

Microsoft.GH-200.v2026-08-17.q48

Exam Code:	GH-200
Exam Name:	GitHub Actions
Certification Provider:	Microsoft
Free Question Number:	48
Version:	v2026-08-17
# of views:	103
# of Questions views:	480
https://www.freecram.net/torrent/Microsoft.GH-200.v2026-08-17.q48.html	

NEW QUESTION: 1

You are reaching your organization's storage limit for GitHub artifacts and packages. What should you do to prevent the storage limit from being reached? (Choose two.)

- A. Delete artifacts from the repositories manually
- B. Disable branch protections in the repository.
- C. Use self-hosted runners for all workflow runs.
- D. Configure the artifact and log retention period.
- E. Configure the repo to use Git Large File Storage.

Answer: ([SHOW ANSWER](#))

Deleting artifacts from repositories manually will free up storage space. Artifacts are typically stored for a limited time by default, but manual cleanup can help manage space.

Configuring the artifact and log retention period allows you to control how long artifacts and logs are retained in your repository. By shortening the retention period, you can prevent unnecessary accumulation of data and manage storage more effectively.

NEW QUESTION: 2

Based on the YAML below, which two statements are correct? (Choose two.)

```

jobs:
  build:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v2
      - uses: actions/setup-node@v1
        with:
          node-version: 12
      - run: npm ci
      - run: npm test

  publish-npm:
    needs: build
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v2
      - uses: actions/setup-node@v1
        with:
          node-version: 12
      - run: npm ci
      - uses: JS-DevTools/npm-publish@v1
        with:
          token: ${ secrets.NPM_TOKEN }

```

- A. This workflow will publish a package to an npm registry.
- B. This workflow will publish a package to GitHub Packages.
- C. This workflow file is using a matrix strategy.
- D. The workflow job publish-npm will only run after the build job passes.

Answer: ([SHOW ANSWER](#))

The publish-npm job includes the JS-DevTools/npm-publish action, which is used to publish an npm package to an npm registry.

The publish-npm job has the needs: build directive, meaning it will only run after the build job successfully completes.

NEW QUESTION: 3

Which statement is true about using default environment variables?

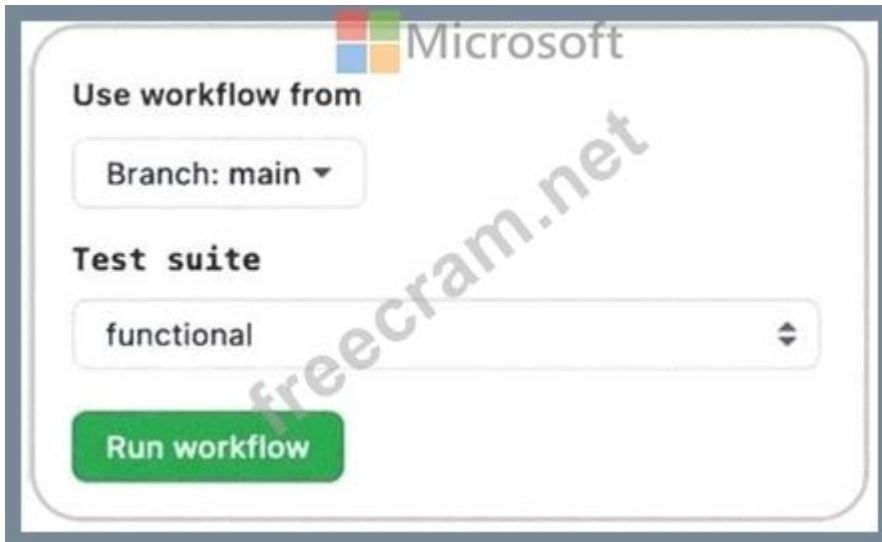
- A. The environment variables can be read in workflows using the ENV: variable_name syntax.
- B. The environment variables created should be prefixed with GITHUB_ to ensure they can be accessed in workflows
- C. The environment variables can be set in the defaults: sections of the workflow
- D. The GITHUB_WORKSPACE environment variable should be used to access files from within the runner.

Answer: D ([LEAVE A REPLY](#))

GITHUB_WORKSPACE is a default environment variable in GitHub Actions that points to the directory on the runner where your repository is checked out. This variable allows you to access files within your repository during the workflow.

NEW QUESTION: 4

As a developer, you want to run a workflow from the Actions tab in GitHub. Which YAML snippet should you use to match the interface in this image?

