Windows Server 2012

Test Lab Guide: Windows Server 2012 R2 Base Configuration

Microsoft Corporation

Abstract

This Microsoft Test Lab Guide (TLG) provides you with step-by-step instructions to create the Windows Base Configuration test lab, using computers running Windows 8.1 or Windows Server 2012 R2. With the resulting test lab environment, you can build test labs based on other Windows Server 2012 R2-based TLGs from Microsoft, TLG extensions in the TechNet Wiki, or a test lab of your own design that can include Microsoft or non-Microsoft products. For a test lab based on physical computers, you can image the drives for future test labs. For a test lab based on virtual machines, you can create snapshots of the base configuration virtual machines. This enables you to easily return to the base configuration test lab, where most of the routine infrastructure and networking services have already been configured, so that you can focus on building a test lab for the product, technology, or solution of interest.

Microsoft

Copyright Information

This document is provided for informational purposes only and Microsoft makes no warranties, either express or implied, in this document. Information in this document, including URL and other Internet Web site references, is subject to change without notice. The entire risk of the use or the results from the use of this document remains with the user. Unless otherwise noted, the example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted herein are fictitious, and no association with any real company, organization, product, domain name, e-mail address, logo, person, place, or event is intended or should be inferred. Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

© 2013 Microsoft Corporation. All rights reserved.
Date of last update: 12/6/2013
Microsoft, Windows, Active Directory, Internet Explorer, and Windows Server are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

All other trademarks are property of their respective owners.

Contents

Introduction

Test Lab Guides (TLGs) allow you to get hands-on experience with new products and technologies using a pre-defined and tested methodology that results in a working configuration. When you use a TLG to create a test lab, instructions tell you what servers to create, how to configure the operating systems and platform services, and how to install and configure any additional products or technologies. A TLG experience enables you to see all of the components and the configuration steps on both the front-end and back-end that go into a single- or multi-product or technology solution.

A challenge in creating useful TLGs is to enable their reusability and extensibility. Because creating a test lab can represent a significant investment of time and resources, your ability to reuse and extend the work required to create test labs is important. An ideal test lab environment would enable you to create a basic lab configuration, save that configuration, and then build out multiple test labs in the future by starting with that basic configuration.

The purpose of this TLG is to enable you to create the Windows Server 2012 R2 Base Configuration test lab, upon which you can build a test lab based on other Windows Server 2012 R2 -based TLGs from Microsoft, TLG extensions in the TechNet Wiki, or a test lab of your own design that can include Microsoft or non-Microsoft products.

Depending on how you deploy your test lab environment, you can image the drives for the Windows Server 2012 R2 Base Configuration test lab if you are using physical computers or you can create snapshots of the Base Configuration test lab virtual machines. This enables you to easily return to baseline configuration where most of the routine client, server, and networking services have already been configured so that you can focus on building out a test lab for the products or technologies of interest. For this reason, make sure that you perform a disk image on each computer if you're using physical computers, or perform virtual machine snapshots if you are using virtual machines after completing all the steps in this TLG.

The Windows Server 2012 R2 Base Configuration TLG is just the beginning of the test lab experience. Other Windows Server 2012 R2-based TLGs or TLG extensions in the TechNet Wiki focus on Microsoft products or platform technologies, but all of them use this Windows Server 2012 R2 Base Configuration TLG as a starting point.

In this guide

This document contains instructions for setting up the Windows Server 2012 R2 Base Configuration test lab by deploying four server computers running Windows Server 2012 R2 and one client computer running Windows 8.1. The resulting configuration simulates a private intranet and the Internet.

Important

The following instructions are for configuring the Windows Server 2012 R2 Base Configuration test lab. Individual computers are needed to separate the services provided on the network and to clearly show the desired functionality. This

configuration is neither designed to reflect best practices nor does it reflect a desired or recommended configuration for a production network. The configuration, including IP addresses and all other configuration parameters, is designed only to work on a separate test lab network.

Note: If you are able to work from a computer-based copy of this document during the lab exercises, and you are running virtual machines in Hyper-V, leverage the Hyper-V clipboard integration feature to paste commands. This will minimize potential errors with mistyped command strings.

