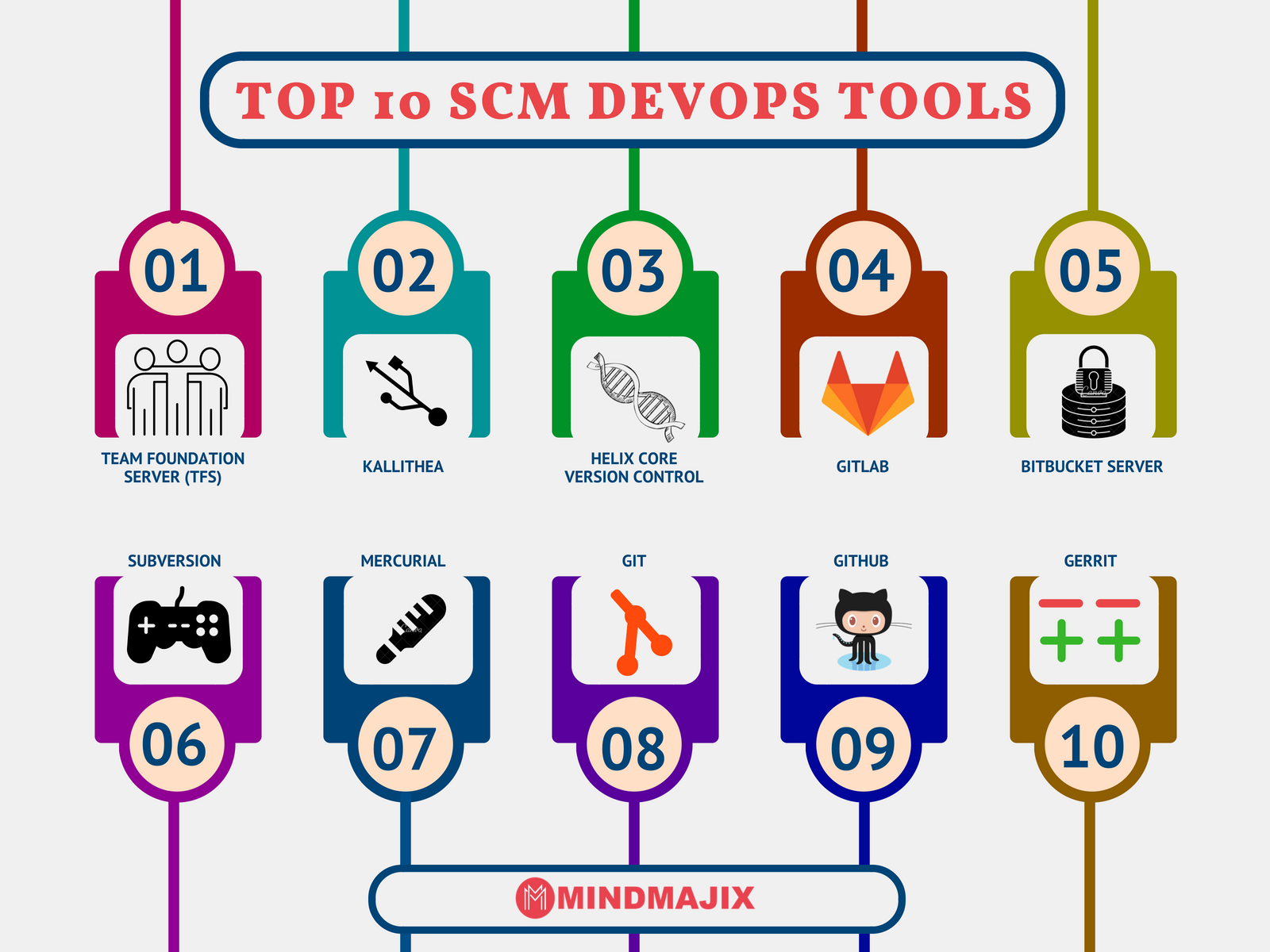
**SCM (Source Code Management) tools**



**GIT HUB:**

**Remote Repository**

**I**

**H**

**G**

**F**

**E**

**D**

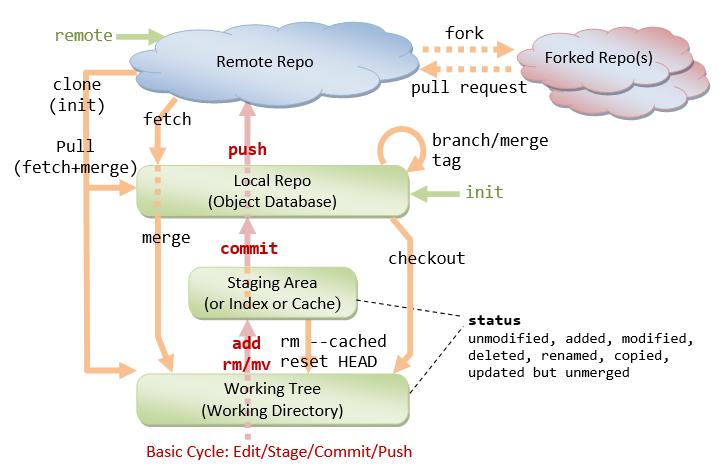
**B**

**C**

**A**

**Working Area**

**Local Repository**



**Install GIT**:

**#apt-get update**

**#apt-get install git-core –y**

**#git –version**

**Initialize git local repository**

**#git init**

**Initialized empty Git repository in /root/.git/**

**#ls –a**

**.git**

**Add User Name and Mail of your GIT:**

**#git config --global user.name <username>**

**#git config --global user.email <git hub user mail>**

**A**: **Clone the repository from GIT HUB**

**#git clone** [**https://github.com/shyam9921/repo1.git**](https://github.com/shyam9921/repo1.git)

**B: Add your file to index Area :**

**#cd repo1**

**#git config --list**

**core.repositoryformatversion=0**

**core.filemode=true**

**core.bare=false**

**core.logallrefupdates=true**

**remote.origin.url=https://github.com/shyam9921/repo1.git**

**remote.origin.fetch=+refs/heads/\*:refs/remotes/origin/\***

**branch.master.remote=origin**

**branch.master.merge=refs/heads/master**

**#vi ex1.php**

**Hi good morning**

**:wq**

**#git status**

**On branch master**

**Your branch is up-to-date with 'origin/master'.**

**Untracked files:**

**(use "git add <file>..." to include in what will be committed)**

**ex1.php**

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

