AZ-400 Q&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.pass4itsure.com/az-400.html

100% Passing Guarantee
100% Money Back Assurance

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

QUESTION 1

DRAG DROP

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

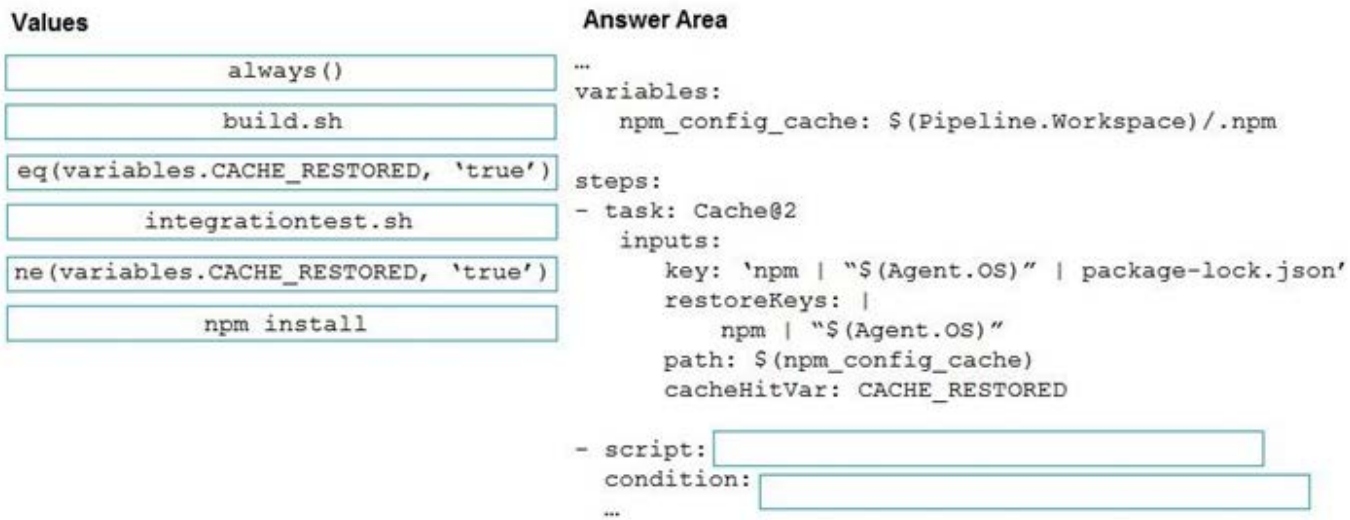
You need to ensure that the dependencies are cached between builds.

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

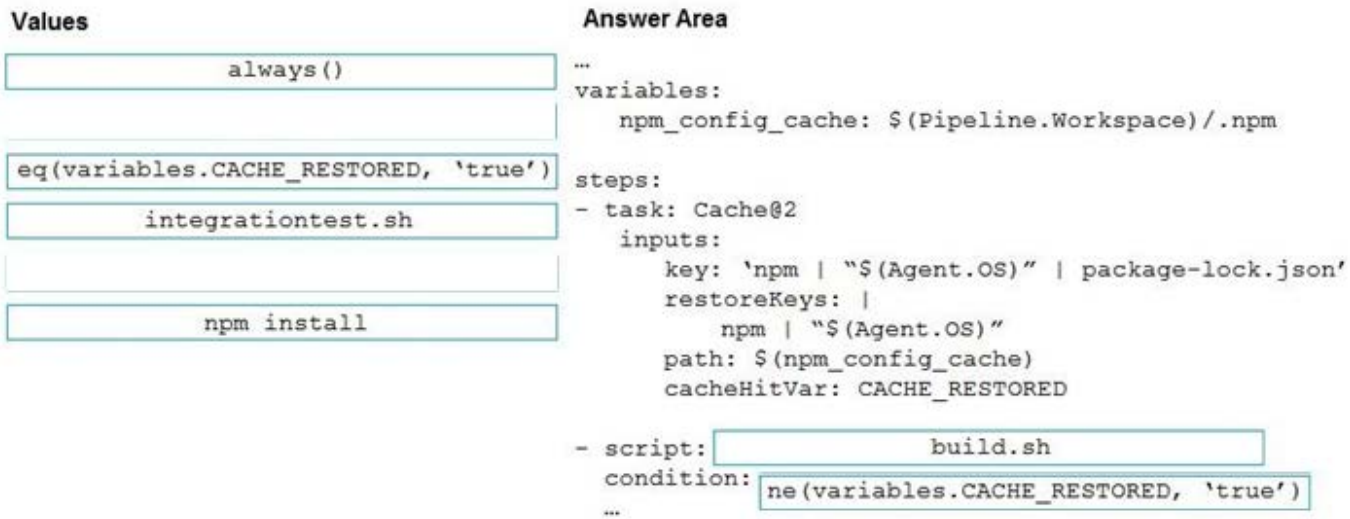
content.

