***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. Make sure you don’t have other application running in your grail local server and make sure you close any previous local running specs or tests.
3. Go back to terminal and go to the virtual-env and activate it
   1. virtualenv qiactive-venv/
   2. source qiactive-venv/bin/activate
4. Go back to the repo link cd $PEARSONPATH
5. make sure you are on the master branch using git checkout master
6. make sure you have all the latest source from the master by using git pull
7. start merging it now safely using tools/git-safemerge origin/content-pull-xx
8. you will be asked to make sure say yes
9. you will be asked to commit with a message you can leave it without message using :wq
10. you push it then to the remote master using ./pre-push --content-pull
11. at the end it will ask you if you are sure to do the push to the master, you type master and hit enter
12. 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. There are times that we can push data to the QA in step 9 as there could be a problem
   1. Error code 302 which is a problem the Jenkin jobs from logging in to the QA using the giving credentials. To avoid this, you login and make some changes like renew the terms policy or changes of password, which is very rare, but still can happen
   2. Error code 500, server problem from pushing datas to the QA because of some criteria been violated. For example, subtest GUID conflict can happen, in which to same subtest having different subtest GUID. This is not allowed and the Jenkins can not override the file and does fail automatically. To avoid this,
      1. You should make sure you have same GUID for same subtest and make sure they are under different test group. You can also set the ID manually.
      2. You can upload it manually by login to the QA. First download the metadata from CIT by going to the specific subtest. Then you login 🡪 admin🡪test🡪godown and choose the upload file.
3. Then you do the deployment process <http://10.25.97.40:4000/fires/deploy-central-qa/>
4. 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/>
5. Then Check for all the items ready to deploys are on QA and move them to “ready to test”
6. 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. During the time of sending and receiving i18n we follow the instruction given in <https://github.com/pearca/qiactive/wiki/Translation-Management-System-(TMS)-Automatic-Tool>
3. There are times deploying war files to QA does not work. Is that case we can do it manually. That is we go to the folder location where the war file exist , workspace/web/ChooseShare/target That way you will find the war file and you download it. Then you open cyberduck and copy it manually by dragging the downloaded war file to webapp folder of the server side opened in the cyberduck. In the cyberduck app you may have many servers listed, you need to open the exact one and go to qiactive/tomcat/webapp location and there where you drop your war file. After that you go back to terminal and go to the server and restart it then the war file deployed manually will be able to extract successfully.

1. 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 MD status and The body can look like :-

If I am asked to build from someone after they made a change

1. If they made change in Central I build the ChooShare <http://10.25.97.40:8080/job/ChooseShare-QA/>
2. If they made a change in Assess I build the push-master-QA <http://10.25.97.40:8080/job/Content-Push-Master-to-QA/>
3. If they made with objective C related staffs – run the ios projects the names with App----

MD Firefighter Tasks:

 Content Pull

 ktea-3

* Math computation

 wais-5

* set relations

 wisc-v-se

* ordfrrd

 wisc-v-uk

* arithmetic
* block design
* comprehension
* figure weights
* information
* matrix reasoning
* picture concepts
* picture span
* similarities
* visual puzzles
* vocabulary

Content Push to QA   
New war file deployed to QA

JIRA RTDs marked as RTT

Legend:

 = Completed  = coming soon

 = tests .subtest

◊ Stims

EOD Firefighter Tasks:

 Content Pull

 wais-5

* set relations
* response booklet coding

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 errors 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.

During the production we will need go through first the approval to do that

1. I need a link like <https://pearson.service-now.com/support/mystoredetail.do?sysparm_document_key=sc_cat_item,a80b77af5de06100839b759a791d7c7d&locationId>=
2. After I get approved I will be able to sign in to the ls link <http://ls.ic.ncs.com/>
3. and then go to the Conan job scheduler using the link <http://releng.ic.ncs.com/conan/requests/job?rt=209151>
   1. when you made a the ticket for the productions the page, scheduled page should look like 

whenever I made a change to my local codes and I try to pull content from the master, I may get code diff message and if did not make any major change and do not want to stop me from pulling my content, I can first go to the terminal and do thses commandes, git reset --hard origin/master.

Running a test locally without the server or running the whole app

1. If I want to run single test, I just have to go to the specific html runnerSpec and run it. Example cd $PEARSONPATH/jasmine-tests/SpecRunner-td-1.html
2. If I want to run the whole test in the master repo I go to the location cd $PEARSONPATH/tools/allJasmine

How to raise ticket to deployment of Q-i 2.9  to [beta1.qiactive.com](http://beta1.qiactive.com/)

1. You should first prepare a pull request for the database for all the springs not deployed since the last deploymen. Prepare it like <https://github.com/pearca/qiactive/pull/5259/>
2. Place a folders full of these database data to the SFTP server so that other teams can access it. Then you raise a ticket for that. In the giving link <https://pearsonstsprod.service-now.com/main/newchange.do>

Injection tms to master and then to release. Whose responsibility is going to be. Iatnn? Today we have to create the new release branch called release 2.9 and we will inject the i18n files to that release as we want to deploy the release into BETA1. Breaking up the central files.

So we will need to do today :-

1. Create a new release called release-2.9
2. We will need to inject the internalization staff called i18n
3. We will then merge it with the master which contains all the releases including previous releases.