The cloning capability applies only to *virtualized* domain controllers, which is itself a newly supported capability in Windows Server 2012.

This section covers the following topic:

□ Configure domain controller cloning

Cloning domain controllers

Beginning with Windows Server 2012, you can now deploy additional domain controllers in a domain by safely copying an existing virtual domain controller. In earlier versions of Windows Server, the fastest way to deploy a new domain controller, physical or virtual, was to start a fresh sysprepped image, promote the new server based on that image to a domain controller, and then complete additional configuration requirements as necessary. Cloning is much faster.

Cloning a domain controller has certain environmental requirements. In addition, cloning a domain controller is a special, specific procedure that includes adding the source virtual machine (VM) to a built-in group account and running certain Windows PowerShell cmdlets prior to exporting and importing.

It’s also worth noting that cloning seems tailor-made for questions on the 70-411 exam. Not only is this feature new in Windows Server 2012, but the cloning prerequisites and the cloning procedure also are made up of details that are just specific enough for the exam. For this reason, be sure to learn these details well, such as the specific names of cmdlets and XML files.

For more information about virtualized domain controllers, see “Test Lab Guide: Demonstrate Virtualized Domain Controller (VDC) in Windows Server ‘8’ Beta” at *http://www.microsoft.com/en-us/download/details.aspx?id=29027.*

Prerequisites for cloning

To clone a domain controller, three different servers need to be running Windows Server 2012:

□ The host server with the Hyper-V server role installed, on which the source VM is running as a guest

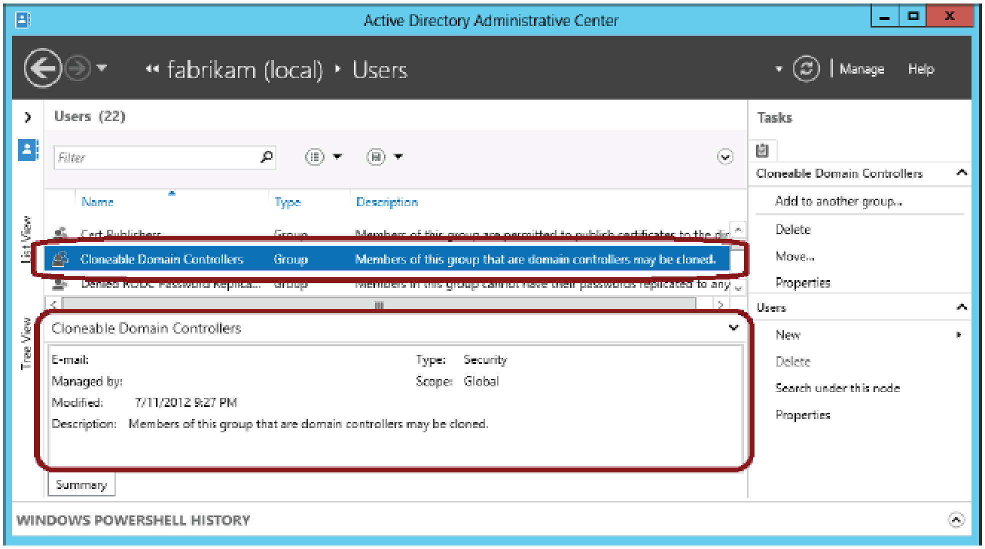
□ A second server, physical or virtual, that is itself a domain controller in the same domain as the VM to be cloned, and that is hosting the PDC Emulator operations master role

□ The source VM to be cloned that is a domain controller, and that cannot be hosting the PDC Emulator operations master role

Note also that the clone domain controller that results from the cloning procedure will be located in the same site as the source domain controller.

Add the source domain controller to the Cloneable Domain Controllers group

If your environment meets the prerequisites, you are ready to begin the cloning procedure. The first step in this procedure is to add the source VM that is a domain controller to the Cloneable Domain Controllers global security group. This built-in group account is new to Windows Server 2012 and is found in the Users container within a domain, as shown in Figure 8-1.