The image shows a GitHub Actions configuration interface. At the top, there's a Microsoft logo. Below it, the text 'Use workflow from' is followed by a dropdown menu showing 'Branch: main'. Underneath, the text 'Test suite' is followed by a dropdown menu showing 'functional'. At the bottom, there is a green button labeled 'Run workflow'.

```
on:
  workflow_dispatch:
    inputs:
      test_suite:
        description: Test suite
        type: choice
        options:
          - functional
          - regression
```

A.

```
on:
  workflow_run:
    inputs:
      test_suite:
        description: Test suite
        type: string
        options:
          - functional
          - regression
```

B.

```

on:
  workflow_dispatch:
    inputs:
      test_suite:
        description: Test suite
        type: choice
        value: functional
        options:
          - regression

```

C.

```

on:
  workflow_run:
    inputs:
      test_suite:
        description: Test suite
        type: choice
        options:
          - functional
          - regression

```

D.

Answer: ([SHOW ANSWER](#))

The first image shows a workflow trigger with an option for the test suite, and the chosen YAML configuration matches this interface. Specifically, the test suite input is defined with `type: choice` and includes the option `value: functional`, which aligns with the visible UI elements in the first image.

NEW QUESTION: 5

You need to perform a task that requires running a custom script in a GitHub Actions workflow. The script must run within the workflow environment without requiring an external container or prebuilt image.

What should you use?

- A. a JavaScript action
- B. a Docker container action
- C. a run step
- D. a composite action

Answer: ([SHOW ANSWER](#))

The simplest correct choice is a run step. A run step executes shell commands or scripts directly on the runner inside the current workflow job environment. It does not require packaging a JavaScript action, building a Docker container, publishing an image, or creating reusable action metadata. A JavaScript action is appropriate when you are building a reusable Node.js-based action. A Docker container action is useful when the execution environment must be packaged with dependencies, but it adds container overhead. A composite action is useful for reusable sequences of steps, but the question asks only for running a custom script inside the workflow

environment. GitHub workflow syntax shows run steps as the native way to execute commands inside a job.

NEW QUESTION: 6

As a DevOps engineer, you are trying to leverage an organization secret in a repo. The value received in the workflow is not the same as that set in the secret. What is the most likely reason for the difference?

- A. There is a different value specified at the repo level.
- B. There is a different value specified at the workflow level.
- C. The Codespace secret doesn't match the expected value.
- D. The Encrypt Secret setting was not configured for the secret.
- E. There is a different value specified at the enterprise level.

Answer: ([SHOW ANSWER](#))

GitHub secrets are defined at different levels: organization, repository, and sometimes at the workflow level.

If a secret is defined at both the organization level and the repository level, the repository-level secret will take precedence. So, if the value of the secret differs between these levels, the workflow will use the value from the repository level instead of the organization level.

NEW QUESTION: 7

You need to make a script to retrieve workflow run logs via the API. Which is the correct API to download a workflow run log?

- A. POST /repos/:owner/:repo/actions/runs/:run_id
- B. GET /repos/:owner/:repo/actions/artifacts/logs
- C. GET /repos/:owner/:repo/actions/runs/:run_id/logs
- D. POST /repos/:owner/:repo/actions/runs/:run_id/logs

Answer: ([SHOW ANSWER](#))

The GET /repos/:owner/:repo/actions/runs/:run_id/logs API endpoint is used to retrieve the logs of a specific workflow run identified by run_id. This is the correct method for downloading logs from a workflow run.

NEW QUESTION: 8

Which action type should be used to bundle a series of run steps into a reusable custom action?

- A. Composite action
- B. Bash script action
- C. Docker container action
- D. JavaScript action

Answer: ([SHOW ANSWER](#))

A composite action allows you to bundle multiple steps into a single reusable action within a workflow. It is composed of multiple run steps or other actions and can be reused across workflows, making it the perfect choice for bundling a series of steps.

NEW QUESTION: 9

You are a developer working on developing reusable workflows for your organization. What keyword should be included as part of the reusable workflow event triggers?

- A. check_run
- B. workflow_run
- C. workflow_call
- D. pull_request

Answer: (SHOW ANSWER)

The workflow_call event is used to trigger a reusable workflow from another workflow. This allows you to create workflows that can be reused in multiple places within your organization, enabling better modularity and reducing duplication.

NEW QUESTION: 10

Disabling a workflow allows you to stop a workflow from being triggered without having to delete the file from the repo. In which scenarios would temporarily disabling a workflow be most useful? (Choose two.)

- A. A workflow sends requests to a service that is down.
- B. A workflow error produces too many, or wrong, requests, impacting external services negatively.
- C. A workflow is configured to run on self-hosted runners
- D. A workflow needs to be changed from running on a schedule to a manual trigger
- E. A runner needs to have diagnostic logging enabled.

Answer: (SHOW ANSWER)

If a workflow depends on an external service that is down, disabling the workflow temporarily will prevent it from running and sending requests to the service, thus avoiding failed requests or unnecessary retries.

If a workflow is causing a negative impact on external services by generating too many requests or incorrect data due to a bug, temporarily disabling the workflow will stop this behavior while the issue is fixed.

