****

**Daily Sanity Trigger & Jenkins/Git Setup**

****

**PRImARY OWNER: D. PRAVEEN Kumar**

**SECONDARY OWNER: DEVOps TEAM (Kalavathi/Manish)**

**CrEATED DATE: 20-JUL-2022**

**Last UPDATED DATE: 08-AUG-2022**

**VERSION: 1.0**

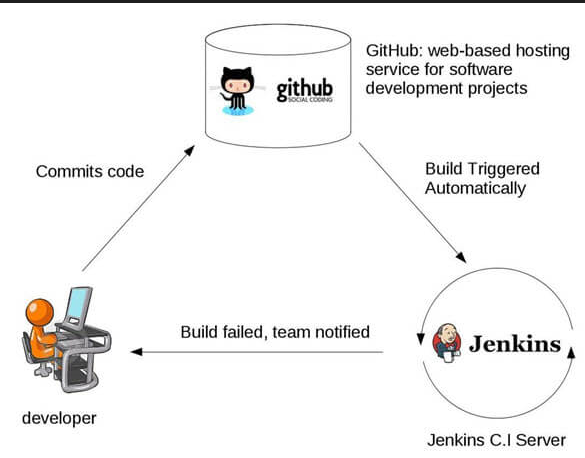
**PRODUCT OWNER:**

**Objective of the document:**

Document comprising End to End process of Jenkins and Git hub. To make sure how Jenkins & Git hub process works effectively as Continuous Integration and Continuous Development.

**Prerequisite:**

1. User should have some knowledge of overall flow of CI and CD process.
2. User should have the basic knowledge of Git Commands.
3. User should have the basic understanding of Java Script and Groovy script.



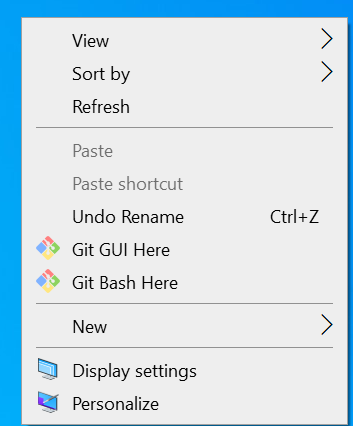
**Step By Step process for GIT Setup:**

Step By Step Installation Steps for Git and Git Bash:

1) Download Git from the official site of Git hub and Install from the windows.

<https://git-scm.com/downloads>

2) Post Installation of git hub, check if Git Bash is installed or by clicking right click on your computer.



3) Get Approval from DevOps Team to access IBM Git Official URL:

<https://github.ibm.com/ibm-cps-ame-platform/ibm-r2r-automation/branches>

4) Steps to Generate SSH Keys

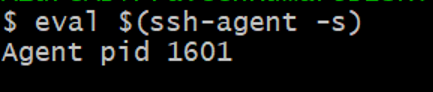
* Right click on your computer open “Git Bash”
* Enter ssh-keygen -t rsa -b 4096 -C “Enter Your IBM Mail Id”



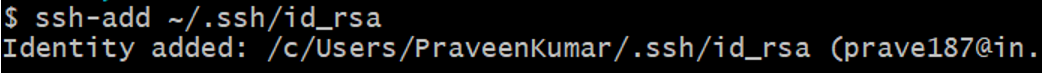
* Keep pressing Enter key, until you get message like “Your SSH key has been saved



* Next Enter, eval $(ssh-agent -s) agent pin will be generated.

