FOSSID SCAN

The Fossid scan is useful to scan the checked in code which is sent to Gerri review, if it contains any GPL contents or Apache licensing code. Based on those it will provide the results success or failure.

**INTRODUCTION**

Once a user commits the code in Git and push the code for Gerrit review. Before the code review, we also want to check if the code contains any GPL contents or Apache licensing code. So this fossid scans the patch set diff with the open source code. This whole process is triggered with the Jenkins job <https://rdkjenkinsdev.stb.r53.xcal.tv/jenkins/job/Fossid-scan-clone/>

The fossid plugin internally looks for the fossid.properties on the Jenkins server and executes the steps according to which it is mentioned in the properties. Here first we are getting the Gerrit patch set difference and saving it a file on the Jenkins server. Then we are copying that file from Jenkins server to the fossid server to a specific location.

On the Fossid server the plugin will be responsible for the scan and this will be performed on that patch. These scan results is the JSON format. Based on the results match\_type: none / partial we decide it is a successful/ failure respectively.

We display the message on the Gerrit review URL on the comments section with the fossid scan results URL were users can go and check the results by themselves. On the Gerrit review URL we are updating the FOSSID-label to -1 or +1 if is a failure or a success respectively.

**Successful Scenario**

* When users will push changes to Git repo (Git commit or else Git amend) and push the changes to origin. A Gerrit event will be triggered and create a Gerrit review URL. The Jenkins job <https://rdkjenkinsdev.stb.r53.xcal.tv/jenkins/job/Fossid-scan-clone/> is also triggered on the push of code to origin and execute the fossid tasks on Jenkins server.
* The scans is being executed on the committed code on the Fossid server.
* The scan will check if there is any licensing code being copied. If there is no GPL contents or Apache licensing code committed, then it finds no matches in the scan results.
* A Scan results will be sent by the fossid plugin to Gerrit. In this case the scan will be successful and the results are being displayed on the Gerrit review in comments section.
* The Fossid-Label is being updated to +1

**Steps to reproduce:**

1. vi file.c

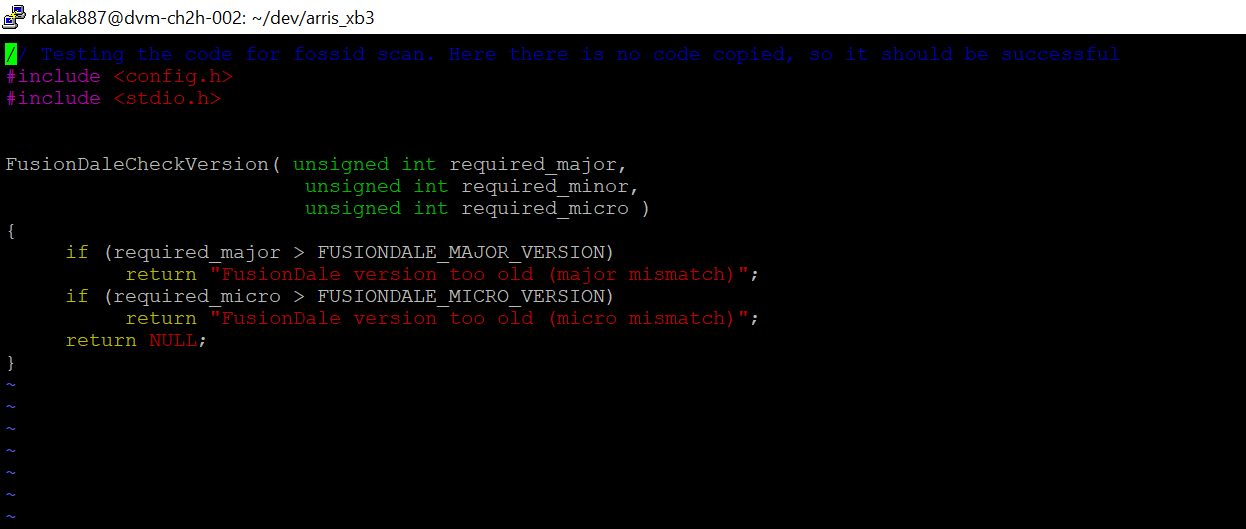
🡪 Add all the necessary code and save the file

1. git add –all
2. git commit –s
3. git push origin HEAD:refs/for/stable2

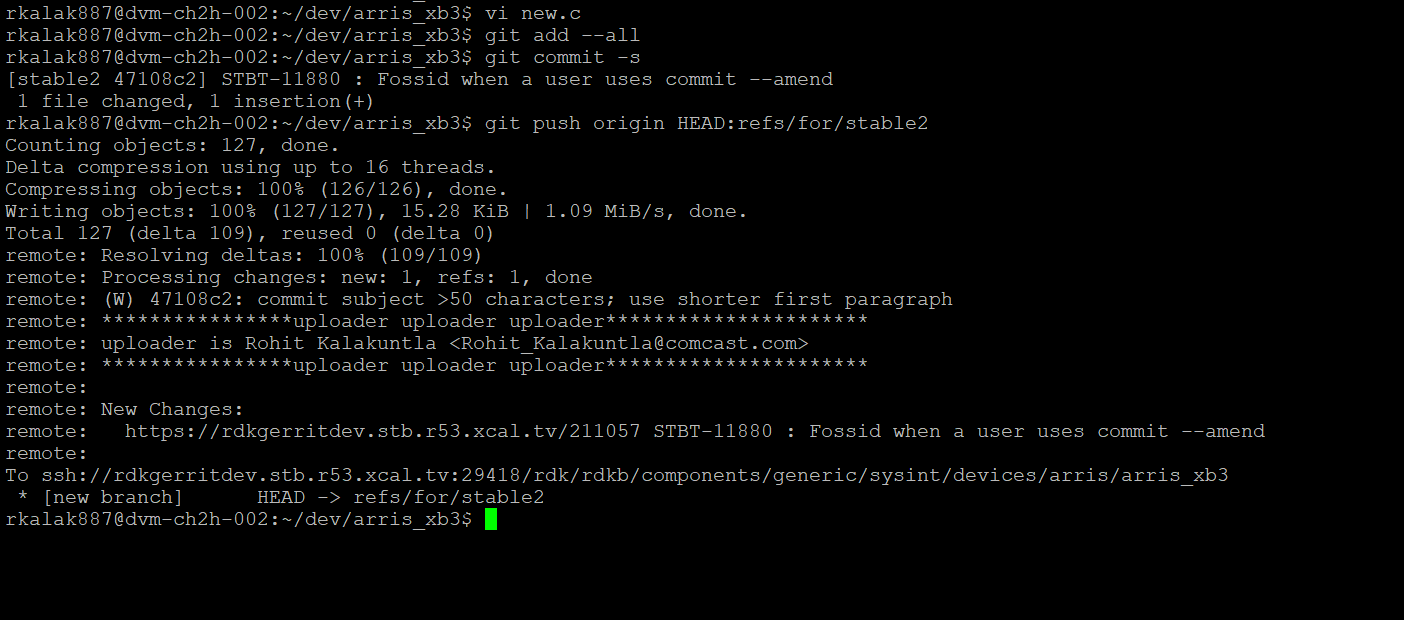
From the review details and Jenkins build job triggered details, we will have all the necessary information.

