- **Vendor:** Microsoft
- **Exam Code:** AZ-400
- **Exam Name:** Implementing and Operating Cisco Data Center Core Technologies (DCCOR)
- **New Updated Questions from Braindump2go (Updated in July/2020)**

**QUESTION 178**
You are deploying a server application that will run on a Server Core installation of Windows Server 2019.
You create an Azure key vault and a secret.
You need to use the key vault to secure API secrets for third-party integrations.
Which three actions should you perform? Each correct answer presents part of the solution.
NOTE: Each correct selection is worth one point.

A. Configure RBAC for the key vault.
B. Modify the application to access the key vault.
C. Configure a Key Vault access policy.
D. Deploy an Azure Desired State Configuration (DSC) extension.
E. Deploy a virtual machine that uses a system-assigned managed identity.

**Answer:** BCE
**Explanation:**
BE: An app deployed to Azure can take advantage of Managed identities for Azure resources, which allows the app to authenticate with Azure Key Vault using Azure AD authentication without credentials (Application ID and Password/Client Secret) stored in the app.
C:
1. Select Add Access Policy.
2. Open Secret permissions and provide the app with Get and List permissions.
3. Select Select principal and select the registered app by name. Select the Select button.
4. Select OK.
5. Select Save.
6. Deploy the app.
References:
https://docs.microsoft.com/en-us/aspnet/core/security/key-vault-configuration

**QUESTION 179**
**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.**
Your company has a project in Azure DevOps for a new web application.
You need to ensure that when code is checked in, a build runs automatically.
Solution: From the Continuous deployment trigger settings of the release pipeline, you enable the Pull request trigger setting.
Does this meet the goal?

A. Yes

B. No

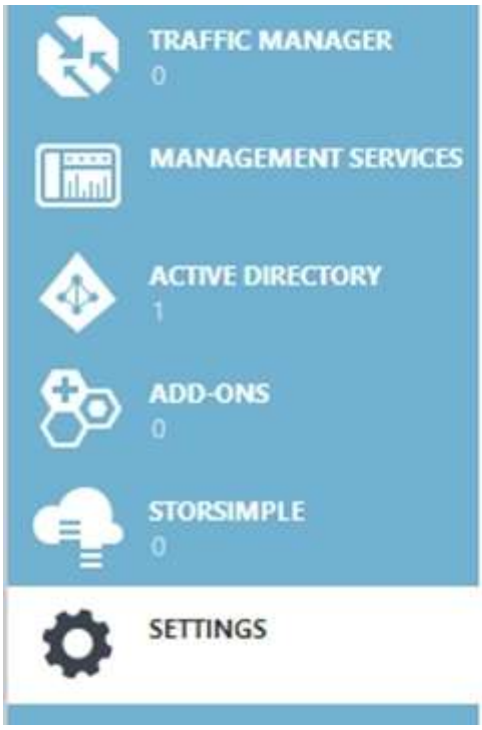
**Answer:** B
**Explanation:**
Instead, In Visual Designer you enable continuous integration (CI) by:
1. Select the Triggers tab.
2. Enable Continuous integration.
Reference:
https://docs.microsoft.com/en-us/azure/devops/pipelines/get-started-designer

**QUESTION 180**
You use WhiteSource Bolt to scan a Node.js application.
The WhiteSource Bolt scan identifies numerous libraries that have invalid licenses. The libraries are used only during development and are not part of a production deployment.
You need to ensure that WhiteSource Bolt only scans production dependencies.
Which two actions should you perform? Each correct answer presents part of the solution.
NOTE: Each correct selection is worth one point.

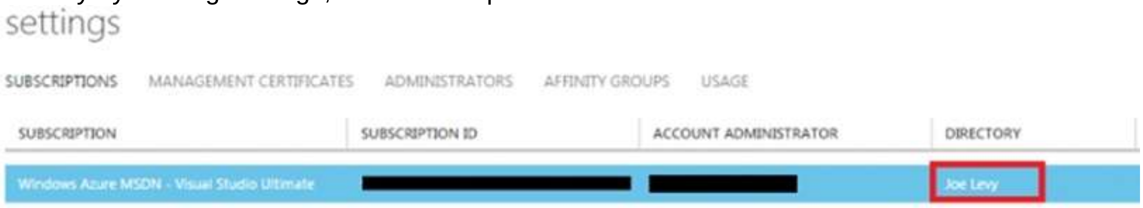
A. Run `npm install` and specify the `--production` flag.
B. Modify the WhiteSource Bolt policy and set the action for the licenses used by the development tools to Reassign.
C. Modify the devDependencies section of the project's Package.json file.
D. Configure WhiteSource Bolt to scan the node_modules directory only.