NEW QUESTION: 11

GitHub-hosted runners support which capabilities? (Choose two.)

- A. automatic patching of both the runner and the underlying OS
- B. automatic file-system caching between workflow runs
- C. support for Linux, Windows, and mac
- D. support for a variety of Linux variations including CentOS, Fedora, and Debian
- E. requiring a payment mechanism (e.g., credit card) to use for private repositories

Answer: (SHOW ANSWER)

GitHub-hosted runners automatically handle patching, meaning they will be kept up to date with the latest security updates and software patches for both the runner environment and the underlying operating system.

GitHub-hosted runners support Linux, Windows, and macOS, giving you flexibility to run workflows on different operating systems without needing to manage your own self-hosted runners.

NEW QUESTION: 12

How can GitHub Actions encrypted secrets be used in if: conditionals within a workflow job?

- A. Set the encrypted secret as a job-level environment variable and then reference the environment variable within the conditional statement.
- B. Create a job dependency that exposes the encrypted secret as a job output, which can then be leveraged in a subsequent dependent job.
- C. Use the secrets context within the conditional statement, e.g. `${{ secrets.MySuperSecret }}`.
- D. Use a workflow command to expose the encrypted secret via a step's output parameter and then use the step output in the job's if: conditional.

Answer: C ([LEAVE A REPLY](#))

GitHub Actions encrypted secrets can be accessed in workflows using the secrets context. You can directly reference the secret within an if: conditional using `${{ secrets.MySuperSecret }}` to determine whether a job or step should run based on the secret's value.

NEW QUESTION: 13

You need to trigger a workflow using the GitHub API for activity that happens outside of GitHub. Which workflow event do you use?

- A. `check_suite`
- B. `workflow_run`
- C. `deployment`
- D. `repository_dispatch`

Answer: D ([LEAVE A REPLY](#))

The `repository_dispatch` event allows you to trigger a workflow in response to external activity. It is commonly used when you need to trigger a workflow from outside GitHub, such as from another system or service, by sending a request to the GitHub API. This event provides flexibility to integrate with various external systems and trigger workflows in a GitHub repository.

NEW QUESTION: 14

How should you install the bats NPM package in your workflow?

```
jobs:
  example-job:
    steps:
      - npm install -g bats
```

A.

```
jobs:
  steps:
    - run: npm install -g bats
```

B.

```
jobs:
  runs-on: ubuntu-latest
  - run: npm install -g bats
```

C.

```
jobs:
  example-job:
    runs-on: ubuntu-latest
    steps:
      - run: npm install -g bats
```

D.

Answer: D ([LEAVE A REPLY](#))

The correct syntax includes specifying the job (example-job), the runner (ubuntu-latest), and the necessary step (npm install -g bats) within the workflow. This ensures that the package is installed properly during the execution of the job.

NEW QUESTION: 15

In which of the following scenarios should you use self-hosted runners? Each correct answer presents a complete solution.

NOTE: Each correct answer is worth one point.

- A. when the workflow jobs must be run on Windows 10
- B. when jobs must run for longer than 6 hours
- C. when you want to use macOS runners
- D. when GitHub Actions minutes must be used for the workflow runs
- E. when a workflow job needs to install software from the local network

Answer: ([SHOW ANSWER](#)**)**

Self-hosted runners are appropriate when GitHub-hosted runner limits or environment restrictions do not meet the workload requirement. GitHub-hosted runner jobs have a 6-hour job execution limit, while self-hosted runner jobs can run for up to 5 days, so jobs that must run longer than 6 hours require self-hosted runners. Self-hosted runners also let an organization control the machine, operating system, software, hardware, and network location. Therefore, if a workflow job needs to install or access software from the local network, a self-hosted runner is the correct choice. macOS and Windows runners are available as GitHub-hosted options, and GitHub Actions minutes apply to GitHub-hosted usage, not self-hosted runner execution.

NEW QUESTION: 16

What will the output be for the following event trigger block in a workflow?

```
on:
  issues:
    types: [opened, edited]
  issue_comment:
    types:
      - created
```

- A. It throws a workflow syntax error, pointing to the types definition in issue_comment event.
- B. It throws a workflow syntax error, pointing to the types definition in issues event.

- C. It runs the workflow when an issue is edited or when an issue comment created.
- D. It runs the workflow when an issue or issue comment in the workflow ' s repository is created or modified.
- E. It runs the workflow when an issue is created or edited, or when an issue or pull request comment is created.

Answer: (SHOW ANSWER)

The provided event trigger block specifies two types of events:

For issues: the workflow triggers on opened or edited issues.

For issue_comment: the workflow triggers when an issue comment is created.

This configuration ensures the workflow will run when either an issue is opened or edited, or an issue comment is created.

