Welcome to Cert007 - Your Ultimate IT Certification Partner

- Real Exam Questions
- Free Updates
- Expert Support
- Instant Access
- Money-Back Guarantee

Visit us at **https://www.cert007.com/** for more information

Exam : **GH-200**

Title : GitHub Actions

Version : DEMO

1.Which scopes are available to define custom environment variables within a workflow file? (Choose three.)
A. the entire workflow, by using env at the top level of the workflow file
B. all jobs being run on a single Actions runner, by using runner.env at the top of the workflow file
C. the entire stage, by using env at the top of the defined build stage
D. within the run attribute of a job step
E. the contents of a job within a workflow, by using jobs.<job_id>.env
F. a specific step within a job, by using jobs.<job_id>.steps[*].env
Answer: A, D, F
Explanation:
You can define environment variables for the entire workflow by using the env key at the top level of the workflow file. These environment variables will be available to all jobs and steps within the workflow.
Environment variables can also be set within the run attribute of a job step, and these variables will be scoped only to that specific step.
You can set environment variables for specific steps within a job by using jobs.<job_id>.steps[*].env, which allows you to define variables that will only be available to that step.

2.Which choices represent best practices for publishing actions so that they can be consumed reliably? (Each correct answer presents a complete solution. Choose two.)
A. Organization name
B. Commit SHA
C. Repository name
D. Tag
E. Default branch
Answer: B, D
Explanation:
GitHub recommends versioning actions with stable references so consumers can deliberately select the code they want to execute. A full commit SHA is the most reliable and secure reference because it identifies one immutable Git commit; GitHub specifically identifies a full SHA as the safest option for stability and security. Tags are also a standard action-release mechanism. Action authors commonly create semantic-version tags such as v1.0.1 and maintain a major-version tag such as v1 so consumers can receive compatible fixes. By contrast, consuming an action directly from its default branch can introduce unexpected changes whenever that branch moves. An organization or repository name identifies where the action lives but does not by itself provide version stability. Therefore, commit SHA and tag are the two appropriate choices.

3.Your organization is managing secrets using GitHub encrypted secrets, including a secret named SuperSecret. As a developer, you need to create a version of that secret that contains a different value for use in a workflow that is scoped to a specific repository named MyRepo.
How should you store the secret to access your specific version within your workflow?
A. Create a duplicate entry for SuperSecret in the encrypted secret store and specify MyRepo as the scope.
B. Create MyRepo_SuperSecret in GitHub encrypted secrets to specify the scope to MyRepo.
C. Create a file with the SuperSecret. information in the .qithub/secrets folder in MyRepo.

D. Create and access SuperSecret from the secrets store in MyRepo.

Answer: B

Explanation:

To scope a secret to a specific repository, you can create a new secret with a name like MyRepo_SuperSecret in the secrets section of the MyRepo repository's settings. This ensures that the secret is specific to that repository and can be used within its workflows.

4.As a DevOps engineer, you are developing a container action. You need to execute a cleanup script after completing the main script execution.

Which code block should be used to define the cleanup script?

A)

```
runs:
  using: 'docker'
  image: 'Dockerfile'
  entrypoint: 'main.sh'
  post: 'cleanup.sh'
```

B)

```
runs:
  using: 'docker'
  image: 'Dockerfile'
  entrypoint: 'main.sh'
  post-if: 'cleanup.sh'
```

C)

```
runs:
  using: 'docker'
  image: 'Dockerfile'
  entrypoint: 'main.sh'
  after: 'cleanup.sh'
```

D)

```
runs:
  using: 'docker'
  image: 'Dockerfile'
  entrypoint: 'main.sh'
  post-entrypoint: 'cleanup.sh'
```

A. Option A

B. Option B

C. Option C

D. Option D

Answer: A

Explanation:

The correct syntax for specifying a cleanup script to be run after the main entry point is executed in a Docker-based GitHub Action is to use the post key. This ensures that cleanup.sh runs after the main script (main.sh) has completed.

5.Which of the following is a valid reusable workflow reference?
A. uses: another-repo/workflow.yml@v1
B. uses: octo-org/another-repo/.github/workflows/workflow.yml@v1
C. uses: octo-org/another-repo/workflow.yml@v1
D. uses: another-repo/.github/workflows/workflow.yml@v1
Answer: B
Explanation:
A reusable workflow stored in another repository must be referenced using the repository owner, repository name, required .github/workflows/directory, workflow filename, and a ref. GitHub documents the syntax as {owner}/{repo}/.github/workflows/{filename}@{ref}.
Option B exactly follows this structure: octo-org identifies the owner, another-repo identifies the repository, the workflow resides under .github/workflows, and v1 supplies the branch, tag, or other valid ref.
Options A and C omit the required .github/workflows path.
Option D includes that directory but omits the repository owner, so it is not a valid cross-repository reference. Reusable workflows are invoked with jobs.<job_id>.uses, rather than through an ordinary step-level action reference. Therefore, option B is the only correctly formed external reusable workflow reference.