**# git add ex1.php**

**#git status**

**On branch master**

**Your branch is up-to-date with 'origin/master'.**

**Changes to be committed:**

**(use "git reset HEAD <file>..." to unstage)**

**new file: ex1.php**

**C: Commit from stage area to local repo:**

**# git commit ex1.php -m "firstcommit of ex1.php"**

**[master 5c67539] firstcommit of ex1.php**

**Committer: root <root@ip-172-31-94-183.ec2.internal>**

**Your name and email address were configured automatically based**

**on your username and hostname. Please check that they are accurate.**

**You can suppress this message by setting them explicitly. Run the**

**following command and follow the instructions in your editor to edit**

**your configuration file:**

**git config --global --edit**

**After doing this, you may fix the identity used for this commit with:**

**git commit --amend --reset-author**

**1 file changed, 1 insertion(+)**

**create mode 100644 ex1.php**

**#git status**

**On branch master**

**Your branch is ahead of 'origin/master' by 1 commit.**

**(use "git push" to publish your local commits)**

**nothing to commit, working directory clean**

**D: Push from Local Repo to Remote Repo:**

**#git push origin master**

**Username for 'https://github.com': shyam9921**

**Password for 'https://shyam9921@github.com':**

**Counting objects: 3, done.**

**Compressing objects: 100% (2/2), done.**

**Writing objects: 100% (3/3), 288 bytes | 0 bytes/s, done.