Valid GH-200 Dumps shared by EduDump.com for Helping Passing GH-200 Exam!

EduDump.com now offer the **newest GH-200 exam dumps**, the EduDump.com GH-200 exam **questions have been updated** and **answers have been corrected** get the **newest** EduDump.com GH-200 dumps with Test Engine here:

<https://www.edudump.com/exams/Microsoft/GH-200/premium/> (**102 Q&As Dumps, 35%OFF**

Special Discount Code: freecram)

NEW QUESTION: 17

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: ' node16 '

start: ' start.js '

start-if: github.event_name == ' push '

main: ' index.js '

B. runs:

using: ' node16 '

pre: ' start.js '

pre-if: github.event_name == ' push '

main: ' index.js '

C. runs:

using: ' node16 '

pre-if: github.event_name == ' push ' then ' start.js '

main: ' index.js '

D. runs:

using: ' node16 '

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

Answer: ([SHOW ANSWER](#))

For JavaScript custom actions, the metadata file supports lifecycle scripts through the runs section. The correct key for a script that runs before the main action entrypoint is pre, and the correct conditional key for controlling whether that pre-script runs is pre-if. Therefore, option B is the only valid metadata structure.

GitHub Actions does not use start, start-if, before, or before-if as action metadata keys. Option C is also invalid because pre-if only defines a condition; it cannot contain both a condition and a script reference in one line. In GitHub Actions exam terms, this tests custom action metadata, JavaScript action lifecycle hooks, and conditional execution before the main action logic.

NEW QUESTION: 18

What metadata file in a custom action defines the main entry point?

- A. action.js
- B. index.js
- C. action.yml
- D. main.yml

Answer: C ([LEAVE A REPLY](#))

The action.yml file is the metadata file in a custom GitHub Action that defines the main entry point, including information such as the inputs, outputs, description, and the runs key that specifies the main entry point (e.g., a script or a Docker container).

NEW QUESTION: 19

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? (Choose three.)

- A. self-hosted runners
- B. actions stored in private repositories in the organization
- C. encrypted secrets
- D. custom Docker actions stored in GitHub Container Registry
- E. actions stored in an organizational partition in the GitHub Marketplace
- F. workflow templates

Answer: ([SHOW ANSWER](#))

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. 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.

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.

NEW QUESTION: 20

Which run: command will set a step's output?

- A. run: echo "MY_OUTPUT=foo" >> \$GITHUB_OUTPUT
- B. run: export MY_OUTPUT=foo
- C. run: echo \${GITHUB_OUTPUT=foo }
- D. run: echo "::set-env name=MY OUTPUT::foo"

Answer: ([SHOW ANSWER](#))

The \$GITHUB_OUTPUT file is used to pass data from one step to another in GitHub Actions. The echo command appends the key-value pair to this file, which sets the output variable (MY_OUTPUT) for the current step.

NEW QUESTION: 21

What is the most secure way to store sensitive information in GitHub Actions workflows?

- A. From the GitHub repository settings, store sensitive data as unencrypted text.
- B. Use GitHub Enterprise Managed User (OIDC) integration to dynamically fetch secrets from a third-party secrets manager.
- C. Use environment variables in the workflow and store sensitive data directly in the variables.
- D. Store the sensitive information as plain text in the workflow YAML file.

Answer: ([SHOW ANSWER](#))

NEW QUESTION: 22

Which of the following statements are true regarding the use of GitHub Actions on a GitHub Enterprise Server instance? (Choose three.)

- A. Use of GitHub Actions on GitHub Enterprise Server requires a persistent internet connection
- B. Actions created by GitHub are automatically available and cannot be disabled
- C. Most GitHub authored actions are automatically bundled for use on GitHub Enterprise Server
- D. Third party actions can be used on GitHub Enterprise Server by configuring GitHub Connect
- E. Actions must be defined in the .github repository
- F. Third party actions can be manually synchronized for use on GitHub Enterprise Server

Answer: ([SHOW ANSWER](#))

GitHub Actions on GitHub Enterprise Server often requires an internet connection, especially for accessing actions from the GitHub Marketplace or third-party actions unless they are manually synced to the server.

To use third-party actions on GitHub Enterprise Server, GitHub Connect can be used to establish a connection between the server and GitHub.com, enabling access to third-party actions.

Third-party actions can also be manually synchronized to the GitHub Enterprise Server, making them available for use in workflows.

NEW QUESTION: 23

What is the minimal syntax for declaring an output named foo for an action?

- ```
outputs:
 foo:
 description: Some value
```

A.
- ```
outputs:  
  - key: foo  
    description: Some value
```

B.
- ```
outputs:
 foo:
 value: Some value
```

C.
- ```
outputs:  
  - key: foo  
    value: Some value
```

D.