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 2

HOTSPOT

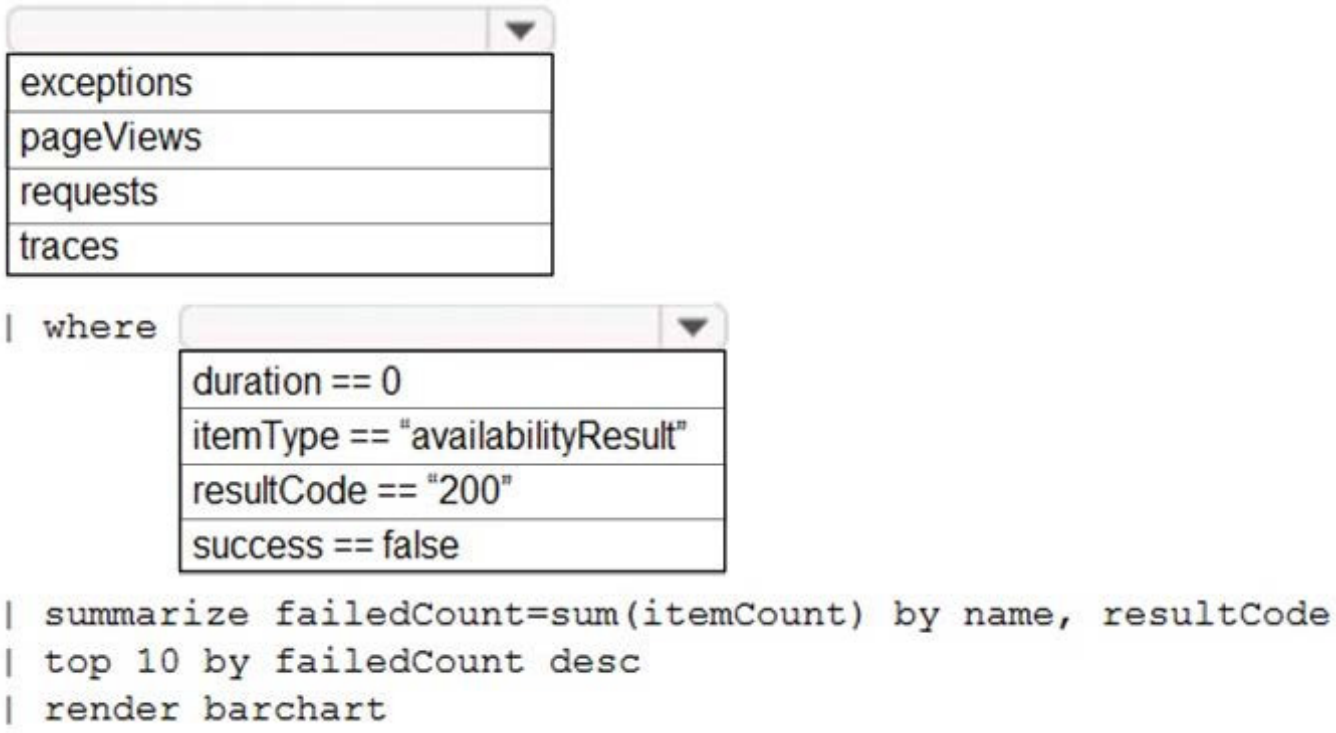
You have an Azure web app named Webapp1.