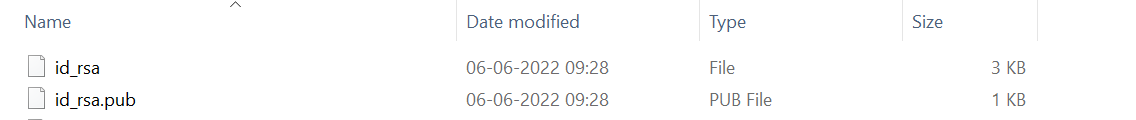


* Next Enter, ssh-add ~/.ssh/id\_rsa , **id\_rsa** file will be generated



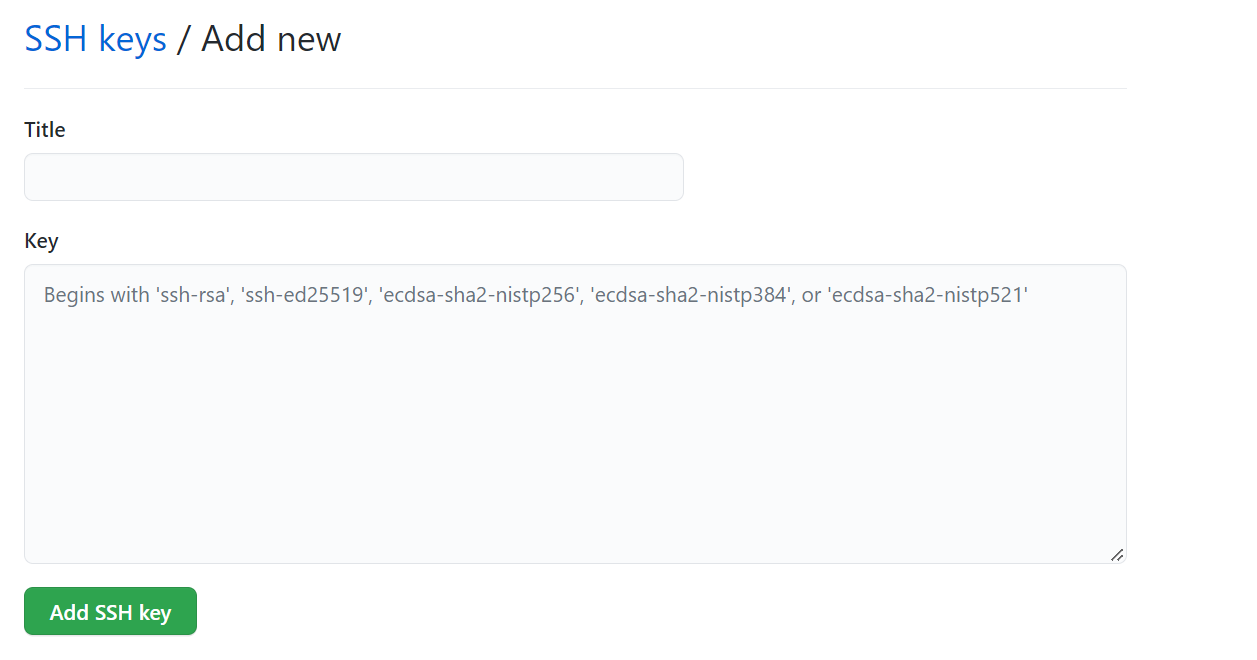
* Next Enter, clip < ~/.ssh/id\_rsa.pub ,**id\_pub** will be generated
* Make sure files are generated in your C drive