Answer: ([SHOW ANSWER](#))

The correct minimal syntax for declaring an output in GitHub Actions is by using the foo key under outputs , and associating it with a value (in this case, Some value). This is the simplest form to define an output in a workflow or action.

NEW QUESTION: 24

What is the simplest action type to run a shell script?

- A. Docker container action
- B. Composite action
- C. Bash script action
- D. JavaScript action

Answer: ([SHOW ANSWER](#))

The simplest official GitHub Actions action type for running shell commands or shell scripts is a composite action. Composite actions allow an action author to combine one or more run steps and reusable workflow commands in an action.yml file without needing to package a Docker image or write a JavaScript action.

Option B is correct because it directly supports shell-based automation. Option A is heavier because Docker container actions require a container image or Dockerfile. Option D is used when the action logic is written in JavaScript and executed with Node.js. Option C is incorrect because "Bash script action" is not an official GitHub Actions action type. This topic checks custom action types and reuse patterns.

NEW QUESTION: 25

Which statement accurately describes using labels to route GitHub Actions workflows to runners?

- A.** A user can assign multiple labels to a self-hosted runner and use the labels in a runs-on to target only runners that match all the specified labels.
- B.** Labels apply only to GitHub-hosted runners and are ignored by self-hosted runners.
- C.** GitHub Actions automatically assigns the runner's IP address as a label, and the label must be used in a runs-on.
- D.** To target a specific runner, the runs-on must match the runner's hostname rather than its labels.

Answer: ([SHOW ANSWER](#))

Self-hosted runners can have default labels and custom labels. These labels are used in the runs-on field to route workflow jobs to runners that match the required criteria. When multiple labels are specified, the runner must match all listed labels before GitHub Actions can assign the job to that runner. Therefore, option A is correct. Option B is false because labels are especially important for self-hosted runners. Option C is incorrect because GitHub does not require the runner IP address as a routing label. Option D is also incorrect because workflows target runners by labels or runner groups, not by hostname. This topic validates runner selection, self-hosted runner configuration, and workflow job routing.

NEW QUESTION: 26

You need to create a reusable GitHub Actions workflow template named ci.yml. The solution must ensure that ci.yml appears on the New workflow interface of GitHub Actions. Where should you store ci.yml?

- A.** .github/workflows
- B.** .github/templates
- C.** .github/workflow-templates
- D.** the root directory of each repository

Answer: ([SHOW ANSWER](#))

Organization starter workflow templates must be stored in a special workflow template directory so that they appear in the GitHub Actions "New workflow" interface. The correct location is .github/workflow-templates, typically in the organization's .github repository. Therefore, option C is correct. Option A, .github/workflows, is where active workflow files are stored inside a repository so they can be triggered by events. It is not the correct location for organization-level starter templates. Option B is not a valid GitHub Actions workflow template directory. Option D is incorrect because storing the file in each repository's root does not publish it as a reusable starter workflow template. This checks workflow template reuse and organization automation standardization.

NEW QUESTION: 27

As a developer, you are using a Docker container action in your workflow. What is required for the action to run successfully?

- A.** The job env must be set to a Linux environment.
- B.** The job runs-on must specify a Linux machine with Docker installed.

- C. The referenced action must be hosted on Docker Hub.
- D. The action must be published to the GitHub Marketplace.

Answer: ([SHOW ANSWER](#))

For a Docker container action to run in a GitHub Actions workflow, the runner must have Docker installed.

The runs-on attribute of the job should specify an environment that supports Docker, typically a Linux environment (e.g., ubuntu-latest), since Docker is widely supported and commonly used in Linux-based environments.

NEW QUESTION: 28

As a developer, you are optimizing a GitHub workflow that uses and produces many different files. You need to determine when to use caching versus workflow artifacts. Which two statements are true? (Choose two.)

- A. Use caching when reusing files that change rarely between jobs or workflow runs.
- B. Use artifacts when referencing files produced by a job after a workflow has ended.
- C. Use caching to store cache entries for up to 30 days between accesses.
- D. Use artifacts to access the GitHub Package Registry and download a package for a workflow

Answer: ([SHOW ANSWER](#))

Caching is ideal for files that change rarely, such as dependencies or build outputs, as it speeds up subsequent workflow runs by reusing previously cached files instead of re-downloading or rebuilding them.

Artifacts are used for persisting files produced during a job that need to be used in later jobs or after the workflow has ended, allowing them to be downloaded or referenced later.

NEW QUESTION: 29

Which workflow command would output the debug message "action successfully debugged"?

- A. `echo :debug::message=action successfully debugged"`
- B. `echo "debug-action successfully debugged"`
- C. `echo "::debug::action successfully debugged"`
- D. `echo ":debug:action successfully debugged:"`

Answer: ([SHOW ANSWER](#))

The `::debug::` syntax is used to output debug messages in GitHub Actions workflows. This command will print the message "action successfully debugged" in the debug logs when the workflow runs.