You need to use an Azure Monitor query to create a report that details the top 10 pages of Webapp1 that failed.

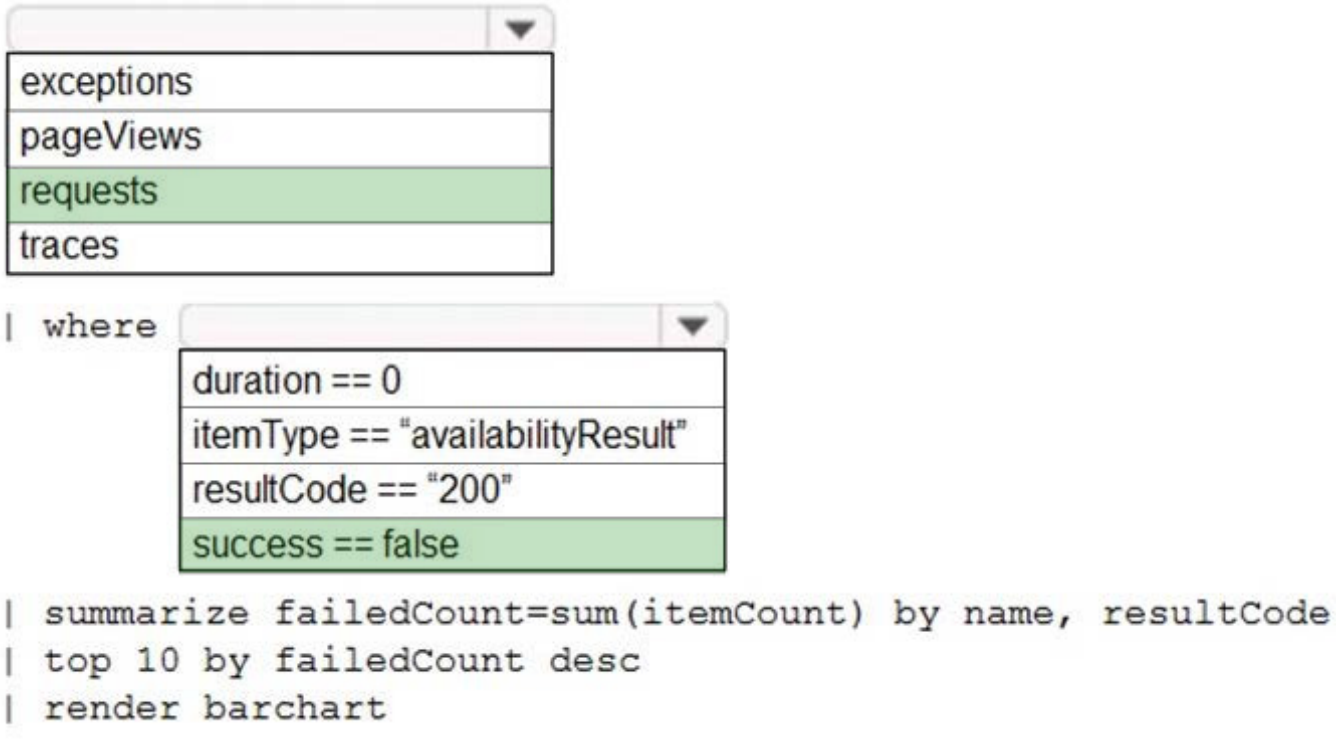
How should you complete the query? To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.

