AZ-400$^{Q\&As}$

Designing and Implementing Microsoft DevOps Solutions

QUESTION 1

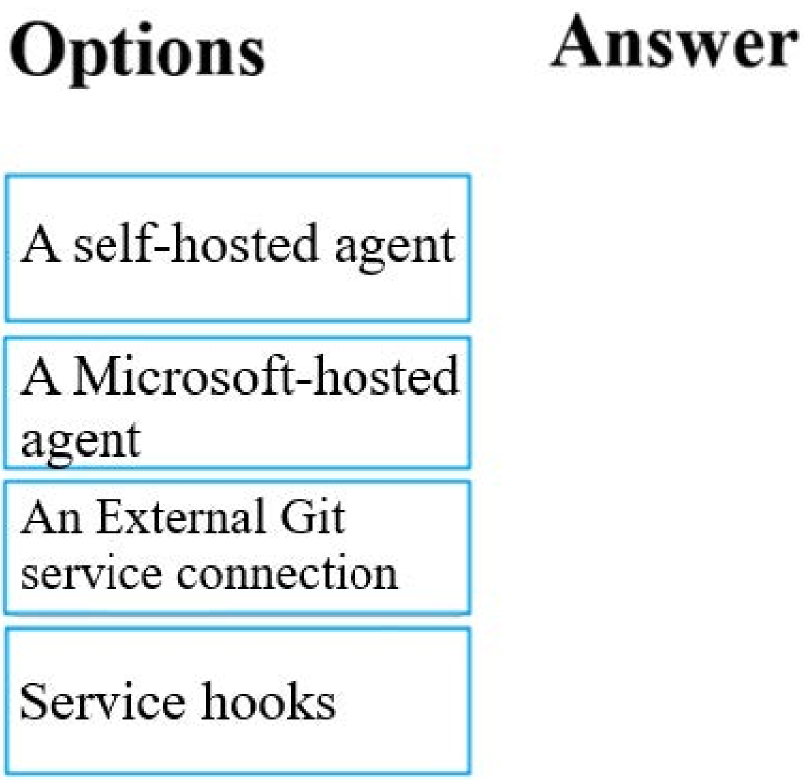
DRAG DROP

You have an on-premises Bitbucket Server with a firewall configured to block inbound Internet traffic. The server is used for Git-based source control.

You intend to manage the build and release processes using Azure DevOps. This plan requires you to integrate Azure DevOps and Bitbucket.

Which of the following will allow for this integration? Answer by dragging the correct options from the list to the answer area.

Select and Place:



