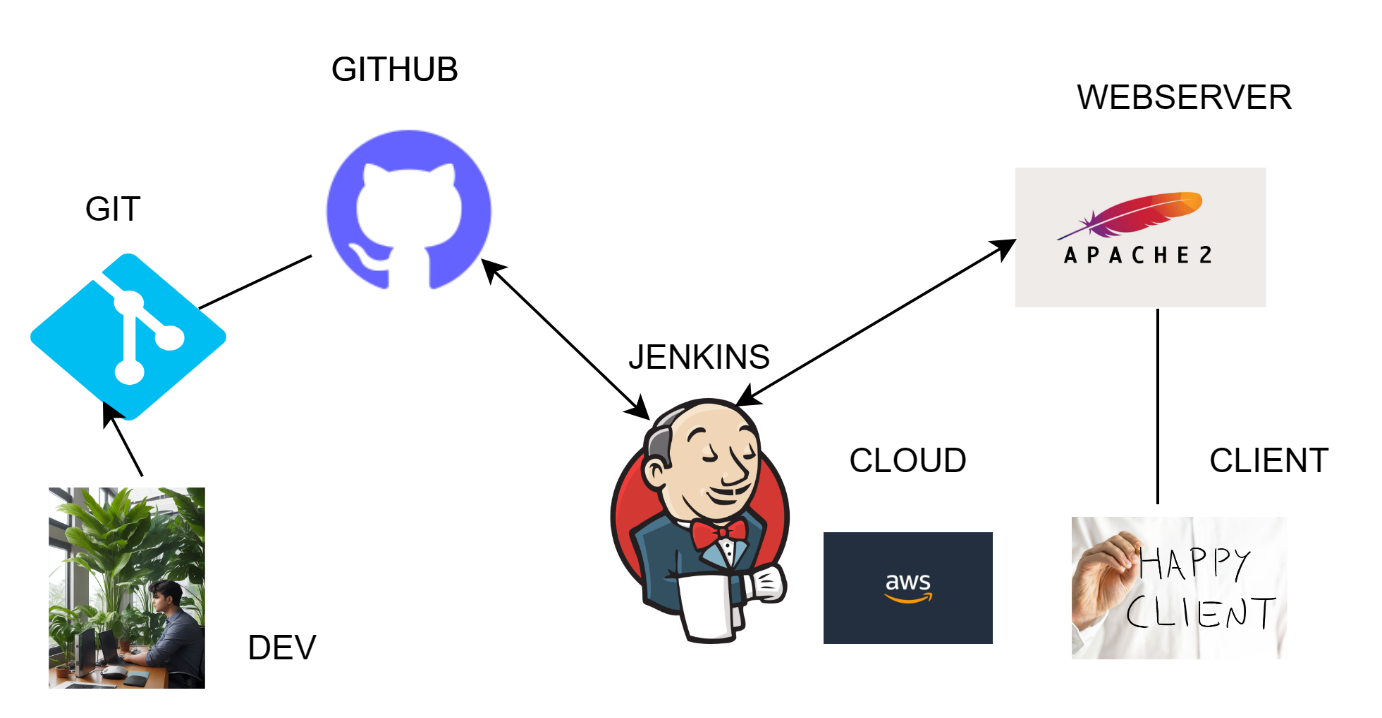
**PROJECT 1 :**

**CREATE A CICD PIPLINE AND DEPLOY A WEBSITE IN APACHE WEBSERVER HOSTED ON AWS USING JENKINS**



STEPS TO BE FOLLOWED .  
FOLLOWS TUTORIALS & GITHUB FOR PROJECT

* JENKINS INTRODUCTION <https://youtu.be/L8B2i_hLwVM?feature=shared>
* JENKINS INSTALATION ON EC2 UBUNTU OS <https://youtu.be/2OyLTG7jesQ?feature=shared>
* PROJECT WEBSITE REPOSITORY : [website1/Aseem-brainwave-html.zip at dev · Aseemakram19/website1 (github.com)](https://github.com/Aseemakram19/website1/blob/dev/Aseem-brainwave-html.zip)

LOGIN TO YOUR EC2 INSTANCE & INSTALL JENKINS , JENKINS PLUGINS , CREATE PIPILINE THROUGH GUI JENKINS <http://ip:8080>

**PREREQUESTION :**

1. INSTALL JENKINS <https://youtu.be/2OyLTG7jesQ?feature=shared>
2. ACCESS JENKINS , CONFIGURE GLOBAL CONFIG TOOLS, INSTALL JENKINS PLUIGN
3. CREATE JOB , FREE STYLE PROJECT
4. PROVIDE CREDENTIALS TO GITHUB INTEGRATION FOR PUBLIC/PRIVATE REPO

2. CREATING A PIPELINE FOR DEVELOPMENT BRANCH

Building in workspace /var/lib/jenkins/workspace/dev-telehealth-pipeline

Open mobaxterm & access severwith credentials.

cd /var/lib/jenkins/workspace/dev-telehealth-pipeline

Pipeline console failure

* ENABLING PERMISSION FOR DEPLOYMENT THROUGH CLI   
    
  Switch to root user

# sudo su –

Change to working direcotry

root@ip-10-0-1-115:/home/ubuntu# cd /var/www

root@ip-10-0-1-115:/var/www# ls -l

total 12

drwxr-xr-x 2 root root 4096 Oct 5 10:02 dev

drwxr-xr-x 2 root root 4096 Oct 5 09:10 html

drwxr-xr-x 2 root root 4096 Oct 5 10:04 qa

root@ip-10-0-1-115:/var/www# groupadd www-data

groupadd: group 'www-data' already exists

root@ip-10-0-1-115:/var/www# ls -la

total 20

drwxr-xr-x 5 root root 4096 Oct 5 09:18 .

drwxr-xr-x 14 root root 4096 Oct 5 09:10 ..

drwxr-xr-x 2 root root 4096 Oct 5 10:02 dev

drwxr-xr-x 2 root root 4096 Oct 5 09:10 html

drwxr-xr-x 2 root root 4096 Oct 5 10:04 qa

This can be useful when you want to grant a user specific permissions or access to resources associated with the "www-data" group, which is often used for web server-related tasks in Linux systems.

