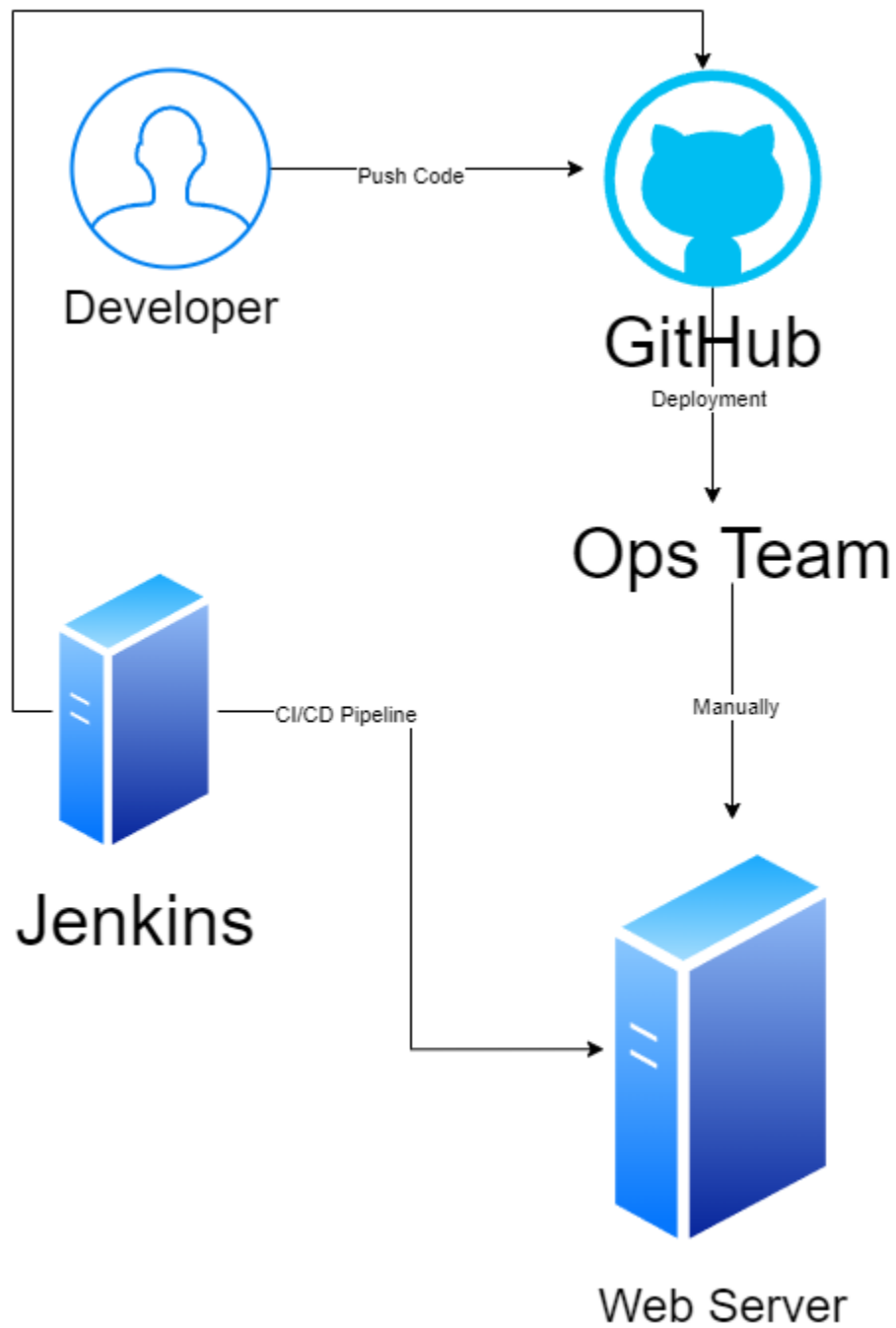