Options

Answer

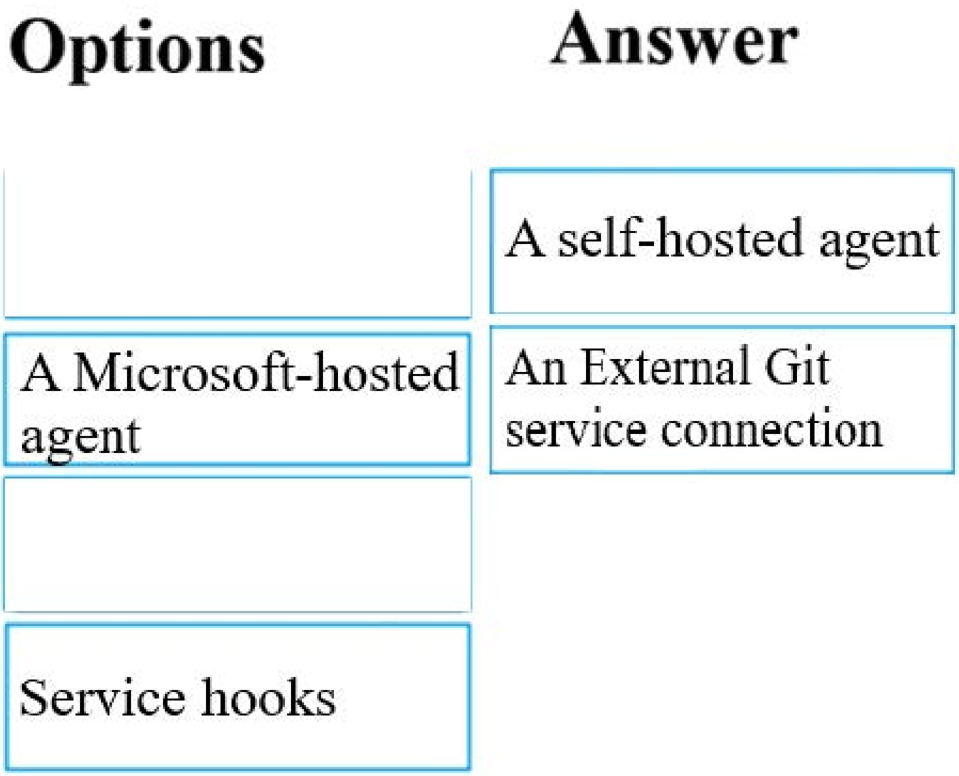
A self-hosted agent

A Microsoft-hosted agent

An External Git service connection

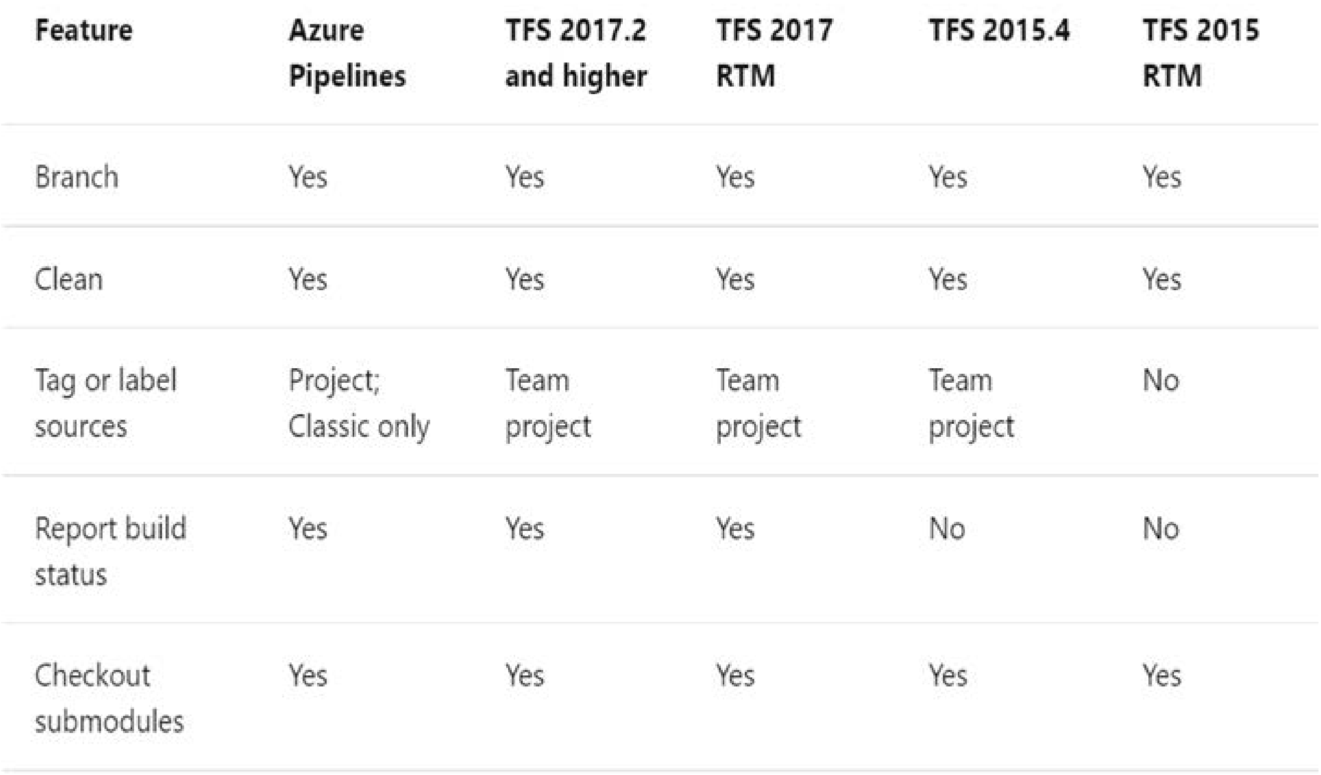
Service hooks

Correct Answer:



Options	Answer
	A self-hosted agent
A Microsoft-hosted agent	An External Git service connection
Service hooks	

Reference: https://docs.microsoft.com/en-us/azure/devops/pipelines/repos/pipeline-options-for-git