Name	Type	Description
Cert Publishers	Group	Members of this group are permitted to publish certificates to the dir
Cloneable Domain Controllers	Group	Members of this group that are domain controllers may be cloned.
Denied RODC Password Replica...	Group	Members in this group cannot have their passwords replicated to any

Cloneable Domain Controllers

E-mail:
Managed by:
Modified: 7/11/2012 9:27 PM
Description: Members of this group that are domain controllers may be cloned.
Type: Security
Scope: Global

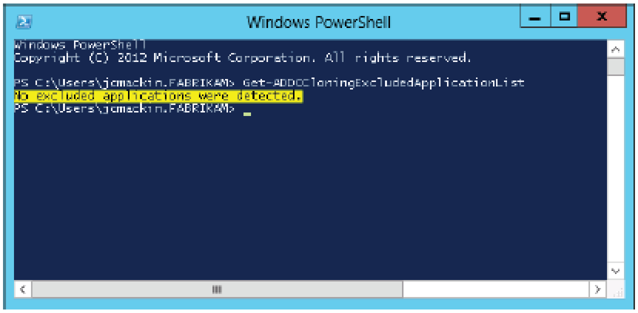
FIGURE 8-1 Windows Server 2012 includes a new Cloneable Domain Controllers global security group.

FIGURE 8-1 Windows Server 2012 includes a new Cloneable Domain Controllers global security group.

To add a user or computer account to a new security group, you can use the Add-ADGroupMember cmdlet. For more information about this cmdlet, visit *http://technet.microsoft.com/en-us/library/ee617210.aspx*.

Review applications with the Get-ADDCCloningExcludedApplicationList cmdlet

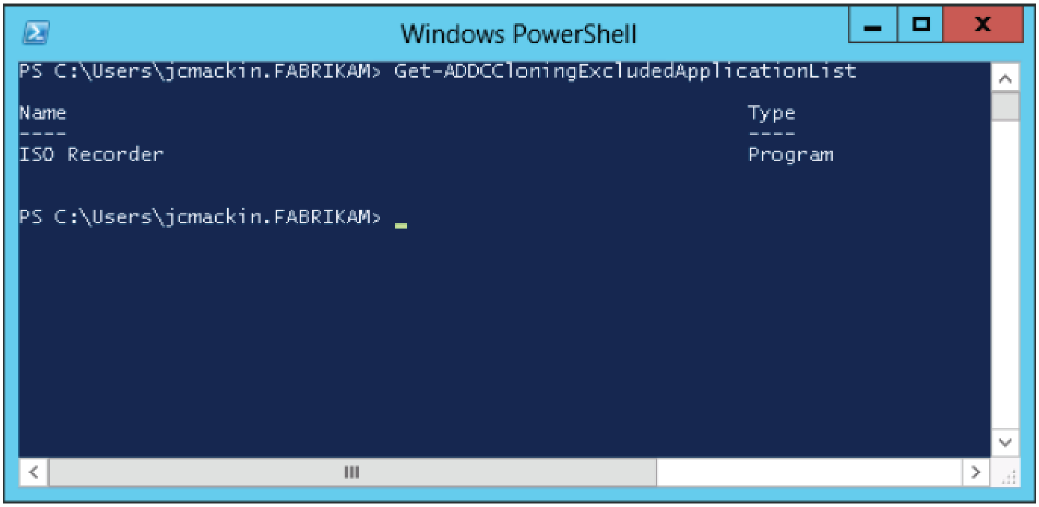
The next step in cloning a virtual domain controller is to run the Get-ADDCCloningExcludedApplicationList cmdlet on the source VM. The purpose of this cmdlet is to present a list of applications or services that are not evaluated for cloning and that are installed on the source VM. If no such applications are found, the cmdlet provides the output shown in Figure 8-2. If an unevaluated application is found, an output similar to the one in Figure 8-3 is displayed.



```
Windows PowerShell
Copyright (C) 2012 Microsoft Corporation. All rights reserved.

PS C:\Users\jcmackin.FABRIKAM> Get-ADDCCloningExcludedApplicationList
No excluded applications were detected.
PS C:\Users\jcmackin.FABRIKAM> _
```