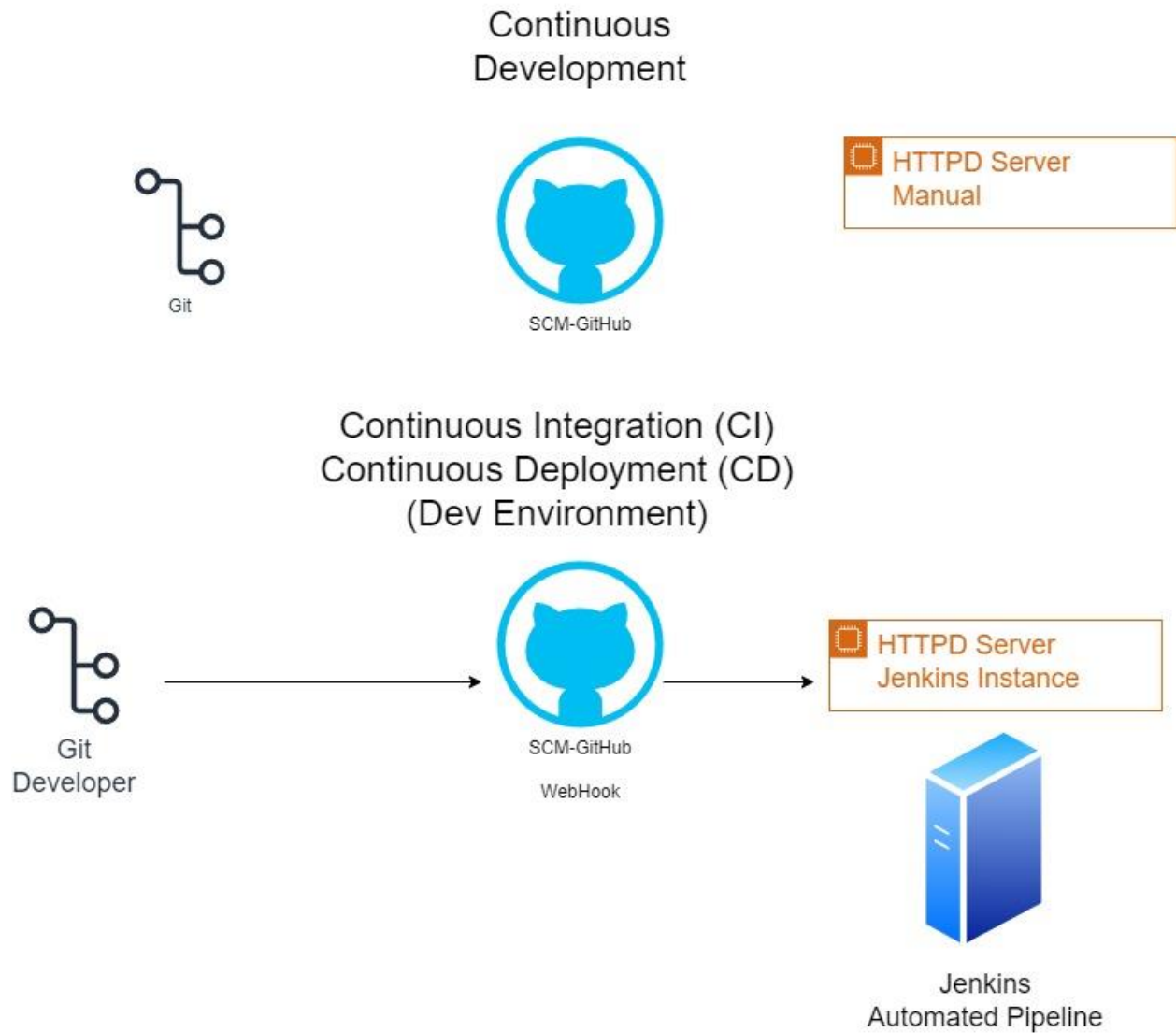