**Screenshots**

Create a file and save it

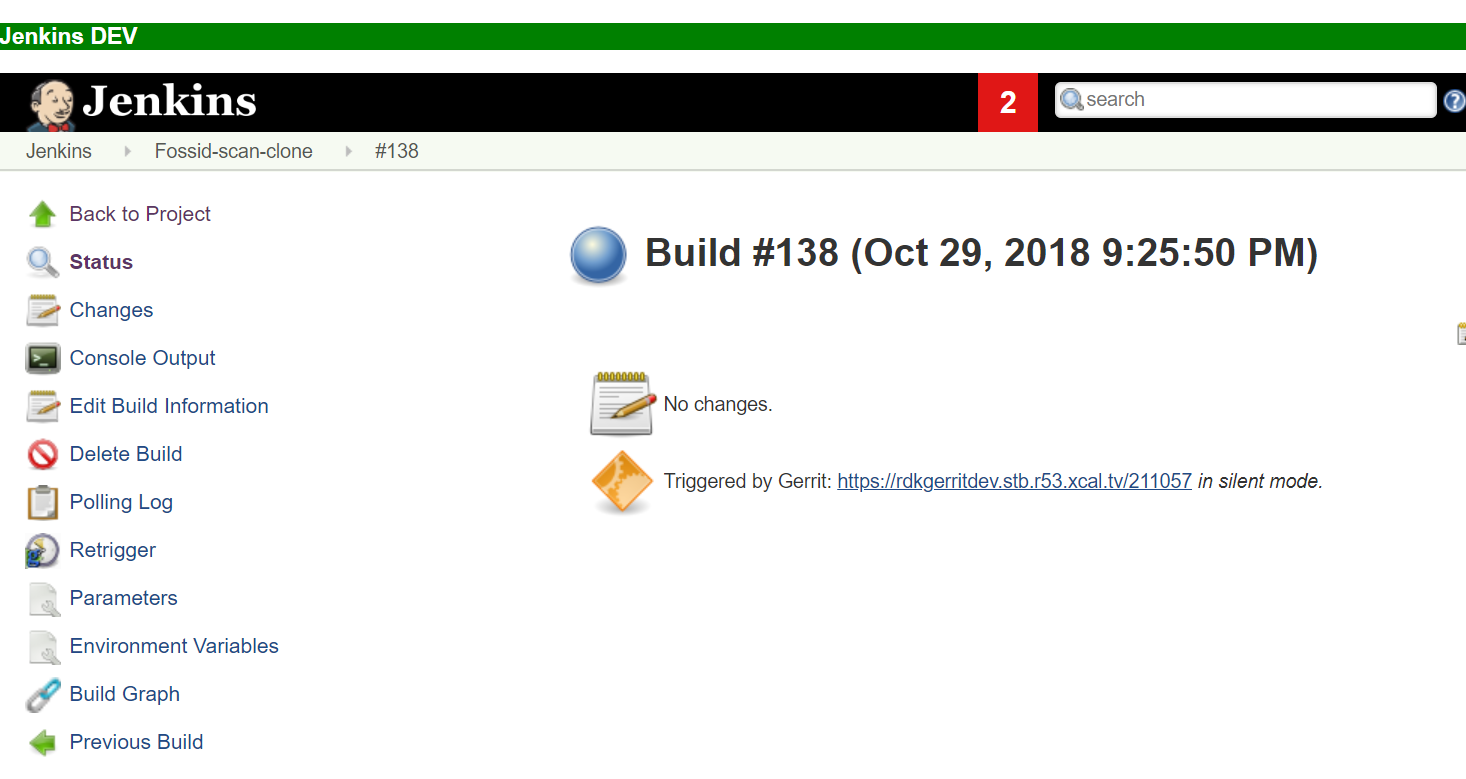


Push the changes.



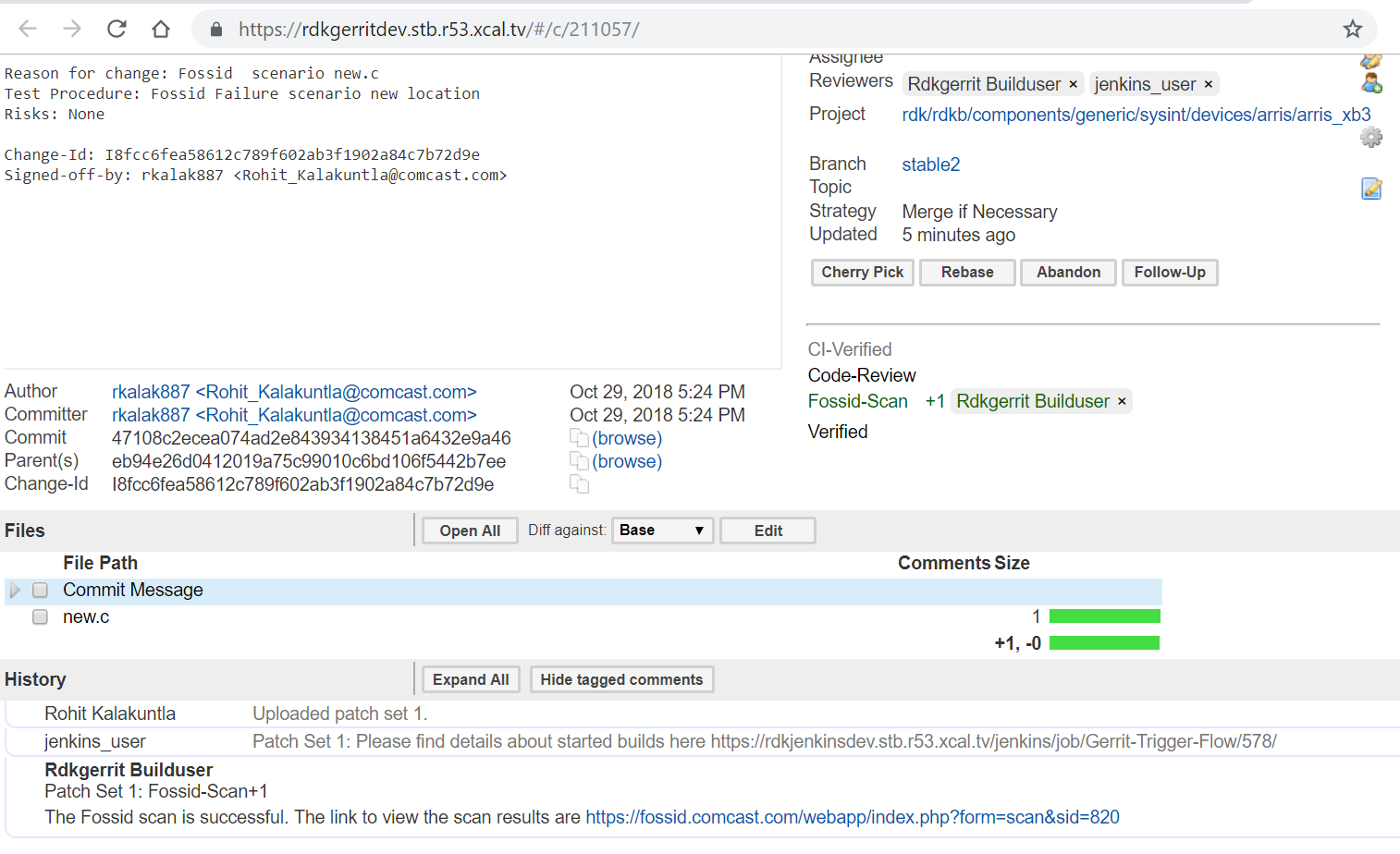
Here the Gerrit review created is <https://rdkgerritdev.stb.r53.xcal.tv/211057>

