Brendan Helms

700695073

Secure Software Engineering

November 24, 2024

1. Step 1: Create a new file.

    1.1. I used the command echo "This is a text5 file for Gerrit review workflow" > text5.txt. This prints the statement within the quotation marks and saves the output to a .txt file called text5.txt. I then used the cat command to read the file.

```
Brendan@LAPTOP-FO6FPRSL MINGW64 /c/gerrit/gerritCodeReview (master)
$ echo "This is a text5 file for Gerrit review workflow" > text5.txt

Brendan@LAPTOP-FO6FPRSL MINGW64 /c/gerrit/gerritCodeReview (master)
$ cat text5.txt
This is a text5 file for Gerrit review workflow
```

2. Step 2: Stage the new file.

    2.1. In order to stage a new file, the ‘git add’ command is used. Staging essentially readies the change to be part of the next commit.

```
Brendan@LAPTOP-FO6FPRSL MINGW64 /c/gerrit/gerritCodeReview (master)
$ git add text5.txt
warning: in the working copy of 'text5.txt', LF will be replaced by CRLF the nex
t time Git touches it
```

    2.2. The git status command can then be used to ensure that the text5.txt file has been properly staged.

```
Brendan@LAPTOP-FO6FPRSL MINGW64 /c/gerrit/gerritCodeReview (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   text5.txt
```

3. Step 3: Commit the file.

    3.1. Now that the file is staged, the next commit will contain this change. The -m option allows for a message to accompany the change. This would allow other developers to see what is included in the new commit. I also again used the git status to verify there is nothing left to commit.

```
Brendan@LAPTOP-FO6FPRSL MINGW64 /c/gerrit/gerritCodeReview (master)
$ git commit -m "Add text5.txt for Gerrit review"
[master 645966b] Add text5.txt for Gerrit review
 1 file changed, 1 insertion(+)
 create mode 100644 text5.txt

Brendan@LAPTOP-FO6FPRSL MINGW64 /c/gerrit/gerritCodeReview (master)
$ git status
On branch master
nothing to commit, working tree clean
```

4. Step 4: Push the commit to the gerrit server

4.1. I first tried to push the commit to the remote repository on the gerrit server. However, I got an error message saying the branch 1 didn't exist.

```
Brendan@LAPTOP-FO6FPRSL MINGW64 /c/gerrit/gerritCodeReview (master)
$ git push gerritdemo HEAD:refs/for/branch1
Enumerating objects: 13, done.
Counting objects: 100% (13/13), done.
Delta compression using up to 8 threads
Compressing objects: 100% (8/8), done.
Writing objects: 100% (12/12), 1.26 KiB | 647.00 KiB/s, done.
Total 12 (delta 2), reused 3 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2)
remote: Processing changes: refs: 1, done
To http://localhost:8090/gerrit-demo.git
 ! [remote rejected] HEAD -> refs/for/branch1 (branch branch1 not found)
error: failed to push some refs to 'http://localhost:8090/gerrit-demo.git'
```

