GIT important commands:

Cloning using git:

- git clone https://github.com/WilmingtonGroup/ComplianceWeek.git

Create the branch and dont push to master branch:

- git checkout -b compdev\_branch

- git push origin compdev\_branch

Branch will be seen in github.com

Push from local server to branch and not to dev(master)

git config --global credential.helper wincred

https://help.github.com/articles/caching-your-github-password-in-git/

Entering your username

git config --global user.name "Vanitha Sophia"

entering your email

git config --global user.email vanitha.sophia@cognizant.com

git status(red)

git add --all

git status(green)

git commit -m "hjg"

git push origin compdev\_branch

find the error log

tail -f staging.compliance.week.com

Step 1: Get a clone of GIT repository to local machine

git clone <download link from GITHUB>

Step 2: Create a branch in local machine

git checkout -b <branch name>

Step 3: Push the code to GIT

git push origin <branch name>

Step 4: SSH the server and create the branch

git fetch && git checkout <branch name>

Errors like:

1. error: Your local changes to the following files would be overwritten by checkout:

docroot/sites/staging.complianceweek.com/settings.php

Please, commit your changes or stash them before you can switch branches.

Aborting

* **git stash**
* **git status**

1. On branch staging

Your branch is behind 'origin/staging' by 22 commits, and can be fast- forwarded.

(use "git pull" to update your local branch)

Untracked files:

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

info.php

* **rm info.php**
* **git fetch && git checkout comp\_branch**

Create branch and commit changes (Important)

1. Branch should be created from master

Git checkout master

1. Pull the changes from git master to local master

Git pull origin master

1. Step again

Git pull origin master

1. Create the new branch

Git checkout –b branch\_name

1. Git add, git commit, git push
2. In ssh dev server,
3. Git fetch && git checkout <branch\_name>

For existing branches which doesn’t have the latest pull

1. Switch to ‘master’ local

Git checkout master

1. Pull the latest changes to local master

git pull origin master

1. Checkout to your branch

Git checkout <branchname>

1. git merge --no-ff master

DELETE option

1. Git fetch && git checkout <branch\_name>

Renaming the branch name in git

git branch -m feature/WDAOS-1469 feature/haplo

Copying the remote folder to desktop:

552898@PC241991 MINGW32 ~

$ scp –r staging.complianceweek.com@5.148.133.153:/srv/web/staging.complianceweek.com/CW\_Haplo\_Files /c/Users/552898/Desktop/WSRdocs

Pwd

/srv/web/staging.complianceweek.com

Copying desktop file to remote desktop:

scp -r /c/Users/552898/Desktop/resourcefiles/ [staging.complianceweek.com@5.148.133.154:/srv/web/staging.complianceweek.com/docroot/sites/default/files](mailto:staging.complianceweek.com@5.148.133.154:/srv/web/staging.complianceweek.com/docroot/sites/default/files)

Copying to private files:

Pwd

/srv/web/staging.complianceweek.com/private-files/

scp -r /c/Users/552898/Desktop/ cwdoc/ [staging.complianceweek.com@5.148.133.154:/srv/web/staging.complianceweek.com/private-files/](mailto:staging.complianceweek.com@5.148.133.154:/srv/web/staging.complianceweek.com/private-files/)

scp -r /c/Users/552898/Desktop/indepth\_reports/ [staging.complianceweek.com@5.148.133.154:/srv/web/staging.complianceweek.com/private-files/](mailto:staging.complianceweek.com@5.148.133.154:/srv/web/staging.complianceweek.com/private-files/)

scp -r /c/Users/552898/Desktop/resourcefiles/ [staging.complianceweek.com@5.148.133.154:/srv/web/staging.complianceweek.com/private-files/](mailto:staging.complianceweek.com@5.148.133.154:/srv/web/staging.complianceweek.com/private-files/)

Copying from dev to desktop:

eval $(ssh-agent -s)

ssh-add ~/compweek

ssh staging.complianceweek.com@5.148.133.153

scp –r [staging.complianceweek.com@5.148.133.153:/srv/web/staging.complianceweek.com/docroot](mailto:staging.complianceweek.com@5.148.133.153:/srv/web/staging.complianceweek.com/docroot)/cogdev\_backup.sql.gz /c/Users/552898/Desktop/db\_backup

Hardware details:

* Ubuntu 16.04
* They're actually all cloud based so they're configured from VM and there's 4 VM CPUs   
  Different RAM on the staging to the Live production servers   
  6G on staging and 18G on prod
* The main difficulty is that there are three nodes with a load balancer
* Each node has its own server, apache ssl, solr, memcache   
  and there's a load balancer that distributes hits
* It's running php fpm 5.6