Extract\_data.py

import os

# Get the environment variable named "TOKEN"

token = os.getenv("TOKEN")

# Print the token value

print(f"Token = {token}")

# Check if the token matches the expected value

if token == "12345":

    print("valid")

else:

    print("invalid")

jenkinsfile

pipeline {

    agent any

    environment {

        PYTHON = "C:\\Users\\tanuj\\AppData\\Local\\Programs\\Python\\Python313\\python.exe"

        PIP = "C:\\Users\\tanuj\\AppData\\Local\\Programs\\Python\\Python313\\Scripts\\pip.exe"

        APP\_TOKEN = credentials('APP\_TOKEN')

    }

    stages {

        stage('Checkout Code') {

            steps {

                checkout scm

            }

        }

        stage('Install Dependencies') {

            steps {

                bat '"%PIP%" install -r requirements.txt'

            }

        }

        stage('Extract Data') {

            steps {

                bat """

                set TOKEN=${env.APP\_TOKEN}

                "${env.PYTHON}" extract\_data.py

                """

            }

        }

    }

    post {

        success {

            echo "✅ Pipeline completed successfully!"

        }

        failure {

            echo "❌ Pipeline failed. Check console output for details."

        }

    }

}

Requirements.txt

requests

pandas

**🧩 1. Create a Pipeline in Jenkins**

1. Open **Jenkins Dashboard → New Item**
2. Enter a name → Select **“Pipeline” → OK**
3. In **Pipeline** section → paste your Jenkinsfile code
4. Click **Save → Build Now**

**🔑 2. Create an App Token (Credential)**

1. Go to **Jenkins Dashboard → Manage Jenkins → Credentials → (Global)**
2. Click **Add Credentials**
3. Choose:
   * **Kind:** *Secret text*
   * **Secret:** *(paste your token value)*
   * **ID:** APP\_TOKEN
4. Click **OK**

**🧠 3. Use in Jenkinsfile**

environment {

APP\_TOKEN = credentials('APP\_TOKEN')

}

**✅ 4. Run Pipeline**

* Click **Build Now** → Jenkins will automatically fetch code, install dependencies, and run your Python script using the stored token securely.

Output:



