**GitHub Actions Lab Guide**

**Lab 1: Workflow YAML Syntax**

**Objective**: Create a basic workflow.

1. Create a new repository in GitHub.
2. Create .github/workflows/ci.yml.
3. Add:

name: CI Demo

on: [push]

jobs:

build:

runs-on: ubuntu-latest

steps:

- uses: actions/checkout@v4

- name: Print Message

run: echo "Hello from GitHub Actions!"

1. Commit and push. Verify the workflow runs in **Actions** tab.

**Lab 2: Re-usable Jobs & Matrix**

**Objective**: Implement a matrix build.

1. In .github/workflows/matrix.yml, add:

name: Matrix Demo

on: [push]

jobs:

test:

runs-on: ${{ matrix.os }}

strategy:

matrix:

os: [ubuntu-latest, windows-latest]

python: [3.8, 3.9]

steps:

- uses: actions/checkout@v4

- uses: actions/setup-python@v5

with:

python-version: ${{ matrix.python }}

- run: python --version

1. Push changes and see parallel jobs execute.

**Lab 3: Cache pip Wheels**

**Objective**: Speed up Python installs.

1. In .github/workflows/cache.yml:

name: Cache Demo

on: [push]

jobs:

build:

runs-on: ubuntu-latest

steps:

- uses: actions/checkout@v4

- uses: actions/setup-python@v5

with:

python-version: '3.10'

- uses: actions/cache@v4

with:

path: ~/.cache/pip

key: ${{ runner.os }}-pip-${{ hashFiles('\*\*/requirements.txt') }}

- run: pip install -r requirements.txt

1. Push twice to observe faster second run.

**Lab 4: OIDC Azure Login**

**Objective**: Authenticate to Azure without secrets.

1. In Azure AD, create an app registration.
2. Configure federated credentials for your GitHub repository.
3. Add the following to .github/workflows/azure-login.yml:

name: Azure OIDC Demo

on: [workflow\_dispatch]

jobs:

login:

runs-on: ubuntu-latest

steps:

- uses: azure/login@v2

with:

client-id: ${{ secrets.AZURE\_CLIENT\_ID }}

tenant-id: ${{ secrets.AZURE\_TENANT\_ID }}

subscription-id: ${{ secrets.AZURE\_SUBSCRIPTION\_ID }}

federated-credentials: true

- run: az account show

1. Dispatch the workflow and verify Azure login.

**Lab 5: Branch Protection Rules**

**Objective**: Secure a branch.

1. Go to repository **Settings → Branches**.
2. Click **Add Rule**.
3. Select main branch.
4. Enable **Require pull request reviews** and **Require status checks**.
5. Test by pushing directly to main (should be blocked).

**Lab 6: Artefact Retention**

**Objective**: Upload and retrieve artefacts.

1. In .github/workflows/artifact.yml:

name: Artifact Demo

on: [push]

jobs:

build:

runs-on: ubuntu-latest

steps:

- run: mkdir output && echo "Hello" > output/hello.txt

- uses: actions/upload-artifact@v4

with:

name: demo-artifact

path: output

retention-days: 5

1. Push changes.
2. In **Actions**, download the uploaded artefact.