- Highlight and right-click a command from this document listed in **bold** text.
- Click **Copy**.
- From the virtual machine menu bar, click **Clipboard**, and then click **Type clipboard text**.

Test lab overview

The Windows Server 2012 R2 Base Configuration test lab consists of the following:

- One computer running Windows Server 2012 R2 named DC1 that is configured as an intranet domain controller, Domain Name System (DNS) server, and Dynamic Host Configuration Protocol (DHCP) server.
- One intranet member server running Windows Server 2012 R2 named APP1 that is configured as a general application and web server.
- One member client computer running Windows 8.1 named CLIENT1 that will switch between Internet and intranet subnets.
- One intranet member server running Windows Server 2012 R2 named EDGE1 that is configured as an Internet edge server.
- One standalone server running Windows Server 2012 R2 named INET1 that is configured as an Internet DNS server, web server, and DHCP server.

The Windows Server 2012 R2 Base Configuration test lab consists of two subnets that simulate the following:

- The Internet, referred to as the Internet subnet (131.107.0.0/24).
- An intranet, referred to as the Corpnet subnet (10.0.0.0/24), separated from the Internet subnet by EDGE1.

Computers on each subnet connect using a physical hub, switch, or virtual switch. See the following figure for the configuration of the Windows Server 2012 R2 Base Configuration test lab.