Create fully automated CI/CD Pipeline and deploy applications



Build a CI/CD Pipeline



Deploy a Amazon Linux Instance, Add http(80), https (443), 8080 in inbound port

Install HTTPD Server

```
yum update -y
```

```
yum install -y httpd
```

```
systemctl start httpd
```

```
systemctl enable httpd
```

Create a index.html file, add content, save it in /var/www/html/ folder

Check it by your public IP address

Go to Apache Server and add necessary permission

```
ls -la /var/www/html/ (only root has permissions)
```

```
ls -al /var/www/html/
```

```
sudo groupadd www-data
```

```
sudo newgrp www-data
```

```
sudo useradd Jenkins
```

```
passwd jenkins
```

```
sudo chown jenkins:www-data /var/www/html/
```

```
ls -al /var/www/html/
```

```
sudo chown jenkins:www-data -R /var/www/html/
```

```
ls -al /var/www/html/
```

```
sudo chmod 2771 /var/www/html/
```

```
sudo chown -R jenkins:www-data /var/www/html
```

```
sudo chmod -R 755 /var/www/html
```

```
ls -al /var/www/html/
```

Install git

yum install git -y

Configure Git

```
$ git config --global user.name "John Doe"
$ git config --global user.email johndoe@example.com
```

Generate SSH key in Linux Machine Add SSH Key in GitHub

ssh-keygen

cd /root/.ssh/

ls

eval "\$(ssh-agent -s)"

Copy Private Key: ssh-add ~/.ssh/id_rsa

cat ~/.ssh/id_rsa.pub

Copy the key and add this in GitHub

Clone the repo using SSH URL

Edit the Code and Push it to GitHub to test

GitHub – Create a Repo

Create a Repo

Add a file in master branch

Go to GitHub account and Create webhook

Go to the GitHub Repo

Select Settings, Select Webhooks

Click Add

Enter jenkinsurl:8080/github-webhook/ in Payload URL: (example: http://52.70.89.157:8080/github-webhook/)

Select content type application/json

Which events: let me select individual events

Pull and Pushes

Click Add

Add a file and GitHub and pipeline will start building

Install Jenkins

<https://www.jenkins.io/doc/tutorials/tutorial-for-installing-jenkins-on-AWS/>

```
sudo yum update -y
```

```
sudo wget -O /etc/yum.repos.d/jenkins.repo https://pkg.jenkins.io/redhat-stable/jenkins.repo
```

```
sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key
```

```
sudo yum upgrade -y
```

```
sudo dnf install java-17-amazon-corretto -y
```

```
sudo yum install jenkins -y
```

```
sudo systemctl enable jenkins
```

```
sudo systemctl start jenkins
```

```
sudo systemctl status jenkins
```

Go to ipaddress:8080

```
sudo cat /var/lib/jenkins/secrets/initialAdminPassword
```

Install default plug-ins

Enter password

Create Jenkins Pipeline

Click New Item1 > Enter name, select Freestyle project

scroll down, select git, copy and paste the repo URL

Enter main or master branch

Scroll down and select **GitHub hook trigger for GITScm polling**

Scroll down to Select Build Steps

Click Execute shell and add the command below

```
rm -rf /var/www/html/*
```

```
cp * -r /var/www/html
```

Scroll down to select apply /save

Testing:

Run the Pipeline by clicking on Build now

Edit code in local git, push code

git add .

git commit -m "added files"

git push origin master

VIM Editor

Press : (colon) to open the prompt bar in the bottom left corner of the window.

Type q! after the colon and hit Enter to exit without saving the changes

Jenkins

Source Code Management

☐ None

☒ Git ?

Repositories ?

Repository URL ?

https://github.com/monzureelahi/test20241.git

Credentials ?

- none -

+ Add ▾

Advanced ▾

Add Repository

Branches to build ?

Branch Specifier (blank for 'any') ?

*/master

Build Triggers

- ☐ Trigger builds remotely (e.g., from scripts) ?
- ☐ Build after other projects are built ?
- ☐ Build periodically ?
- ☒ GitHub hook trigger for GITScm polling ?
- ☐ Poll SCM ?

Build Environment

- ☐ Delete workspace before build starts
- ☐ Use secret text(s) or file(s) ?
- ☐ Add timestamps to the Console Output
- ☐ Inspect build log for published build scans
- ☐ Terminate a build if it's stuck
- ☐ With Ant ?

Build Steps

≡ Execute shell ?

Command

See [the list of available environment variables](#)

```
rm -rf /var/www/html/*
cp * -r /var/www/html
```

GitHub WebHook

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://52.90.237.101:8080/github-webhook/

Content type

application/json

Secret

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.