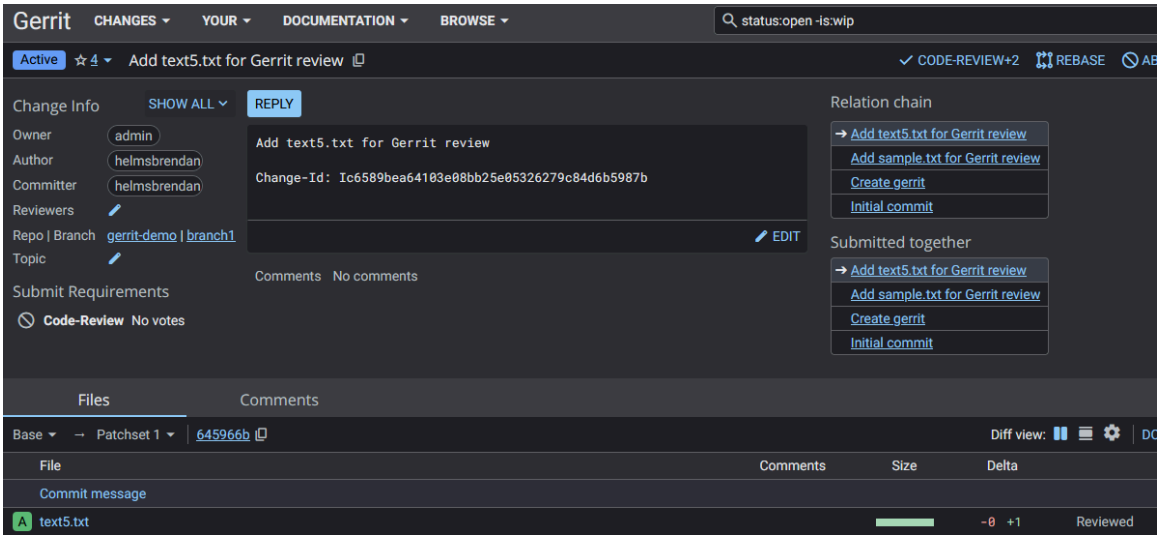
4.2. To fix this I created the branch 1 inside of the gerrit server. I then used the command fetch gerritdemo to notify the local repository of the changes.

```
Brendan@LAPTOP-FO6FPRSL MINGW64 /c/gerrit/gerritCodeReview (branch1)
$ git fetch gerritdemo
From http://localhost:8090/gerrit-demo
 * [new branch]      branch1    -> gerritdemo/branch1

Brendan@LAPTOP-FO6FPRSL MINGW64 /c/gerrit/gerritCodeReview (branch1)
$ git branch -r
  gerritdemo/branch1
  gerritdemo/main
  gerritdemo/master
  origin/HEAD -> origin/main
  origin/main
```

4.3. I had to disable the Change-ID requirement from the remote repository settings to get the push to work. Below you can see the success message from the git terminal as well as the text5.txt inside of the gerrit remote repository.

```
Brendan@LAPTOP-FO6FPRSL MINGW64 /c/gerrit/gerritCodeReview (branch1)
$ git push gerritdemo HEAD:refs/for/branch1
Enumerating objects: 13, done.
Counting objects: 100% (13/13), done.
Delta compression using up to 8 threads
Compressing objects: 100% (8/8), done.
Writing objects: 100% (12/12), 1.26 KiB | 647.00 KiB/s, done.
Total 12 (delta 2), reused 3 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2)
remote: Processing changes: refs: 1, new: 4, done
remote: warning: pushing without Change-Id is deprecated
remote:
remote: SUCCESS
remote:
remote:   http://localhost:8090/c/gerrit-demo/+/1 Initial commit [NEW]
remote:   http://localhost:8090/c/gerrit-demo/+/2 Create gerrit [NEW]
remote:   http://localhost:8090/c/gerrit-demo/+/3 Add sample.txt for Gerrit revi
ew [NEW]
remote:   http://localhost:8090/c/gerrit-demo/+/4 Add text5.txt for Gerrit revie
w [NEW]
remote:
To http://localhost:8090/gerrit-demo.git
 * [new reference]   HEAD -> refs/for/branch1
```



5. Step 5: Create a Reviewer

5.1. Before adding a reviewer I had to create a new user which had a few steps I had to follow. The ability to directly create a new user from the GUI is disabled so I had

to use the command line. To do this you must ssh to the remote repository. However, I got a permission error as seen below.