NEW QUESTION: 30

As a developer, you need to use GitHub Actions to deploy a microservice that requires runtime access to a secure token. This token is used by a variety of other microservices managed by different teams in different repos. To minimize management overhead and ensure the token is secure, which mechanisms should you use to store and access the token? (Choose two.)

- A.** Store the token in a configuration file in a private repository. Use GitHub Actions to deploy the configuration file to the runtime environment.
- B.** Store the token as a GitHub encrypted secret in the same repo as the code. Create a reusable custom GitHub Action to access the token by the microservice at runtime.
- C.** Use a corporate non-GitHub secret store (e.g., HashiCorp Vault) to store the token. During deployment, use GitHub Actions to store the secret in an environment variable that can be accessed at runtime.
- D.** Store the token as a GitHub encrypted secret in the same repo as the code. During deployment, use GitHub Actions to store the secret in an environment variable that can be accessed at runtime.
- E.** Store the token as an organizational-level encrypted secret in GitHub. During deployment, use GitHub Actions to store the secret in an environment variable that can be accessed at runtime.

Answer: ([SHOW ANSWER](#))

Using a corporate secret store like HashiCorp Vault provides a secure, centralized location for sensitive information. GitHub Actions can then retrieve and store the token securely during deployment by setting it as an environment variable, ensuring the token remains secure and accessible at runtime.

Storing the token as an organizational-level encrypted secret in GitHub ensures it is accessible across multiple repositories, minimizing management overhead. GitHub Actions can then use this secret during deployment by setting it as an environment variable, allowing the microservice to access it securely at runtime.

NEW QUESTION: 31

Which syntax correctly accesses a job output (output1) of an upstream job (job1) from a dependent job within a workflow?

- A.** `${{needs.job1.outputs.output1}}`
- B.** `${{needs.job1.output1}}`
- C.** `${{depends.job1.output1}}`
- D.** `${{job1.outputs.output1}}`

Answer: ([SHOW ANSWER](#))

The needs context is used to reference the outputs of jobs that are dependencies of the current job. In this case, `needs.job1.outputs.output1` correctly accesses the output of output1 from the job job1 in the dependent job.

Valid GH-200 Dumps shared by EduDump.com for Helping Passing GH-200 Exam!

EduDump.com now offer the **newest GH-200 exam dumps**, the EduDump.com GH-200 exam **questions have been updated** and **answers have been corrected** get the **newest** EduDump.com GH-200 dumps with Test Engine here:

NEW QUESTION: 32

Which choices represent best practices for publishing actions so that they can be consumed reliably? (Choose two.)

- A. repo name
- B. tag
- C. commit SHA
- D. organization name
- E. default branch

Answer: (SHOW ANSWER)

Using a tag is a best practice because tags are immutable and represent a fixed version of your action. By referencing tags, consumers of your action can be assured they are using a stable and specific version of the action, which helps in avoiding issues with breaking changes.

The commit SHA is another reliable way to specify a particular version of an action. By referencing a specific commit SHA, consumers can ensure they are using exactly the code that was written at that moment, avoiding the potential for changes in the future.

NEW QUESTION: 33

Where should workflow files be stored to be triggered by events in a repository?

- A. .github/workflows/
- B. .github/actions/
- C. Nowhere; they must be attached to an act on in the GitHub user interface
- D. anywhere
- E. .workflows/

Answer: (SHOW ANSWER)

Workflow files must be stored in the .github/workflows/ directory of the repository. This is the standard location for GitHub Actions workflow files, and workflows in this directory are automatically triggered by events defined in the file, such as pushes, pull requests, or other GitHub events.

NEW QUESTION: 34

Which of the following statements are true regarding the use of GitHub Actions on a GitHub Enterprise Server instance? (Choose three.)

- A. Third party actions can be manually synchronized for use on GitHub Enterprise Server
- B. Actions must be defined in the .github repository
- C. Actions created by GitHub are automatically available and cannot be disabled
- D. Most GitHub authored actions are automatically bundled for use on GitHub Enterprise Server
- E. Third party actions can be used on GitHub Enterprise Server by configuring GitHub Connect
- F. Use of GitHub Actions on GitHub Enterprise Server requires a persistent internet connection

Answer: ([SHOW ANSWER](#))

NEW QUESTION: 35

Which of the following commands will set the \$FOO environment variable within a script, so that it may be used in subsequent workflow job steps?

- A. run: echo "::set-env name=FOO::bar"
- B. run: echo "FOO=bar" >> \$GITHUB_ENV
- C. run: echo \${{ \$FOO=bar }}
- D. run: export FOO=bar

Answer: B ([LEAVE A REPLY](#))

The \$GITHUB_ENV environment variable is used to set environment variables that persist across steps in a workflow job. By echoing FOO=bar into \$GITHUB_ENV, the variable FOO will be available in subsequent steps within the same job.

