Git Hook

Task

The Nautilus application development team was working on a git repository /opt/apps.git which is cloned under /usr/src/kodekloudrepos directory present on Storage server in Stratos DC. The team want to setup a hook on this repository, please find below more details:

- Merge the feature branch into the master branch`, but before pushing your changes complete below point.
- Create a post-update hook in this git repository so that whenever any changes are pushed to the master branch, it creates a release tag with name release-2023-06-15, where 2023-06-15 is supposed to be the current date. For example if today is 20th June, 2023 then the release tag must be release-2023-06-20. Make sure you test the hook at least once and create a release tag for today's release.
- Finally remember to push your changes.

Problem Understanding

- We are supposed to create a web hook called “post-update” in remote bare repository.
- “post-update” hook should be triggered after every push to master(remote) which will initiate a process of creating a tag
- The tag should consist of todays date, Ex- if today is 20th June, 2023 then the release tag must be release-2023-06-20.

Solution

1. SSH from jumphost to storage server
2.
   ```
   SSH natasha@172.16.238.15
   ```
3. Change to directory /opt/apps.git (your remote directory)
   ```
   cd /opt/apps.git
   ```
4. Enable appropriate user permissions to directory
   ```
   sudo chown -R natasha:natasha /opt/apps.git
   ```
5. Change to hooks directory
   ```
   cd /opt/apps.git/hooks
   ```
6. Create file called post-update
   ```
   vi post-update
   ```
7. Copy paste below content as code and save the file by pressing Esc > :wq! after you copy the below content
   ```
   #!/bin/bash
   current_date=$(date +"%Y-%m-%d")
   ```

```
release_name="release-$current_date"
exec git tag "$release_name"
```

8. Give Executable permission for this “post-update” file we created

```
chmod +x post-update
```

9. Change to local repository cloned @ /usr/src/kodekloudrepos

```
cd /usr/src/kodekloudrepos
```

10. Assign appropriate user permission to directory with below code and enter user(natasha) password(Bl@kW)

```
sudo chown -R natasha:natasha /usr/src/kodekloudrepos
```

11. Change to .git repo @ /usr/src/kodekloudrepos/blog

```
cd /usr/src/kodekloudrepos/blog #repleace blog with your cloned repo folder name
```

12. Add blog repo to git safe repos by executing command below #(OPTIONAL)

```
git config --global --add safe.directory @  #replace blog with your cloned repo folder name
```

13. Approve and assign commits made to user natasha #(OPTIONAL)

```
git config --global user.email "natasha@stratos.xfusioncorp.com"
git config --global user.name  "natasha"
```

14. Determine which branch you are in when you are @ /usr/src/kodekloudrepos/blog

```
git branch
```

15. Move to Master (Local) branch

```
git checkout master
```

16. Fetch changes from remote(master) to local (master) #(use any one ONLY of the following below comands

```
 git pull origin master
 git fetch origin master   # Downloads changes from remote master to origin/master
git merge origin/master   # Merges changes from origin/master into your local master
```

17. Merge updates which are saved in feature to master branch of you local cloned repo @/usr/src/kodekloudrepos/blog

```
git merge feature
```

18. Push merge to remote master

```
sudo git push origin master
```

19. Check to check if tags are created #(Use ONLY 1 of the following)

```
sudo git fetch --tags
sudo git tag
```