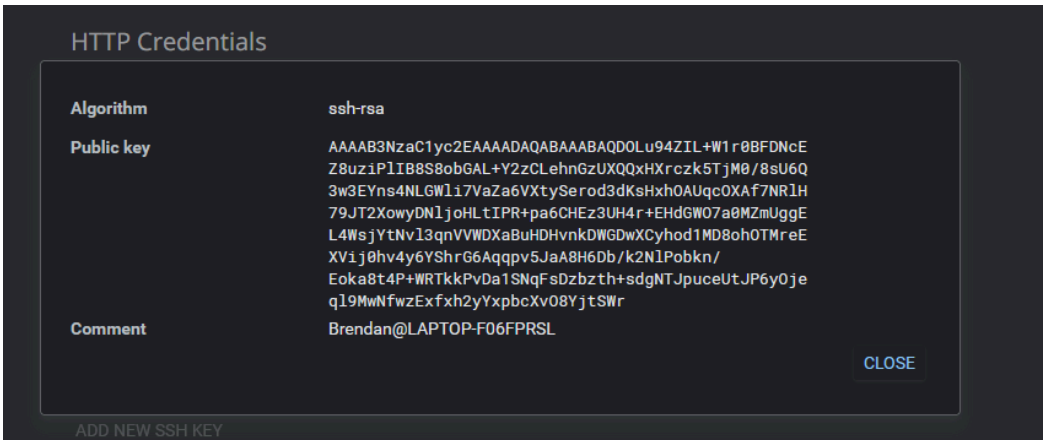
NOTE: port 29418 is standardly used for gerrit ssh

```
Brendan@LAPTOP-FO6FPRSL MINGW64 /c/gerrit/gerritCodeReview (branch1)
$ ssh -p 29418 admin@localhost
admin@localhost: Permission denied (publickey).
```

5.2. After doing some research I discovered that this was due to an ssh key not being added to the gerrit server. I didn't have an ssh key pair yet so I had to generate one. I found that this command generates a keypair and saves them into two files id_rsa and id_rsa.pub. I added the public key from id_rsa.pub to the gerrit server's 'SSH key section' so that the admin account could be properly authenticated and allow ssh communication from my git bash terminal.

NOTE: -t : encryption method; -b number of bits

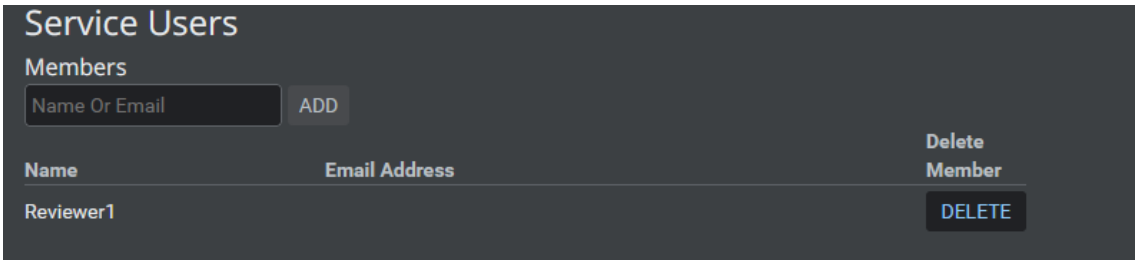
```
Brendan@LAPTOP-FO6FPRSL MINGW64 /c/gerrit/gerritCodeReview (branch1)
$ ssh-keygen -t rsa -b 2048
Generating public/private rsa key pair.
```



5.3. Now it is time to add the new user to the gerrit server. This user will serve as a reviewer to the changes made in earlier steps of this project. To do this I utilize the ssh connection which was made possible by adding the ssh key to the gerrit

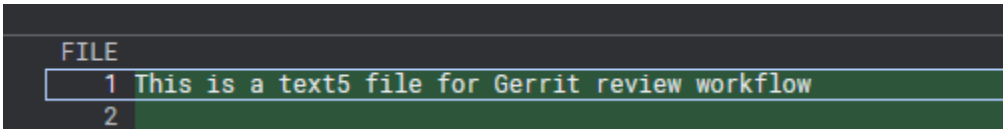
server. I had to create a passphrase when I created the keypair and was prompted to enter it here to allow for the new account to be created. I temporarily added the new ‘Reviewer1’ account to the ‘Service Users’ so it can be seen on gerrit.