Feature	Azure Pipelines	TFS 2017.2 and higher	TFS 2017 RTM	TFS 2015.4	TFS 2015 RTM
Branch	Yes	Yes	Yes	Yes	Yes
Clean	Yes	Yes	Yes	Yes	Yes
Tag or label sources	Project; Classic only	Team project	Team project	Team project	No
Report build status	Yes	Yes	Yes	No	No
Checkout submodules	Yes	Yes	Yes	Yes	Yes

QUESTION 2

HOTSPOT

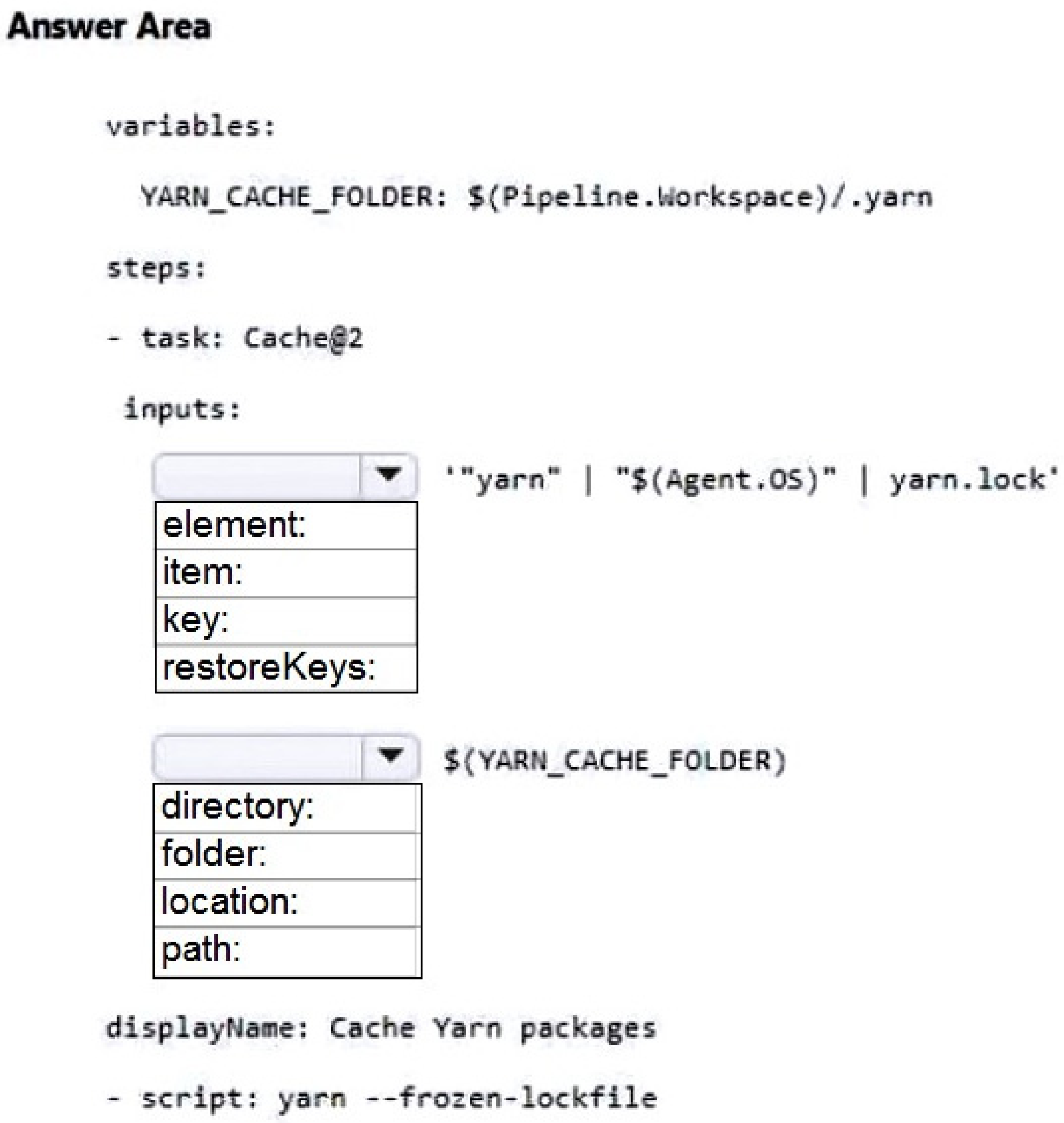
You have an Azure subscription that contains Azure DevOps build pipelines.