**

**Total 3 (delta 1), reused 0 (delta 0)**

**remote: Resolving deltas: 100% (1/1), completed with 1 local object.**

**To https://github.com/shyam9921/repo1.git**

**2cadf4a..5c67539 master -> master**

**#git log**

**commit 5c67539a630bb6fbb6c88af589cf5fab993f8973**

**Author: root <root@ip-172-31-94-183.ec2.internal>**

**Date: Sun Mar 24 14:08:39 2019 +0000**

**firstcommit of ex1.php**

**#git log --oneline**

**5c67539 firstcommit of ex1.php**

**To see remote repository:**

**#git remote –v**

**origin https://github.com/shyam9921/repo1.git (fetch)**

**origin https://github.com/shyam9921/repo1.git (push)**

**To add Remote repository from console:**

**In case you are not clone the repository the you set the path for your remote repository**

**#cd**

**#vi abc.txt**

**#git add abc.txt**

**#git commit abc.txt –m “new code of pub”**

**#git remote –v**

**It shows empty**

**#git push origin master**

**ERROR:** **fatal: 'origin' does not appear to be a git repository**

**fatal: Could not read from remote repository.**

**Please make sure you have the correct access rights**

**and the repository exists.**

**Add Remote:**

**#** **git remote add origin** [**https://github.com/shyam9921/repo1.git**](https://github.com/shyam9921/repo1.git)

**#git remote –v**

**origin https://github.com/shyam9921/repo1.git (fetch)**

**origin https://github.com/shyam9921/repo1.git (push)**

**This repo already created in git hub**.