FIGURE 8-2 Output of Get-ADDCCloningExcludedApplicationList revealing no unsafe applications.



```
PS C:\Users\jcmackin.FABRIKAM> Get-ADDCCloningExcludedApplicationList

Name                                                        Type
----                                                        ----
ISO Recorder                                                Program


PS C:\Users\jcmackin.FABRIKAM> _
```

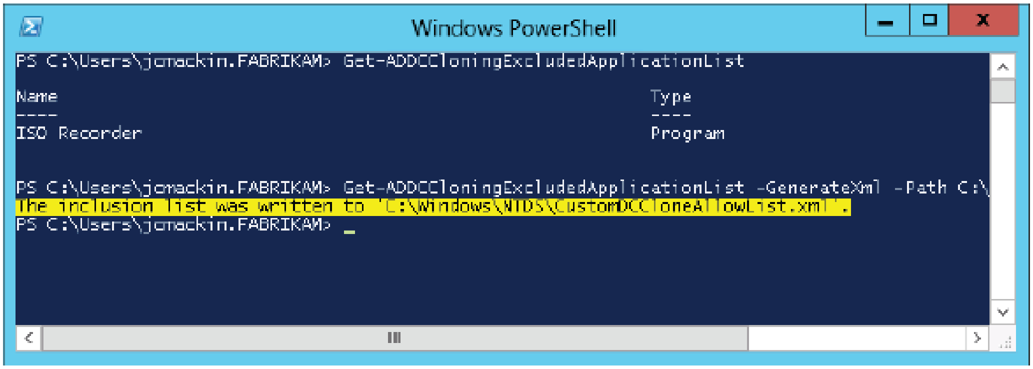
FIGURE 8-2 Output of Get-ADDCCloningExcludedApplicationList revealing no unsafe applications.

FIGURE 8-3 Output of Get-ADDCCloningExcludedApplicationList revealing a potentially unsafe application.

If the cmdlet returns a list of services and installed programs, review the list. Consult the software vendor of each application to determine whether it can be safely cloned. If any applications or services in the list cannot be safely cloned, you must uninstall them from the source domain controller at this point.

Next, if you determine that the services or applications returned by the Get-ADDCCloningExcludedApplicationList cmdlet are safe for cloning, you can add them to an inclusion list called CustomDCCloneAllowList.xml. To do so, use the -GenerateXml option with the same cmdlet. For example, the following command generates the excluded application list as a file named CustomDCCloneAllowList.xml at the specified folder path (C:\Windows\NTDS) and forces overwrite if a file by that name is found to already exist at that path location:

The output of this command is shown in Figure 8-4, and the contents of the CustomDCCloneAllowList.xml file are shown in Figure 8-5.