NEW QUESTION: 36

Which workflow event is used to manually trigger a workflow run?

- A. create
- B. workflow_dispatch
- C. workflow_run
- D. status

Answer: ([SHOW ANSWER](#))

The workflow_dispatch event is used to manually trigger a workflow run in GitHub Actions. You can specify this event in the workflow file to allow users to manually trigger the workflow from the GitHub UI, often with optional input parameters.

NEW QUESTION: 37

As a developer, you have a 10-MB data set that is required in a specific workflow. Which steps should you perform so the dataset is stored encrypted and can be decrypted during the workflow? (Choose three.)

- A. Encrypt the dataset.
- B. Leverage the actions/download-secret action in the workflow.
- C. Store the dataset in a GitHub encrypted secret.
- D. Store the encryption keys in a GitHub encrypted secret.
- E. Compress the dataset
- F. Commit the encrypted dataset to the same repository as the workflow
- G. Create a GitHub encrypted secret with the Large object option selected and upload the dataset.

Answer: ([SHOW ANSWER](#))

First, the dataset should be encrypted before being stored. This ensures that the data is protected when stored in a repository.

The encrypted dataset can be stored in a GitHub secret, ensuring it is securely kept and not exposed publicly.

The encryption key needed to decrypt the dataset should also be stored in a GitHub secret to maintain security during the workflow, allowing access only when needed.

NEW QUESTION: 38

Without the need to use additional infrastructure, what is the simplest and most maintainable method for configuring a workflow job to provide access to an empty PostgreSQL database?

- A. Use service containers with a Postgres database from Docker hub.
- B. Run the actions/postgres action in a parallel job.
- C. It is currently impossible to access the database with GitHub Actions.
- D. Dynamically provision and deprovision an environment.

Answer: A ([LEAVE A REPLY](#))

GitHub Actions supports the use of service containers, which allows you to spin up a PostgreSQL database (or any other service) in a Docker container during your workflow. You can pull a PostgreSQL image from Docker Hub, and the container will automatically be available to your workflow job. This method requires no additional infrastructure and is easy to configure and maintain, as you simply define the container in the workflow file.

NEW QUESTION: 39

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?

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

A.

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

B.

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

C.

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

D.

Answer: ([SHOW ANSWER](#))

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.

NEW QUESTION: 40

You create a self-hosted runner labeled as runner1.
You need to ensure that a GitHub Actions workflow job runs only on runner1.
Which YAML statement should you use?

- A. runs-on: [self-hosted, linux-xlarge]
- B. runs-on: - self-hosted - ubuntu-latest
- C. runs-on: [self-hosted, runner1]
- D. runs-on: self-hosted

Answer: ([SHOW ANSWER](#))

To target a specific self-hosted runner, the job must use runs-on with labels that match that runner. A self-hosted runner normally has the self-hosted label plus default labels and any custom labels you assign. Since the runner is labeled runner1, the workflow should specify both self-hosted and runner1:

runs-on: [self-hosted, runner1]. GitHub Actions will then route the job only to a self-hosted runner matching all specified labels. Option A targets a different custom label. Option B is malformed and also mixes a self-hosted label with a GitHub-hosted runner label. Option D targets any self-hosted runner, not specifically runner1. GitHub documents that an array of runs-on labels requires a runner matching all listed labels.

NEW QUESTION: 41

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: ([SHOW ANSWER](#))

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.

NEW QUESTION: 42

Which default GitHub environment variable indicates the owner and repository name?

- A. REPOSITORY NAME
- B. GITHUB REPOSITORY
- C. ENV REPOSITORY
- D. GITHUB WORKFLOW REPO

Answer: ([SHOW ANSWER](#))

The `GITHUB_REPOSITORY` environment variable contains the owner and repository name in the format `owner/repository`. It is automatically provided by GitHub Actions and can be used to reference the repository in workflows.

NEW QUESTION: 43

You are reaching your organization's storage limit for GitHub artifacts and packages. What should you do to prevent the storage limit from being reached?

- A. via the `.github` repository owned by the organization
- B. via repositories owned by the organization
- C. via the GitHub Marketplace
- D. via a repository owned by a third party

Answer: ([SHOW ANSWER](#))

To prevent reaching the storage limit for GitHub artifacts and packages, you should manage and clean up artifacts and packages stored in repositories owned by your organization. This includes deleting unnecessary artifacts and managing the lifecycle of packages, as they contribute directly to your organization's storage quota.

NEW QUESTION: 44

As a DevOps engineer, you need to execute a deployment to different environments like development and testing based on the labels added to a pull request. The deployment should use the `releases` branch and trigger only when there is a change in the files under `apps` folder. Which code block should be used to define the deployment workflow trigger?