The Jenkins build job has been triggered

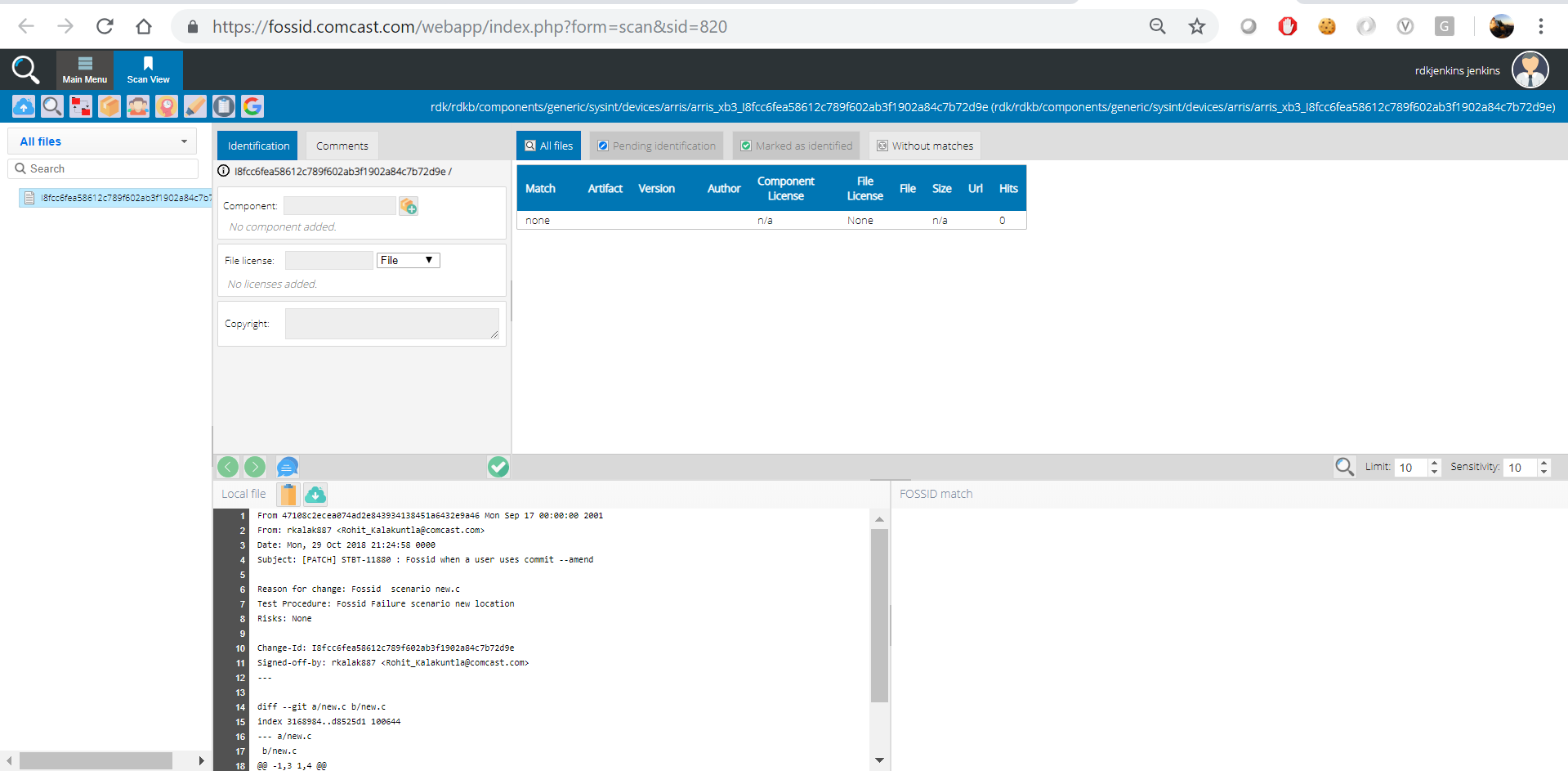


The scan results are being sent to Gerrit review. The URL in this case was

<https://rdkgerritdev.stb.r53.xcal.tv/211057>.



Here we see the Fossid-scan label +1 and also in comments “The Fossid scan is successful. The link to view the scan results are <https://fossid.comcast.com/webapp/index.php?form=scan&sid=820>”



Here in the Fossid scan results there is no match. Match: none for the change pushed.

**Failure scenario**

* When users will push changes to Git repo. An event will be triggered and initiate the job on Jenkins server.
* The scans is being executed on the committed code on the Fossid server.
* The scan will check if there is any licensing code being copied. If there is GPL contents or Apache licensing code committed, then it finds the match in the scan results.
* A Scan results will be sent by the fossid plugin to Gerrit. In this case the scan will be failure and the results are being displayed on the Gerrit review in comments section.
* The Fossid-Label is being updated to -1