You need to implement pipeline caching by using the cache task.

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

NOTE: Each correct selection is worth one point.

Hot Area:



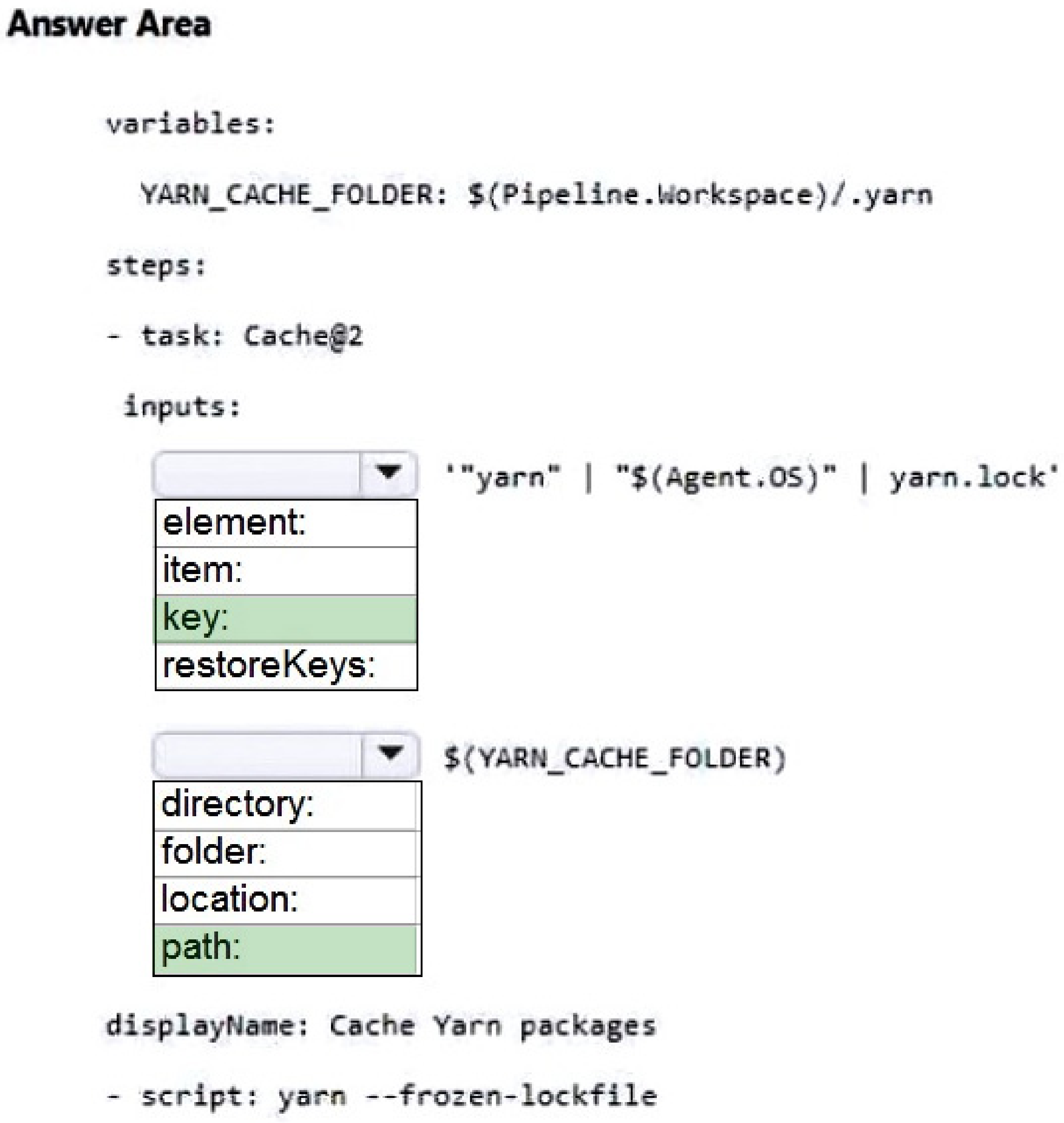
Answer Area

```
variables:
  YARN_CACHE_FOLDER: $(Pipeline.Workspace)/.yarn
steps:
- task: Cache@2
 inputs:
   [▼] '"yarn" | "$(Agent.OS)" | yarn.lock'
   [▼] $(YARN_CACHE_FOLDER)
displayName: Cache Yarn packages
- script: yarn --frozen-lockfile
```