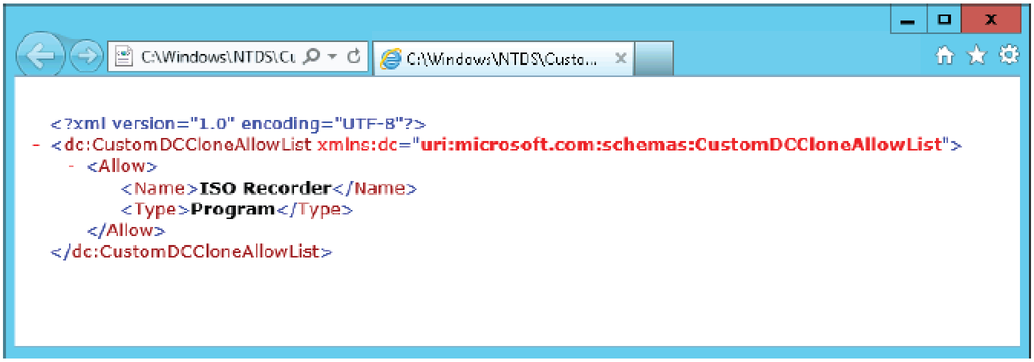


```
PS C:\Users\jcmackin.FABRIKAM> Get-ADDCCloningExcludedApplicationList

Name                                                        Type
----                                                        ----
ISO Recorder                                                Program


PS C:\Users\jcmackin.FABRIKAM> Get-ADDCCloningExcludedApplicationList -GenerateXml -Path C:\
The inclusion list was written to 'C:\Windows\NTDS\CustomDCCloneAllowList.xml'.
PS C:\Users\jcmackin.FABRIKAM> _
```

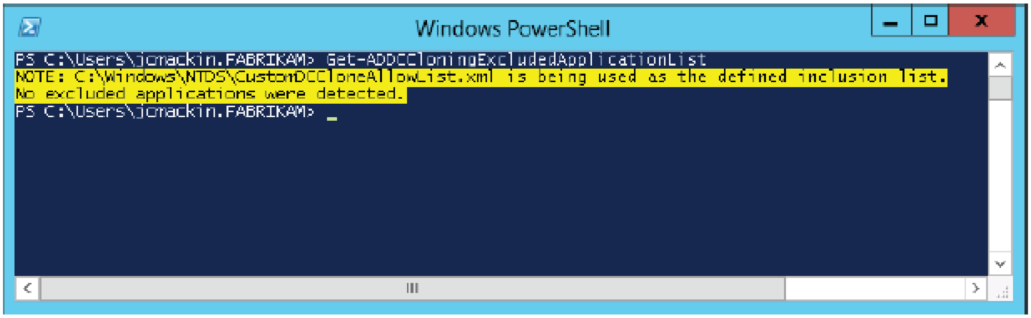
FIGURE 8-4 Adding detected applications to the inclusion list.



```
  <?xml version="1.0" encoding="UTF-8"?>
- <dc:CustomDCCloneAllowList xmlns:dc="uri:microsoft.com:schemas:CustomDCCloneAllowList">
  - <Allow>
        <Name>ISO Recorder</Name>
        <Type>Program</Type>
    </Allow>
  </dc:CustomDCCloneAllowList>
```

FIGURE 8-5 The CustomDCCloneAllowList.xml file.

After you perform this step, the Get-ADDCCloningExcludedApplicationList cmdlet will provide the output shown in Figure 8-6.



```
PS C:\Users\jcmackin.FABRIKAM> Get-ADDCCloningExcludedApplicationList
NOTE: C:\Windows\NTDS\CustomDCCloneAllowList.xml is being used as the defined inclusion list.
No excluded applications were detected.
PS C:\Users\jcmackin.FABRIKAM> _
```

FIGURE 8-6 The output of Get-ADDCCloningExcludedApplicationList after adding a detected application to the inclusion list.

Note that if any programs originally returned by the Get-ADDCCloningExcludedApplicationList cmdlet are not added to the inclusion list (CustomDCCloneAllowList.xml), the next step will fail.

For more information about the Get-ADDCCloningExcludedApplicationList cmdlet, visit *http://technet.microsoft.com/en-us/library/hh852291*.