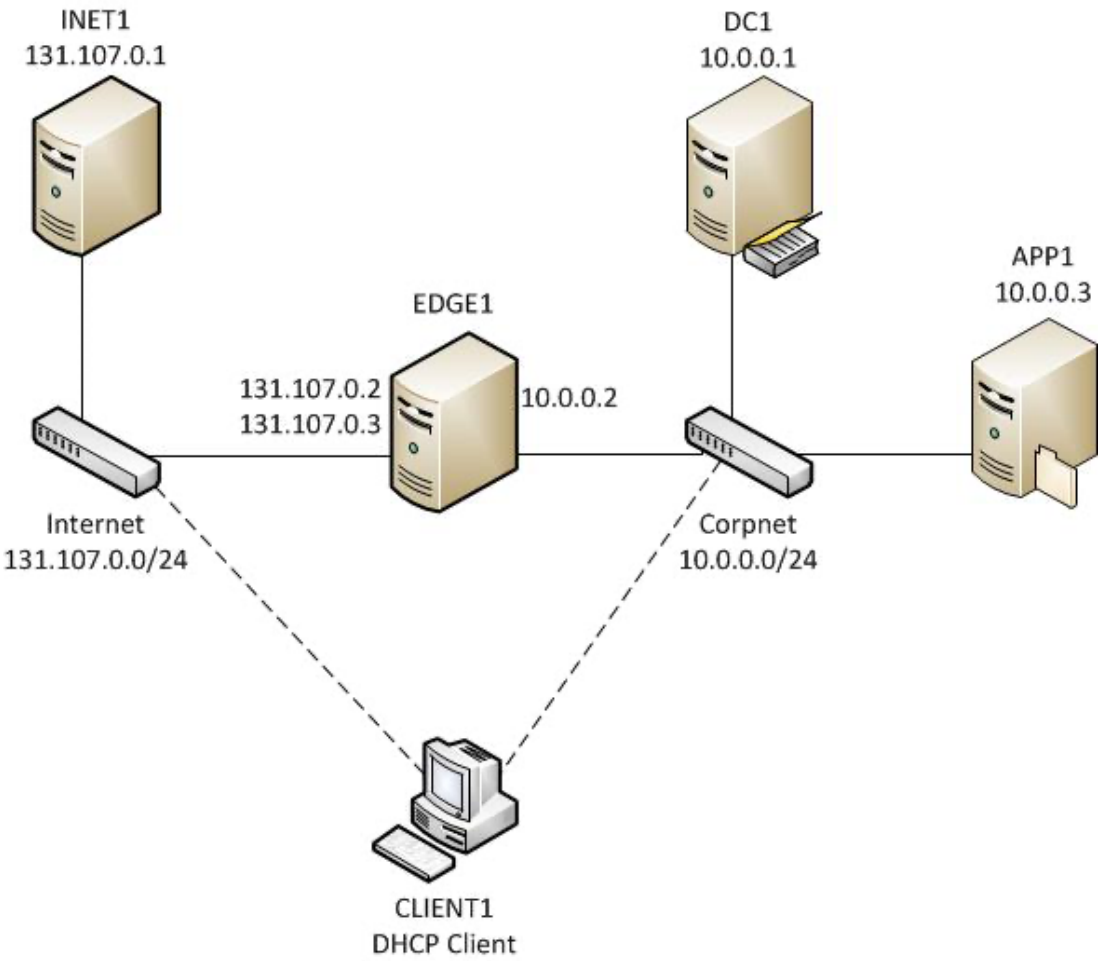

Figure 1 Windows Server 2012 R2 Base Configuration

This document describes how to build out the Windows Server 2012 R2 Base Configuration test lab in two sections:

- Steps for configuring the Corpnet subnet (DC1, APP1, and CLIENT1)
- Steps for configuring the Internet subnet (EDGE1 and INET1)

There are some TLGs that require only the Corpnet subnet. However, it is strongly recommended that you build out both subnets if you ever plan to test technologies, products, or solutions that include access to Corpnet servers and services from the Internet. The Windows Server 2012 R2 Base Configuration test lab environment consisting of both subnets can be saved and reused for other TLGs. By building out both the Corpnet and Internet subnets, you will have a reusable snapshot of the entire Windows Server 2012 R2 Base Configuration test lab that can be used for many TLGs, which has the starting Windows Server 2012 R2 Base Configuration test lab in a unified and consistent state.

Hardware and software requirements

The following are the minimum required components of the test lab:

- The product disc or files for Windows Server 2012 R2.
- The product disc or files for Windows 8.1.
- Four computers that meet the minimum hardware requirements for Windows Server 2012 R2. One of these computers (EDGE1) has two network adapters installed.
- One computer that meets the minimum hardware requirements for Windows 8.1.
- If you wish to deploy the Base Configuration test lab in a virtualized environment, your virtualization solution must support Windows Server 2012 R2 64-bit virtual machines. The server hardware must support the amount of RAM required to run the virtual operating systems included in the Base Configuration test lab and any other virtual machines that may be required by additional TLGs.

Important
Run Windows Update on all computers or virtual machines either during the installation or immediately after installing the operating systems. After running Windows Update, you can isolate your physical or virtual test lab from your production network.

User account control

When you configure the Windows 2012 R2 or Windows 8.1 operating system, you are required to click **Continue** or **Yes** in the **User Account Control** (UAC) dialog box for some tasks. Several of the configuration tasks require UAC approval. When you are prompted, always click **Continue** or **Yes** to authorize these changes. Alternatively, see the Appendix of this guide for instructions about how to set the UAC behavior of the elevation prompt for administrators.

Steps for Configuring the Corpnet Subnet

There are 3 steps to setting up the Corpnet subnet of the Windows Server 2012 R2 Base Configuration test lab.

1. Configure DC1.
2. Configure APP1.
3. Configure CLIENT1.

Note
You must be logged on as a member of the Domain Admins group or a member of the local Administrators group on each computer to complete the tasks described in this guide.

The following sections provide details about how to perform these steps.

Step 1: Configure DC1

DC1 provides the following services:

- A domain controller for the corp.contoso.com Active Directory Domain Services (AD DS) domain
- A DNS server for the corp.contoso.com DNS domain
- A DHCP server for the Corpnet subnet

DC1 configuration consists of the following:

- Install the operating system
- Configure TCP/IP
- Install Active Directory and DNS
- Install DHCP
- Create a user account in Active Directory

Install the operating system on DC1

First, install Windows Server 2012 R2 as a standalone server.

To install the operating system on DC1

1. Start the installation of Windows Server 2012 R2.
2. Follow the instructions to complete the installation, specifying a strong password for the local Administrator account. Log on using the local Administrator account.
3. Connect DC1 to a network that has Internet access and run Windows Update to install the latest updates for Windows Server 2012 R2.
4. Connect DC1 to the Corpnet subnet.

Configure TCP/IP properties on DC1

Next, configure the TCP/IP protocol with a static IP address of 10.0.0.1 and the subnet mask of 255.255.255.0.

