***FireFighter work***

The first thing you will need to do is follow the documentation: -

1. Go to Jenkins dashboard and follow up the already scheduled content pull project if it is already started, like the automatic schedule one, if not you start it manually.
2. Once it is finished its task check the status.
3. Once it is failed you can follow up accordingly
4. Once it is succeeded or you can see it in the console log, you follow what it said at the end of the console with good messages.
5. Then go to the page what changes have been done in the link:- <https://github.com/pearca/qiactive/pull/5138/files>
6. Checkout every files added or modified during the build process
7. There are files that cannot be shown in the GitHub and you can check it out in the Terminal
   1. Got the repo directory first cd $PEARSONPATH
   2. You can checkout to the latest branch created by Jenkins and make sure you can access it. You can make sure using git fetch origin content-pull-QA-XX
   3. Switch the branch to the new branch created during the job’s build by like git checkout content-pull-QA-XX you can find this name in:-<https://github.com/pearca/qiactive/pulls>
   4. Then see the difference made as compare to the master using the command git difftool -x json-diff master:filename filename
   5. After you check the confusions you can comment what you observed like share that you are satisfied or happy with the changes and submit it.
   6. After that someone from the review team will approve the result.
8. Once it is approved we do the merge and push to the final branch.

**Merging the approved branch to the Master branch**

1. Block first anybody from going to do the merge as you are doing merge by going to the link <https://qa.qiactive.com/get-banana.html>
2. Go back to terminal and go to the virtual-env and activate it
   1. virtualenv qiactive-venv/
   2. source qiactive-venv/bin/activate
3. Go back to the repo link cd $PEARSONPATH
4. make sure you are on the master branch using git checkout master
5. make sure you have all the latest source from the master by using git pull
6. start merging it now safely using tools/git-safemerge origin/content-pull-xx
7. you will be asked to make sure say yes
8. you will be asked to commit with a message you can leave it without message using :wq
9. you push it then to the remote master using ./pre-push --content-pull
10. at the end it will ask you if you are sure to do the push to the master, you type master and hit enter
11. make sure to release your banana at the end

**make sure you running the virtualBox server**

**in the terminal make sure the grails and py is in the right direction and running correctly.**

1. Next comes is content-push-QA by going to the Jenkins using the link <http://10.25.97.40:8080/job/Content-Push-Master-to-QA/>
2. Then you do the deployment process <http://10.25.97.40:4000/fires/deploy-central-qa/>
3. The deployment to the CIT is like optional and you can do it more upon demand as it runs every day automatically <http://10.25.97.40:4000/fires/deploy-cit-qa/>
4. Then Check for all the items ready to deploys are on QA and move them to “ready to test”
5. When you are deploying, in step 10, you need to make group aware of your action in case someone is doing or working in it same repo and you can email them saying like: with subject :- QA Central Recycling in 20 Minutes and with a body :- Please let me know if you want me to wait.

Thanks .

1. Then follow what is next as per the info in <http://10.25.97.40:4000/guides/firefighter-guide/>
2. Once I am done with all the steps I should be able to build the release job in the link <http://10.25.97.40:8080/job/Content-Push-Release-2.8-to-QA/> which is mostly I do whenever I finish my steps and whenever I get the any request for content push from any member of the team.

When you see all the building and testing are gone well and successfully start sending emails to team with all the process completed. This is to be done most of the time twice. So you can update them one in mid-day and second at the end of the day. The subject can be like :- FF MID-DAY status and The body can look like :-

MD Firefighter Tasks:

 Content Pull

 celf-5

* + - * ages-5-8-formulated sentences
      * ages-9-21-formulated sentences

 rbans-form-a

* + - * figure copy
      * figure recall

 wais-5

* + - * set relations
      * similarities
      * vocabulary

 wms-5

* + - * adult-spatial addition

Content Push to QA   
 New war file deployed to QA

JIRA RTDs marked as RTT

Legend:

 = Completed  = coming soon

 = tests .subtest

EOD Firefighter Tasks:

 Content Pull

 test-vt

* + - * vt subtest

 wisc-v-se

 wms-5

* + - * adult-tfls

 Content Push to QA   
 New war file deployed to QA

JIRA RTDs marked as RTT

Legend:

 = Completed  = tests .subtest

Under the content pull all the sub-content can be found like in <https://github.com/pearca/qiactive/pull/5138> this will be repeated at the end of the day with current or latest updates.

While content pull is running you can follow in the build box server to follow what the Jasmine is doing during the test. Using the link 10.25.97.40 . Refer to the spread sheet link to get the full link and credential info. You can follow the steps in <http://10.25.97.40:4000/fires/mac-build-box/>

Once it is done as well, you can go to the <https://cit.qa.qiactive.com/choose/test/edit/151> to check what is made or if everything is going well.

Errors could happen: -

1. There are errirs that can happen because of some scripts we write in the command shell and not being able to use it or access it. Or if the request reply with a null…in this case we can disable the export, like wms5 symbol span
2. In content-push-release there could be user credentials conflicts, which will send an error with 403 conflicts.
3. A project can be failed also if another job is going through.
4. There are times the grails do not start easily; you can try to start it locally by going to the choose-share directory like
   1. cd $PEARSONPATH/web/choose-share
   2. then grails run-app
   3. or you can check separately for the grails by grails or grails –version
   4. if this does not work, you will need to go to the .bash\_profile if it is added to the PATH root.