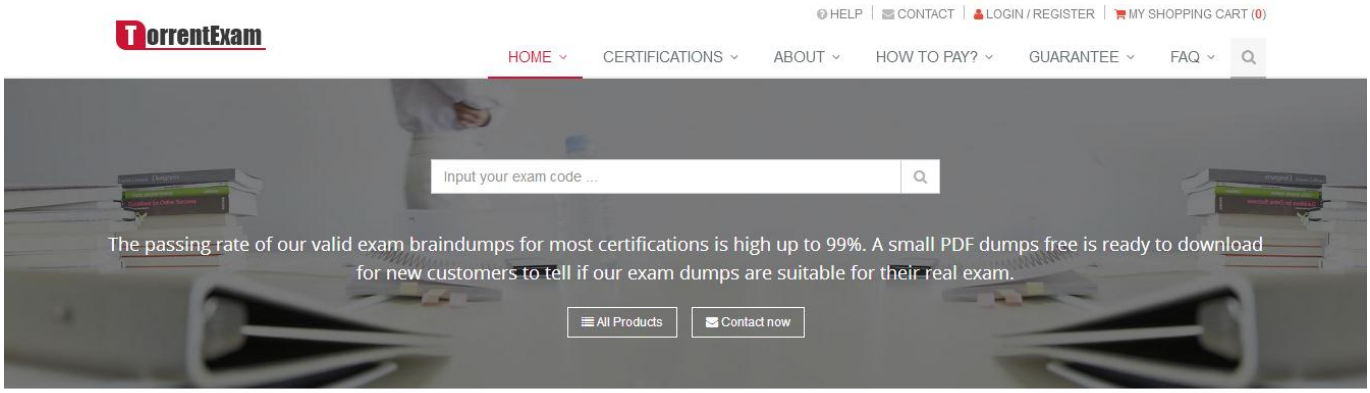
TorrentExam



Why Choose Us

QUALITY AND VALUE

RealExamFree Practice Exams are written to the highest standards of technical accuracy, using only certified subject matter experts and published authors for development - no all dumps.

TESTED AND APPROVED

We are committed to the process of vendor and third party approvals. We believe professionals and executives alike deserve the confidence of quality coverage these authorizations provide.

EASY TO PASS

If you prepare for the exams using our RealExamFree testing engine, It is easy to succeed for all certifications in the first attempt. You don't have to deal with all dumps or any free torrent / rapidshare all stuff.

TRY BEFORE BUY

RealExamFree offers free demo of each product. You can check out the interface, question quality and usability of our practice exams before you decide to buy.

Customer Reviews

I wasted a lot of money and failed twice. Thanks to HPE0-J78 exam collection I pass now.

Noel

realexamfree is a reliable company. I pass exam at first shot. Many thanks

Julie

Pass BIMF.EN successfully. Really good dumps. It saves me a lot of time. Wonderful!

Ahern

online test engine is very useful for me,because i could practice the C-TERP10-67 question dumps in my phone when i was waititng or on the bus even without internet,i could make the most of my time.Last week,i passed the C-TERP10-67. so i want to share the realexamfree with you guys,hope you will get a good result in test.

Carl

http://www.torrentexam.com

Best Exam Bootcamp & Excellent VCE Torrent & Satisfying Dumps Torrent

Exam : **GH-200**

Title : GitHub Actions

Vendor : Microsoft

Version : DEMO

NO.1 You need to create new workflows to deploy to an unfamiliar cloud provider. What is the fastest and safest way to begin?

A. Create a custom action to wrap the cloud provider's CLI.

B. Search GitHub Marketplace for verified actions published by the cloud provider.

C. Use the actions/jenkins-plugin action to utilize an existing Jenkins plugin for the cloud provider.

D. Search GitHub Marketplace for actions created by GitHub.

E. Download the CLI for the cloud provider and review the associated documentation.

Answer: B

Explanation:

Searching the GitHub Marketplace for verified actions published by the cloud provider is the quickest and safest approach. Many cloud providers offer verified GitHub Actions that are maintained and optimized to interact with their services. These actions typically come with the correct configurations and best practices, allowing you to get started quickly without reinventing the wheel.

Note: About GitHub Marketplace for apps

GitHub Marketplace where you can share your apps with everyone.

