**Python-For-DevOps**

**Connect with GitHub using Python**

* To connect with GitHub using Python, you can use the GitHub API, which allows you to interact with repositories,  
  issues, pull requests, and more. One of the most commonly used libraries for this purpose is PyGithub.
* **Steps to Connect with GitHub Using Python:**
* Install PyGithub.
* Generate a Personal Access Token.
* Use PyGithub to Connect.

**Clone a GitHub repository using Python**

* To clone a GitHub repository using Python, you can use the git command through Python's **subprocess** module

**or**   
use the **GitPython** library, which provides an easier interface for interacting with Git repositories.

* **Method 1:** Using **subprocess** Module: This method runs the **git** command-line tool through Python.   
  You need to have git installed on your system for this to work.
* Import the subprocess module:  
  This module is used to spawn new processes, connect to their input/output/error pipes, and obtain their return codes.
* Run the git clone command:  
  Use **subprocess.run()** to execute the git clone command that clones the repository.

**Python Boto3**

* **Boto3** is the Amazon Web Services (AWS) Software Development Kit (SDK) for Python. It allows developers to write   
  software that makes use of services like Amazon S3, EC2, DynamoDB, and more.
* **Step 1: Install Boto3,** before using **Boto3**, you need to install it via pip:
* pip install boto3
* **Step 2: AWS Credentials Setup**
* **Boto3** uses your AWS credentials to connect to AWS services. To authenticate Boto3, you must provide the following:
* **Access Key ID**
* **Secret Access Key**
* **AWS Region**

**Launching an EC2 Instance using boto3.resource**

* Step-by-Step Process for Launching an EC2 Instance Using Boto3 Resource
* Install Boto3.
* Configure AWS Credentials
* Launch an EC2 Instance
* Additional Customization Options

**CI/CD Automation with Jenkins and Python**

* Create one AWS instance and setup **JAVA** AND **JENKINS**

**Installation of Java**

**sudo apt update**

**sudo apt install fontconfig openjdk-17-jre**

**java -version**

**Installation of Jenkins**

**sudo wget -O /usr/share/keyrings/jenkins-keyring.asc \**

**https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key**

**echo "deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc]" \**

**https://pkg.jenkins.io/debian-stable binary/ | sudo tee \**

**/etc/apt/sources.list.d/jenkins.list > /dev/null**

**sudo apt-get update**

**sudo apt-get install jenkins**

* Once the Jenkins setup then install required plugins like **GIT.**
* Create one git repository and add ec2instnace creation python script in file.

[**https://github.com/devops-catchup/Python-CICD/blob/master/ec2instanceboto3.py**](https://github.com/devops-catchup/Python-CICD/blob/master/ec2instanceboto3.py)

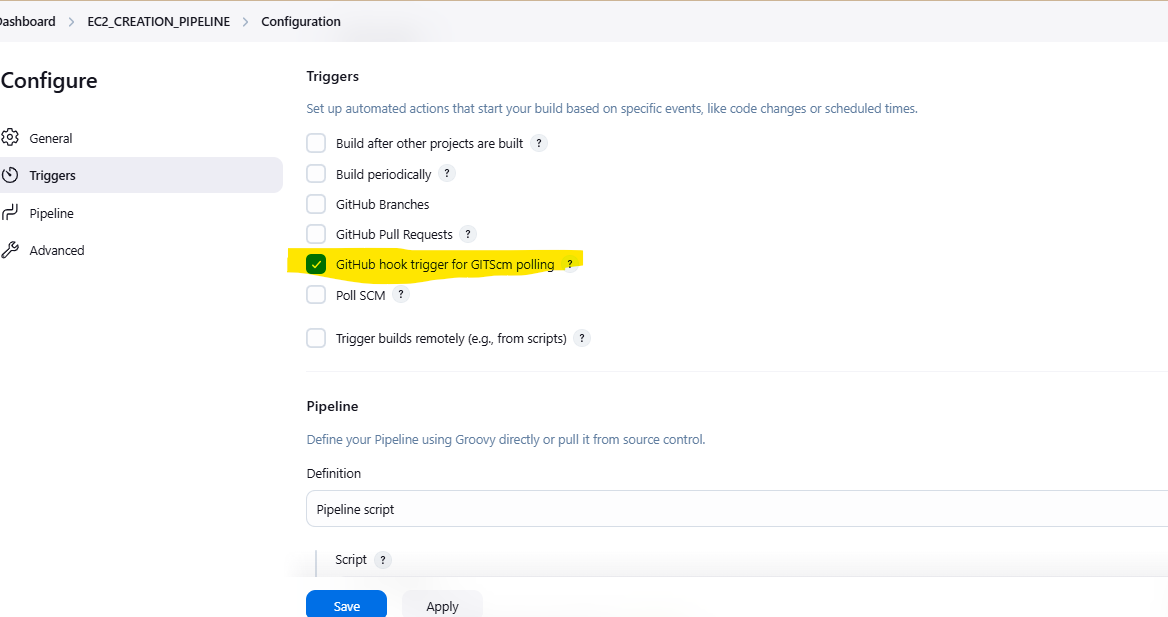
* Create a Basic Jenkins Pipeline job for above Python Projects also add access and secret key in Jenkins credentials. Below is the pipeline script.

