* **Development branch Saloni** (usually called ‘develop’)  
  This is your main development branch where all the changes destined for the next release are placed, either by directly committing small changes or by merging other branches (e.g. feature branches) into this branch.
* **Production branch** (usually called ‘master’)  
  This branch represents the latest released / deployed codebase. Only updated by merging other branches into it.
* **Feature branches** (usually prefixed with ‘feature/’)  
  When you start work on anything non-trivial, you create a feature branch. When finished, you’ll merge this branch back into the development branch to queue it for the next release.
* **Release branches** (usually prefixed with ‘release/’)  
  When you’re about to package a new release, you create a release branch from the development branch. You can commit to it during your preparation for a release, and when it’s ready to be deployed, you merge it into both the development branch and the master branch (to indicate that the release has been deployed).

Generate SSH for removing efforts to entering credentials agaian again

1. In Git Bash, paste the text below, substituting in your GitHub email address.
2. ssh-keygen -t rsa -b 4096 -C "your\_email@example.com"
3. # Creates a new ssh key, using the provided email as a label
4. Generating public/private rsa key pair.
5. When you're prompted to "Enter a file in which to save the key," press Enter. This accepts the default file location.
6. Enter a file in which to save the key (/Users/you/.ssh/id\_rsa): [Press enter]
7. At the prompt, type a secure passphrase. For more information, see ["Working with SSH key passphrases"](https://help.github.com/articles/working-with-ssh-key-passphrases).
8. Enter passphrase (empty for no passphrase): [Type a passphrase]

Enter same passphrase again: [Type passphrase again

**If you are using another terminal prompt**, such as [Git for Windows](https://git-for-windows.github.io/), turn on ssh-agent:

# start the ssh-agent in the background

eval $(ssh-agent -s)

Agent pid 59566

Add your SSH key to the ssh-agent:

ssh-add ~/.ssh/id\_rsa

|  |  |
| --- | --- |
| Check the path where you have generated the public key.. you can also copy the id\_rsa by using this command  **clip < ~/.ssh/id\_rsa.pub** | |
|  | **cat ~/.ssh/id\_rsa.pub**  then you can copy your ssh key |

To clean : Clear /Ctrl L

Commit git commit –m ‘’

Push git push -u

Branch git checkout –b [branch name]

Git branch –a (all branches present)

git remote add origin <https://github.com/salonirajguru/xyz.git> -- add repo to remote

:

git status

git checkout master

git add ‘add.txt’

git status

git commit –m ‘Added add.txt to master’

git checkout –b develop

git push origin develop--------to push files to server

git merge develop

git reset HEAD~1

git revert HEAD

git cherry-pick C2 C4