**Steps to reproduce the scenario:**

1. vi file.c

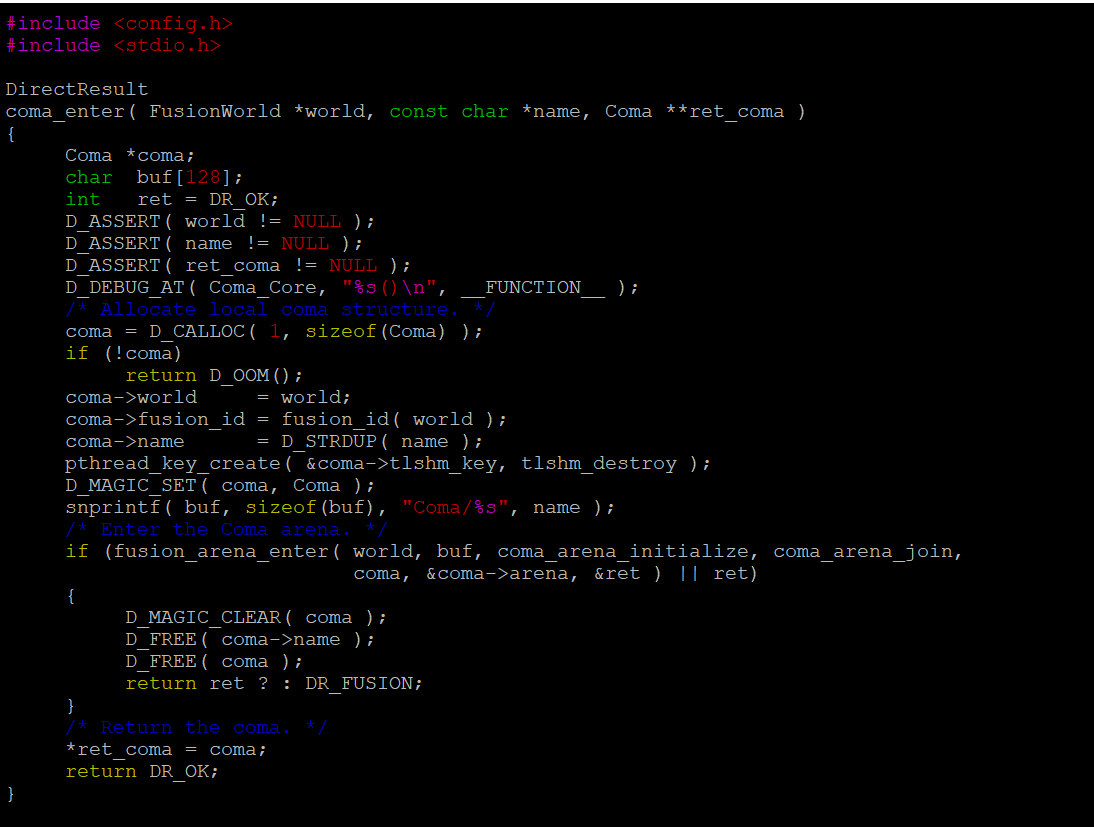
🡪 Add all the necessary GPL code or apache code and save the file

1. git add –all
2. git commit –s
3. git push origin HEAD:refs/for/stable2

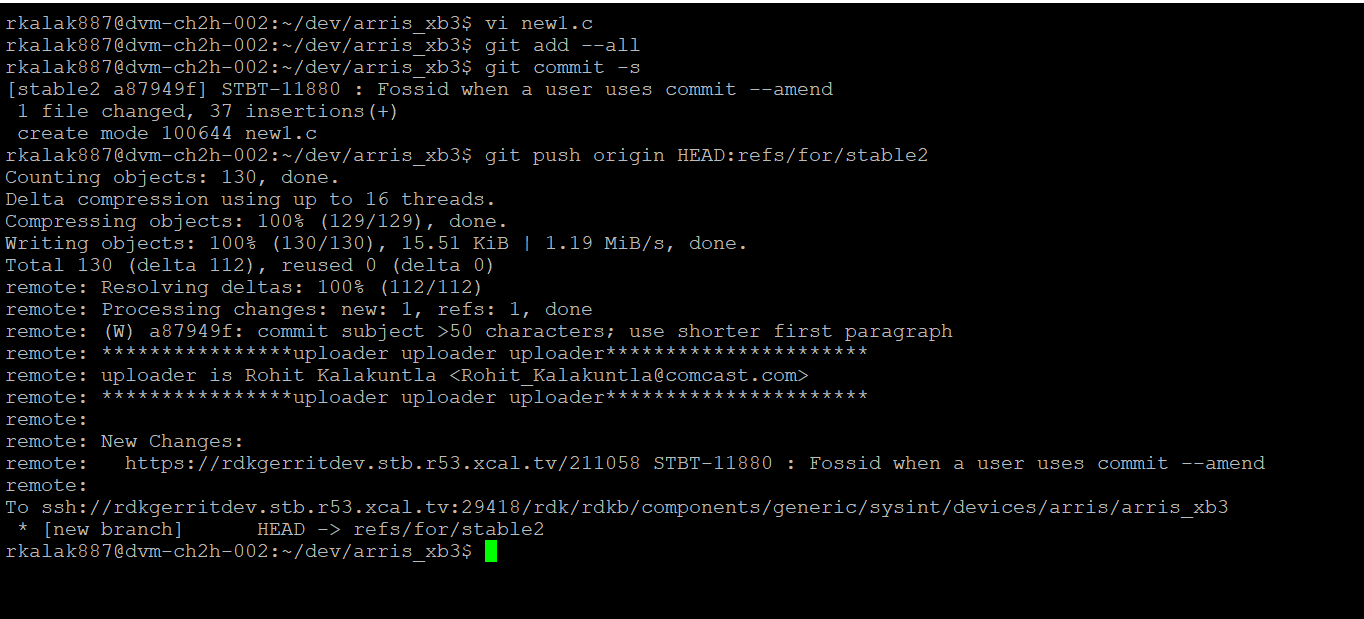
From the review details and Jenkins build job triggered details, we will have all the necessary information.

