AZ-400Q&As

Designing and Implementing Microsoft DevOps Solutions

Pass Microsoft AZ-400 Exam with 100% Guarantee

Free Download Real Questions & Answers **PDF** and **VCE** file from:

https://www.pass2lead.com/az-400.html

100% Passing Guarantee
100% Money Back Assurance

Following Questions and Answers are all new published by Microsoft Official Exam Center

Instant Download After Purchase
100% Money Back Guarantee
365 Days Free Update
800,000+ Satisfied Customers

QUESTION 1

Note: This question is part of a series of questions that present the same scenario. Each question in the series contains a unique solution that might meet the stated goals. Some question sets might have more than one correct solution, while others might not have a correct solution.

After you answer a question in this section, you will NOT be able to return to it. As a result, these questions will not appear in the review screen.

You manage a project in Azure DevOps.

You need to prevent the configuration of the project from changing over time.

Solution: Add a code coverage step to the build pipelines.

Does this meet the goal?

A. Yes

B. No

Correct Answer: B

Instead implement Continuous Assurance for the project.

Reference: https://azsk.azurewebsites.net/04-Continous-Assurance/Readme.html

QUESTION 2

DRAG DROP

You have an Azure DevOps pipeline that is used to deploy a Node.js app.

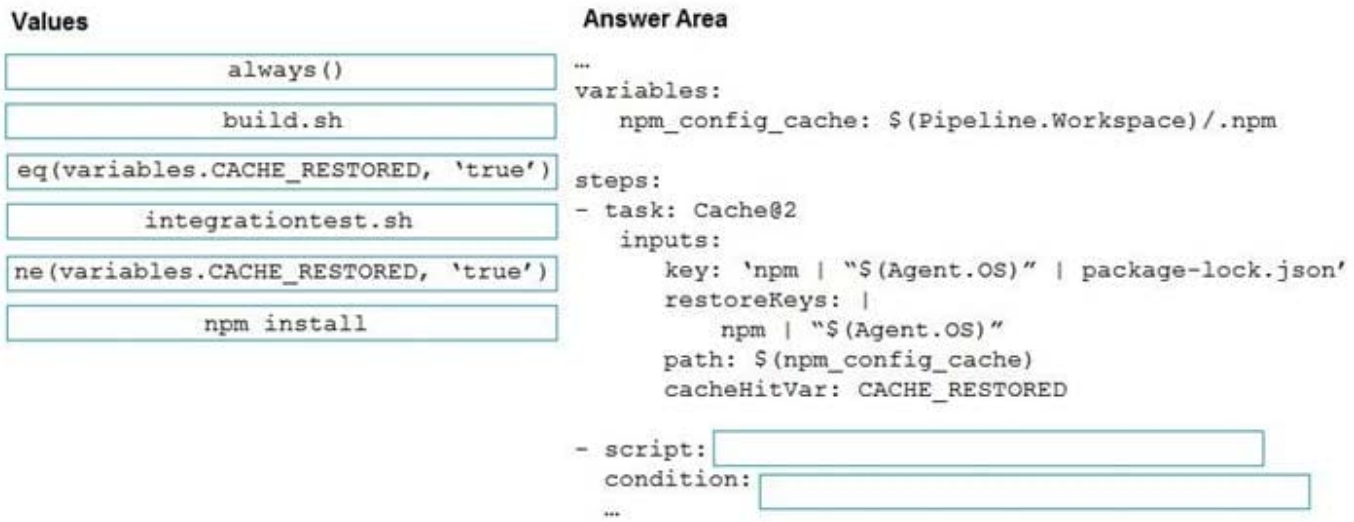
You need to ensure that the dependencies are cached between builds.

How should you configure the deployment YAML? To answer, drag the appropriate values to the correct targets. Each value may be used once, more than once, or not at all. You may need to drag the split bar between panes or scroll to view

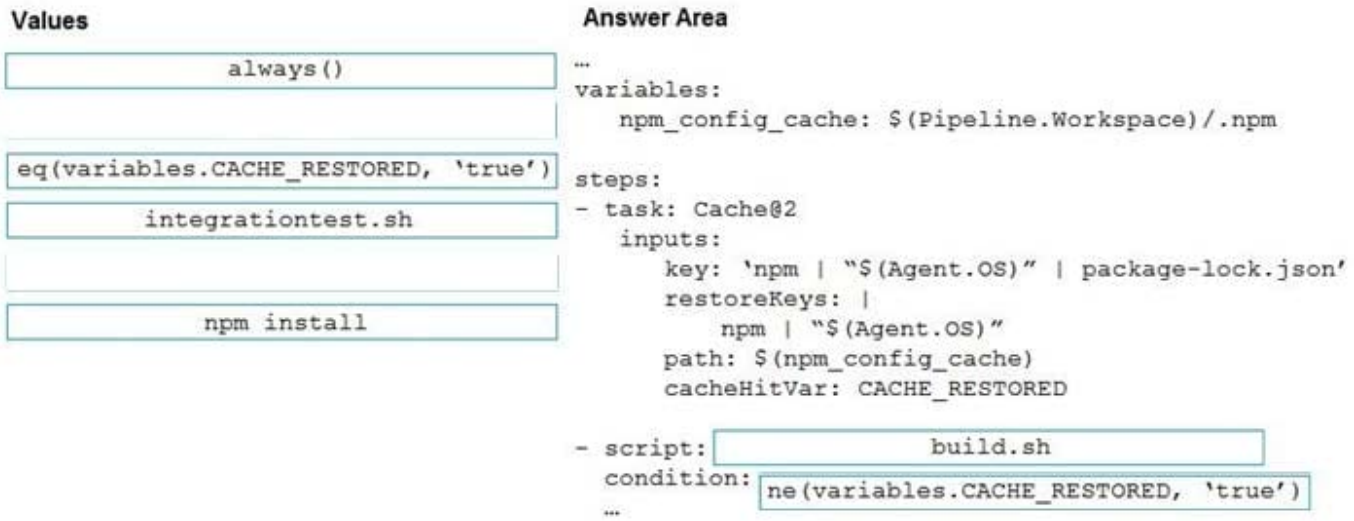
content.

NOTE: Each correct selection is worth one point.

Select and Place:



Correct Answer:



Explanation:

Box 1: build.sh Conditioning on cache restoration In some scenarios, the successful restoration of the cache should cause a different set of steps to be run. For example, a step that installs dependencies can be skipped if the cache was restored. This is possible using the cacheHitVar task input. Setting this input to the name of an environment variable will cause the variable to be set to true when there\\'s a cache hit, inexact on a restore key cache hit, otherwise it will be set to false. This variable can then be referenced in a step condition or from within a script.

In the following example, the install-deps.sh step is skipped when the cache is restored:

YAML

steps:

-task: Cache@2

inputs:

key: mykey | mylockfile

restoreKeys: mykey

path: $(Pipeline.Workspace)/mycache

cacheHitVar: CACHE_RESTORED

-

script: install-deps.sh condition: ne(variables.CACHE_RESTORED, \\'true\\')

-

script: build.sh

Box 2: ne(variables.CACHE_RESTORED, \\'true\\')

Incorrect:

*

always() condition: always() # this step will always run, even if the pipeline is canceled

*

.integrationtest.sh

Build base images in integration tests.

*

npm install

npm install downloads a package and it\\'s dependencies.

When run without arguments, npm install downloads dependencies defined in a package.json file and generates a node_modules folder with the installed modules.

Reference: https://learn.microsoft.com/en-us/azure/devops/pipelines/release/caching

---

QUESTION 3

You have an Azure DevOps organization named Contoso.

You need to receive Microsoft Teams notifications when work items are updated.

What should you do?