- element:
- item:
- key:
- restoreKeys:

- directory:
- folder:
- location:
- path:

Correct Answer:



Answer Area

```
variables:
  YARN_CACHE_FOLDER: $(Pipeline.Workspace)/.yarn
steps:
- task: Cache@2
 inputs:
   [▼] '"yarn" | "$(Agent.OS)" | yarn.lock'
   element:
   item:
   key:
   restoreKeys:

   [▼] $(YARN_CACHE_FOLDER)
   directory:
   folder:
   location:
   path:
displayName: Cache Yarn packages
- script: yarn --frozen-lockfile
```

Box 1: key:

Box 2: path:

Example:

Here\\'s an example showing how to cache dependencies installed by Yarn:

YAML

Copy

variables: YARN_CACHE_FOLDER: $(Pipeline.Workspace)/.yarn steps:

-task: Cache@2

inputs:

```
key: \\'"yarn" | "$(Agent.OS)" | yarn.lock\\'

restoreKeys: |

"yarn" | "$(Agent.OS)"

"yarn"

path: $(YARN_CACHE_FOLDER)

displayName: Cache Yarn packages
```

-script: yarn --frozen-lockfile In this example, the cache key contains three parts: a static string ("yarn"), the OS the job is running on since this cache is unique per operating system, and the hash of the yarn.lock file that uniquely identifies the set of dependencies in the cache.

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

QUESTION 3

You have 50 Node.js-based projects that you scan by using WhiteSource. Each project includes Package.json, Package-lock.json, and Npm-shrinkwrap.json files.

You need to minimize the number of libraries reports by WhiteSource to only the libraries that you explicitly reference. What should you do?

A. Configure the File System Agent plug-in.

B. Add a devDependencies section to Package-lock.json.

C. Configure the Artifactory plug-in.

D. Delete Package-lock.json.

Correct Answer: B

Separate Your Dependencies

Within your package.json file be sure you split out your npm dependencies between devDependencies and (production) dependencies. The key part is that you must then make use of the --production flag when installing the npm packages.

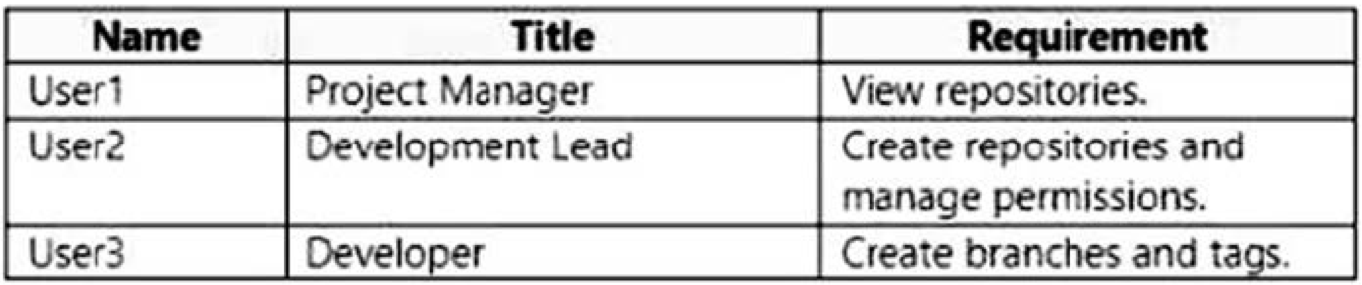
The --production flag will exclude all packages defined in the devDependencies section.

References: https://blogs.msdn.microsoft.com/visualstudioalmrangers/2017/06/08/manage-your-open-source-usage-and-security-as-reported-by-your-cicd-pipeline/

QUESTION 4

DRAG DROP

You have an Azure subscription that contains a project in Azure DevOps named Project1. You have three Azure Active Directory (Azure AD) users that require access to Project1 as shown in the following table.



Name	Title	Requirement
User1	Project Manager	View repositories.
User2	Development Lead	Create repositories and manage permissions.
User3	Developer	Create branches and tags.

