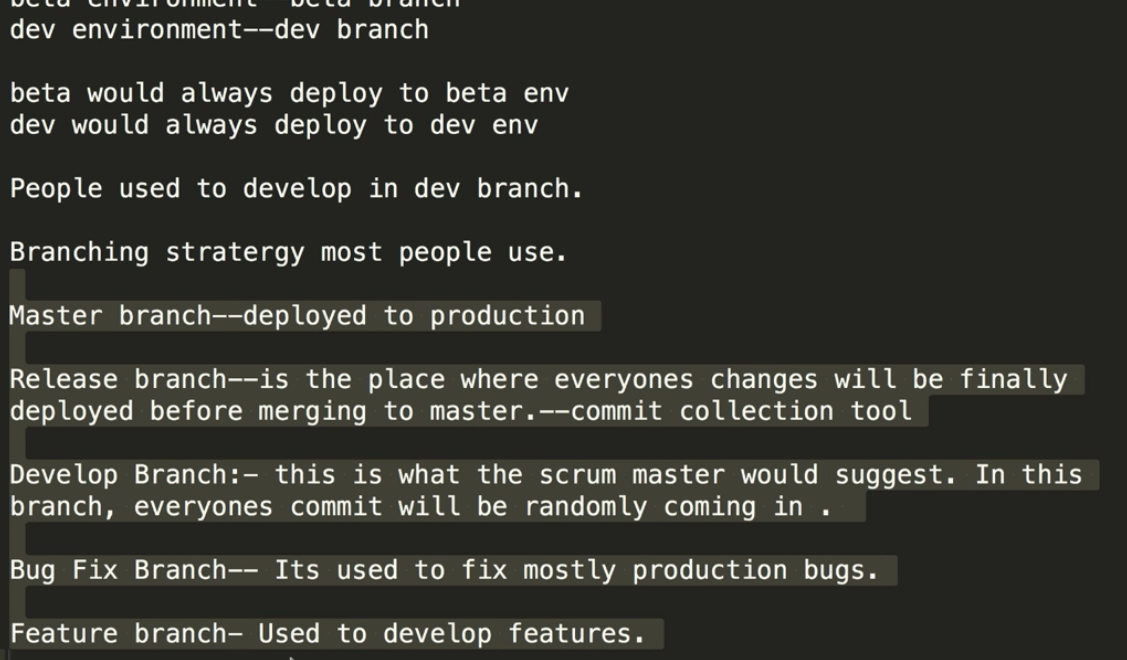
[**https://docs.gitlab.com/ee/user/project/integrations/jira.html**](https://docs.gitlab.com/ee/user/project/integrations/jira.html)

**this integration helps in marking the issues in jira fixed automatically**



**Branching strategy**





***important command for git in ubantu machine***

*Below command remove the git from the ubantu machine*

*apt-get remove git -y*

*Below command installthe git in the ubantu machine*

*apt-get install git -y*

Important links for GIT

<https://git-scm.com/downloads>

*set incremental pull in jinkins using git*