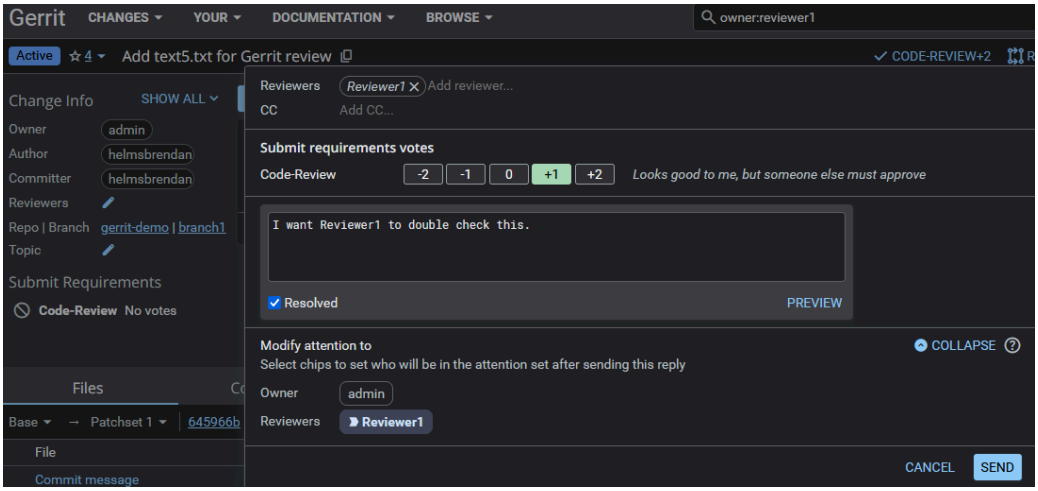
```
Brendan@LAPTOP-FO6FPRSL MINGW64 /c/gerrit/gerritCodeReview (branch1)
$ ssh -p 29418 admin@localhost gerrit create-account reviewer1 --full-name "Revi
ewer1"
Enter passphrase for key '/c/Users/Brendan/.ssh/id_rsa':
```



6. Step 6: Review the code and send a request to another reviewer

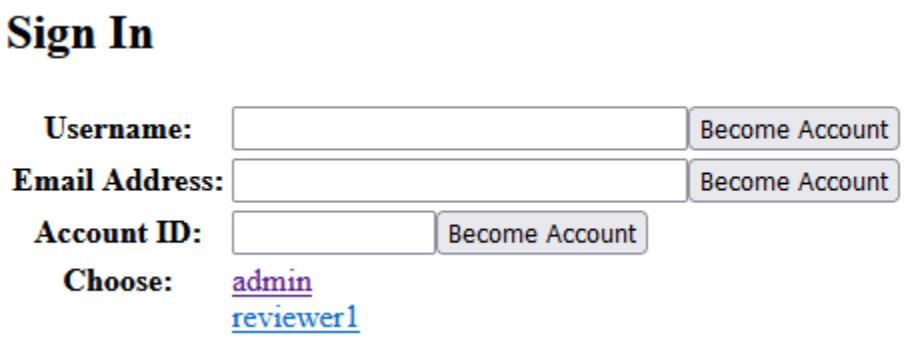
6.1. Logged in as admin, I reviewed the code (which is the single line seen below). I decided that it looks fine and reviewed it. I added a new reviewer (Reviewer1 from step 5) and gave it the +1 rating which suggests someone else must approve.