**Screenshots**

Create a file and save it

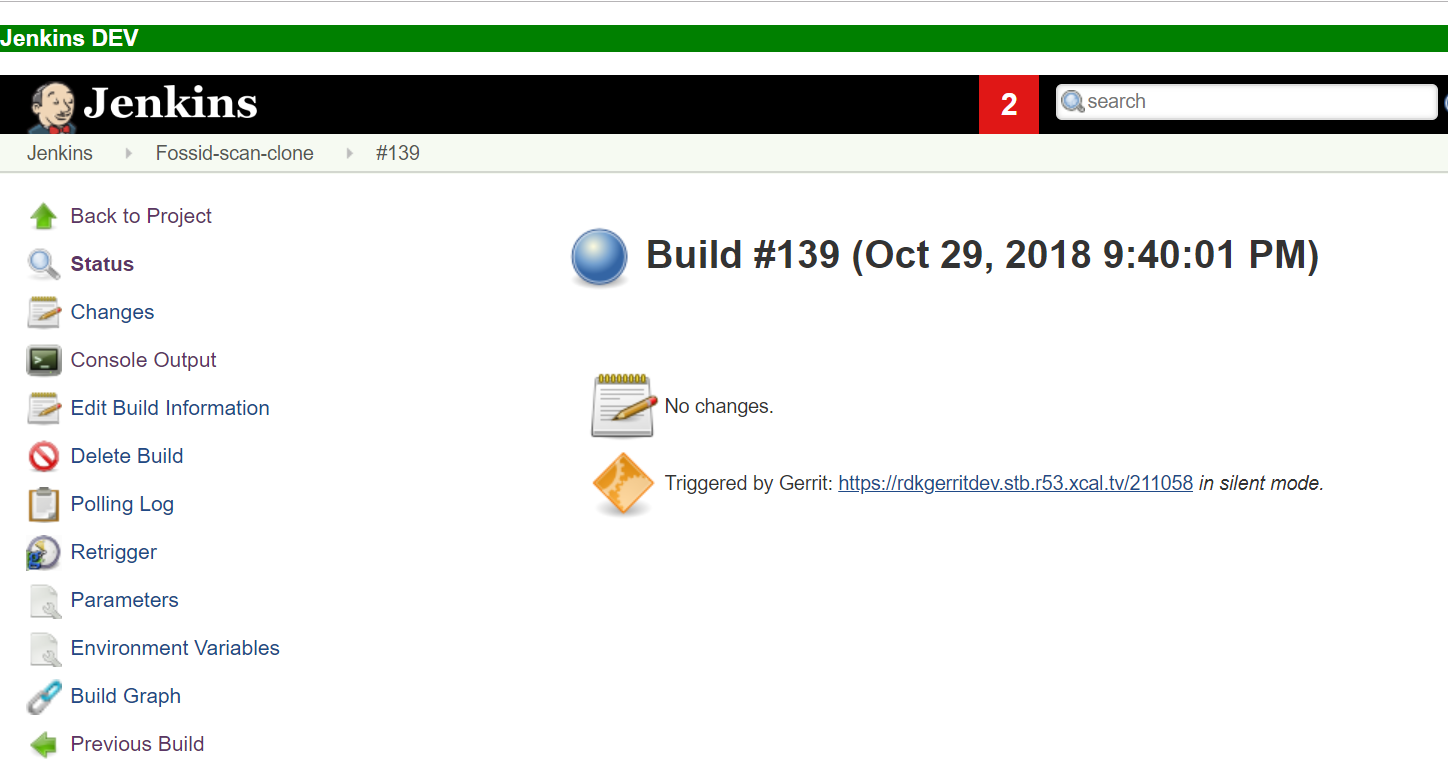


Push the changes.



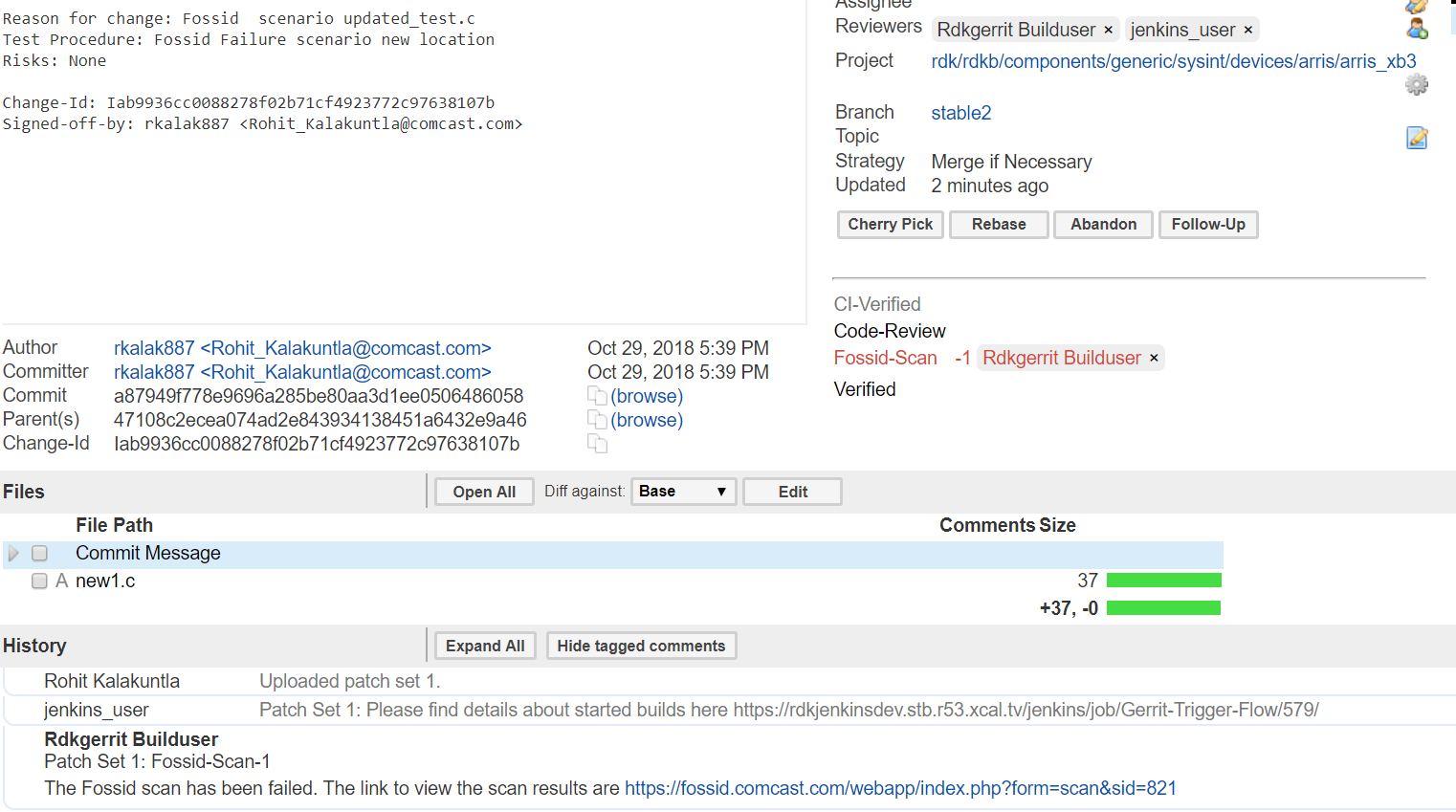
Here the Gerrit review created is <https://rdkgerritdev.stb.r53.xcal.tv/211058> and

The Jenkins build job has been triggered.