GitHub Marketplace connects you to developers who want to extend and improve their GitHub workflows. You can list free and paid tools for developers to use in GitHub Marketplace. GitHub Marketplace offers developers two types of tools: GitHub Actions and Apps, and each tool requires different steps for adding it to GitHub Marketplace.

GitHub Actions

Anyone can publish an action in GitHub Marketplace. GitHub verifies some partner organizations and these are shown as verified creators.

Reference:

https://docs.github.com/en/enterprise-cloud@latest/apps/github-marketplace/github-marketplace-overview/about-github-marketplace-for-apps

NO.2 For which type of GitHub Actions workflows should you use a self-hosted runner?

A. workflows that require access to private internal network resources

B. workflows that have no specific environment configurations and must be triggered on a GitHub-hosted infrastructure

C. short-lived workflows that do NOT require custom dependencies

D. workflows that require the latest operating system and preinstalled software versions

Answer: A

Explanation:

Self-hosted Runners: These run on your own infrastructure (on-premises or cloud). They are primarily used when you need specific environment configurations, such as custom hardware (GPUs), specific operating systems, or access to internal resources behind a firewall.

Using self-hosted runners is a standard solution for accessing private internal network resources, as they can be placed directly within your on-premises or cloud-based private network.

Incorrect:

[Not B]

For GitHub Actions workflows with no specific environment configurations that must be triggered on GitHub-hosted infrastructure, you should typically use GitHub-hosted runners, not self-hosted ones.

Reference:

https://github.blog/enterprise-software/ci-cd/when-to-choose-github-hosted-runners-or-self-

hosted-runners-with-github-actions

NO.3 Which workflow command would output the debug message "action successfully debugged"?
A. echo "::debug::action successfully debugged"
B. echo "::debug::message=action successfully debugged"
C. echo "debug=action successfully debugged"
D. echo ":debug:action successfully debugged:"
Answer: B
Explanation:
To output a specific debug message in GitHub Actions, use the ::debug:: workflow command.
Usage Syntax
You can issue this command by printing it to standard output (stdout) during a workflow step:
echo "::debug::{message}"
Reference:
https://docs.github.com/en/actions/reference/workflows-and-actions/workflow-commands

NO.4 In a workflow triggered by a pull request comment, which GitHub Actions context property contains the event payload used to access the comment text and the pull request number?
A. github.repository
B. github.event
C. github.event_path
D. github.job
Answer: B
Explanation:
In a workflow that runs on a pull request comment the event context contains the webhook payload. With github.event you can read the structured fields for the comment text and the pull request number. The comment text is available under the comment body field and the pull request number is available under the issue number for comment events or under the pull request number for pull request events. This makes github.event the direct and convenient source for these values.

NO.5 You are a DevOps engineer working on a custom action. You want to conditionally run a script at the start of the action, before the main entrypoint. Which code block should be used to define the metadata file for your custom action?
A. runs:
using: 'nodel6'
pre-if: github.event_name == 'push' then 'start.js'
main: 'index.js'
B. runs:
using: 'nodel6'
pre: 'start.js'
pre-if: github.event_name == 'push'
main: 'index.js'
C. runs:
using: 'nodel6'
start: 'start.js'

start-if: github.event_name == 'push'
main: 'index.js'

D. runs:
using: 'nodel6'
before: 'start.js'
before-if: github.event_name == 'push'
main: 'index.js'

Answer: B

Explanation:
The pre: line before the pre-if: line.
Note: runs.pre-if
Optional
Allows you to define conditions for the pre: action execution. The pre: action will only run if the conditions in pre-if are met. If not set, then pre-if defaults to always(). In pre-if, status check functions evaluate against the job's status, not the action's own status.
Note that the step context is unavailable, as no steps have run yet.
In this example, cleanup.js only runs on Linux-based runners:
pre: 'cleanup.js'
pre-if: runner.os == 'linux'
Reference:
https://docs.github.com/en/actions/reference/workflows-and-actions/metadata-syntax

NO.6 Your organization needs to simplify reusing and maintaining automation in your GitHub Enterprise Cloud. Which components can be directly reused across all repositories in an organization ?
(Each correct answer presents a complete solution. Choose three.)