Do this step using Windows PowerShell

To configure TCP/IP on DC1

1. In Server Manager, click **Local Server** in the console tree. Click the link next to **Ethernet**.

   Note

   The link may not immediately appear. Wait for the network interfaces to be enumerated.
2. In **Network Connections**, right-click **Ethernet**, and then click **Properties**. Note that the "Ethernet" interface name may be different on your computer.
3. Click **Internet Protocol Version 4 (TCP/IPv4)**, and then click **Properties**.
4. Select **Use the following IP address**. In **IP address**, type **10.0.0.1**. In **Subnet mask**, type **255.255.255.0**. Select **Use the following DNS server addresses. In Preferred DNS server, type 127.0.0.1.**
5. Click **OK** and then close the Ethernet Properties dialog.
6. Close the **Network Connections** window.
7. From the **Tools** menu in Server Manager, click **Windows PowerShell**.
8. To configure the firewall to allow ICMPv4 ping packets, type the following commands and press ENTER after each command.
   New-NetFirewallRule –DisplayName “Allow ICMPv4-In” –Protocol ICMPv4
   New-NetFirewallRule –DisplayName “Allow ICMPv4-Out” –Protocol ICMPv4 –Direction Outbound
9. Close the Windows PowerShell window.
10. In Server Manager, click **Local Server** in the console tree. Click the link next to **Computer name** in the Properties tile.
11. On the **Computer Name** tab of the System Properties dialog, click **Change**.
12. In **Computer name**, type **DC1**, click **OK** twice, and then click **Close**. When you are prompted to restart the computer, click **Restart Now**.
13. After restarting, login using the local Administrator account.

Windows PowerShell equivalent commands

The following Windows PowerShell cmdlet or cmdlets perform the same function as the preceding procedure. Enter each cmdlet on a single line, even though they may appear word-wrapped across several lines here because of formatting constraints. Note that the "Ethernet" interface name may be different on your computer. Use the **ipconfig /all** command to list all the interfaces.

```
New-NetIPAddress -InterfaceAlias Ethernet -IPAddress 10.0.0.1 -AddressFamily IPv4 -PrefixLength 24
Set-DnsClientServerAddress -InterfaceAlias Ethernet -ServerAddresses 127.0.0.1
New-NetFirewallRule –DisplayName “Allow ICMPv4-In” –Protocol ICMPv4
New-NetFirewallRule –DisplayName “Allow ICMPv4-Out” –Protocol ICMPv4 –Direction Outbound
Rename-Computer DC1
Restart-Computer
```

Configure DC1 as a domain controller and DNS server

Next, configure DC1 as a domain controller and DNS server for the corp.contoso.com domain.
Do this step using Windows PowerShell

To configure DC1 as a domain controller and DNS server

1. Launch **Server Manager**.
2. On the **Dashboard** screen, under **Configure this local server**, click **Add roles and features**.
3. Click **Next** three times to get to the server role selection screen.
4. In the **Select Server Roles** dialog, select **Active Directory Domain Services**. Click **Add Features** when prompted, and then click **Next**.
5. In the **Select features** dialog, click **Next**.
6. In the **Active Directory Domain Services** dialog, click **Next**.
7. In the **Confirm installation selections** dialog, click **Install.** Wait for the installation to complete.
8. In the **Installation progress** dialog, click the **Promote this server to a domain controller** link.

   Note: If you close the "Installation progress" dialog before it presents the promotion link, click the gray **Tasks** flag in the upper right section of **Server Manager.** When the installation is complete you will see the **Promote this server to a Domain Controller** link.
9. In the **Deployment Configuration** dialog, select **Add a new forest**. In the **Root domain name** field, type **corp.contoso.com**. Click **Next**.
10. In the **Domain Controller Options** dialog, leave the default values, specify a strong DSRM password twice, and then click **Next** four times to accept default settings for DNS, NetBIOS, and directory paths.
11. In the **Review Options** dialog, review your selections and then click **Next**.

    Note: You can also click the **View script** button to review and save the Windows PowerShell commands that Server Manager will run during DC Promotion.
12. In the **Prerequisites Check** dialog, allow the validation to complete and verify that no errors are reported. Since this is the first DNS server deployment in the forest, you can safely ignore all warnings regarding DNS delegation. Click **Install** to start the domain controller promotion. Allow the installation to complete.
13. Allow the domain controller to restart. After the server restarts, logon using the CORP\Administrator credentials.

