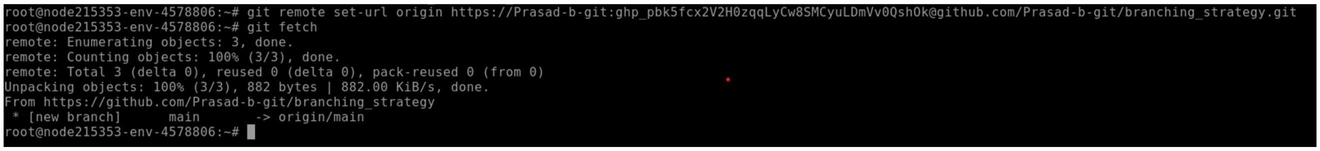
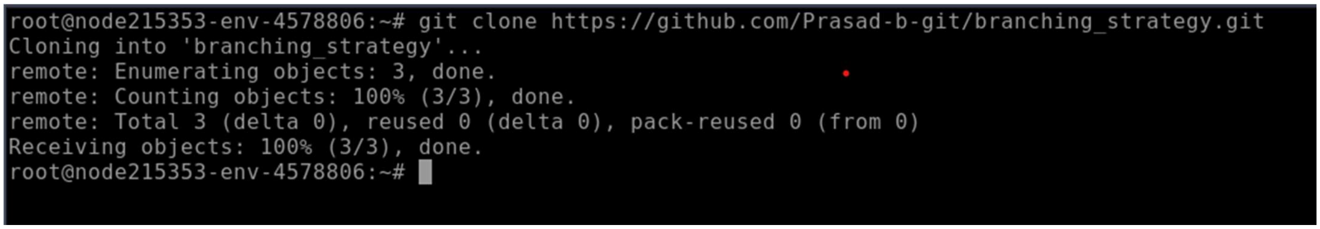
GIT
Branching Strategy & merge Conflict

Step 1: Login into GitHub Via Access Token



```
root@node215353-env-4578806:~# git remote set-url origin https://Prasad-b-git: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@github.com/Prasad-b-git/branching_strategy.git
root@node215353-env-4578806:~# git fetch
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (3/3), 882 bytes | 882.00 KiB/s, done.
From https://github.com/Prasad-b-git/branching_strategy
 * [new branch]      main       -> origin/main
root@node215353-env-4578806:~# 
```

Step 2: Clone the repo in the local system.



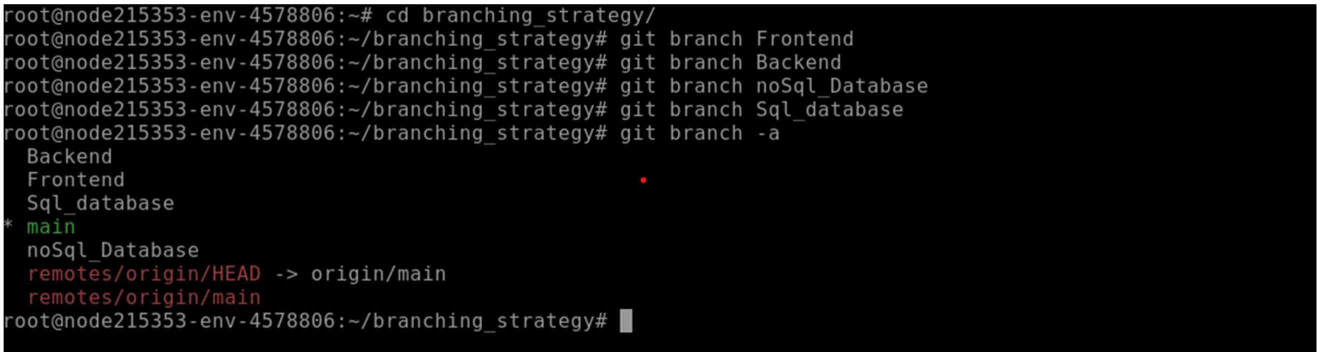
```
root@node215353-env-4578806:~# git clone https://github.com/Prasad-b-git/branching_strategy.git
Cloning into 'branching_strategy'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.
root@node215353-env-4578806:~# 
```

Step 3: To apply branching Strategy, Need to create branches for the repo.

* By use of branching strategy, Developers can collab with each other.

* It enable parallel development, where multiple features and bugs fixes can be worked simultaneously.

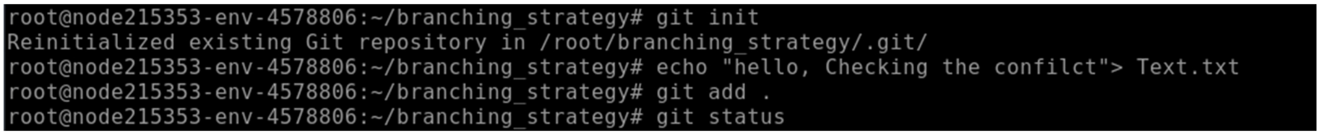
* Changes can be reviewed through pull requests or merge requests before being integrated into the main branch.