Hot Area:



Correct Answer:



Box 1: requests

Failed requests (requests/failed):

The count of tracked server requests that were marked as failed.

Kusto code:

requests

| where success == \\'False\\'

Box 2: success == false

QUESTION 3

DRAG DROP

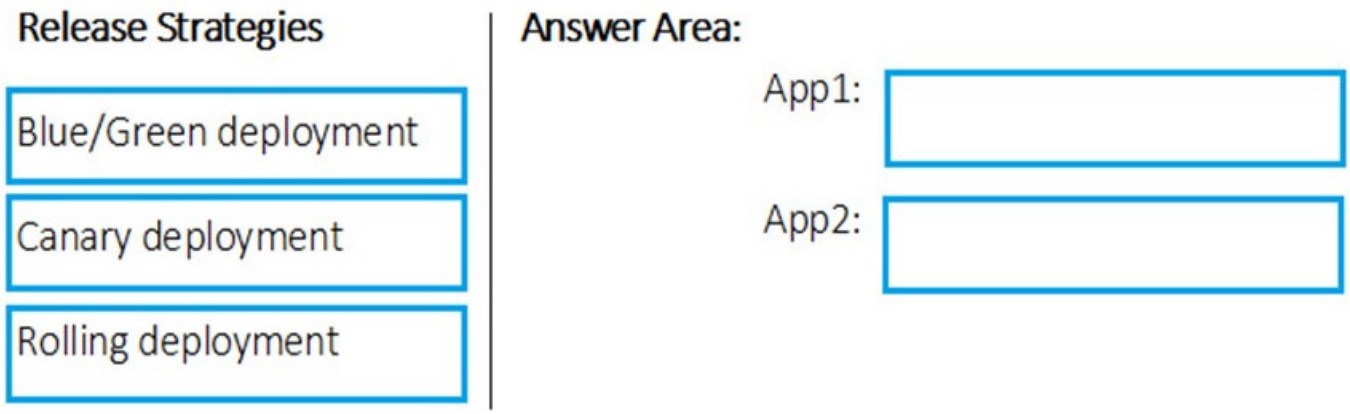
You are defining release strategies for two applications as shown in the following table.

Application name	Goal
App1	Failure of App1 has a major impact on your company. You need a small group of users, who opted in to a testing App1, to test new releases of the application.
App2	You need to minimize the time it takes to deploy new releases of App2, and you must be able to roll back as quickly as possible.

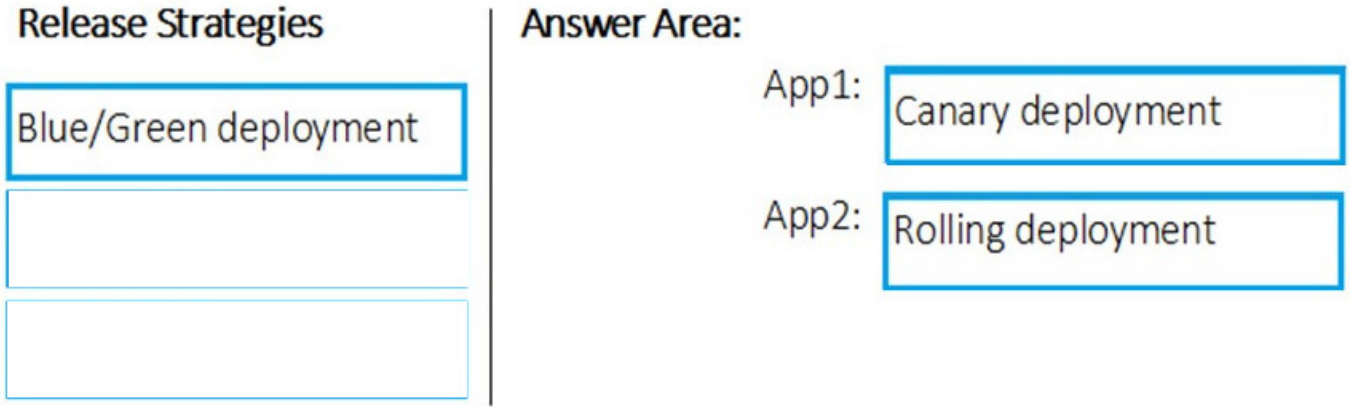
Which release strategy should you use for each application? To answer, drag the appropriate release strategies to the correct applications. Each release strategy may be used once, more than once, or not at all. You may need to drag the split bar between panes or scroll to view content.

