

GitHub Push and Commit Change Documentation

1. **Commit Changes:** To begin, I need to make modifications to the files that are stored in my local repository and then commit those changes. Make use of the commands listed below:

```
git add .                # Add all changes to the staging area
# Commit changes with a message
git commit -m "My commit message here"
```

2. **Push Changes to GitHub:** After I have committed I have changes locally, I will be able to push them to my GitHub repository by using the command that is shown below:

```
# Replace branch_name with the name of My branch
git push origin branch_name
```

3. **Change Commit:** To modify the commit message or modify the commit itself (for example, by adding more modifications), I have the option to use the `--amend` option in conjunction with the `git commit` command. By using this command, I am able to make changes to the most recent commit.

```
git commit --amend -m "New commit message"
```

If I want to amend the commit with additional changes:

```
git add .                # Add any additional changes
# Amend the commit with new changes
#without changing the commit message
git commit --amend --no-edit
```

4. **Force Push:** In the event that I have made changes to the commit and need to bring the remote branch up to date with the modifications, I will be required to force push since I will be rewriting the commit history. When force pushing, use caution since it has the potential to erase modifications made on the remote branch.

```
git push origin branch_name --force
```

Alternatively, I can use the shorthand for force pushing:

```
git push origin branch_name -f
```