A. actions stored in an organizational partition in the GitHub Marketplace

B. custom Docker actions stored in GitHub Container Registry

C. self-hosted runners

D. encrypted secrets

E. workflow templates

F. actions stored in private repositories in the organization

Answer: DEF

Explanation:
[D] Encrypted secrets can be accessed across repositories in the same organization, making it easy to store sensitive data (like API keys or tokens) securely while allowing multiple workflows to access them.
[E] Workflow templates allow you to create reusable templates for workflows that can be shared across repositories within the organization. This makes it easier to standardize processes and automate them across multiple projects.
[F] Actions stored in private repositories within the organization can be reused across all repositories by referencing them in workflows. This ensures a centralized way of maintaining custom actions.
Reference:
https://docs.github.com/en/actions/how-tos/reuse-automations/reuse-workflows

NO.7 As a developer, what two options should you recommend to implement standards for automation reuse? Each correct answer presents a complete solution. NOTE: Each correct answer is worth one point.

A. Create reusable actions and workflows that can be called from other workflows.

B. Create a marketplace partition to publish reusable automation for the company.

C. Store shared corporate actions in subfolders in a defined and documented internally accessible repository.

D. Create workflow templates and store them in the organization's .github repository.

Answer: AC

Explanation:

[A]

Implementing standards for reusable actions and workflows is the most effective way to reduce duplication and maintain consistency across your GitHub organization.

Reusable Workflows (workflow_call)

Best for standardizing entire processes (e.g., a complete CI/CD pipeline).

Location: Defined in .github/workflows/ of a central repository.

Trigger: Use the on: workflow_call trigger to define inputs, secrets, and outputs.

Usage: Called from another workflow using the uses keyword (e.g., owner/repo/.github/workflows/ci.yml@v1).

Benefit: They allow you to enforce security scans, build steps, and deployment gates across multiple repositories from a single source

[C]

Implementing a marketplace partition for internal GitHub Actions is a key strategy for scaling automation securely and efficiently across a company. This centralized approach ensures that developers use vetted, high-quality automation while adhering to corporate standards.

Strategic Implementation of Internal Reuse

To implement standards for automation reuse effectively, focus on these core components:

[C] Internal Actions Marketplace: Create a dedicated organization or specific repositories to host and display shared actions. This "marketplace" acts as a curated directory of approved automations, preventing the use of unverified third-party actions from the public marketplace.

Workflow Templates: Store standardized workflow templates in your organization's .github repository. These templates appear in the "Actions" tab when a user creates a new workflow, ensuring consistency across different teams.

[A] Reusable Workflows: Design modular workflows that can be called by other repositories using the uses keyword. Unlike standard actions, these can manage entire jobs and runners, making them ideal for standardizing complex CI/CD pipelines.

Internal Repository Sharing: Set repository visibility to Internal and configure Actions permissions to allow access from other repositories in the organization or enterprise.

Reference:

https://www.incredibuild.com/blog/best-practices-to-create-reusable-workflows-on-github-actions

https://xebia.com/blog/setting-up-an-internal-github-actions-marketplace

NO.8 Which of the following scenarios would require the use of self-hosted runners instead of GitHub- hosted runners?

A. running more than the three concurrent workflows supported by GitHub-hosted runners

B. performing builds on macOS
C. exceeding 50,000 monthly minutes of build time
D. using specialized hardware configurations required for workflows
E. using Docker containers as part of the workflow
Answer: D
Explanation:
The use of self-hosted runners is required when your workflows depend on specialized hardware configurations that are not available through standard GitHub-hosted runners. While GitHub offers "larger runners" with GPU support for certain paid plans, self-hosted runners provide the most flexibility for highly specific or proprietary hardware needs.
Scenarios Requiring Specialized Hardware
Specific GPU Models: Workflows for AI/ML training or intensive graphics rendering that require specific NVIDIA or other specialized GPU architectures.
Alternative CPU Architectures: When you must build or test on specific ARM processors, legacy x86 32-bit systems, or other non-standard architectures not supported by GitHub's managed pool.
High-Resource Requirements: Tasks needing massive amounts of RAM (beyond 256 GB) or high-core counts for extreme parallel processing.
Custom Peripherals: Workflows that need physical access to hardware connected via USB, PCIe, or other local interfaces (e.g., embedded systems testing or hardware-in-the-loop).
Reference:
https://docs.github.com/en/actions/how-tos/manage-runners/self-hosted-runners/use-in-a-workflow