[**https://github.com/devops-catchup/Python-CICD/blob/master/Jenkinsfile**](https://github.com/devops-catchup/Python-CICD/blob/master/Jenkinsfile)

* Deploying Python Code via Jenkins Jobs build the job and aws instance will be launch in aws account.

**Setup GitHub Webhooks in Jenkins**

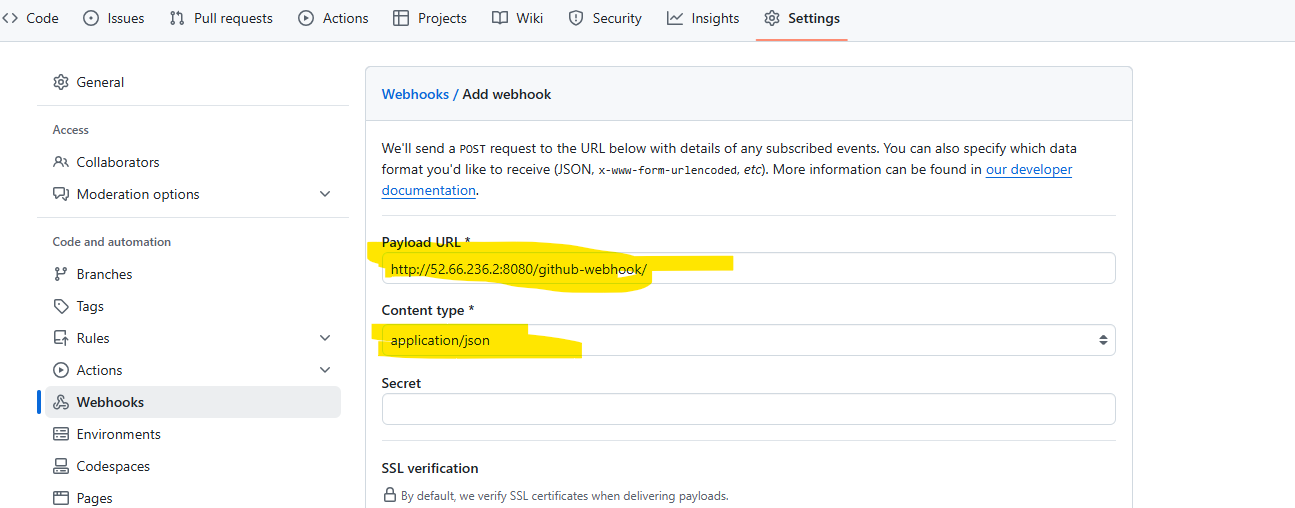
* Go to required Jenkins job and enable the below trigger options

****

Then apply and save, and copy the Jenkins url and add this (**github-webhook/**) after 8080/ like below

[**http://52.66.236.2:8080/github-webhook/**](http://52.66.236.2:8080/github-webhook/)

and add this url in required repository settings 🡪 webhooks 🡪 ADD Webhooks

****

And make changes in repository file and in Jenkins job will be trigger