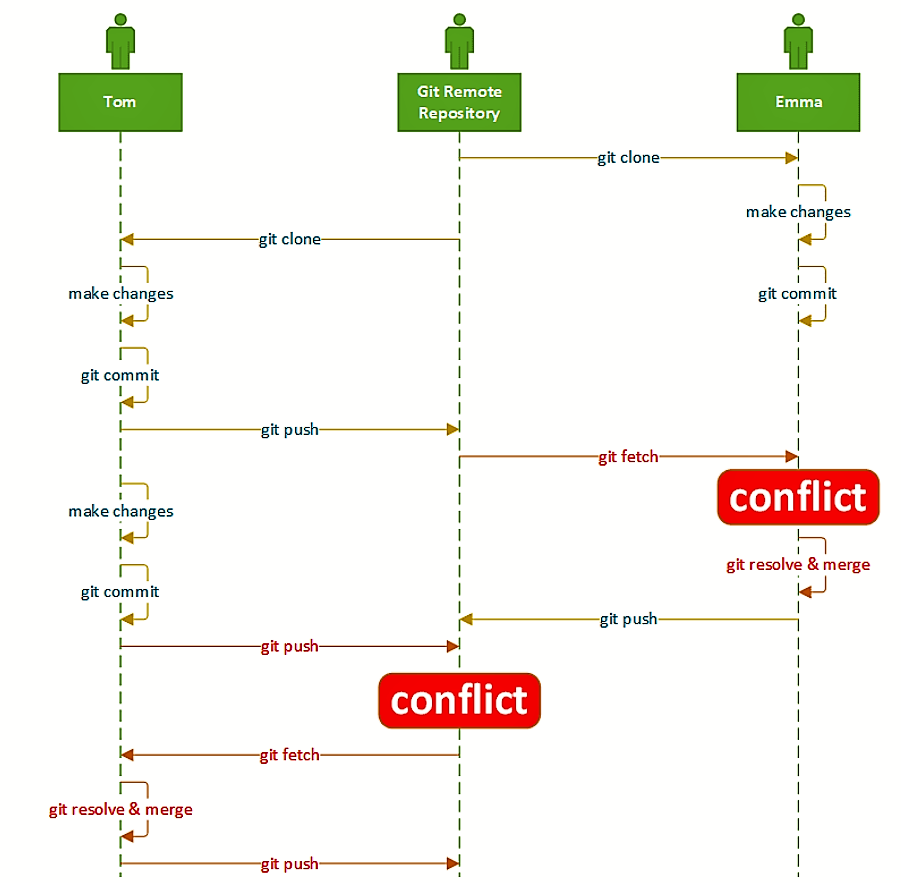
Git command **:**

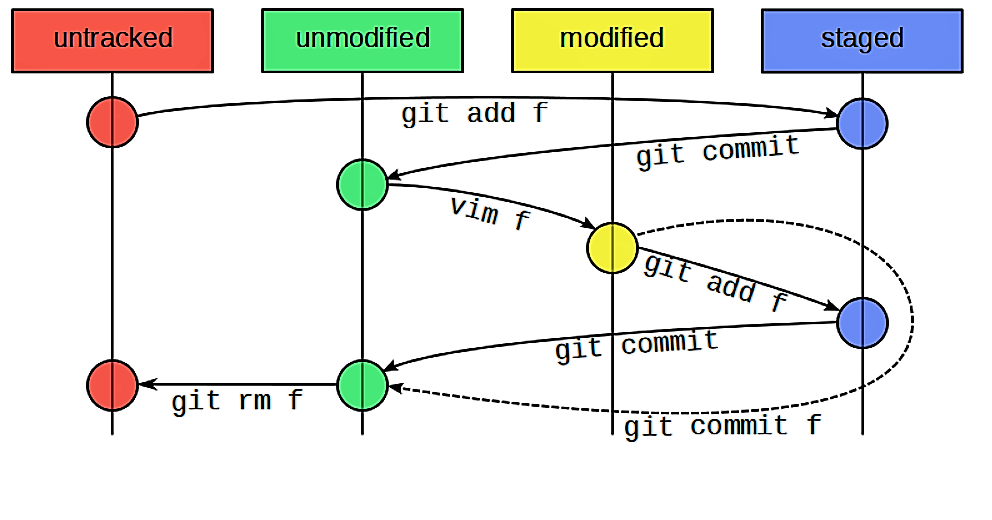
* git config --global user.name "jagdish"
* git config --global user.email "pkjagdish4501@gmail.com"
* git config --list
* git config user.name
* git config user.email
* git status
* git init
* git add –a
* git add .
* git add <directory or file name>
* git commit -m "hello.txt changed first time"
* git log
* git restore hello.txt
* rm -rf .git
* git clone <https://github.com/CodeWithHarry/TextUtils-React.git>
* touch .gitignore
* \*.log
* dir/
* /dir/

//if we add this ^ file or folder to .gitignore then change in this file or directory not traced byGIT

* git diff //show difference between working and staging area
* git diff --staged
* git commit -a -m "this is to skipping staging area"
* git rm third.txt //for deleting
* git mv hello.txt rename.txt //renaming purpose
* git rm --cached new.txt
* git log –stat
* git log --pretty=oneline
* git log --pretty=short
* git log --pretty=full
* git log –since=2.day
* git log --pretty=format:"%h --%an"
* git commit --amend
* git checkout -- second.txt
* git checkout -f
* git remote add origin git@github.com:jagdish4501/TextUtils-React.git
* git remote
* git remote -v
* git push -u origin master
* git config --global alias.st status
* git config --global alias.unstage 'restore --staged --'
* git unstage second.txt
* git checkout -b devlop //making a new branch
* git checkout master
* git checkout devlop
* git branch
* git merge devlop //this commond marge devlop branch to master
* git branch -v //commit hash and commit message
* git branch --merged //show branches which is merged
* git branch --no-merged //show branch which are not merged
* git branch -d devlop //D
* git fetch

Git merge conflict:

****



**GENERATING SSH key:**

* ssh-keygen -t rsa -b 4096 -C "pkjagdish4501@gmail.com"
* eval $(ssh-agent -s)
* ssh-add ~/.ssh/id\_rsa
* tail ~/.ssh/id\_rsa.pub