**Answer:** AC
**Explanation:**
A: To resolve NPM dependencies, you should first run "npm install" command on the relevant folders before executing the plugin.
C: All npm packages contain a file, usually in the project root, called package.json - this file holds various metadata relevant to the project. This file is used to give information to npm that allows it to identify the project as well as handle the project's dependencies. It can also contain other metadata such as a project description, the version of the project in a particular distribution, license information, even configuration data - all of which can be vital to both npm and to the end users of the package.
Reference:
https://whitesource.atlassian.net/wiki/spaces/WD/pages/34209870/NPM+Plugin
https://nodejs.org/en/knowledge/getting-started/npm/what-is-the-file-package-json

**QUESTION 181**
SIMULATION
You plan to deploy a runbook that will create Azure AD user accounts.
You need to ensure that runbooks can run the Azure PowerShell cmdlets for Azure Active Directory.
To complete this task, sign in to the Microsoft Azure portal.
**Answer:**
Azure Automation now ships with the Azure PowerShell module of version 0.8.6, which introduced the ability to non-interactively authenticate to Azure using OrgId (Azure Active Directory user) credential-based authentication. Using the steps below, you can set up Azure Automation to talk to Azure using this authentication type.
Step 1: Find the Azure Active Directory associated with the Azure subscription to manage:
1. Log in to the Azure portal as the service administrator for the Azure subscription you want to manage using Azure Automation. You can find this user by logging in to the Azure portal as any user with access to this Azure subscription, then clicking Settings, then Administrators.



2. Note the name of the directory associated with the Azure subscription you want to manage. You can find this directory by clicking Settings, then Subscriptions.

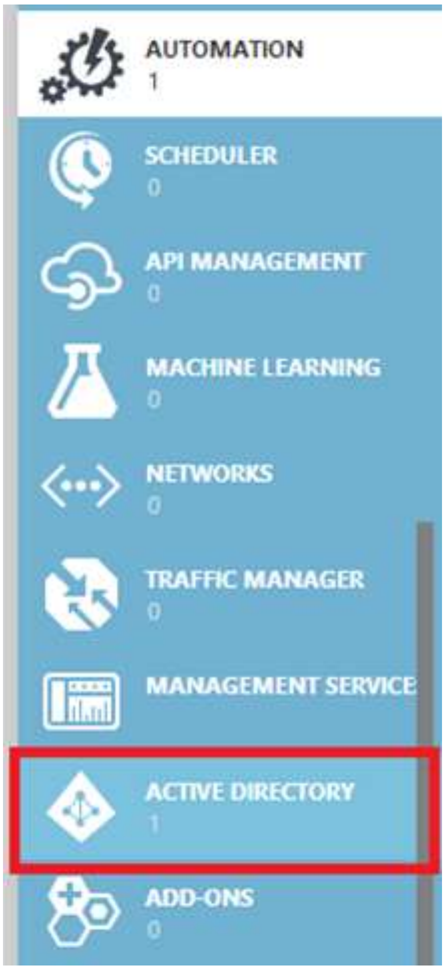


settings

SUBSCRIPTIONS MANAGEMENT CERTIFICATES ADMINISTRATORS AFFINITY GROUPS USAGE

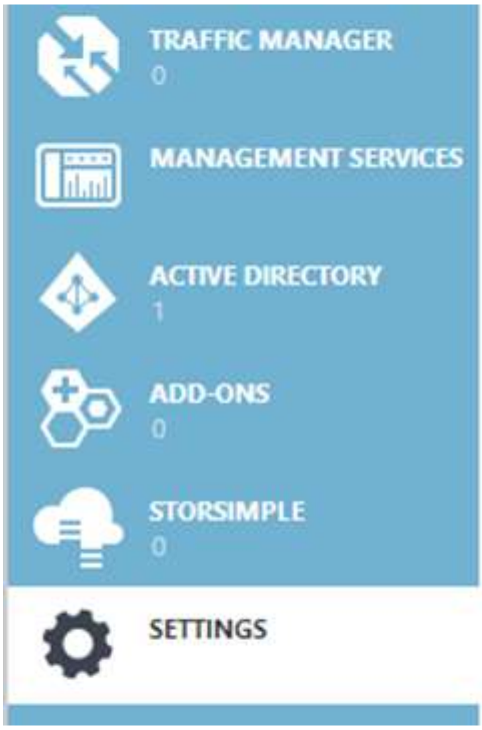
| SUBSCRIPTION | SUBSCRIPTION ID | ACCOUNT ADMINISTRATOR | DIRECTORY |
|---|---|---|---|
| Windows Azure MSDN - Visual Studio Ultimate | [redacted] | [redacted] | Joe Levy |

Step 2: Create an Azure Active Directory user in the directory associated with the Azure subscription to manage:
You can skip this step if you already have an Azure Active Directory user in this directory. and plan to use this OrgId to manage Azure.
1. In the Azure portal click on Active Directory service.