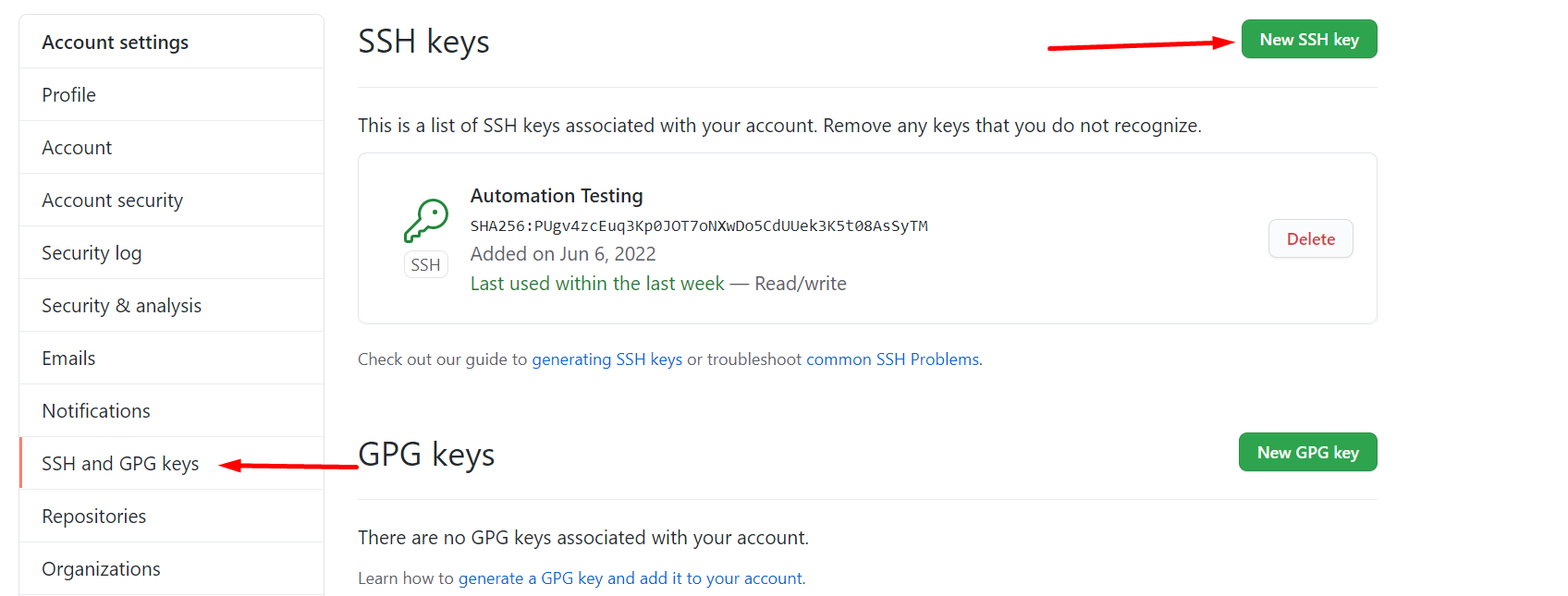
C:\Users\Your Name\.ssh



5) Once SSH keys generated and need to be configured in Git Hub settings.

* Open IBM Git Hub Site
* Go to Your profile and click settings
* Click SSH and GPG Keys
* Click on New SSH Keys
* Copy the content inside the file id\_rsa.pub and paste under New SSH keys, **make sure there is no spaces from the top and button of the content**
* Then Submit, Your SSH file will be generated.

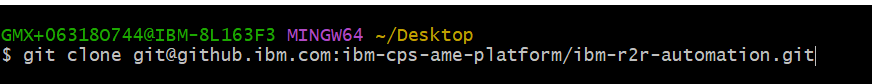


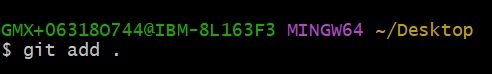


6)Step by Step process to work with Git Hub remote repository.

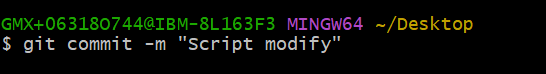
* Create New folder in local and name it as proper convention

E.g., Sanity-Production-Oriflame

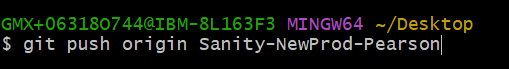
* Clone a Git remote repository to local folder 
* Once local repository created, Go inside that folder
* Place all your files into local repository and add.



* Once files are added, then Get commit



* Once Commit, Next step is push to remote repository



* After pushing the code, make sure you get successful message like “Meta Data is Transferred in to remote repository”

=============Ending of Git Installation & Set Up==============

**Step By Step process for Jenkin Setup - CI & CD:**

1) Get Approval from DevOps to access Jenkins

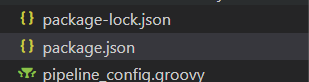
<https://gbsjenkins.edst.ibm.com/job/R2R/job/ibm-r2r-automation/>

2) Make sure you get an access for **IBM R2R Automation Branch**

3) Go inside a local git repository and install all node dependencies by entering **npm init** in the terminal or Command Prompt



4) Post installation of node dependencies, make sure package.json file will be created in your folder.



5) Need to create two files which is required to work with Jenkins CI/CD

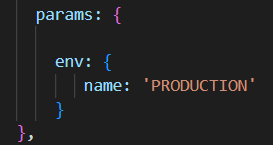
* Jenkinsfile
* Pipeline\_config.groovy

Those files are written in **Groovy Script & Java Script,** which is one time set up so that next time you can use those files in anywhere to work with Jenkins. Those files will be available in box link.

<https://ibm.ent.box.com/folder/167763794187>

6) In Jenkins it a mandatory to write a code to accept parameter by adding few steps in your code

**Add this code in your config file**



Instead of giving the sheet name, we need to pass it as parameter

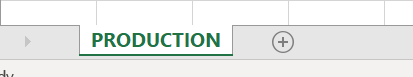
**Add this code in your spec file by removing excel sheet name and replace with parameter**



7) In Input data sheet, the sheet names should be in such a way that both the sheet name and the parameter in the Jenkins file should be matched.