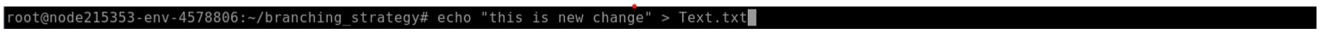
```
root@node215353-env-4578806:~# cd branching_strategy/
root@node215353-env-4578806:~/branching_strategy# git branch Frontend
root@node215353-env-4578806:~/branching_strategy# git branch Backend
root@node215353-env-4578806:~/branching_strategy# git branch noSql_Database
root@node215353-env-4578806:~/branching_strategy# git branch Sql_database
root@node215353-env-4578806:~/branching_strategy# git branch -a
  Backend
  Frontend
  Sql_database
* main
  noSql_Database
  remotes/origin/HEAD -> origin/main
  remotes/origin/main
root@node215353-env-4578806:~/branching_strategy# 
```

Step 4: Learning the merge and resolving it.

*Created the file with name text.txt with some content inside at both branches Frontend and Main.

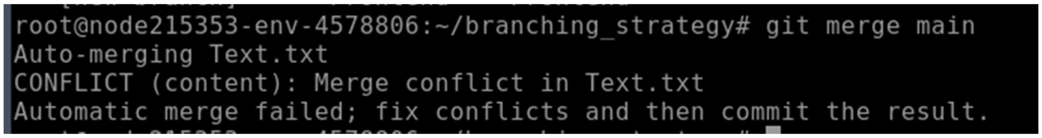


```
root@node215353-env-4578806:~/branching_strategy# git init
Reinitialized existing Git repository in /root/branching_strategy/.git/
root@node215353-env-4578806:~/branching_strategy# echo "hello, Checking the confilct"> Text.txt
root@node215353-env-4578806:~/branching_strategy# git add .
root@node215353-env-4578806:~/branching_strategy# git status
```



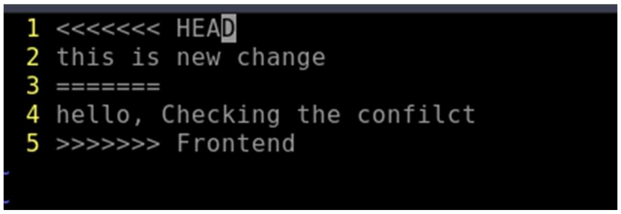
```
root@node215353-env-4578806:~/branching_strategy# echo "this is new change" > Text.txt
```

* Attempted to merge the main branch but encountered an error due to file naming issues



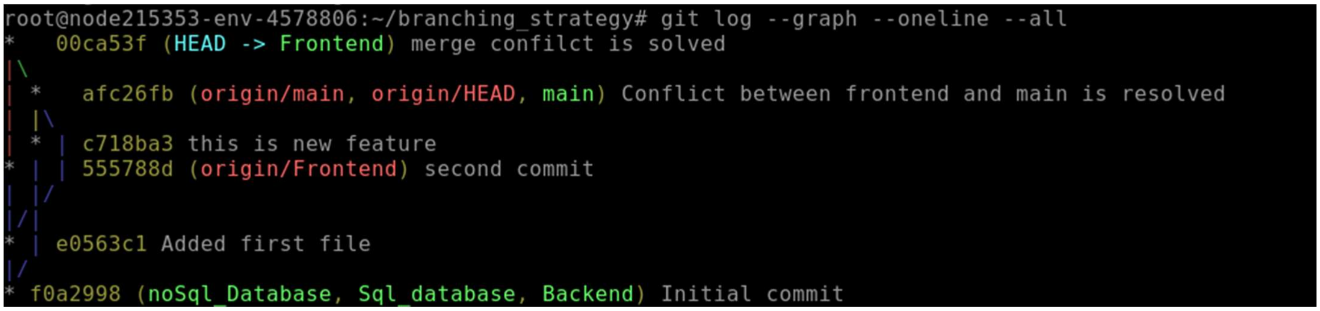
```
root@node215353-env-4578806:~/branching_strategy# git merge main
Auto-merging Text.txt
CONFLICT (content): Merge conflict in Text.txt
Automatic merge failed; fix conflicts and then commit the result.
```

* To resolve the merge issue, changes were made by editing the file.



```
<<<<<<< HEAD
this is new change
=======
hello, Checking the confilct
>>>>>>> Frontend
```

* The git conflict is solve.



```
root@node215353-env-4578806:~/branching_strategy# git log --graph --oneline --all
*   00ca53f (HEAD -> Frontend) merge confilct is solved
|\
| *   afc26fb (origin/main, origin/HEAD, main) Conflict between frontend and main is resolved
| |\
| * | c718ba3 this is new feature
* | | 555788d (origin/Frontend) second commit
| |/
|/|
* | e0563c1 Added first file
|/
* f0a2998 (noSql_Database, Sql_database, Backend) Initial commit
```