**GIT REPOSITORY WORKFLOW**

**Version 1.0**

**Created by: Vivek Katoch**

**Dated: 25/12/2017**

1. **Install Git Client on local machine**

Download and install latest git.exe from following link <https://git-scm.com/download/win>

Run executable file for set up

1. **Access GitHub account**

Access your account at <https://github.com/> using LDAP credentials

1. **Create new local GitHub repository**

Go to drive [C:,D:] using GitBash

Create new folder with your project name “docservice”

$ mkdir docservice

Change your current directory to project folder.

$ cd docservice

1. **Initialize and Config Git repository at your local machine**

Run following to initialize your repository

$ git init

Config your Git to identify your commits.

$ git config -–global user.name “vkatoch2000”

$ git config -–global user.email “v\_katoch2000@yahoo.co.in”

Verify your configuration using command:

$ git config –list

1. **Set remote for your local repository**

Create new private project [with same name you have local repository (recommended)]at <https://ecgit01.corp.edifecs.com/> using LDAP credentials

Copy “https remote URL“provided by it.

Run following command:

$ git remote add origin “https remote URL “

Verify new remote URL

$ git remote –v

1. **Push local changes to remote repository**

$ create new file for commit.

$ git add. or git add <filename> [it will stage your untracked files changes]

$ git commit –m “any file commit message”

$ git commit –am “any file commit message” [ it will stage only your tracked file’s changes]

$ git push –u origin master

1. **Create local branch**

$ git branch [list down all the branches & it will show \* before current branch]

Command to create local branch and switch to that branch

$ git checkout –b docservice-dev

Check the changes

$ git branch

1. **Push local branch to remote**

$ git push origin docservice-dev

1. **Resolve commit conflicts**
2. When changes are staged but not commit.

Do some changes in file locally and stash

$ git stash or git stash save "add style to our site"

$ git pull origin <branch-name

$ git stash apply

Above command may show you merge error & resolve them.

1. When changes are commit but not pushed.
   1. Changes are in different files

$ git push origin <branch-name>

Above command will show you **conflict error** and ask you to pull remote changes first

$ git pull origin <branch-name>

Prompt to auto merge screen

Change your message [Optional] and save and exit the screen

:wq

Now push the changes

* 1. Changes in same file but different lines

$ git push origin <branch-name>

Above command will show you **conflict error** and ask you to pull remote changes first.

$ git pull origin <branch-name>

Prompt to auto merge screen

Change your message [Optional] and save and exit the screen

:wq

Now push the changes

* 1. Changes in same place in same files when auto merging failed

$ git push origin <branch-name>

Above command will show you **conflict error** and ask you to pull remote changes first.

$ git pull origin <branch-name>

Above command will show you **auto merging fail error** and ask you resolve conflicts manually

$ nano <conflict-file-name>

Edit & resolve conflicts manually save the file.

Commit and push resolved conflict changes.

1. **Update dev branch from master branch**

Commit some changes to master branch & commit the changes.

Switch to dev branch

$ git checkout <branch-name>

Command to update <branch-name> from master branch

$ git pull origin master

Conflicts may occur, resolve them

1. **Merge your branch to master repository**

Switch to dev branch

Pull changes from dev & master branch respectively

Switch to master branch

Pull changes from master branch

Create merge request from dev branch

$ git merge --no-ff --no-commit <branch-name>

$ git commit –m “merge branch to master”

$ git push origin master

1. **Delete dev branch from local & remote**

Remove from local using command

$ git branch -d **<**branch-name>

Remote from remote using command

$ git push origin --delete <branch-name>

1. **To check GIT logs**

* git log –oneline

1. **To Update code from older commits**

* git log –oneline [list commit Ids]
* git checkout <commitID> <filename>

1. **To Remove files from GIT repository**

* git rm file1.txt
* git commit -m "remove file1.txt"

Note \* - But if you want to remove the file only from the Git repository and not remove it from the filesystem, use:

* git rm --cached file1.txt
* git commit -m "remove file1.txt"
* git push origin <branch-name>

1. **Unable to access remote repository : 403 error**

* git config --global --unset credential.helper
* git config --system --unset credential.helper