Run the New-ADDCCloneConfigFile cmdlet on the source VM

The New-ADDCCloneConfigFile cmdlet runs a list of prerequisite checks on the source VM and generates a clone configuration file, DCCloneConfig.xml, if all the checks succeed. The clone configuration file includes settings that you have specified for the new clone VM, such as an IP address configuration and computer name, by using parameters with the cmdlet. If the command runs successfully, the DCCloneConfig.xml file is saved in a location that will automatically configure the clone with these settings when you later start the clone for the first time, so you don't need to look for the file or move it from its default location. (If you don't specify a name for the clone, one will be chosen automatically. If you don't specify a static IP

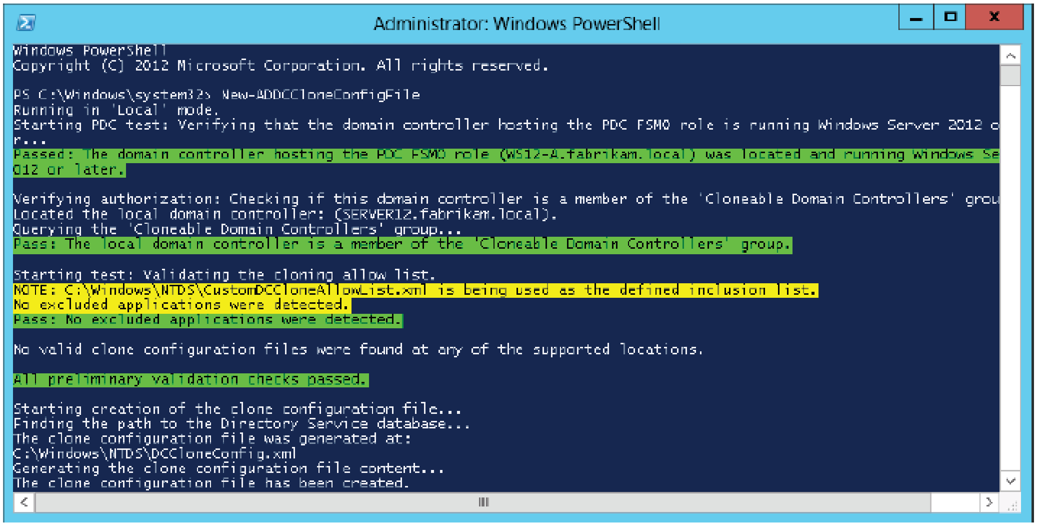
configuration, it will be set dynamically.)
The checks succeed if the answers to the following three questions are all “yes”:

□ Is the PDC Emulator operations master role hosted on a domain controller running Windows Server 2012?

□ Is this computer a member of the Cloneable Domain Controllers group?

□ Are all programs and services originally listed in the output of the Get-ADDCCloningExcludedApplicationList cmdlet now captured in CustomDCCloneAllowList.xml?

A successful check of a source domain controller is shown in Figure 8-7.



```
Windows PowerShell
Copyright (C) 2012 Microsoft Corporation. All rights reserved.

PS C:\Windows\system32> New-ADDCCloneConfigFile
Running in 'Local' mode.
Starting PDC test: Verifying that the domain controller hosting the PDC FSMO role is running Windows Server 2012 o
r...
Passed: The domain controller hosting the PDC FSMO role (WS12-A.fabrikam.local) was located and running Windows Se
012 or later.

Verifying authorization: Checking if this domain controller is a member of the 'Cloneable Domain Controllers' grou
Located the local domain controller: (SERVER12.fabrikam.local).
Querying the 'Cloneable Domain Controllers' group...
Pass: The local domain controller is a member of the 'Cloneable Domain Controllers' group.

Starting test: Validating the cloning allow list.
NOTE: C:\Windows\NTDS\CustomDCCloneAllowList.xml is being used as the defined inclusion list.
No excluded applications were detected.
Pass: No excluded applications were detected.

No valid clone configuration files were found at any of the supported locations.

All preliminary validation checks passed.

Starting creation of the clone configuration file...
Finding the path to the Directory Service database...
The clone configuration file was generated at:
C:\Windows\NTDS\DCCloneConfig.xml
Generating the clone configuration file content...
The clone configuration file has been created.
```

FIGURE 8-7 The output of the New-ADDCCloneConfigFile cmdlet.