NO.9 What are the two types of environment protection rules you can configure? Each correct answer presents a complete solution. NOTE: Each correct answer is worth one point.
A. artifact storage
B. wait timer
C. branch protections
D. required reviewers
Answer: BD
Explanation:
In GitHub Actions, you can configure four primary types of environment protection rules to control how and when deployments proceed to a specific environment.
These rules include:
[D] Required Reviewers: Specify up to six people or teams who must approve a deployment before it can proceed. Only one reviewer from the list needs to approve to unlock the deployment.
[B] Wait Timer: Set a mandatory delay (from 0 to 43,200 minutes) that must pass after a job is triggered before it can run.
Deployment Branches and Tags: Restrict deployments to only occur from specific branches or those matching certain tag patterns (e.g., main, release/*, or v*).
Custom Deployment Protection Rules: Enable third-party systems or automated tools (via GitHub Apps) to gate deployments based on external data, such as security scan results or ticket status.
Reference:
https://oneuptime.com/blog/post/2026-01-25-github-actions-environment-protection-rules/view

NO.10 A single secret must be accessed by workflows in specific repositories. What is the best way to create the secret?

A. Create an environment secret at the organization level and leverage that environment in each of the specified repositories.

B. Create an organization secret, specify Selected repositories as the Repository access, and select the required repositories.

C. Create the secret in one of the repositories, check the Share secret option, and select the required repositories.

D. Store the secret in a supported external key vault. Configure OpenID Connect (OIDC) to allow access to the external vault and link the secret from the external key vault in each of the specific repositories.

Answer: B

Explanation:

Creating secrets for an organization

When creating a secret or variable in an organization, you can use a policy to limit access by repository. For example, you can grant access to all repositories, or limit access to only private repositories or a specified list of repositories.

To specify that the secret should be available to selected repositories within the organization, use the --repos or -r flag.

gh secret set --org ORG_NAME SECRET_NAME --repos REPO-NAME-1, REPO-NAME-2 Note: REST API endpoints for GitHub Actions Secrets Use the REST API to interact with secrets in GitHub Actions.

* Set selected repositories for an organization secret

Replaces all repositories for an organization secret when the visibility for repository access is set to selected. The visibility is set when you Create or update an organization secret.

Request example

Put /orgs/{org}/actions/secrets/{secret_name}/repositories

Reference:

https://docs.github.com/en/actions/how-tos/write-workflows/choose-what-workflows-do/use-secrets

https://docs.github.com/en/rest/actions/secrets?apiVersion=2022-11-28#set-selected- repositories -for-an-organization-secret

NO.11 In GitHub Actions, how is the success or failure of a step determined by its exit code?

A. Exit codes are ignored and a maintainer sets the result manually

B. Zero exit code means success and any nonzero means failure

C. Success is controlled by continue-on-error rather than the exit code

D. Every exit code is a failure

Answer: B

Explanation:

GitHub Actions evaluates each step by the exit status of the process it runs. The runner marks a step successful when the command finishes with an exit status of 0. If the command returns any other code then the step is marked as failed and the job may stop depending on your workflow configuration.

NO.12 As a developer, you have configured an IP allow list on a GitHub organization. Which effects

does the IP allow list have on GitHub Actions? (Each answer presents a complete solution. Choose two.)

A. You can use standard GitHub-hosted runners since their IP addresses are automatically allowed.

B. You can use self-hosted runners with known IP addresses.

C. You must allow GitHub Actions's IP address ranges in order to use marketplace actions.

D. You can use GitHub-hosted larger runners since they can be configured with static IP addresses.

Answer: BD

Explanation:

Using GitHub Actions with an IP allow list. If you use an IP allow list and would also like to use GitHub Actions, you must use self-hosted runners or GitHub-hosted larger runners with static IP address ranges To allow your self-hosted or larger hosted runners to communicate with GitHub, add the IP address or IP address range of your runners to the IP allow list that you have configured for your enterprise.

Reference:

https://docs.github.com/en/enterprise-cloud@latest/organizations/keeping-your-organization-secure/managing-security-settings-for-your-organization/managing-allowed-ip-addresses-for- your -organization