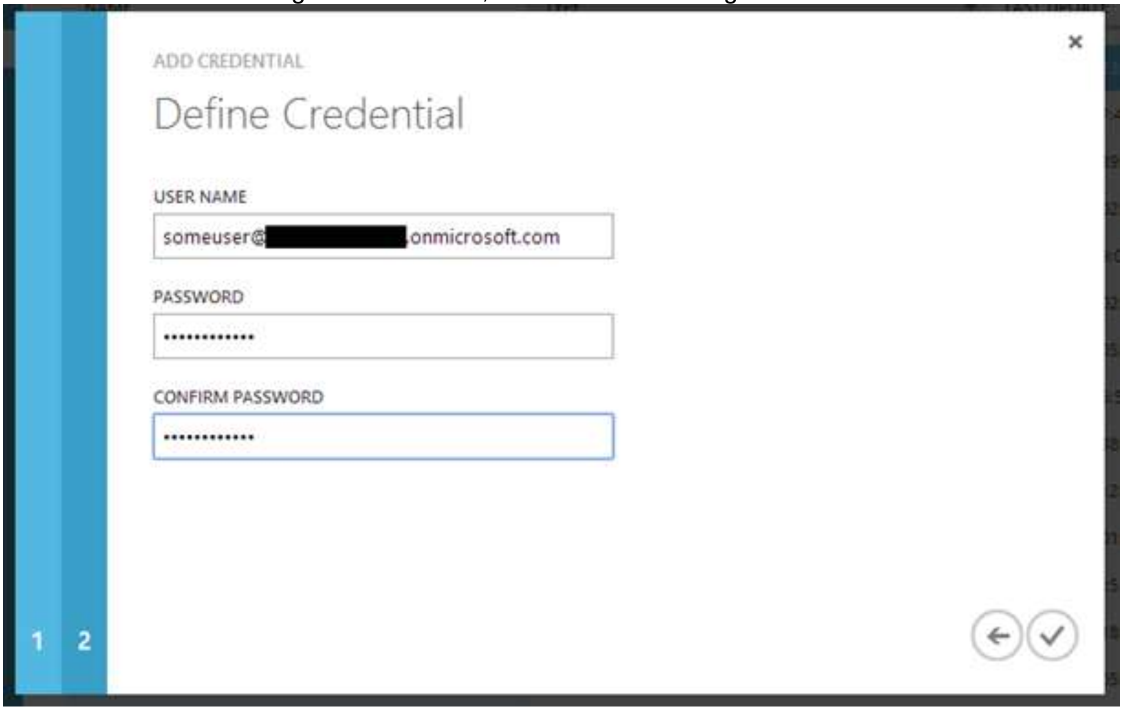
2. Click the directory name that is associated with this Azure subscription.
3. Click on the Users tab and then click the Add User button.
4. For type of user, select “New user in your organization.” Enter a username for the user to create.
5. Fill out the user’s profile. For role, pick “User.” Don’t enable multi-factor authentication. Multi-factor accounts cannot be used with Azure Automation.
6. Click Create.
7. Jot down the full username (including part after @ symbol) and temporary password.

Step 3: Allow this Azure Active Directory user to manage this Azure subscription.
1. Click on Settings (bottom Azure tab under StorSimple)



2. Click Administrators
3. Click the Add button. Type the full user name (including part after @ symbol) of the Azure Active Directory user you want to set up to manage Azure. For subscriptions, choose the Azure subscriptions you want this user to be able to manage. Click the check mark.

Step 4: Configure Azure Automation to use this Azure Active Directory user to manage this Azure subscription
Create an Azure Automation credential asset containing the username and password of the Azure Active Directory user that you have just created. You can create a credential asset in Azure Automation by clicking into an Automation Account and then clicking the Assets tab, then the Add Setting button.



Define Credential

USER NAME
someuser@[redacted]onmicrosoft.com

PASSWORD
•••••••••••••

CONFIRM PASSWORD
•••••••••••••

Note: Once you have set up the Azure Active Directory credential in Azure and Azure Automation, you can now manage Azure from Azure Automation runbooks using this credential.
References:
https://azure.microsoft.com/sv-se/blog/azure-automation-authenticating-to-azure-using-azure-active-directory/

**QUESTION 182**

You use a Git repository in Azure Repos to manage the source code of a web application. Developers commit changes directly to the master branch.