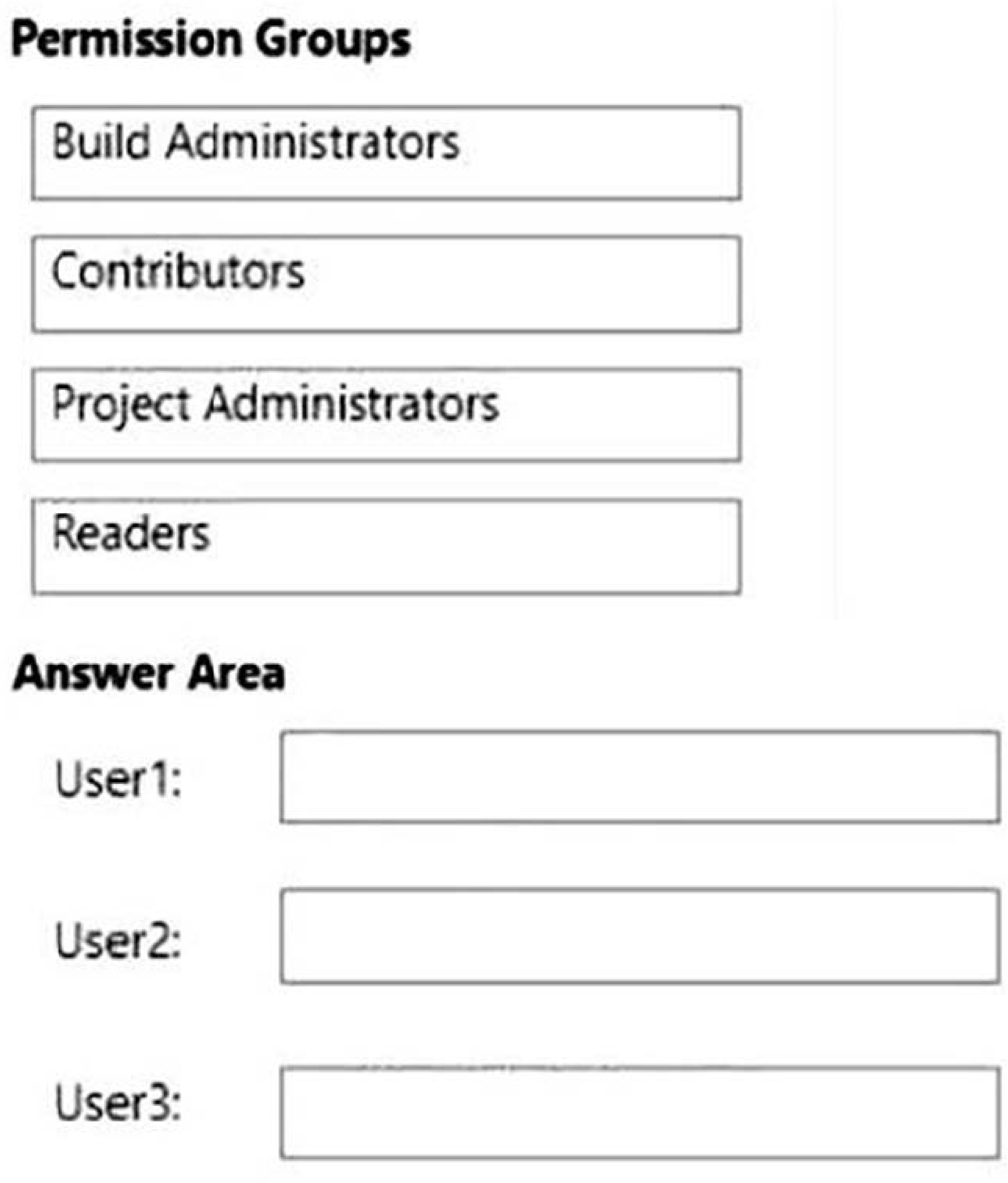
You need to ensure that the users have the appropriate permissions. The solution must use the principle of least privilege.

To which permission group in Azure DevOps should you add each user? To answer, drag the appropriate permission groups to the correct users. Each permission group 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:



Permission Groups

- Build Administrators
- Contributors
- Project Administrators
- Readers

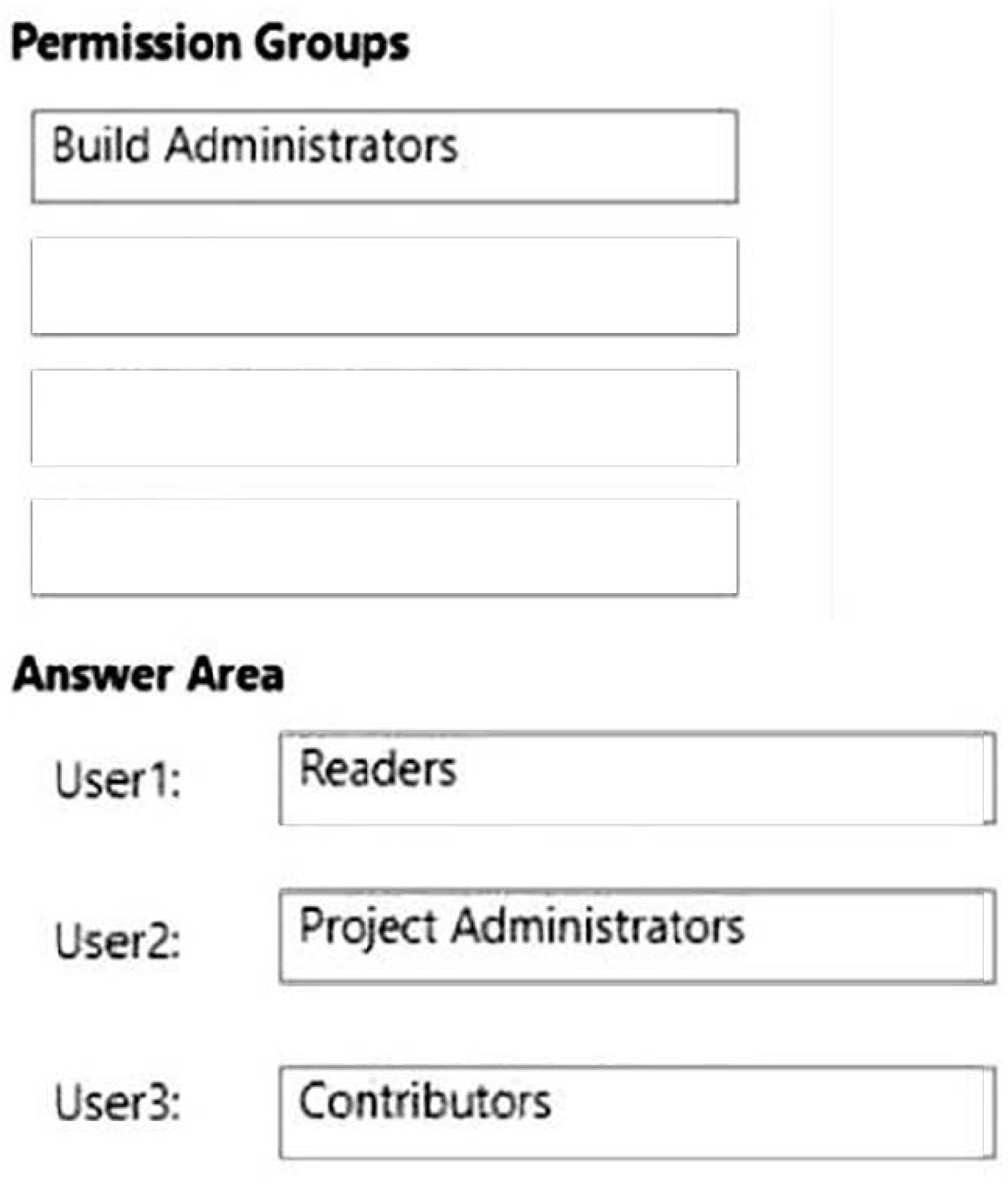
Answer Area

User1:

User2:

User3:

Correct Answer:



Permission Groups

Build Administrators

Answer Area

User1:	Readers
User2:	Project Administrators
User3:	Contributors

QUESTION 5

Your company has an Azure DevOps project,

The source code for the project is stored in an on-premises repository and uses on an on-premises build server.

You plan to use Azure DevOps to control the build process on the build server by using a self-hosted agent.

You need to implement the self-hosted agent.

You download and install the agent on the build server.

Which two actions should you perform next? Each correct answer presents part of the solution.

A. From Azure, create a shared access signature (SAS).

B. From the build server, create a certificate, and then upload the certificate to Azure Storage.

C. From the build server, create a certificate, and then upload the certificate to Azure Key Vault.

D. From DevOps, create a personal access token (PAT).

E. From the build server, run config.cmd.

Correct Answer: DE

https://docs.microsoft.com/en-us/azure/devops/pipelines/agents/v2-windows?view=azure-devops (Get PAT, run config)