**#git pull origin <branchname>**

**#git push origin master**

**BRANCHES:**

**Git branch is helpful for work without directly on main environment**

**Again Coming To Cloning Repo:**

**#cd repo1**

**To check Branches:**

**#git branch**

**\*master**

**To Create Branches**

**#git branch <branchname>**

**#git branch br1**

**#git branch**

**\*master \* Active branch**

**br1**

**#git checkout br1**

**To active br1**

**#git branch**

**\*br1**

**Master**

**#vi file**

**Hello to every one**

**:wq**

**#git add file**

**#git commit file file –m “bashcode”**

**#git push origin master**

**ERROR: Username for 'https://github.com': shyam9921**

**Password for 'https://shyam9921@github.com':**

**To https://github.com/shyam9921/repo1.git**

**! [rejected] master -> master (fetch first)**

**error: failed to push some refs to 'https://github.com/shyam9921/repo1.git'**

**hint: Updates were rejected because the remote contains work that you do**

**hint: not have locally. This is usually caused by another repository pushing**

**hint: to the same ref. You may want to first integrate the remote changes**

**hint: (e.g., 'git pull ...') before pushing again.**

**hint: See the 'Note about fast-forwards' in 'git push --help' for details.**

**root@ip-172-31-94-183:~/repo1# git push origin br1**

**Username for 'https://github.com': shyam9921**

**Password for 'https://shyam9921@github.com':**

**Counting objects: 27, done.