Auto-Deployment with Git Hooks

On Server Side

- Read about Git hooks here : https://git-scm.com/book/en/v2/Customizing-Git-Git-Hooks
- SSH to server you wish to deploy the code :
    - *ssh -i <location_of_identity_file> <username>@<hostname>*
- Initialize an empty project directory where the code will be pushed :
    - *mkdir your_project_name*
    - *Eg: mkdir* ***booholDevAutoDeploy***
- Initialize a bare repo on the server. Read more about bare repos here (Git Init Command)
    - Create a directory *repos* so that you can create multiple repos in future
        - *mkdir repos && cd repos*
    - Create a new directory and *cd* to it
        - *mkdir your_project.git && cd your_project.git*

      Example i created a **booholDev.git** directory for my project
    - Initialize a bare repo here
        - *git init --bare*
    - Run ***ls*** to see what is inside directory
    - You will see different directories and files.
    - Change to the hooks directory
        - *cd hooks*
    - We will be modifying the *post-receive* hook in this folder which will run our custom script whenever data is pushed in this repo. *post-receive* is appended with *.sample* filename. Rename it to *post-receive* to make it functional. Alternatively you can also create a new file named *post-receive (Skip this step if you don't see post-receive.sample file there)*
        - *mv post-receive.sample post-receive*
    - If you do not see post-receive.sample in the hooks folder, try to create a new one by following command, or if it is already there, try to edit it:
        - *sudo nano post-receive*
    - Add following lines:

```
#!/bin/sh
read oldrev newrev refname
echo “I love hooks”;
echo “Received a push request on branch: $refname”;
GIT_WORK_TREE=<path_to_your_project_directory>
export GIT_WORK_TREE
git checkout -f <branch_name_to_deploy>;
sudo npm install
sudo pm2 restart <process_id>;
echo “Finished, server has been restarted.”
```

- Sample :

```
#!/bin/sh
read oldrev newrev refname
echo “I love hooks”;
echo “Received a push request on branch: $refname”;
GIT_WORK_TREE=/home/ubuntu/booholDevAutoDeploy
export GIT_WORK_TREE
git checkout -f master;
sudo pm2 restart 0;
echo “Finished, server has been restarted.”
```

- Replace ***<path_to_your_project_directory>*** with the directory in which you are deploying your project.Replace ***<branch_name_to_deploy>*** with the name of your release branch(It’s **master** for most cases); Replace **<process_id>** with process id of your pm2 process

- If you want to update your code only when you push to a certain branch, place your code in a if else construct:

```
#!/bin/sh
echo “I love hooks”;
if(“$refname” == “master”)
then
echo “Received a push request on branch: $refname”;
GIT_WORK_TREE=/home/ubuntu/booholDevAutoDeploy
export GIT_WORK_TREE
git checkout -f master;
sudo pm2 restart 0;
echo “Finished, server has been restarted.”
else
echo “Not deployed”;
fi
```

- You can also run a custom script by writing its name in this file. Example :

  ```
  #!/bin/sh
  echo “Running script”;
  /location/to/your/script;
  ```

- Finally exit **nano editor** saving the changes.
- Change file perm on *post-receive* to make it executable:
  - *chmod +x post-receive*
- Now you are done on the server side

On Dev side

- Change to your git working directory containing your code and where git is initialized
- Add a new remote pointing to the bare repo you created on the server
  - *git remote add <remote_name> <protocol://user@hostname/path/to/your/repo>*
- Where remote name is any name you find suitable. In my case it is:
  - *git remote add ec2 ssh://ubuntu@eiya.mx:/home/ubuntu/repos/booholDev.git*
- Now whenever you push to above remote post-receive hook on the server is triggered and accordingly the script is run. Example :
  - git push ec2 <branch_name>
- You will also need to add ssh key of your local machine on the server. Generate a ssh key on your machine by following this link upto Step 3
- ssh to your server
  - *ssh -i <location_of_identity_file> <username>@<hostname>*
- Add the ssh key:
  - View existing keys added to your server
    - *cat ~/.ssh/authorized_keys*
  - Edit this file:
    - *sudo nano ~/.ssh/authorized_keys*
  - Add the key copied from your dev machine. Save and exit the file

PS: The directory structure on my ec2 server is as follows:

```
ubuntu@ip-172-31-0-117:~$ ls
booholDevAutoDeploy  node_modules  repos
```

```
ubuntu@ip-172-31-0-117:~$
ubuntu@ip-172-31-0-117:~$ cd repos/
ubuntu@ip-172-31-0-117:~/repos$ ls
booholDev.git
ubuntu@ip-172-31-0-117:~/repos$
```

Credits : Harekam Singh <harekamsingh@hotmail.com>