Jenkins Setup and Pipeline Configuration Documentation

This document provides step-by-step instructions to set up Jenkins, configure the environment, and create a pipeline for converting DOCX files to PDF and sending an email with the converted file.

1. Install Jenkins

Step 1: Update the System

```
sudo apt update && sudo apt upgrade -y
```

Step 2: Install Java

Jenkins requires Java. Install the OpenJDK package:

```
sudo apt install openjdk-21-jdk -y
```

Verify Java installation:

```
java -version
```

Step 3: Add Jenkins Repository

```
curl -fsSL https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key | sudo
tee "/usr/share/keyrings/jenkins-keyring.asc" > /dev/null
```

```
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
```

Step 4: Install Jenkins

```
sudo apt update
sudo apt install jenkins -y
```

Step 5: Start and Enable Jenkins

```
sudo systemctl start jenkins
sudo systemctl enable jenkins
```

Step 6: Access Jenkins Web Interface

1. Open your browser and go to `http://<server-ip>:8080`.
2. Use the following command to get the initial admin password:

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

3. Install the recommended plugins and create an admin user.

---

2. Configure Jenkins Server

Step 1: Install Required Plugins

- Git plugin
- Pipeline plugin

Go to `Manage Jenkins` > `Manage Plugins` > `Available` and install the plugins.

Step 2: Set Up SSH Credentials

1. Go to `Manage Jenkins` > `Credentials` > `Global` > `Add Credentials`.
2. Add your SSH private key for GitHub access.

---

3. Configure the Environment

Step 1: Install System Dependencies

```
sudo apt install python3 python3-venv libreoffice -y
```

Step 2: Set Up the Virtual Environment

1. Create a directory for your pipeline:

   ```
   mkdir -p /var/lib/jenkins/pipeline
   ```

2. Create and activate the virtual environment:

```
python3 -m venv /var/lib/jenkins/venv
source /var/lib/jenkins/venv/bin/activate
```

3. Install Python dependencies:

```
pip install --upgrade pip
pip install yagmail
```

4. Deactivate the virtual environment:

```
deactivate
```

Step 3: Install LibraOffice

Install LibraOffice in your server.

4. Prepare the Scripts

1. `convert_docx_to_pdf.sh`

Save this script in `/var/lib/jenkins/pipeline/convert_docx_to_pdf.sh`:

```
#!/bin/bash

# Check if a file is provided
if [ -z "$1" ]; then
  echo "Usage: $0 <path-to-docx-file>"
  exit 1
fi

# Check if LibreOffice is installed
if ! command -v libreoffice &> /dev/null
then
    echo "LibreOffice could not be found, please install it first."
    exit 1
fi

# Convert the DOCX to PDF using LibreOffice
libreoffice --headless --convert-to pdf "$1"

if [ $? -eq 0 ]; then
  echo "Conversion successful: $(basename "$1" .docx).pdf"
else
  echo "Conversion failed."
```

```
    exit 1
fi
```

Make the script executable:

```
chmod +x /var/lib/jenkins/pipeline/convert_docx_to_pdf.sh
```

2. send_email.py

Save this script in /var/lib/jenkins/pipeline/send_email.py:

```
import sys
import yagmail

if len(sys.argv) != 2:
    print("Usage: python3 send_email.py <pdf-file>")
    sys.exit(1)

pdf_file = sys.argv[1]
recipient = "recipient@example.com"
sender = "your_email@example.com"
sender_password = "your_password"

try:
    yag = yagmail.SMTP(user=sender, password=sender_password)
    yag.send(to=recipient, subject="Converted PDF", contents="Find the
attached PDF.", attachments=pdf_file)
    print(f"Email sent successfully to {recipient}")
except Exception as e:
    print(f"Failed to send email: {e}")
    sys.exit(1)
```

---

5. Create the Jenkins Pipeline

Pipeline Script

Use the following pipeline script:

```
pipeline {
    agent any

    parameters {
        string(name: 'DOCX_FILE_NAME', defaultValue: 'example.docx',
description: 'Name of the DOCX file to convert')
    }
```

```
    environment {
        GIT_REPO = 'git@github.com:mawais003/private-repo.git'
        SSH_CREDENTIALS_ID = 'jenkins'
        TARGET_DIR = '/var/lib/jenkins/pipeline'
        VENV_PATH = '/var/lib/jenkins/venv'
    }

    stages {
        stage('Git Pull') {
            steps {
                checkout([
                    $class: 'GitSCM',
                    branches: [[name: '*/main']],
                    doGenerateSubmoduleConfigurations: false,
                    extensions: [],
                    submoduleCfg: [],
                    userRemoteConfigs: [[
                        url: env.GIT_REPO,
                        credentialsId: env.SSH_CREDENTIALS_ID
                    ]]
                ])
            }
        }

        stage('Copy DOCX File') {
            steps {
                sh """
                cp \${WORKSPACE}/\${DOCX_FILE_NAME} ${TARGET_DIR}
                """
            }
        }

        stage('Set up Virtual Environment') {
            steps {
                sh """
                if [ ! -f "${VENV_PATH}/bin/activate" ]; then
                    python3 -m venv ${VENV_PATH}
                fi
                """
            }
        }

        stage('Install Dependencies') {
            steps {
                sh """
                ${VENV_PATH}/bin/pip install --upgrade pip
                ${VENV_PATH}/bin/pip install yagmail
                """
            }
        }

        stage('Convert DOCX to PDF') {
            steps {
                sh """
```

```
                cd ${TARGET_DIR}
                bash ./convert_docx_to_pdf.sh \${DOCX_FILE_NAME}
                """
            }
        }

        stage('Send Email') {
            steps {
                script {
                    def pdfFileName = DOCX_FILE_NAME.replace('.docx',
'.pdf')
                    sh """
                    ${VENV_PATH}/bin/python
/var/lib/jenkins/pipeline/send_email.py ${TARGET_DIR}/${pdfFileName}
                    """
                }
            }
        }
    }
}
```

---

6. Test the Pipeline

1. Start a new Jenkins job.
2. Use the pipeline script provided.
3. Pass the required parameter (`DOCX_FILE_NAME`).
4. Run the job and ensure all steps complete successfully.