**

**Compressing objects: 100% (20/20), done.**

**Writing objects: 100% (27/27), 2.35 KiB | 0 bytes/s, done.**

**Total 27 (delta 9), reused 0 (delta 0)**

**remote: Resolving deltas: 100% (9/9), completed with 1 local object.**

**remote:**

**remote: Create a pull request for 'br1' on GitHub by visiting:**

**remote: https://github.com/shyam9921/repo1/pull/new/br1**

**remote:**

**To https://github.com/shyam9921/repo1.git**

**\* [new branch] br1 -> br1**

**#git checkout master**

**#git log --oneline**

**5c67539 firstcommit of ex1.php**

**Branch combines with other Branch:**

**3 ways:**

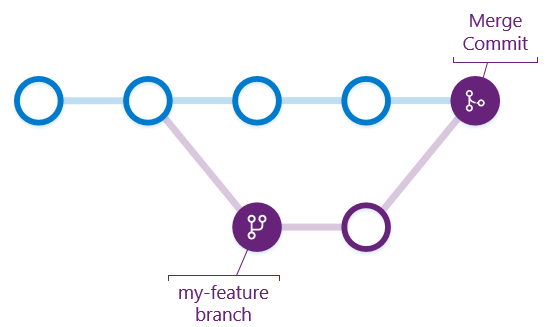
**1. git merge**

**2. git rebase**

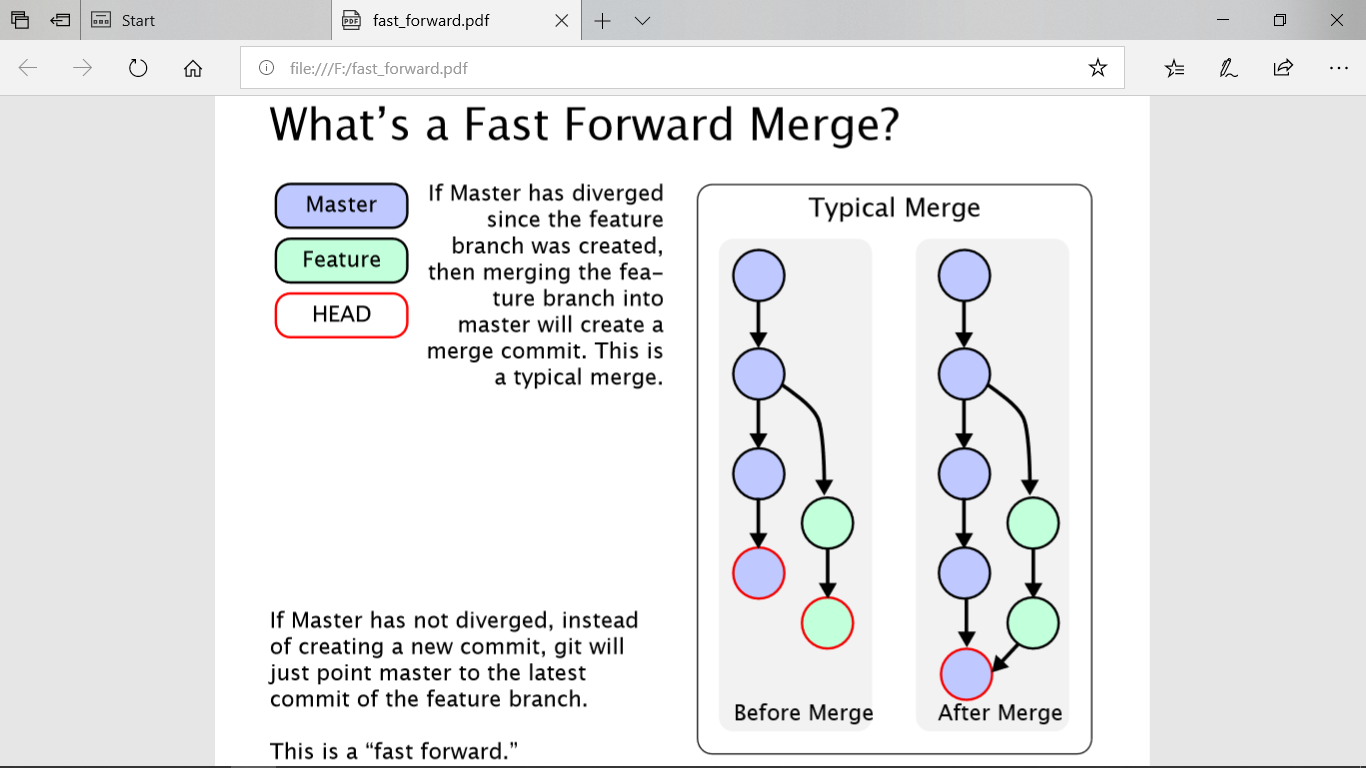
**3. git squash**

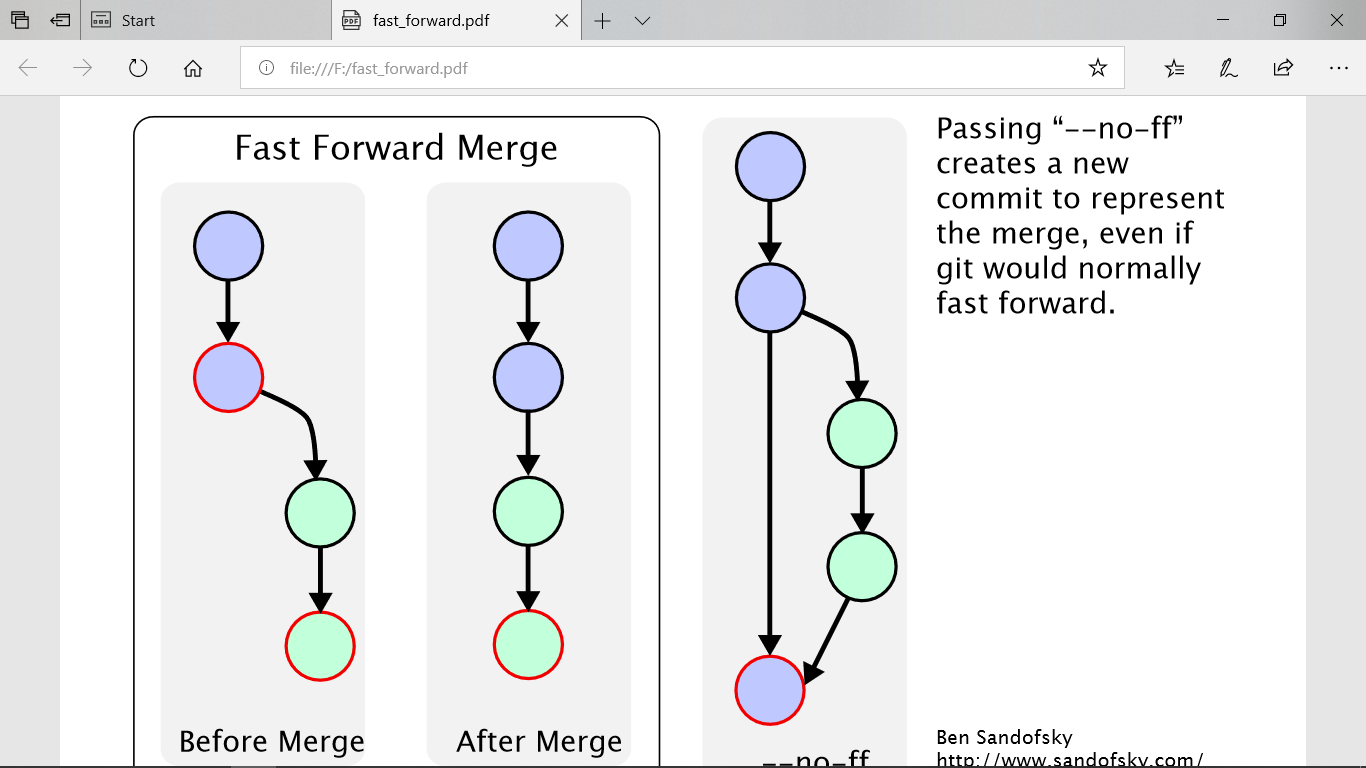
**1.git merge:**

**It is nothing but modifications of one branch merge to another branch at the time new commit id generated.**



**Fast forward merge no commit id generated.**

****

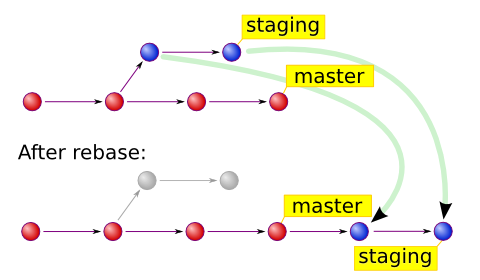
****

**#git merge br1 --no-ff merge commit generated**

**#git merge br1 fast forward merge and no merge commit generated**

**#git push origin master**

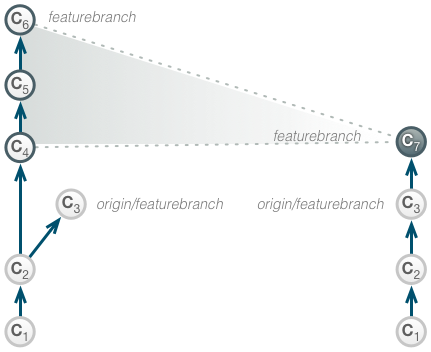
**GIT REBASE:**



**#git rebase <branchname>**

**GIT SQUASH:**

**The term squashing refers to consolidating multiple commits into one single commit, for the purpose of simplifying the project history.**