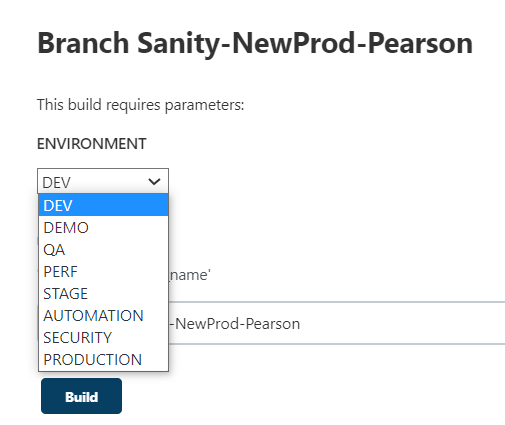


Input Data Sheet Name - Sample



8) Once all the setup is ready, push it to git hub repository.

9) Post pushing, make sure repository is created in Jenkins and select parameter and click build



10) Once script is completed, you can see allure report will also be generated.



==========================Ending of Jenkins Set Up==========================

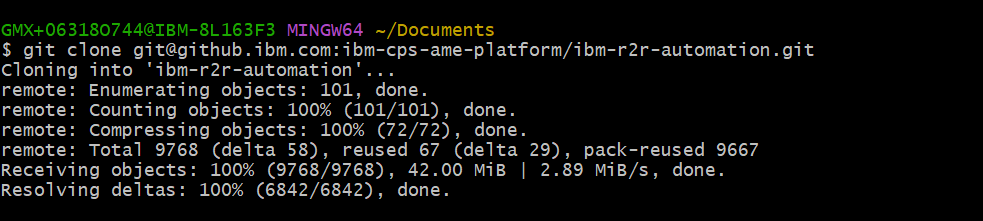
**Step By Step process for Jenkins Manual Trigger – Adhoc Request**

**Note:**

* Below Steps are covered for the first-time user who wants to trigger sanity manually.
* Users who already set up manual steps kindly proceed from the step 9 to step 21.

1. Please make sure, above all the steps are completed to proceed further to work with manual trigger of Jenkins
2. Clone Respective branch from the git which you are going to clone locally for script update, push, commit and pull
3. Open Git bash and enter

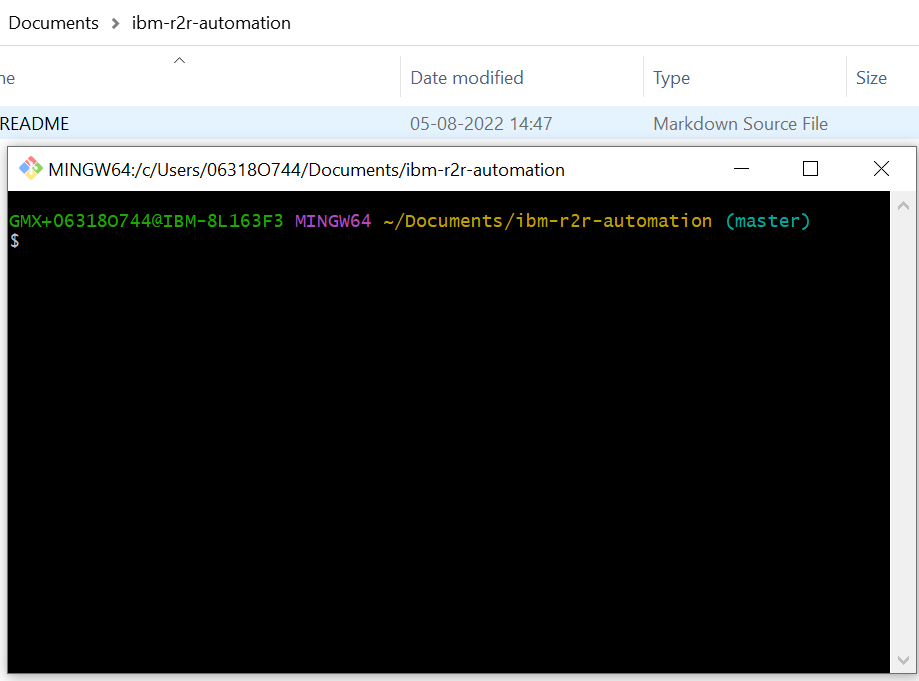
**git clone** [**https://github.ibm.com/ibm-cps-ame-platform/ibm-r2r-automation**](https://github.ibm.com/ibm-cps-ame-platform/ibm-r2r-automation)