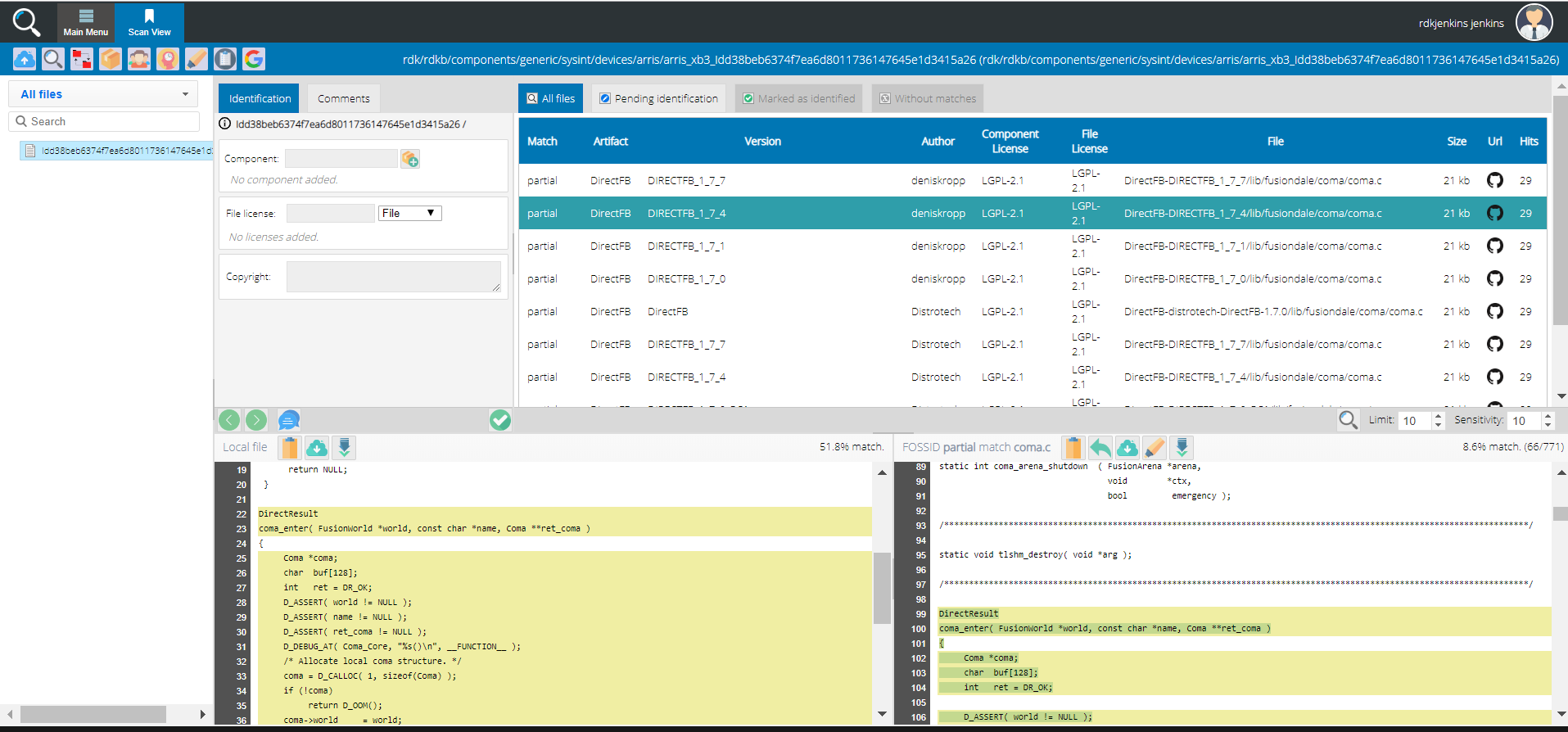


The scan results are being sent to Gerrit review. The URL in this case was

<https://rdkgerritdev.stb.r53.xcal.tv/211058>



Here we see the Fossid-scan label -1 and also in comments “The Fossid scan has been failed. The link to view the scan results are <https://fossid.comcast.com/webapp/index.php?form=scan&sid=821>”



Here in the Fossid scan results there is a GPL match. Match: partial for the change pushed. So it is displayed the code from which it has been matched and what part code has been matched. This will help the user to correct that particular code.

**Correcting a Failure scenario to make it Successful.**

* Once the user is aware of the reason for Fossid scan failure, he has to re-correct it.
* So he has to update the commited code accordingly.
* The user in this case will do a git commit –amend as it will be updated in same review id.
* When users will push changes to Git repo. An event will be triggered and initiate the job on Jenkins server.
* The scans is being executed on the committed code on the Fossid server.
* The scan will check if there is any licensing code being copied. If there is GPL contents or Apache licensing code committed, then it finds the match in the scan results.
* A Scan results will be sent by the FOSSID plugin to Gerrit. If it finds a match it will be again failure or if there is no match it will be a success. The results are being displayed on the Gerrit review in comments section.
* The Fossid-Label is being updated to +1 or -1 based on the success or failure respectively.

**Steps to reproduce the scenario:**

1. vi file.c

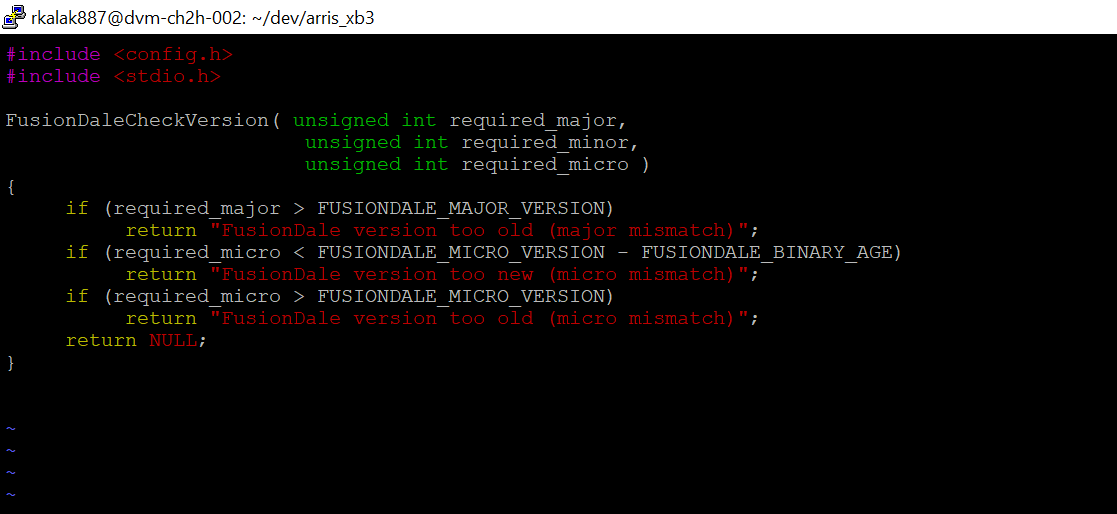
🡪 Update all the necessary code to avoid the GPL code or apache code and save the file