The command essentially adds the user "jenkins" to the "www-data" group without removing them from any other groups they might be a part of.

usermod: This is the command itself, used to modify user accounts.

-aG: These are options used with usermod:

-a: This option appends the user to the supplementary group(s) without removing them from their existing group(s).

-G: This option specifies the group(s) to which you want to add the user.

www-data: This is the name of the group you want to add the user "jenkins" to. In this case, you are adding the user "jenkins" to the "www-data" group.

jenkins: This is the username of the user you want to modify.

# root@ip-10-0-1-115:/var/www# usermod -aG www-data jenkins

# root@ip-10-0-1-115:/var/www# ls -la

total 20

drwxr-xr-x 5 root root 4096 Oct 5 09:18 .

drwxr-xr-x 14 root root 4096 Oct 5 09:10 ..

drwxr-xr-x 2 root root 4096 Oct 5 10:02 dev

drwxr-xr-x 2 root root 4096 Oct 5 09:10 html

drwxr-xr-x 2 root root 4096 Oct 5 10:04 qa

# root@ip-10-0-1-115:/var/www# sudo chown jenkins:www-data dev

The sudo chown jenkins:www-data html command is used to change the ownership of a directory or file to the user "jenkins" and the group "www-data

# root@ip-10-0-1-115:/var/www# sudo chown jenkins:www-data dev

# root@ip-10-0-1-115:/var/www# ls -la

total 20

drwxr-xr-x 5 root root 4096 Oct 5 09:18 .

drwxr-xr-x 14 root root 4096 Oct 5 09:10 ..

drwxr-xr-x 2 jenkins www-data 4096 Oct 5 10:02 dev

drwxr-xr-x 2 root root 4096 Oct 5 09:10 html

drwxr-xr-x 2 root root 4096 Oct 5 10:04 qa

check to switch the to jenkins user

root@ip-10-0-1-115:/var/www# su jenkins

jenkins@ip-10-0-1-115:/var/www$ cd dev

jenkins@ip-10-0-1-115:/var/www/dev$ ls

index.html

jenkins@ip-10-0-1-115:/var/www/dev$ cd ..

jenkins@ip-10-0-1-115:/var/www$ exit

exit

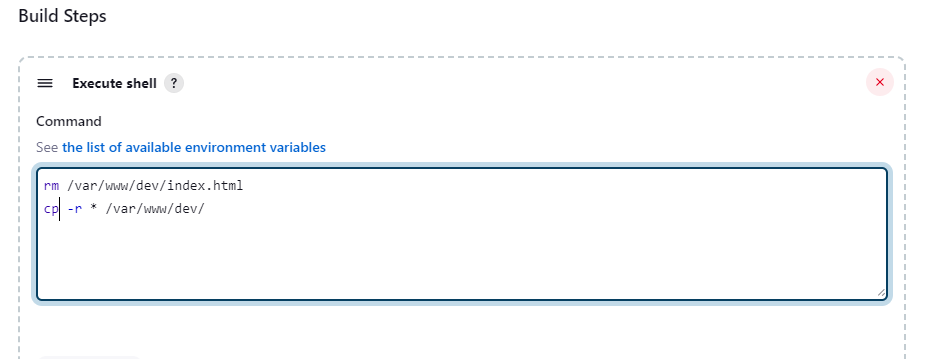
command "sudo chmod -R 2771 dev/" recursively changes the permissions of the "dev" directory and its subdirectories, setting the permissions to 2771. The '2' at the beginning indicates the setgid permission, which means newly created files and directories within "dev" will inherit the group ownership of the parent directory, ensuring consistent group ownership. The '771' represents read, write, and execute permissions for the owner of the directory and group, while others have read and execute permissions. This command is often used for shared directories where multiple users need access and collaborative permissions management.

root@ip-10-0-1-115:/var/www# sudo chmod -R 2771 dev/

* **Shell Commands**

rm /var/www/dev/index.html

cp -r \* /var/www/dev/



**Create a webhook for auto CICD**

Webhooks / Manage webhook

Settings

Recent Deliveries

We’ll send a POST request to the URL below with details of any subscribed events. You can also specify which data format you’d like to receive (JSON, x-www-form-urlencoded, etc). More information can be found in our developer documentation.

* Payload URL

<http://52.200.58.29:8080/github-webhook/>

* Content type

application/json

* Secret –

insert token issue for GitHub- Jenkins’s integration

If you've lost or forgotten this secret, you can change it, but be aware that any integrations using this secret will need to be updated. —

Which events would you like to trigger this webhook?

Just the push event.

Send me everything.

Let me select individual events.

Pull requests

Pull request assigned, auto merge disabled, auto merge enabled, closed, converted to draft, demilestoned, dequeued, edited, enqueued, labeled, locked, milestoned, opened, ready for review, reopened, review request removed, review requested, synchronized, unassigned, unlabeled, or unlocked.

Pushes

Git push to a repository.

unresolved.