A. on:

`pull_request_label:`

`branches:`

`- ' releases '`

paths:

- ' apps/** '

B. on:

pull_request_review:

types: [labeled]

branches:

- ' releases '

paths:

- ' apps/** '

C. on:

pull_request:

types: [labeled]

branches:

- ' releases/** '

paths:

- ' apps '

D. on:

pull_request:

types: [labeled]

branches:

- ' releases '

paths:

- ' apps/** '

Answer: (SHOW ANSWER)

The correct trigger must use the pull_request event because the workflow is based on activity on a pull request. The specific activity is a label being added, so types: [labeled] is required. The deployment must target the releases branch, so the branch filter must be branches: - ' releases '. The workflow must also run only when files under the apps folder change, so paths: - ' apps/** ' is the correct recursive path filter. Option A is invalid because pull_request_label is not a GitHub Actions event.

Option B uses the wrong event, pull_request_review. Option C uses an incorrect branch/path combination. GitHub workflow syntax supports event activity types, branch filters, and path filters together.

NEW QUESTION: 45

Which code version is the default for scheduled triggered workflows?

A. Scheduled workflows run on the:

B. latest commit and branch on which the workflow was triggered,

C. latest commit from the branch named schedule,

D. latest commit from the branch named main,

E. specified commit and branch from the workflow YAML file,

F. latest commit on the default or base branch

Answer: ([SHOW ANSWER](#))

Scheduled workflows in GitHub Actions are triggered at specified times, and they run on the latest commit of the branch that triggers the workflow. This means the workflow will run on the most recent commit on the branch that was active at the time the scheduled event occurs.

NEW QUESTION: 46

As a developer, you need to add the correct syntax to allow the following workflow file to be triggered by multiple types of events.

Which two code blocks should you add starting at line 5? Each correct answer presents a complete solution.

NOTE: Each correct answer is worth one point.

4 name: Node CI/CD

5

6

7

8 jobs:

9 build:

10 runs-on:

11 steps:

12 - uses: actions/checkout@v2

13 - uses: actions/setup-node@v1

14 with:

15 node-version: 12

16 - run: npm ci

17 - run: npm test

18

A. on:

env:

- 'prod'

- 'qa'

- 'test'

B. on: [push, commit]

C. on: [push, pull_request]

D. on:

branches:

- 'main'

- 'dev'

E. on:

push:

branches:

```
- main
release:
types:
- created
F. on:
schedule:
- cron: ' */15 * * * * '
initiate:
- ' main '
```

Answer: (SHOW ANSWER)

A workflow can be triggered by multiple events either by using array syntax, such as on: [push, pull_request], or by configuring each event separately when filters or activity types are needed. Option C is valid because both push and pull_request are supported workflow events. Option E is also valid because it configures two event triggers: push filtered to the main branch and release filtered to the created activity type. Option B is invalid because commit is not a GitHub Actions workflow event.

Option A wrongly puts environments under on. Option D defines branch filters without an event. Option F uses initiate, which is not a valid event. GitHub documents both multiple-event array syntax and separate event configuration for filters.

Valid GH-200 Dumps shared by EduDump.com for Helping Passing GH-200 Exam!

EduDump.com now offer the **newest GH-200 exam dumps**, the EduDump.com GH-200 exam **questions have been updated** and **answers have been corrected** get the **newest** EduDump.com GH-200 dumps with Test Engine here:

<https://www.edudump.com/exams/Microsoft/GH-200/premium/> (102 Q&As Dumps, **35%OFF**

Special Discount Code: freecram)

NEW QUESTION: 47

Which default GitHub environment variable indicates the name of the person or app that initiated a workflow?

- A. ENV_ACTOR
- B. GITHUB_WORKFLOW_ACTOR
- C. GITHUB_ACTOR
- D. GITHUB_USER

Answer: (SHOW ANSWER)

The GITHUB_ACTOR environment variable indicates the name of the person or app that initiated the workflow. This variable is automatically provided by GitHub in the workflow and can be used to identify the user or application triggering the workflow.

NEW QUESTION: 48

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 .github/secrets folder in MyRepo.
- D. Create and access SuperSecret from the secrets store in MyRepo.

Answer: (SHOW ANSWER)

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.

Valid GH-200 Dumps shared by EduDump.com for Helping Passing GH-200 Exam!

EduDump.com now offer the **newest GH-200 exam dumps**, the EduDump.com GH-200 exam **questions have been updated** and **answers have been corrected** get the **newest**

EduDump.com GH-200 dumps with Test Engine here:

<https://www.edudump.com/exams/Microsoft/GH-200/premium/> (102 Q&As Dumps, **35%OFF**

Special Discount Code: freecram)