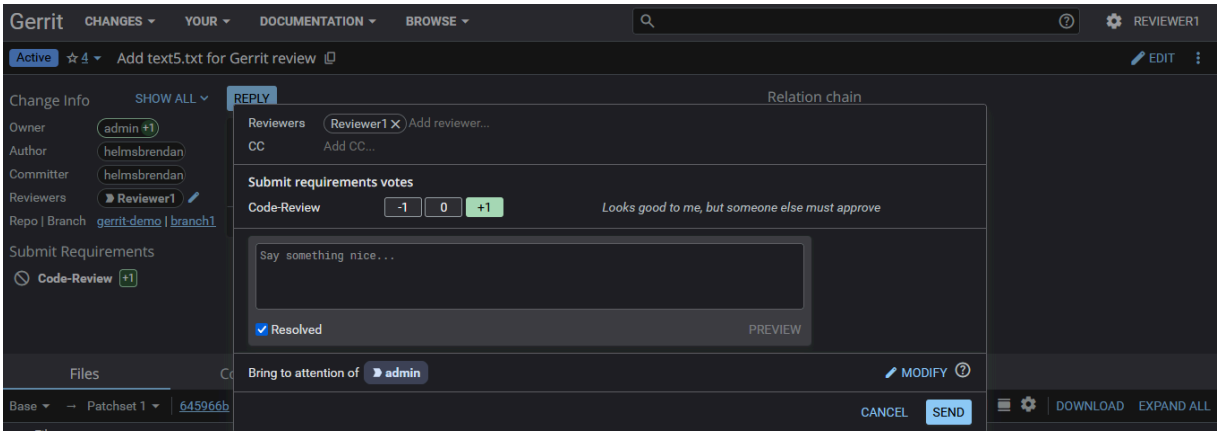


7. Step 7: Login as another reviewer and review the code.

7.1. I logged out of the admin account and clicked the ‘sign in’ button in the top right. There is now a second account available to login as. This is the reviewer1 account created in step 5. I clicked the reviewer1 to login.

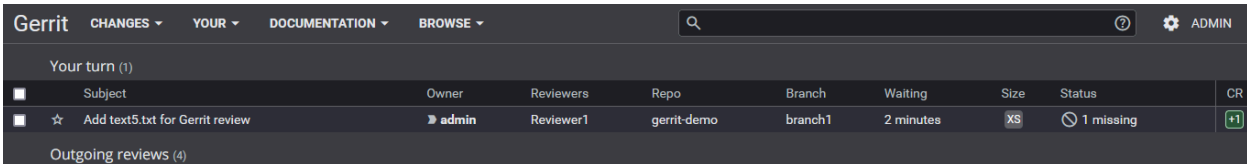


7.2. I went to the change that the admin user wanted me to review. I clicked the reply button and gave it a +1. Since, I am not the admin I can’t give it the official go ahead (+2). My review will notify the admin user that all is well and do officially accept the change.



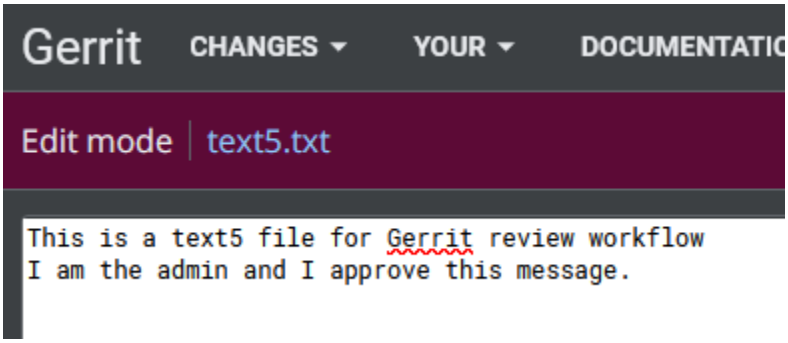
8. Step 8: Log in as admin again

8.1. I again logged in as an the administrator. Upon logging in, there is now a new tab on the Changes page called ‘Your Turn’. This is due to the Reviewer1 user notifying the admin to accept the change.