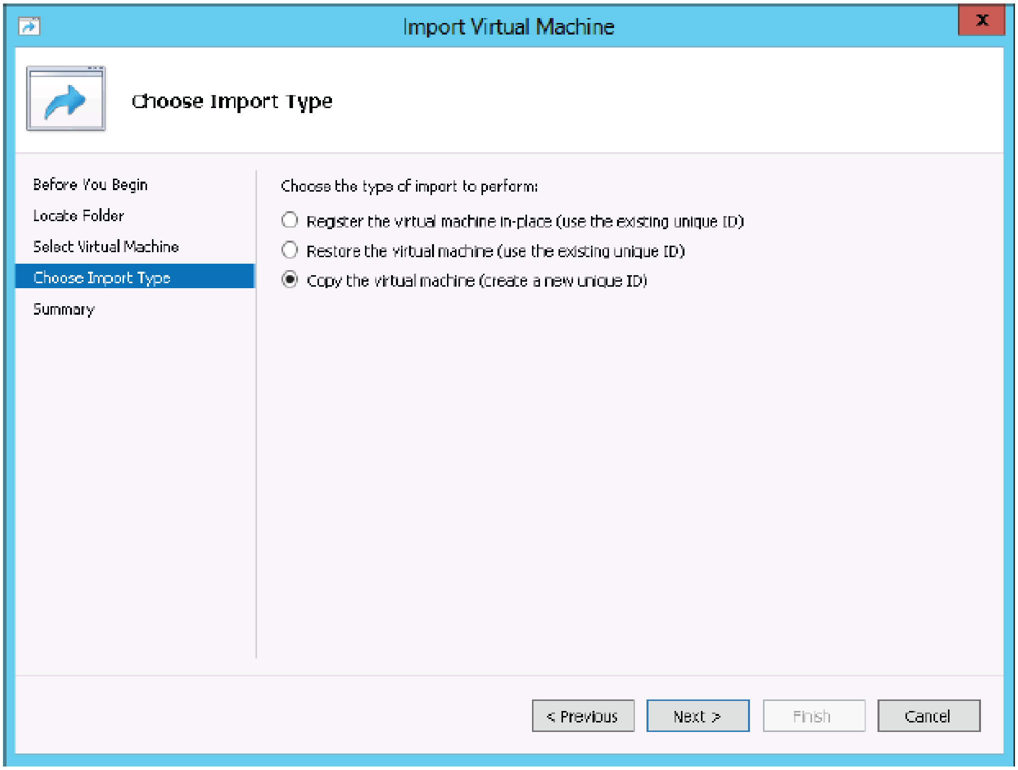
For more information about the New-ADDCCloneConfigFile cmdlet, visit *http://technet.microsoft.com/en-us/library/jj158947*.

Export and then import the VM of the source domain controller

To clone the VM, first shut it down. Then, you can use the Export command in Hyper-V Manager to copy the VM files to a location you choose. To export the VM using Windows PowerShell instead, use the Export-VM cmdlet as in the following example:

At this point, *you must delete all the snapshots in the Snapshots subdirectory of the exported VM*. If desired, you can then copy the exported VM and its associated files to another computer running Windows Server 2012 that has the Hyper-V role installed.

Next, you can use the Import command in Hyper-V Manager to import the exported VM. Use the Copy The Virtual Machine (Create A New Unique ID) option when importing the VM, as shown in Figure 8-8. To perform this step in Windows PowerShell, use the Import-VM cmdlet as in the following example: Finally, after importing the copy of the source VM, you can restart the source VM, and start the new clone VM.



Import Virtual Machine

Choose Import Type

Choose the type of import to perform:

○ Register the virtual machine in-place (use the existing unique ID)

○ Restore the virtual machine (use the existing unique ID)

● Copy the virtual machine (create a new unique ID)

FIGURE 8-8 Creating a unique ID when importing an image allows you to use the source image again.