**#vi abc1**

**Hihello**

**:wq**

**#git add abc1**

**#git commit abc1 –m “code hy in abc1”**

**With out asking username and password :**

**#cd**

**#ssh-keygen**

**(enter 4times)**

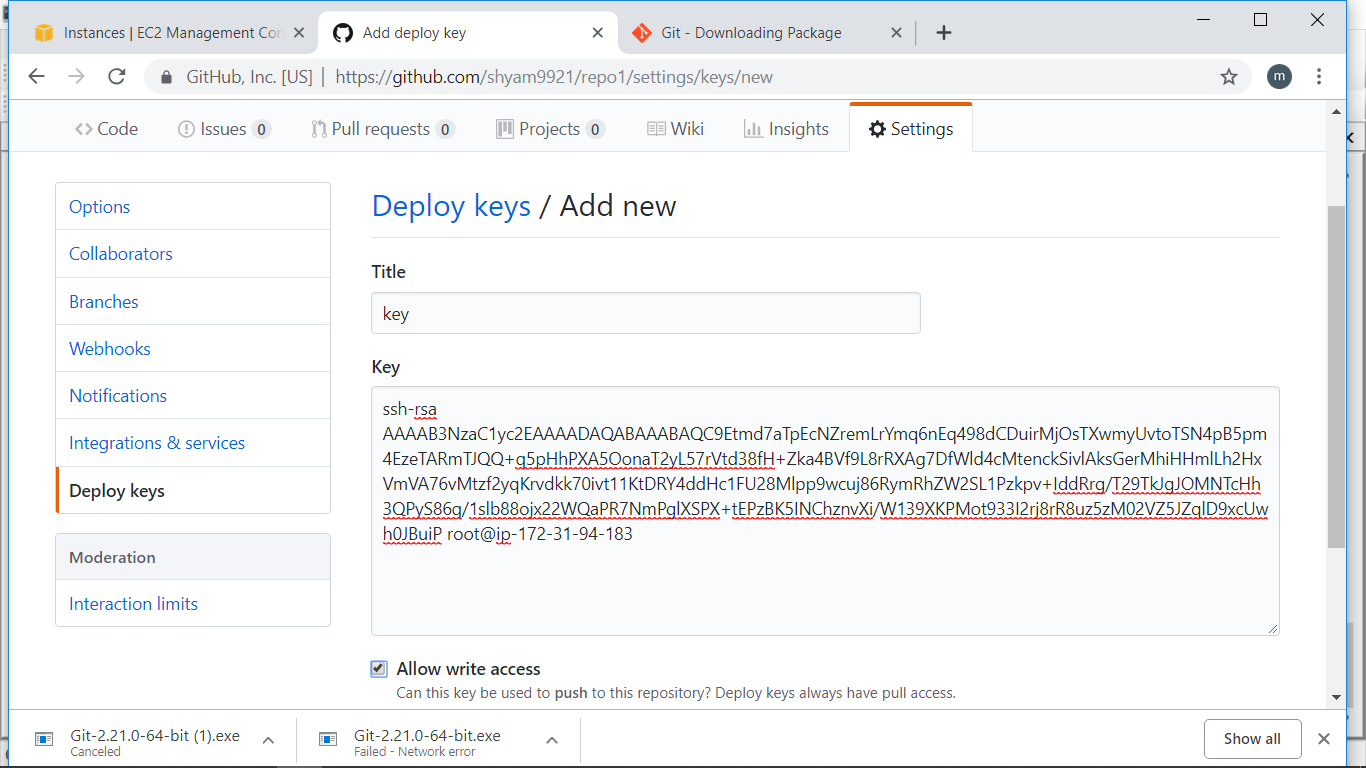
**#ls –a**

**#cd .ssh**

**#cat id\_rsa.pub**

**ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQC9Etmd7aTpEcNZremLrYmq6nEq498dCDuirMjOsTXwmyUvtoTSN4pB5pm4EzeTARmTJQQ+q5pHhPXA5OonaT2yL57rVtd38fH+Zka4BVf9L8rRXAg7DfWld4cMtenckSivlAksGerMhiHHmlLh2HxVmVA76vMtzf2yqKrvdkk70ivt11KtDRY4ddHc1FU28Mlpp9wcuj86RymRhZW2SL1Pzkpv+IddRrg/T29TkJgJOMNTcHh3QPyS86q/1slb88ojx22WQaPR7NmPglXSPX+tEPzBK5INChznvXi/W139XKPMot933I2rj8rR8uz5zM02VZ5JZqlD9xcUwh0JBuiP root@ip-172-31-94-183**

**Copy that and paste on :**

****

**Set the url of key:**

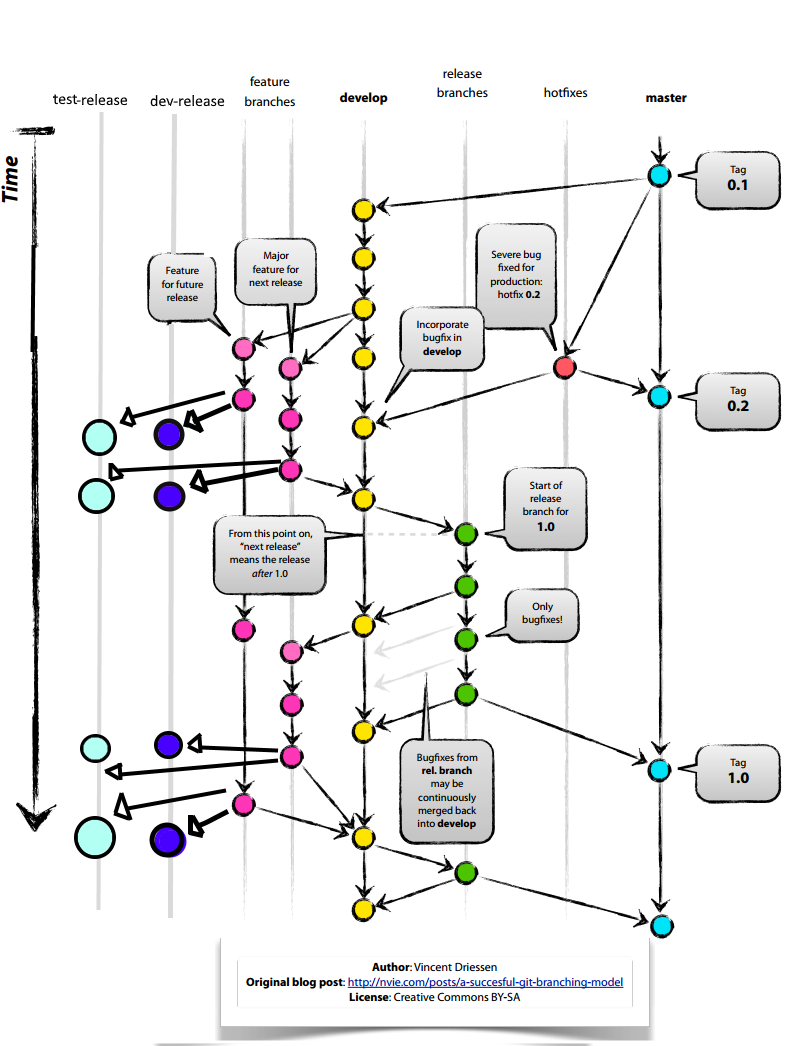
**# git remote set-url origin** [**git@github.com:shyam9921/repo1.git**](mailto:git@github.com:shyam9921/repo1.git)