1. git add –all
2. git commit --amend
3. git push origin HEAD:refs/for/stable2

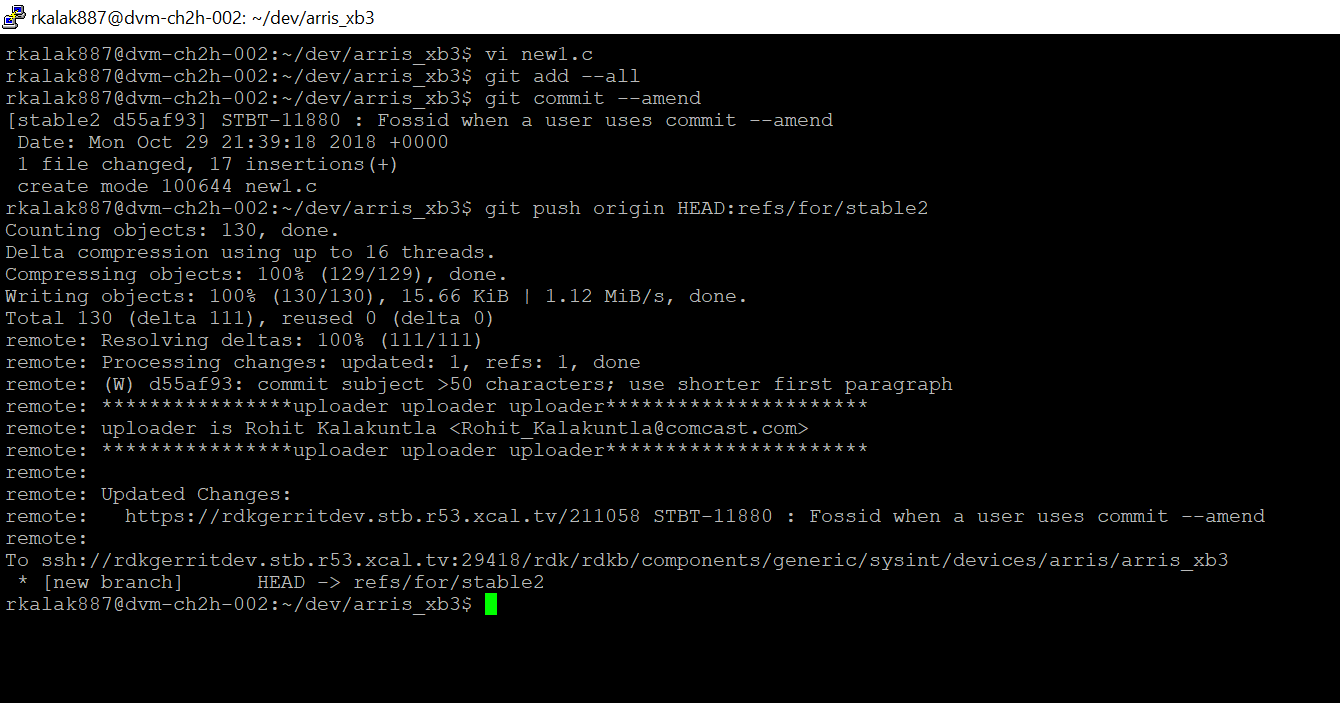
From the review details and Jenkins build job triggered details, we will have all the necessary information.

**Screenshots**

Updating the file which contains the GPL code and making necessary changes and saving it.