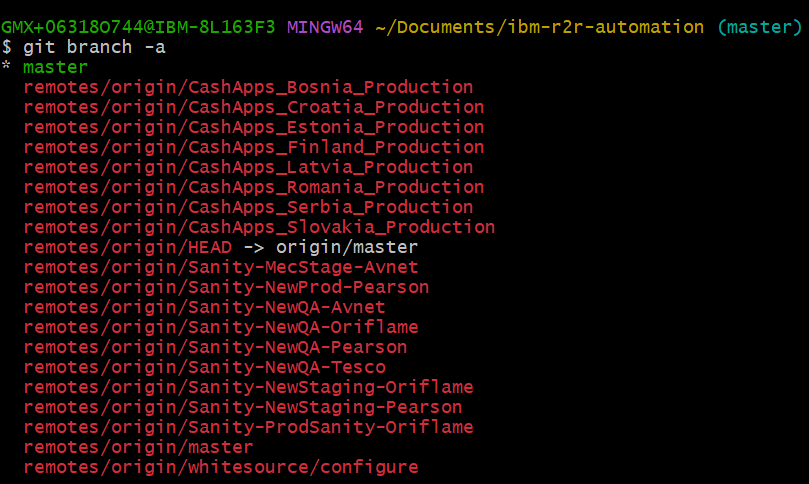
1. Once git clone is done, you could see “ibm-r2r-automation” folder created in your local directory.



1. Once local directory is created, go inside that folder and open Git bash



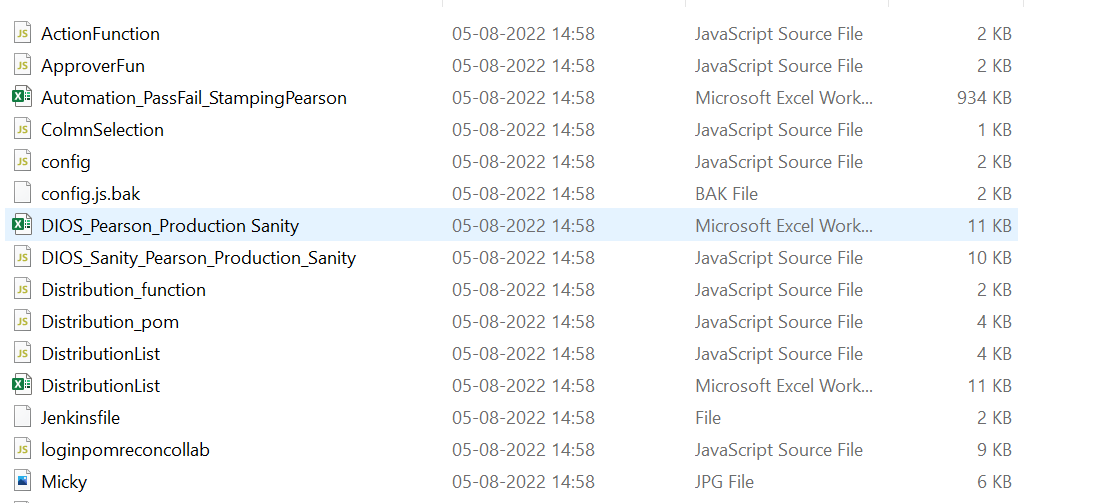
1. In Git bash enter **git branch -a** to view list of branches in the git repository



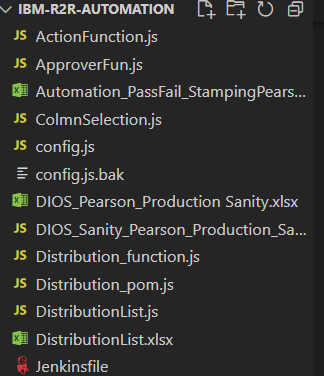
1. Next steps you need to pull remote repository branch to local directory for that you need to open git bash and enter **git pull origin -b “Branch name”**



1. After pulling the branch from the git, you could see remote repository branch files are exported into the folder “ibm-r2r-automation”

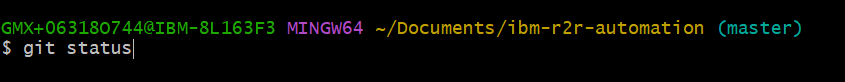


1. Open visual studio code and open folder with the local directory “ibm-r2r-auromation” and update the sanity script and recon date sheet accordingly.



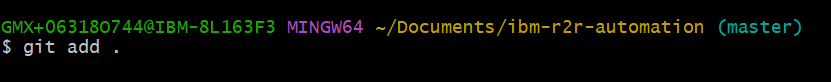
1. Once update is done, go into the local git repository” ibm-r2r-automation”, open git bash and enter **git status** in the git bash

**Note: Git status** commend is used to check the modified file. If you want to check which files you are modified and yet to push into git, then you can use this commend.