**#git push origin master**

**The authenticity of host 'github.com (192.30.253.113)' can't be established. RSA key fingerprint is SHA256:nThbg6kXUpJWGl7E1IGOCspRomTxdCARLviKw6E5SY8.Are you sure you want to continue connecting (yes/no)? yes**

**Only first time**

**BRANCH STRATEGY:**



**In my company we use:**

**Master branch**

**Release branch**

**Dev branch**

**Feature branches**

**Hot fix branch**

**TAGS:**

**Tag is nothing but named snapshot. It contains some meta data**

**In my Organization I am using tags for on which commit id we release new version of application.**

**#git tag <tag name> <commit id>**

**#git tag –list**

**#git push origin <tagname>**

**How to delete tags:**

**#git tag –d <tag name> delete locally**

**#git push origin –delete <tag name> delete remotely**

**GIT STASH:**

**Git stash is temporarily pause the work. For suppose you have work in particular file at the time your lead assigned some hot fixes to you at the time it’s coming.**

**#git stash save <filename>**

**#git stash ---list**

**Stash{0}**

**Stash{1}**

**#vi file12**

**Hi hello dear**

**:wq**

**#git stash save file12**

**ONCE YOUR WORK DONE:**

**Apply the stash and delete the stash**

**#git stash apply <stash id>**

**#git stash drop <stash id>**

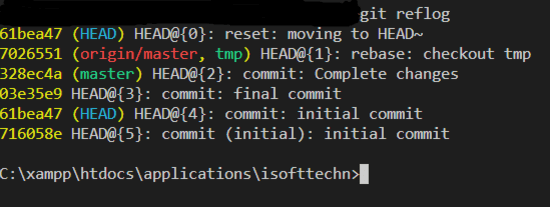
**#git stash pop <stash id>**

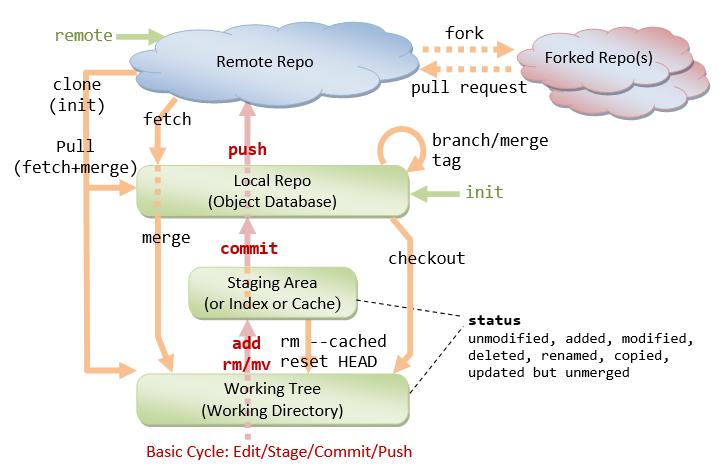
**Drop + apply**

**REFLOG:**

**The reflog command will give you a good history of what’s been happening on the head of your branches. Run it, then find the line that refers to the state that you want to get back to:**

**#git reflog**





**Git reset commitid**

**Git reset –soft commitid**

* **Git reset –soft commitid**

**Commitid deleted and it’s add to stage**

* **Git reset commitid commitid not deleted but code add to workspace**