Remember for the 70-411 exam that you need to delete snapshots before importing and that you need to choose the Copy The Virtual Machine (Create New Unique ID) option when you import the VM.

Objective summary

□ Windows Server 2012 allows you to clone a virtualized domain controller for rapid deployment of a new domain controller.

□ The source VM must be a member of the Cloneable Domain Controllers global security group.

□ Three computers must be running Windows Server 2012: the host server running Hyper-V, the guest VM that is the domain controller to be cloned, and a third domain controller that owns the PDC Emulator operations master role for the domain.

□ You need to use the Get-ADDCCloningExcludedApplicationList cmdlet to determine whether any applications on the source domain controller have not yet been determined to be safe for cloning. You must either uninstall such applications or add them to the inclusion list by running the same cmdlet again with the -GenerateXml switch.

□ Next, run the New-ADDCCloneConfigFile cmdlet to run prerequisite checks to determine whether the domain controller is ready to be cloned.

□ When the domain controller passes the prerequisite checks, use Hyper-V Manager or the Export-VM and Import-VM cmdlets to copy the VM. Be sure to delete the snapshots of the exported VM before you import. When importing, choose the option to copy the VM and create a new unique ID.

Objective review

Answer the following questions to test your knowledge of the information in this objective. You can find the answers to these questions and explanations of why each answer choice is correct or incorrect in the “Answers” section at the end of the chapter.

1. You are a network administrator for Fabrikam.com. The Fabrikam.com network includes a private cloud built on six physical servers with the Hyper-V role installed, three of which are running Windows Server 2012, and three of which are running Windows Server 2008 R2. These physical servers host a total of 24 virtualized guest servers, including domain controllers, infrastructure servers, database servers, and application servers. All VMs are members of the Fabrikam.com domain.

One of the virtualized domain controllers is named DC1. DC1 is hosted on a physical server named HYPV1. You want to clone DC1 to add another domain controller to the Fabrikam.com domain.

Which of the following is NOT a requirement for cloning DC1?

A. DC1 must own the PDC emulator role.

B. HYPV1 must be running Windows Server 2012.

C. DC1 must be running Windows Server 2012.

D. All programs and services originally listed in the output of the Get-ADDCCloningExcludedApplicationList cmdlet when it is run on DC1 must be added to the Allow list in CustomDCCloneAllowList.xml.

2. You want to clone a domain controller named VDC1 that is running Windows Server 2012 within
a VM. VDC1 is hosted on a server named Host01, which is also running Windows Server 2012.

VDC1 belongs to the Contoso.local domain and owns both the PDC Emulator and Infrastructure Master operations master roles for that domain. Contoso.local currently includes just one other domain controller, named VDC2. VDC2 is running Windows Server 2008 R2 and is the only global catalog server in Contoso.local.
You want to create a new domain controller named DC3 that is based on a clone of VDC1. Which steps do you need to take before you can achieve this? (Choose all that apply.)
A. Move the Infrastructure Master operations master role to VDC2.

B. Make VDC1 a global catalog server.

C. Upgrade VDC2 to Windows Server 2012.

D. Make Host01 a member of the Cloneable Domain Controllers global security group.

E. Make VDC1 a member of the Cloneable Domain Controllers global security group.

3. You want to clone a domain controller named DCA.fabrikam.local that is running Windows Server 2012 in a VM. DCA is hosted on a server named HV01, which is also running Windows Server 2012.

When you run the cmdlet Get-ADDCCloningExcludedApplicationList on DCA, the output displays the name of a single application, App1. You want to ensure that App1 is made available on all future domain controllers that result from cloning DCA. You have verified that App1 is safe for cloning.
What should you do next?
A. Export the DCA VM.

B. Add App1 to the CustomDCCloneAllowList.xml file.

C. Run the New-ADDCCloneConfigFile cmdlet.

D. Run the New-VirtualDiskClone cmdlet