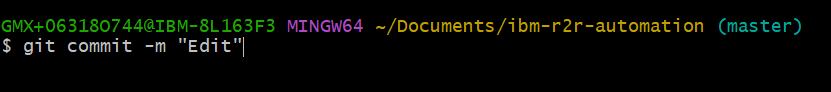
1. Next, you need to add modified file, for that enter

**git add “file name” or git add .** in the git bash



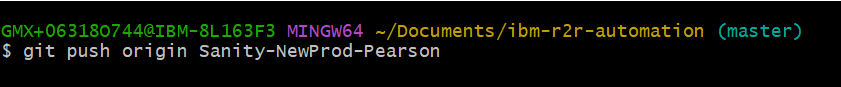
1. Next you need to commit file for that enter

**Git commit -m “Reason for commit”** in the git bash



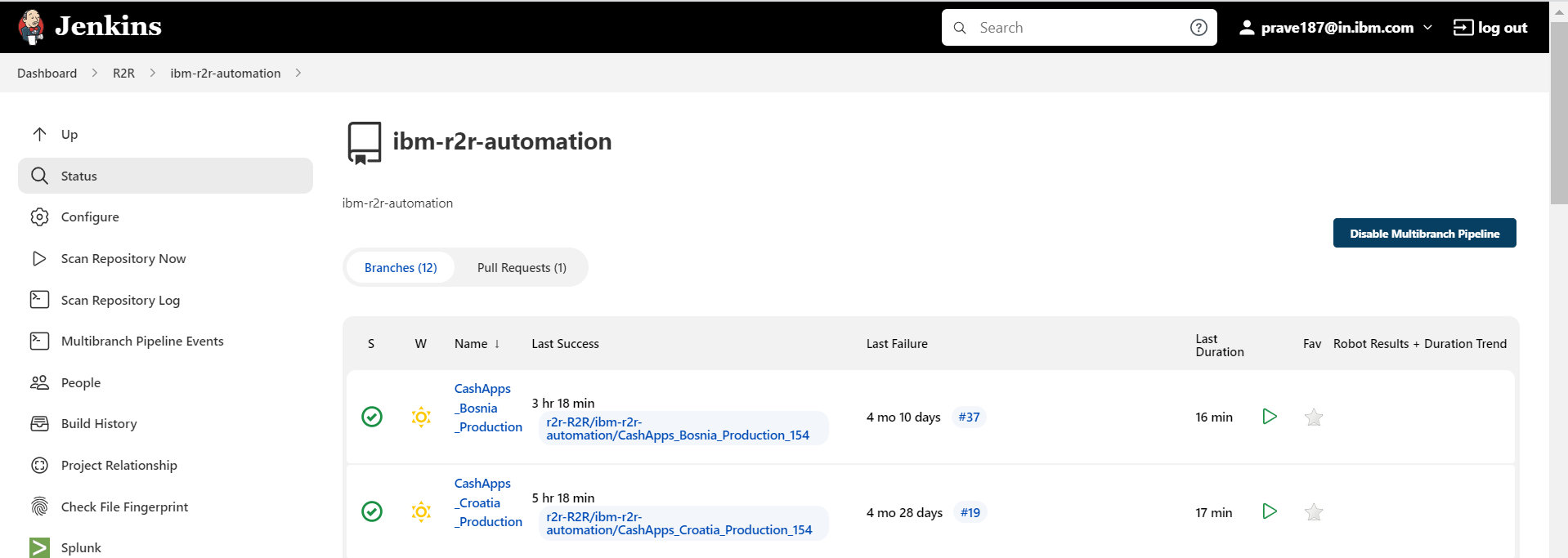
1. Next, once commit is done, you need to push files in to remote repository for that enter

**Git push origin -m “Reason for push”** in the git bash

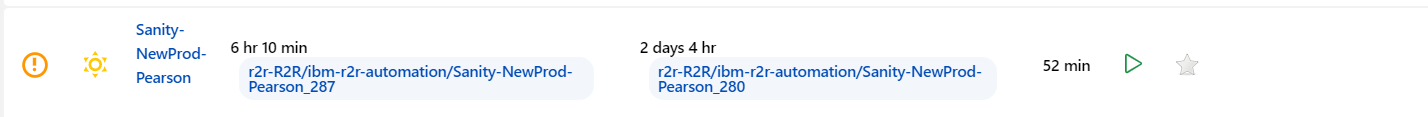


1. Once push is done, Open Jenkins and go in to ibm-r2r-automation

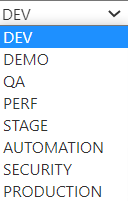
[**https://gbsjenkins.edst.ibm.com/job/R2R/job/ibm-r2r-automation/**](https://gbsjenkins.edst.ibm.com/job/R2R/job/ibm-r2r-automation/)