NOTE: Each correct selection is worth one point.

Select and Place:



Correct Answer:



App1: Canary deployment

With canary deployment, you deploy a new application code in a small part of the production infrastructure. Once the application is signed off for release, only a few users are routed to it. This minimizes any impact.

With no errors reported, the new version can gradually roll out to the rest of the infrastructure.

App2: Rolling deployment:

In a rolling deployment, an application\\'s new version gradually replaces the old one. The actual deployment happens over a period of time. During that time, new and old versions will coexist without affecting functionality or user experience.

This process makes it easier to roll back any new component incompatible with the old components.

Incorrect Answers:

Blue/Green deployment

A blue/green deployment is a change management strategy for releasing software code. Blue/green deployments, which may also be referred to as A/B deployments require two identical hardware environments that are configured exactly the

same way. While one environment is active and serving end users, the other environment remains idle.

Blue/green deployments are often used for consumer-facing applications and applications with critical uptime requirements. New code is released to the inactive environment, where it is thoroughly tested. Once the code has been vetted, the

team makes the idle environment active, typically by adjusting a router configuration to redirect application program traffic. The process reverses when the next software iteration is ready for release.

References:

https://dev.to/mostlyjason/intro-to-deployment-strategies-blue-green-canary-and-more-3a3

QUESTION 4

DRAG DROP

You have an Azure Repos repository named repo1.

You delete a branch named features/feature11.

You need to recover the deleted branch.

Which three commands should you run in sequence? To answer, move the appropriate commands from the list of commands to the answer area and arrange them in the correct order.