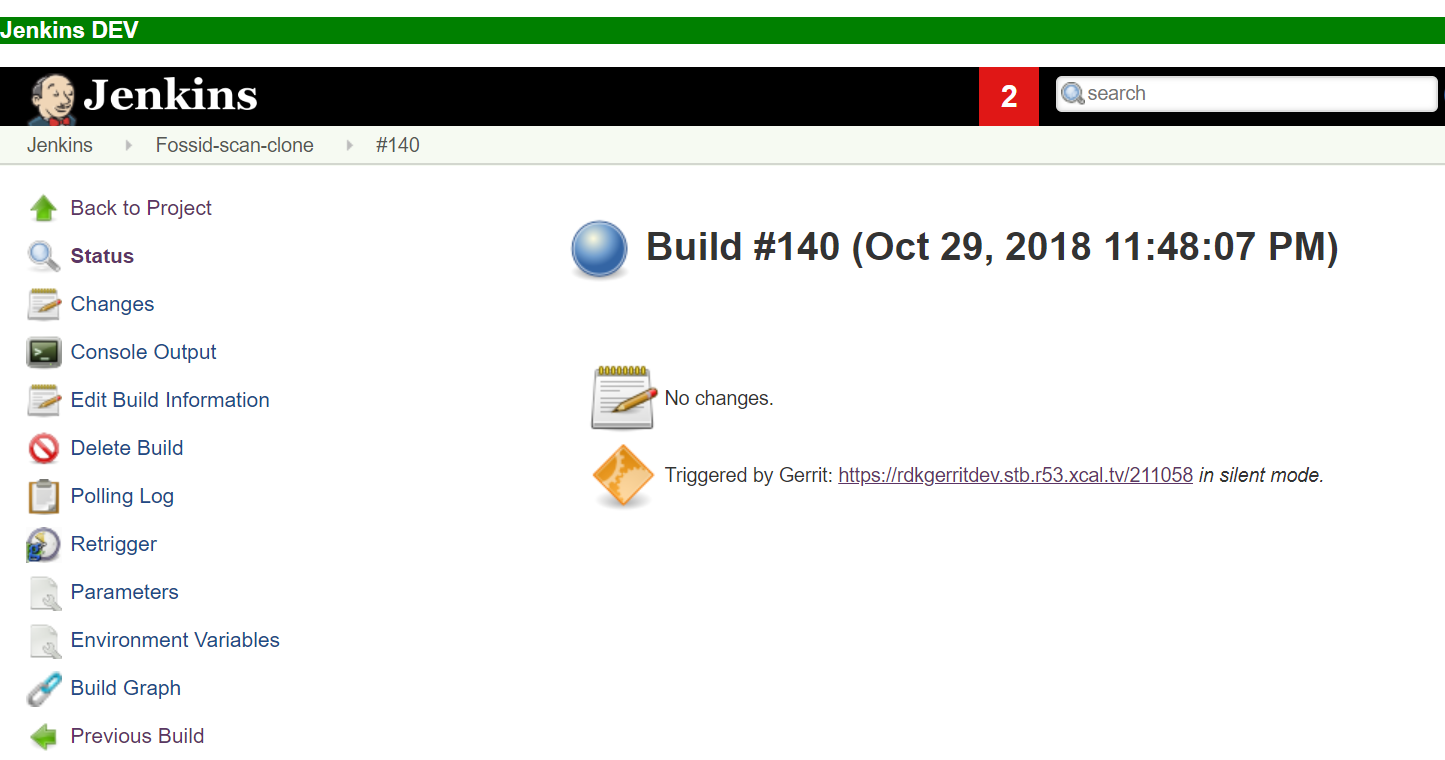


Amend the changes

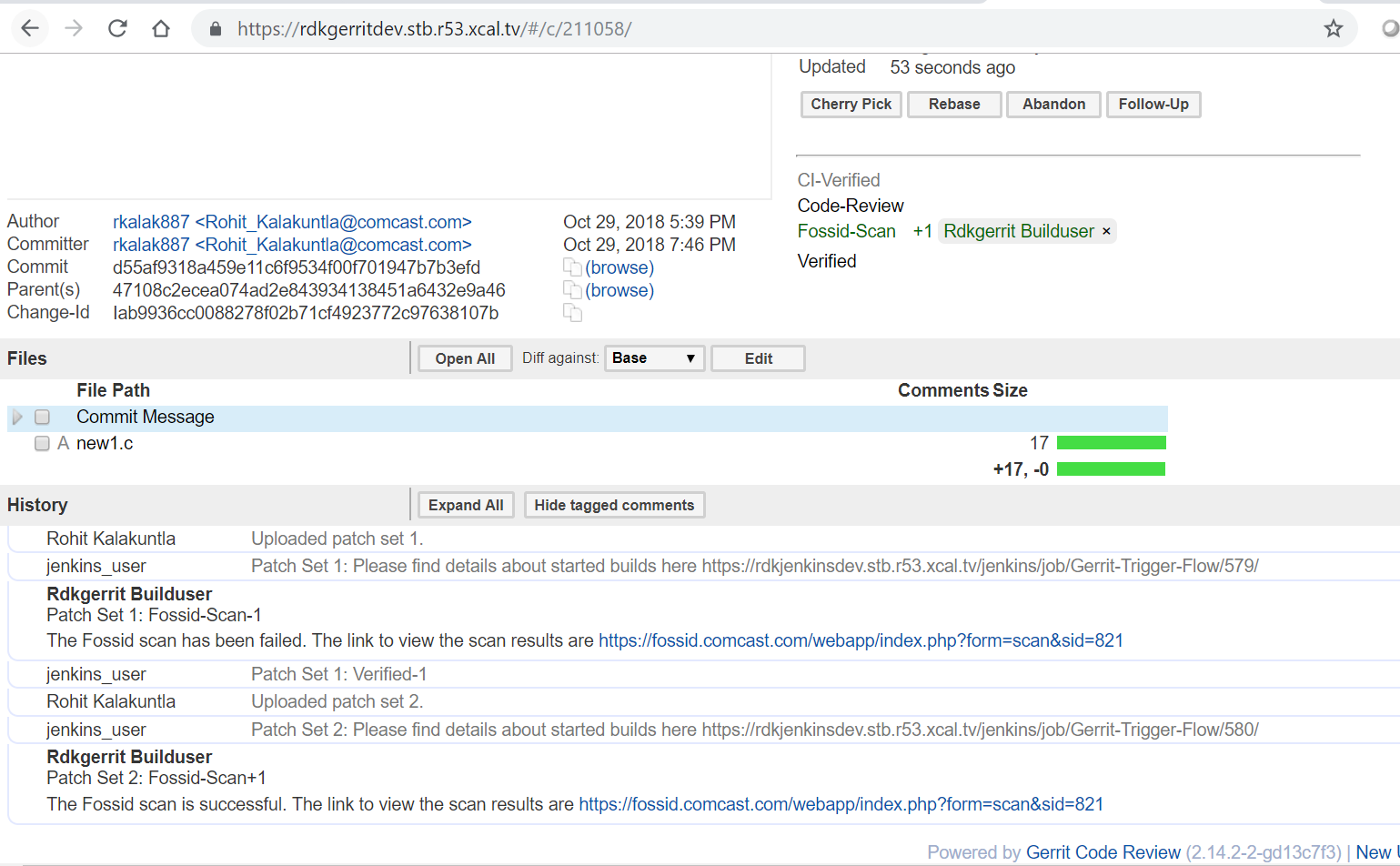


Here the same Gerrit review has updated <https://rdkgerritdev.stb.r53.xcal.tv/211058> and

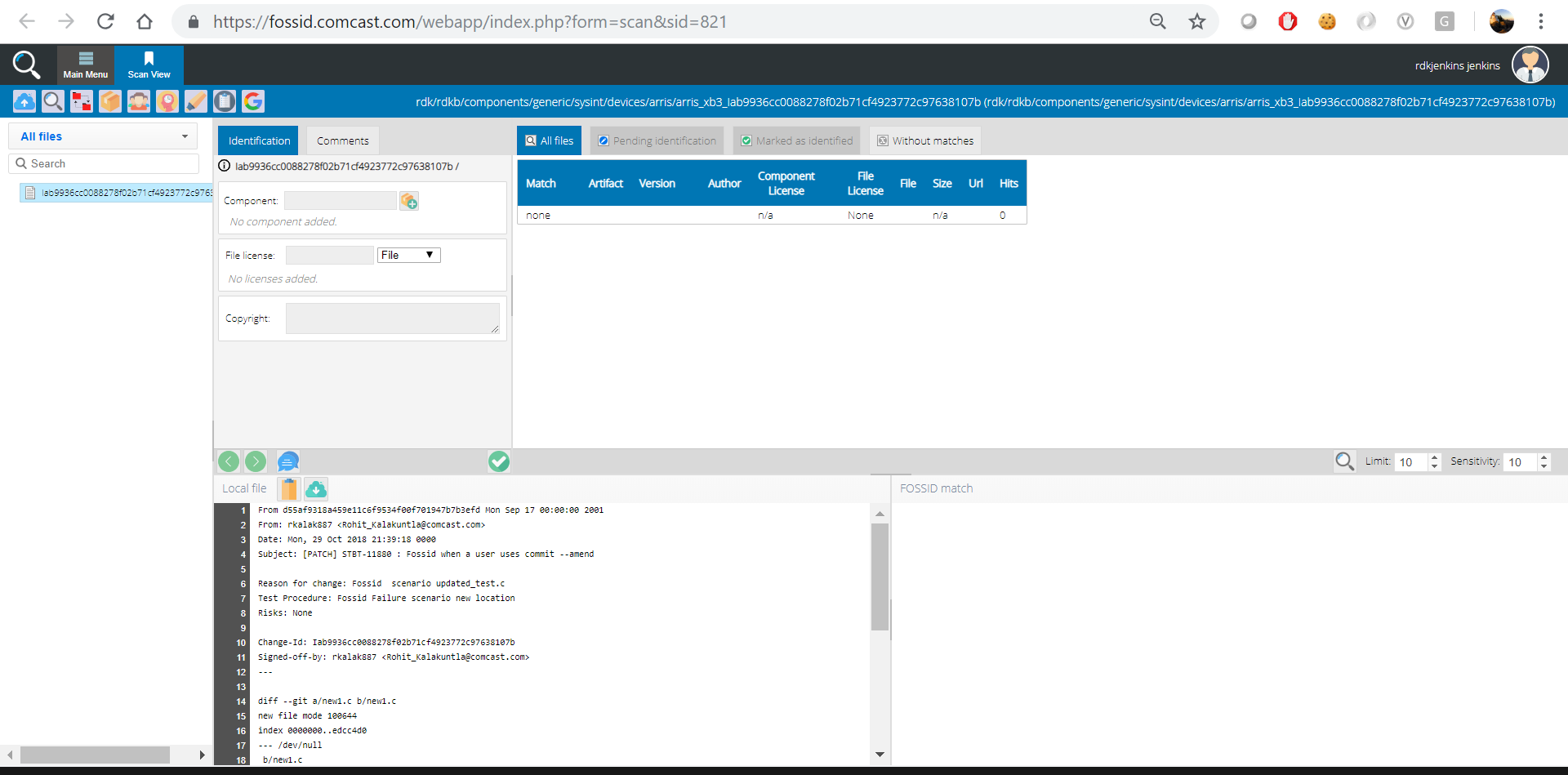
The Jenkins build job has been triggered.



The scan results are being sent to Gerrit review. The URL is the same because we have done a git amend <https://rdkgerritdev.stb.r53.xcal.tv/211058>



Here we see the Fossid-scan label +1 in the same Gerrit review and also in comments “The Fossid scan is successful. The link to view the scan results are <https://fossid.comcast.com/webapp/index.php?form=scan&sid=821>” This is the same scan id for the same Gerrit review.

Here in the Fossid scan results there is no match as the user has updated the code and amend it.