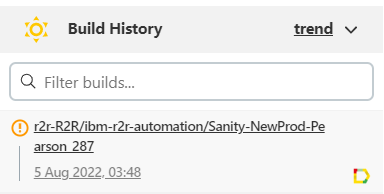
1. Next go into Jenkins’s branch where you are going to trigger manual sanity



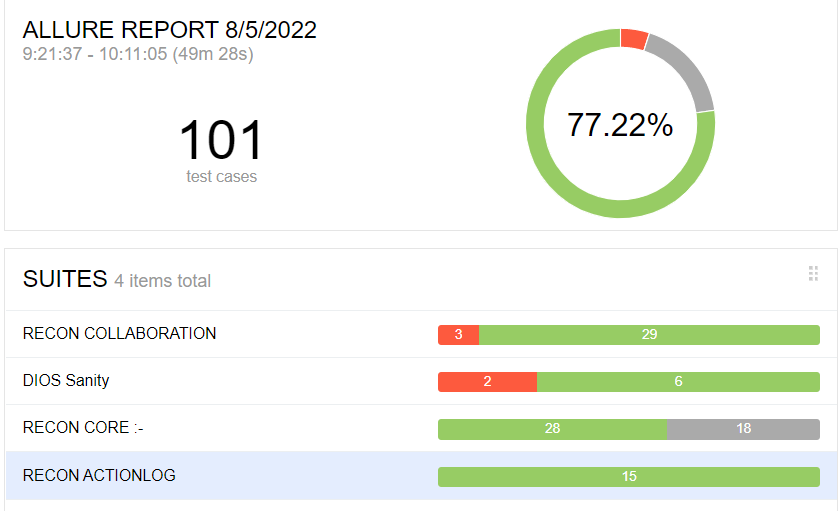
16.Next, click on **build with parameter** and click on **environment** drop down and click on build



17.After build trigger, you could see build is running in the build history.



18.post completion of build you could see allure report icon right side of the build and click on that icon, it will open allure report page.



19.Next, take screen shot of all the report and store it into your local repository and place it into box folder.

[**https://ibm.ent.box.com/folder/167904022201**](https://ibm.ent.box.com/folder/167904022201)

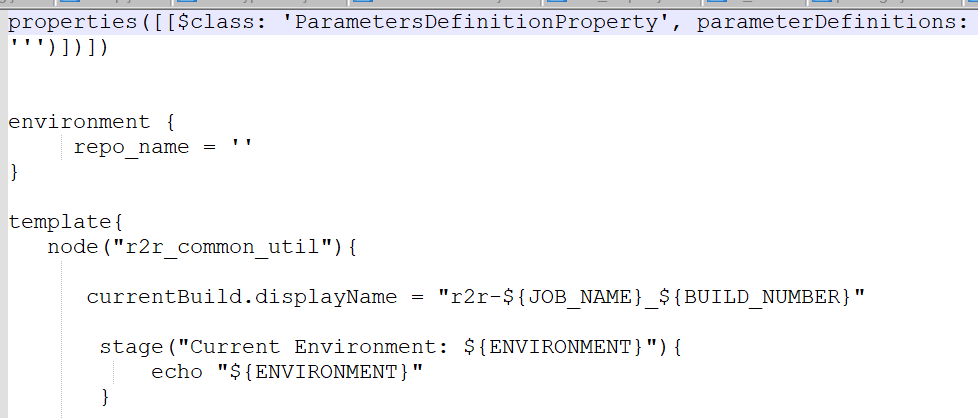
20.If sanity is passed, send sanity mail to all the stack holders which includes, pass/Fail count, Status summary, execution summary, client and program details.

21. If sanity is Failed, please raise a defect with sufficient screen shots to respective defect owner, and Send sanity mail with includes defect number, pass/Fail count, Status summary, execution summary, client and program details.

**Step By Step process for Jenkins Automatic Scheduler – Daily regular sanity**

**Note:** Below Steps are covered for the first-time user who wants to schedule sanity automatically.

1. Before proceedings with Jenkins’s scheduler, you must complete the steps of Jenkins manual trigger.
2. Go to local git repository “ibm-r2r-automation” folder and open the Jenkins file.