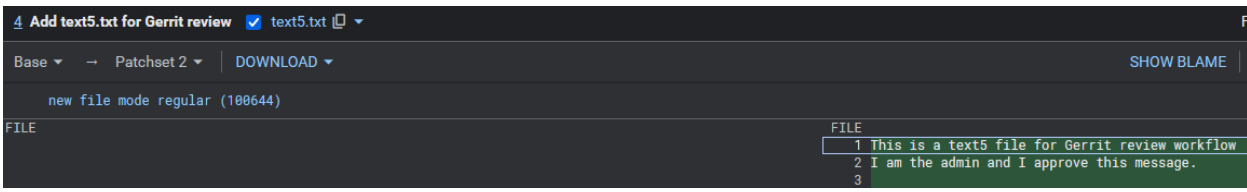


9. Step 9: Make a Small Change in the Text File and Make a Final Approval

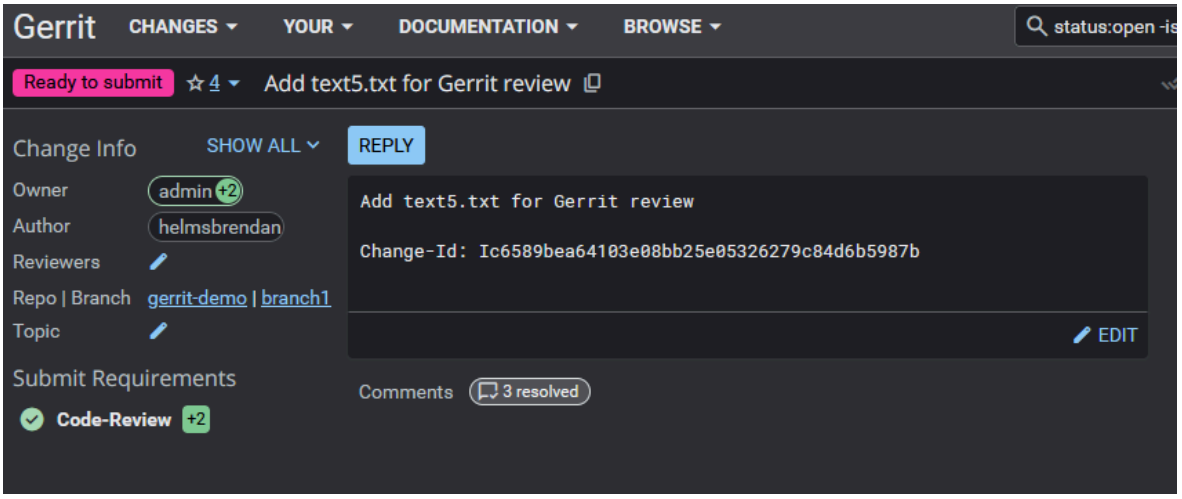
9.1. Logged in as admin, I opened the change again and decided to make a change. I opened the text5.txt directly from the interface and selected the edit option. I added the second line of code as seen below.



9.2. I clicked the save and publish button in the top right corner to save it as Patchset2 as seen below.



9.3. Finally, I gave it the final +2 approval to signify that it is ready to submit.



10. Step 10: In Local Repository, Fetch Patch Set 2

10.1. I used the command below to fetch the changes from the remote server. The git fetch gerritdemo refs/changes/XX/12345/2 command took me a long time to figure out. The XX is the last two digits of the change number. However, my change number is only one digit (4) so I had to enter '04' for it to fetch the change.

```
Brendan@LAPTOP-FO6FPRSL MINGW64 /c/gerrit/gerritCodeReview (branch1)
$ git fetch gerritdemo refs/changes/4/4/2
fatal: couldn't find remote ref refs/changes/4/4/2
```

```
Brendan@LAPTOP-FO6FPRSL MINGW64 /c/gerrit/gerritCodeReview (branch1)
$ git fetch gerritdemo refs/changes/04/4/2
remote: Counting objects: 4, done
remote: Finding sources: 100% (3/3)
remote: Total 3 (delta 0), reused 0 (delta 0)
Unpacking objects: 100% (3/3), 485 bytes | 60.00 KiB/s, done.
From http://localhost:8090/gerrit-demo
 * branch            refs/changes/04/4/2 -> FETCH_HEAD
```

11. Step 11: Merge Patch set 2 into the current branch

11.1. I then merged the set into the branch1. I did this beforehand already on the GUI so it gave the message below. I did redo the previous steps to get the correct output as seen in the second screenshot.

```
Brendan@LAPTOP-FO6FPRSL MINGW64 /c/gerrit/gerritCodeReview (branch1)
$ git fetch gerritdemo refs/changes/04/4/2
From http://localhost:8090/gerrit-demo
 * branch            refs/changes/04/4/2 -> FETCH_HEAD

Brendan@LAPTOP-FO6FPRSL MINGW64 /c/gerrit/gerritCodeReview (branch1)
$ git merge FETCH_HEAD
Already up to date.
```

```
Merge commit 'refs/changes/10/10/1' of http://localhost:8090/gerrit-demo into br
```

12. Step 12: Create a commit point

12.1. I finally created an empty checkpoint commit to mark the point in which the review was completed as seen below.

```
Brendan@LAPTOP-FO6FPRSL MINGW64 /c/gerrit/gerritCodeReview (branch1)
$ git commit --allow-empty -m "Checkpoint after Gerrit review workflow"
[branch1 01e5c58] Checkpoint after Gerrit review workflow
```