**GitHub Actions Study Guide**

**1. Workflow YAML Syntax**

* **Definition**: Workflows are automation definitions written in YAML files located in .github/workflows/.
* **Core Components**:
  + name: Identifies the workflow.
  + on: Specifies events that trigger the workflow (e.g., push, pull\_request, schedule).
  + jobs: Groups of steps executed in a defined environment.
  + runs-on: Specifies the OS/runner.
  + steps: Actions or shell commands to run.
* **Example Workflow**:

name: CI Pipeline

on: [push, pull\_request]

jobs:

build:

runs-on: ubuntu-latest

steps:

- uses: actions/checkout@v4

- uses: actions/setup-python@v5

with:

python-version: '3.10'

- run: pip install -r requirements.txt

**2. Re-usable Jobs & Matrix**

* **Re-usable Jobs**:
  + Define jobs in one workflow that can be called from others.
  + Useful for consistency across multiple repositories.

jobs:

test:

uses: org/repo/.github/workflows/test.yml@main

with:

python-version: '3.10'

* **Matrix Strategy**:
  + Run jobs in parallel with varying configurations.

strategy:

matrix:

python: [3.8, 3.9, 3.10]

os: [ubuntu-latest, windows-latest]

**3. Cache pip Wheels**

* **Purpose**: Speed up workflows by caching Python dependencies.
* **Implementation**:

- uses: actions/cache@v4

with:

path: ~/.cache/pip

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

restore-keys: |

${{ runner.os }}-pip-

* **Best Practice**: Use hashFiles on dependency files for cache invalidation.

**4. OIDC Azure Login**

* **Overview**: Authenticate with Azure without storing long-lived credentials.
* **Setup**:
  + Configure Azure to trust GitHub OIDC tokens.
  + Use azure/login action with federated credentials.

- 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

**5. Branch Protection Rules**

* **Purpose**: Maintain repository integrity.
* **Typical Rules**:
  + Require pull request review.
  + Enforce passing status checks.
  + Disallow force pushes and deletions.
* **Configuration**: Settings → Branches → Add Rule.

**6. Artefact Retention**

* **Purpose**: Store build/test artefacts for debugging or releases.
* **Example**:

- uses: actions/upload-artifact@v4

with:

name: build-output

path: ./dist

retention-days: 7

* **Default**: 90 days, can be reduced to save storage.

**7. Study Tips**

* Practice writing workflows from scratch.
* Experiment with matrix builds for multiple environments.
* Test cache effectiveness by running workflows twice.
* Configure a sandbox Azure account for OIDC login.
* Apply branch protection rules in a test repository.
* Upload and download artefacts to understand retention effects.