1. In the Jenkins file, go to pipeline Triggers at the right-hand side

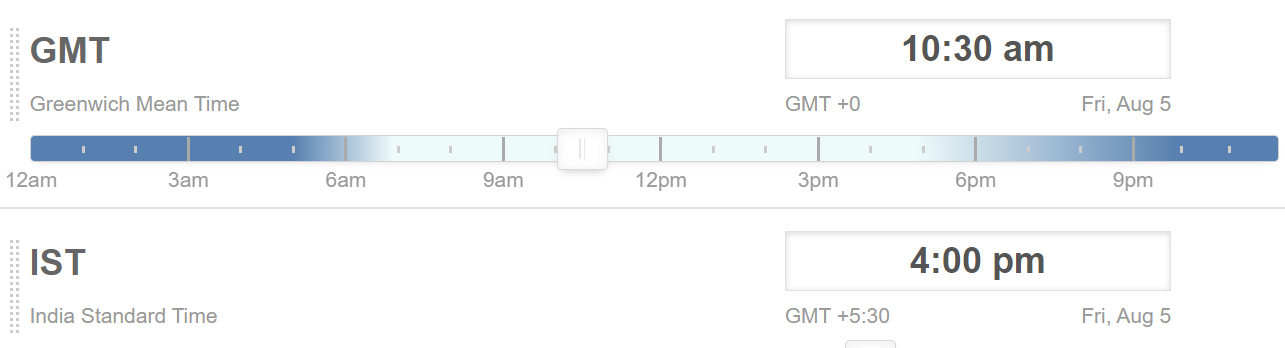


1. In the Jenkins we are following CRON format for scheduling the timing and also you need to convert GMT to IST format as server is following GMT format. Please refer below document to learn about CRON table.

<https://en.wikipedia.org/wiki/Cron>

1. Once you are ready with time then you must convert GMT to IST format, for that please refer below URL for conversion

<https://savvytime.com/converter/gmt-to-ist>

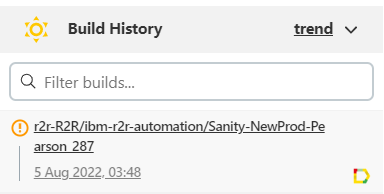


1. Once converted into IST format then come to Jenkins’s file, go to pipeline Triggers, and give the time

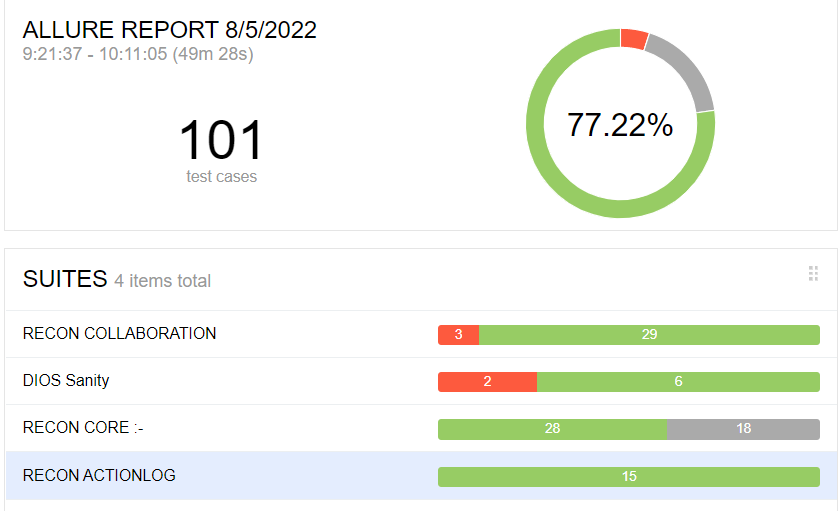
**Note**: While providing the time you must give **minute first then following hours.**



1. Once scheduled timers, follow the above steps “**Jenkins Manual Trigger – Adhoc Request**” for pushing the files in to remote git repository from the step 10 to step 13.
2. Once Timers is scheduled, build will run as per the scheduled time.
3. After build triggers, you could see build is running automatically as per the scheduled time.



1. post completion of build you could see allure report icon right side of the build and click on that icon, it will open allure report page.



1. Next, take screen shot of all the report and store it into your local repository and place it into box folder.

[**https://ibm.ent.box.com/folder/167904022201**](https://ibm.ent.box.com/folder/167904022201)

1. If sanity is passed, send sanity mail to all the stack holders which includes, pass/Fail count, Status summary, execution summary, client and program details.
2. If sanity is Failed, please raise a defect with sufficient screen shots to respective defect owner, and Send sanity mail with includes defect number, pass/Fail count, Status summary, execution summary, client and program details.