**📘 Workflow Documentation: SF Environment Setup**

**🧾 Overview**

The **SF Environment Setup** workflow automates setting up a Salesforce environment in CI/CD. It installs the Salesforce CLI, authenticates into a Salesforce org using JWT-based login, and runs an Apex script.

This is particularly useful for automating tasks such as pre-deployment setup, testing Apex code, or preparing metadata.

**⚙️ Trigger**

yaml

CopyEdit

on:

workflow\_dispatch:

inputs:

environment:

description: 'Salesforce org'

required: true

* This is a **manual trigger** (workflow\_dispatch).
* Requires a user to input the Salesforce org alias or username when manually launching the workflow.

**🧱 Job Summary**

yaml

CopyEdit

jobs:

sf-env-setup:

runs-on: ubuntu-latest

* Defines a job called sf-env-setup that runs on the latest Ubuntu runner.

**🧩 Steps Breakdown**

**1. Checkout Code**

yaml

CopyEdit

- name: Checkout code

uses: actions/checkout@v3

* Clones the repository code into the GitHub runner environment.

**2. Install Node.js**

yaml

CopyEdit

- name: Install Node.js

uses: actions/setup-node@v3

with:

node-version: "20.0.0"

* Installs Node.js 20, which is required to run the Salesforce CLI.

**3. Install Salesforce CLI**

yaml

CopyEdit

- name: Install Salesforce CLI

run: |

npm install -g @salesforce/cli@2.49.7

sf --version

* Installs a specific version of the Salesforce CLI globally via npm.
* Verifies the installation by printing the version.

**4. Install sfdx-git-delta Plugin**

yaml

CopyEdit

- name: Install SGD Delta Plugin

run: echo 'y' | sf plugins install sfdx-git-delta

* Installs the sfdx-git-delta plugin, which is useful for delta deployments.
* Uses echo 'y' to automatically accept prompts.

**5. Authenticate to Salesforce Using JWT**

yaml

CopyEdit

- name: Authenticate Salesforce Org

run: |

sf auth jwt grant \

-r https://test.salesforce.com \

-i ${{ secrets.SF\_CONSUMER\_KEY }} \

-f ${GITHUB\_WORKSPACE}/assets/devopskeys/server.key \

-o ${{ secrets.SF\_USERNAME }}

* Authenticates into a Salesforce org using **JWT-based OAuth**.
* Requires:
  + Connected App’s **Consumer Key** (SF\_CONSUMER\_KEY)
  + **Username** of the org (SF\_USERNAME)
  + Private key located at assets/devopskeys/server.key

**6. Execute Apex Script**

yaml

CopyEdit

- name: Execute Apex Script and Capture Output

id: execute\_apex

run: |

result=$(sf apex run --file ${GITHUB\_WORKSPACE}/scripts/test1.apex --target-org ${{ github.event.inputs.environment }})

echo "$result" > apex\_output.json

echo "Raw JSON Response:"

echo "$result"

* Executes the specified Apex script against the provided org alias or username.
* Captures the command output to apex\_output.json for later use.
* Echoes the raw output to the workflow logs for visibility.

**🔐 Required Secrets**

Ensure the following secrets are added to your GitHub repository:

| **Secret Name** | **Purpose** |
| --- | --- |
| SF\_CONSUMER\_KEY | Consumer key of your connected Salesforce app |
| SF\_USERNAME | Salesforce username (e.g., ci-user@example.com) |

**📁 Expected File Structure**

arduino

CopyEdit

.

├── .github/

│ └── workflows/

│ └── sf-env-setup.yml

├── assets/

│ └── devopskeys/

│ └── server.key

├── scripts/

│ └── test1.apex