A. From Azure DevOps, configure a service hook subscription

B. From Microsoft Teams, configure a connector

C. From the Microsoft Teams admin center, configure external access

D. From Microsoft Teams, add a channel

E. From Azure DevOps, install an extension

Correct Answer: A

Service hooks let you run tasks on other services when events happen in your Azure DevOps projects. For example, create a card in Trello when a work item is created or send a push notification to your team\\'s mobile devices when a build

fails. You can also use service hooks in custom apps and services as a more efficient way to drive activities when events happen in your projects.

Note: Service hook publishers define a set of events. Subscriptions listen for the events and define actions to take based on the event. Subscriptions also target consumers, which are external services that can run their own actions, when an

event occurs.

Reference:

https://docs.microsoft.com/en-us/azure/devops/service-hooks/overview

QUESTION 4

You have a project in Azure DevOps named Project1. Project1 contains a build pipeline named Pipe1 that builds an application named Appl.

You have an agent pool named Pool1 that contains a Windows Server 2019-based self-hosted agent.

Pipe1 uses Pool1.

You plan to implement another project named Project2. Project2 will have a build pipeline named Pipe2 that builds an application named App2.

App1 and App2 have conflicting dependencies.

You need to minimize the possibility that the two build pipelines will conflict with each other. The solution must minimize infrastructure costs.

What should you do?

A. Create two container jobs.

B. Change the self-hosted agent to use Red Hat Enterprise Linux (RHEL) 8.

C. Add another self-hosted agent

D. Add a Docker Compose task to the build pipelines.

Correct Answer: A

QUESTION 5

HOTSPOT

You have an Azure Kubernetes Service (AKS) pod.

You need to configure a probe to perform the following actions:

1.

Confirm that the pod is responding to service requests.

2.

Check the status of the pod four times a minute.

3.

Initiate a shutdown if the pod is unresponsive.

How should you complete the YAML configuration file? To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.

Hot Area:

Answer Area

```
apiVersion: v1
kind: Pod
metadata:
     labels:
          test: readiness-and-liveness
     name: readiness-http
spec:
     containers:
     - name: container1
       image: k8s.ger.io/readiness-and-lieveness
       args:
       - /server
```

[livenessProbe: | readinessProbe: | ShutdownProbe: | startupProbe:]

```
    httpGet:
         path: /checknow
         port: 8123
         httpHeaders:
         - name: Custom-Header
           value: CheckNow
```

[initialDelaySeconds: 15 | periodSeconds: 15 | timeoutSeconds: 15]

Correct Answer:

Answer Area

```
apiVersion: v1
kind: Pod
metadata:
     labels:
          test: readiness-and-liveness
     name: readiness-http
spec:
     containers:
     - name: container1
       image: k8s.ger.io/readiness-and-lieveness
       args:
       - /server
```

livenessProbe:
readinessProbe:
ShutdownProbe:
startupProbe:

```
    httpGet:
         path: /checknow
         port: 8123
         httpHeaders:
         - name: Custom-Header
           value: CheckNow
```

initialDelaySeconds: 15
periodSeconds: 15
timeoutSeconds: 15

Box 1: readinessProbe:

For containerized applications that serve traffic, you might want to verify that your container is ready to handle incoming requests. Azure Container Instances supports readiness probes to include configurations so that your container can\\'t be

accessed under certain conditions.

Incorrect Answers:

livenessProbe: Containerized applications may run for extended periods of time, resulting in broken states that may need to be repaired by restarting the container. Azure Container Instances supports liveness probes so that you can

configure your containers within your container group to restart if critical functionality is not working.

Box 2: periodSeconds: 15

The periodSeconds property designates the readiness command should execute every 15 seconds.

Reference:

https://docs.microsoft.com/en-us/azure/container-instances/container-instances-readiness-probe

Latest AZ-400 Dumps AZ-400 PDF Dumps AZ-400 Study Guide