**### Python App Hosting On Jenkins ###**

**# Setup of Jenkins in EC2 Instance :**

**Step-I : Install Jenkins On Ec2 instance (Linux) :**

- Select t2.small instance type while creating Instance.

sudo yum update

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

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

Jenkins is now installed and running on your EC2 instance. To configure Jenkins:

Connect to http://<your\_server\_public\_DNS>:8080 from your browser. You will be able to access Jenkins through its management interface:

**Use the following command to display password:**

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

**Step-II : Make changes in /lib/systemd/system/jenkins.service file** & **Resize /tmp :**

sudo nano /lib/systemd/system/jenkins.service

-Add this line at the end of [Service] :

Environment="JAVA\_OPTS=-Djava.io.tmpdir=/var/jenkins\_tmp"

-Add this line at the end of file :

WantedBy=multi-user.target

sudo mkdir -p /var/jenkins\_tmp

sudo chown jenkins:jenkins /var/jenkins\_tmp

sudo chmod 700 /var/jenkins\_tmp

1) Check current size of /tmp :

df -h /tmp

2) Resize it :

sudo mount -o remount,size=2G /tmp

3) Add this line in the /etc/fstab file and save it .

tmpfs /tmp tmpfs defaults,size=2G,noatime,nosuid 0 0

4) Remount /tmp to apply the changes:

sudo mount -o remount /tmp

5) Verify the change :

df -h /tmp

sudo rebbot

sudo systemctl restart Jenkins

**Step-III : Install Git**

sudo yum install git -y

git --version

# GitHub Repository Link : <https://github.com/Divyap8/pythonapp>

**Step-IV : Install Python3 & pip3 :**

sudo yum install python3

sudo yum install python3-pip

python3 --version

pip3 –version

**Step-V : Clone repository and install requirements.txt :**

git clone <https://github.com/Divyap8/pythonapp.git>

sudo chown -R jenkins:jenkins /home/ec2-user/pythonapp

sudo chmod -R 755 /home/ec2-user/pythonapp

ls -ld /home/ec2-user/pythonapp

cd pythonapp

python3 --version

pip3 –version

# Create a file /etc/systemd/system/flaskapp.service and add below content to that file and save it .

[Unit]

Description=Flask App

After=network.target

[Service]

User=ec2-user

WorkingDirectory=/home/ec2-user/pythonapp

ExecStart=/usr/bin/python3 /home/ec2-user/pythonapp/app.py

Restart=always

[Install]

WantedBy=multi-user.target

sudo systemctl daemon-reload

sudo systemctl enable flaskapp.service

sudo systemctl start flaskapp.service

sudo service jenkins start

cd pythonapp

python3 -m venv venv

source venv/bin/activate

pip3 install -r requirements.txt

python3 -m unittest discover -s test

**# On Jenkins Website :**

**Step-I : Plugins :**

Install Git Plugin on Jenkins and add it in Tools

**Step-II : Build Job :**

1. On the Jenkins Dashboard, click on "New Item".

Select "Freestyle project" and give it a name.

Click OK to proceed to the job configuration page**.**

1. In the job configuration page, go to the Source Code Management section.

Select Git.

Add your repository URL.

Specify a branch (e.g., main,master) in the Branch to build field**.**

1. Build Triggers : GitHub hook trigger for GITScm polling
2. Build Steps – Execute Shell :

cd $WORKSPACE

pip install -r requirements.txt

**Step-III : Test Job :**

1. On the Jenkins Dashboard, click on "New Item".

Select "Freestyle project" and give it a name.

Click OK to proceed to the job configuration page**.**

1. In the job configuration page, go to the Source Code Management section.

Select Git.

Add your repository URL.

Specify a branch (e.g., main,master) in the Branch to build field**.**

1. Build Triggers : Build after other projects are built[?](http://13.229.233.4:8080/job/pythontest/configure)

Projects to watch : Select your build job

Trigger only if build is stable

1. Build Steps – Execute Shell :

#!/bin/bash

cd /home/ec2-user/pythonapp

pip install -r requirements.txt --quiet

python3 -m unittest discover -s test

**Step-IV : Deploy Job :**

1. On the Jenkins Dashboard, click on "New Item".

Select "Freestyle project" and give it a name.

Click OK to proceed to the job configuration page**.**

1. In the job configuration page, go to the Source Code Management section.

Select Git.

Add your repository URL.

Specify a branch (e.g., main,master) in the Branch to build field**.**

1. Build Triggers : Build after other projects are built[?](http://13.229.233.4:8080/job/pythontest/configure)

Projects to watch : Select your test job

Trigger only if build is stable

1. Build Steps – Execute Shell :

#!/bin/bash

cd /home/ec2-user/pythonapp

source venv/bin/activate

pip install -r requirements.txt

1. Again Build Steps – Execute Shell :

#!/bin/bash

cd /home/ec2-user/pythonapp

git pull origin main

1. Again Build Steps – Execute Shell :

#!/bin/bash

pkill -f "python3 app.py"

nohup python3 app.py > flask.log 2>&1 &

**# Setup of Liveserver :**

sudo yum update -y

sudo yum install python3 python3-pip -y

pip3 install virtualenv

sudo dnf install nginx -y

sudo systemctl start nginx

sudo systemctl enable nginx

sudo systemctl status nginx

sudo dnf install git -y

git --version

git clone https://github.com/Divyap8/pythonapp.git

cd /home/ec2-user/pythonapp

virtualenv venv

source venv/bin/activate

pip install -r requirements.txt

python3 app.py

# Nginx proxy :

sudo nano /etc/nginx/nginx.conf

-Inside server block :

server\_name Liveserver PublicIP;

location / {

proxy\_pass http://127.0.0.1:5000;

proxy\_set\_header Host $host;

proxy\_set\_header X-Real-IP $remote\_addr;

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

proxy\_set\_header X-Forwarded-Proto $scheme;

}