You need to implement a change management procedure that meets the following requirements:

```
- The master branch must be protected, and new changes must be built in the feature
branches first.
- Changes must be reviewed and approved by at least one release manager before each
merge.
- Changes must be brought into the master branch by using pull requests.
```

What should you configure in Azure Repos?

A. branch policies of the master branch
B. Services in Project Settings
C. Deployment pools in Project Settings
D. branch security of the master branch

**Answer:** A

**Explanation:**

Branch policies help teams protect their important branches of development. Policies enforce your team's code quality and change management standards.

Reference:

https://docs.microsoft.com/en-us/azure/devops/repos/git/branch-policies

**QUESTION 183**

**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 plan to update the Azure DevOps strategy of your company.

You need to identify the following issues as they occur during the company's development process:

```
- Licensing violations
- Prohibited libraries
```

Solution: You implement continuous integration.

Does this meet the goal?

A. Yes
B. No

**Answer:** A

**Explanation:**

WhiteSource is the leader in continuous open source software security and compliance management.

WhiteSource integrates into your build process, irrespective of your programming languages, build tools, or development environments. It works automatically, continuously, and silently in the background, checking the security, licensing, and quality of your open source components against WhiteSource constantly- updated definitive database of open source repositories.

Reference:

https://azuredevopslabs.com/labs/vstsextend/whitesource/

**QUESTION 184**

**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 plan to update the Azure DevOps strategy of your company. You need to identify the following issues as they occur during the company's development process:

```
- Licensing violations
- Prohibited libraries
```

Solution: You implement pre-deployment gates.
Does this meet the goal?

A. Yes
B. No

**Answer:** B
**Explanation:**
Instead use implement continuous integration.
Note: WhiteSource is the leader in continuous open source software security and compliance management. WhiteSource integrates into your build process, irrespective of your programming languages, build tools, or development environments. It works automatically, continuously, and silently in the background, checking the security, licensing, and quality of your open source components against WhiteSource constantly-updated definitive database of open source repositories.
Reference:
https://azuredevopslabs.com/labs/vstsextend/whitesource/

**QUESTION 185**
**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 plan to update the Azure DevOps strategy of your company.
You need to identify the following issues as they occur during the company's development process:

```
- Licensing violations
- Prohibited libraries
```

Solution: You implement automated security testing.
Does this meet the goal?

A. Yes
B. No

**Answer:** B
**Explanation:**
Instead use implement continuous integration.
Note: WhiteSource is the leader in continuous open source software security and compliance management. WhiteSource integrates into your build process, irrespective of your programming languages, build tools, or development environments. It works automatically, continuously, and silently in the background, checking the security, licensing, and quality of your open source components against WhiteSource constantly-updated definitive database of open source repositories.
Reference:
https://azuredevopslabs.com/labs/vstsextend/whitesource/

**QUESTION 186**
**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.**
Your company uses Azure DevOps to manage the build and release processes for applications.
You use a Git repository for applications source control.
You need to implement a pull request strategy that reduces the history volume in the master branch.
Solution: You implement a pull request strategy that uses fast-forward merges.
Does this meet the goal?

A. Yes

B. No

**Answer:** A
**Explanation:**
No fast-forward merge - This option merges the commit history of the source branch when the pull request closes and creates a merge commit in the target branch.
Reference:
https://docs.microsoft.com/en-us/azure/devops/repos/git/branch-policies

**QUESTION 187**
**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.**
Your company uses Azure DevOps to manage the build and release processes for applications.
You use a Git repository for applications source control.
You need to implement a pull request strategy that reduces the history volume in the master branch.
Solution: You implement a pull request strategy that uses squash merges.
Does this meet the goal?

A. Yes
B. No

**Answer:** B
**Explanation:**
Instead use fast-forward merge.
Note:
Squash merge - Complete all pull requests with a squash merge, creating a single commit in the target branch with the changes from the source branch.
No fast-forward merge - This option merges the commit history of the source branch when the pull request closes and creates a merge commit in the target branch.
Reference:
https://docs.microsoft.com/en-us/azure/devops/repos/git/branch-policies

**QUESTION 188**
**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.**
Your company uses Azure DevOps to manage the build and release processes for applications.
You use a Git repository for applications source control.
You need to implement a pull request strategy that reduces the history volume in the master branch.
Solution: You implement a pull request strategy that uses an explicit merge.
Does this meet the goal?

A. Yes
B. No

**Answer:** B
**Explanation:**
Instead use fast-forward merge.
Note:
No fast-forward merge - This option merges the commit history of the source branch when the pull request closes and creates a merge commit in the target branch.
Reference:
https://docs.microsoft.com/en-us/azure/devops/repos/git/branch-policies