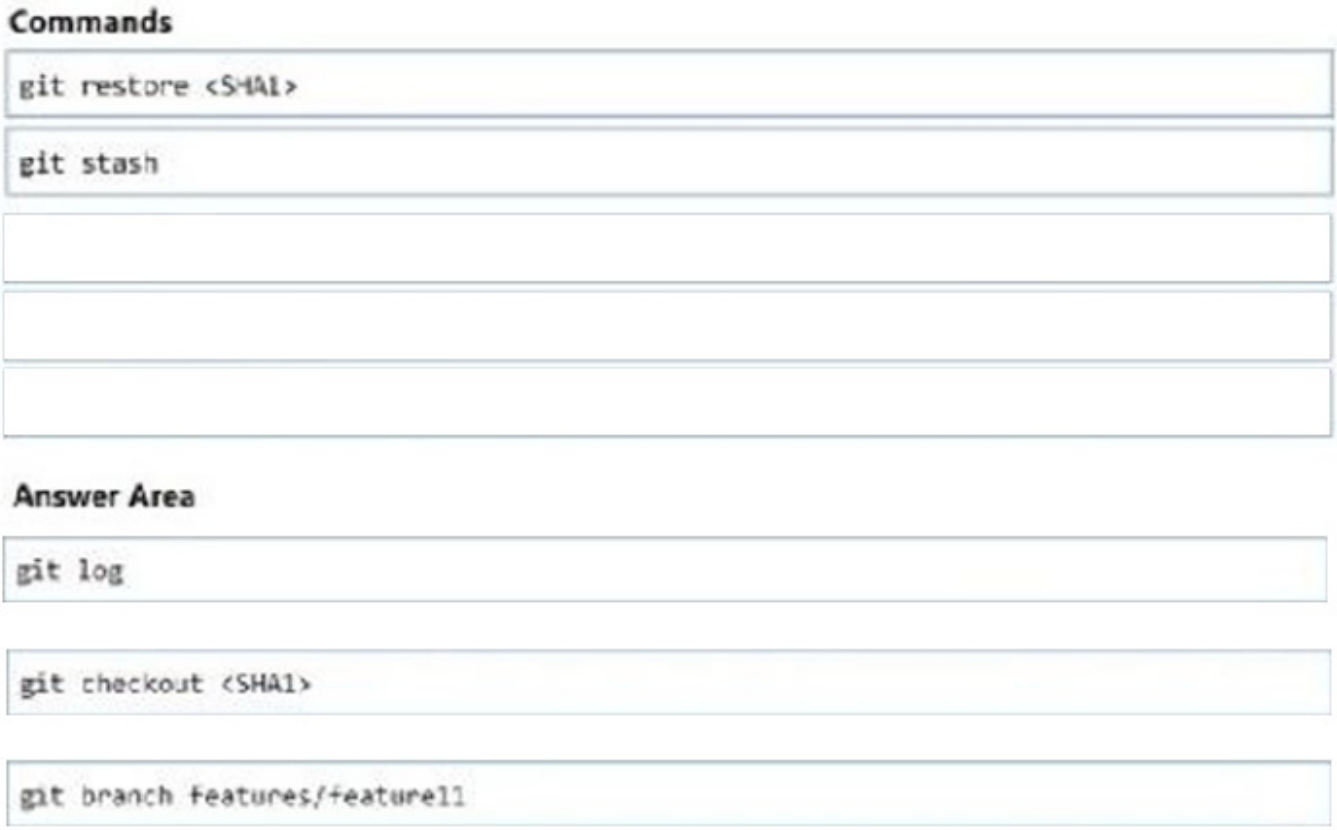
Select and Place:

Commands

git restore <SHA1>
git stash
git log
git checkout <SHA1>
git branch features/feature11

Answer Area

Correct Answer:



QUESTION 5

You have a project in Azure DevOps named Project1.

You implement a Continuous Integration/Continuous Deployment (CI/CD) pipeline that uses PowerShell Desired State Configuration (DSC) to configure the application infrastructure.

You need to perform a unit test and an integration test of the configuration before Project1 is deployed.

What should you use?

A. the PSScriptAnalyzer tool

B. the Pester test framework

C. the PSCodeHealth module

D. the Test-DscConfiguration cmdlet

Correct Answer: B

Explanation:

Pester is a testing and mocking framework for PowerShell.

Pester provides a framework for writing and running tests. Pester is most commonly used for writing unit and integration tests, but it is not limited to just that. It is also a base for tools that validate whole environments, computer deployments,

database configurations and so on.

Example: Building a Continuous Integration and Continuous Deployment pipeline with DSC

This example demonstrates how to build a Continuous Integration/Continuous Deployment (CI/CD) pipeline by using PowerShell, DSC, and Pester.

IntegrationTests: Runs the Pester integration tests.

UnitTests: Runs the Pester unit tests.

Incorrect:

* PSScriptAnalyzer PSScriptAnalyzer is a static code checker for PowerShell modules and scripts. PSScriptAnalyzer checks the quality of PowerShell code by running a set of rules. The rules are based on PowerShell best practices identified by PowerShell Team and the community. It generates DiagnosticResults (errors and warnings) to inform users about potential code defects and suggests possible solutions for improvements.

PSScriptAnalyzer ships with a collection of built-in rules that check various aspects of PowerShell code such as:

The presence of uninitialized variables Use of PSCredential type Use of Invoke-Expression And many more

Reference: https://pester.dev/docs/quick-start/#what-is-pester https://learn.microsoft.com/en-us/azure/devops/pipelines/release/dsc-cicd https://github.com/PowerShell/PSScriptAnalyzer