Windows PowerShell equivalent commands

The following Windows PowerShell cmdlet or cmdlets perform the same function as the preceding procedure. Enter each cmdlet on a single line, even though they may appear word-wrapped across several lines here because of formatting constraints.

```
Install-WindowsFeature AD-Domain-Services -IncludeManagementTools
Install-ADDSForest -DomainName corp.contoso.com
```

Install and configure DHCP on DC1

Next, configure DC1 as a DHCP server so that CLIENT1 can automatically configure itself when it connects to the Corpnet subnet.

Do this step using Windows PowerShell

To install and configure the DHCP server role on DC1

1. In the **Dashboard** console of Server Manager, under **Configure this local server**, click **Add roles and features**.
2. Click **Next** three times to get to the server role selection screen.
3. In the **Select server roles** dialog, select **DHCP Server**, click **Add Features** when prompted, and then click **Next**.
4. In the **Select features** dialog, click **Next**.
5. Click **Next** on the DHCP Server screen, and then click **Install**.
6. Allow the installation to complete, and then in the Results window, click the link for **Complete DHCP configuration**.

   Note: If you close the "Installation progress" dialog before it presents the **Complete DHCP configuration** link, click the gray **Tasks** flag in the upper right section of **Server Manager.** When the installation is complete you will see the **Complete DHCP configuration** link.
7. In the DHCP Post-Install configuration wizard, click **Next**, and then click **Commit**.
8. On the Summary page, click **Close**.
9. In the Add Roles and Features Wizard, click **Close**.
10. From the **Tools** menu in Server Manager, click **DHCP**.
11. In the DHCP console tree, expand **dc1.corp.contoso.com**, and click **IPv4**. Right-click **IPv4**, and click **New Scope**.
12. Click **Next** in the New Scope Wizard.
13. Type **Corpnet** for scope name, and then click **Next**.
14. Next to **Start IP Address**, type **10.0.0.100**, next to **End IP Address**, type **10.0.0.200**, and next to **Subnet Mask**, type **255.255.255.0**.
15. Click **Next** eight times to accept all scope option default settings, and then click **Finish**.
16. Close the DHCP Manager console.

Windows PowerShell equivalent commands

The following Windows PowerShell cmdlet or cmdlets perform the same function as the preceding procedure. Enter each cmdlet on a single line, even though they may appear word-wrapped across several lines here because of formatting constraints.

```
Install-WindowsFeature DHCP -IncludeManagementTools
Netsh DHCP Add SecurityGroups
Restart-Service DhcpServer
Add-DhcpServerInDC -DnsName dc1.corp.contoso.com
Add-DhcpServerv4Scope -name "Corpnet" -StartRange 10.0.0.100 -EndRange 10.0.0.200 -SubnetMask 255.255.255.0
Set-DhcpServerv4OptionValue -DnsDomain corp.contoso.com -DnsServer 10.0.0.1
```

Create a user account in Active Directory on DC1

Next, create a user account in Active Directory that will be used when logging in to CORP domain member computers.

Do this step using Windows PowerShell

To create a user account in Active Directory

1. From the **Tools** menu in Server Manager, click **Active Directory Administrative Center**.
2. In the console tree, click the arrow to expand **corp (local)**, and then double-click **Users**. This adds Users as a recent navigation link in the console tree.
3. In the **Tasks** pane, click **New**, and then click **User**.
4. In the Create User dialog, type **User1** next to **Full name** and type **User1** next to **User SamAccountName logon: corp** (both required fields indicated by the red asterisk icon).
5. In **Password**, type the password that you want to use for this account, and in **Confirm password**, type the password again.
6. Under Password options, select **Other password options**, and select **Password never expires**.
7. Scroll down to access the **Member of** section of the Create User dialog, and click **Add**. Type **Domain Admins; Enterprise Admins**, and then click **OK**.
8. Click **OK** to close the Create User dialog.
9. Exit the Active Directory Administrative Center.
10. Sign out of DC1 as the Administrator user (right-click the **Start** icon, point to **Shut down or sign out**, and then click **Sign out**).
11. Sign in using the **User1** account.

Windows PowerShell equivalent commands

The following Windows PowerShell cmdlet or cmdlets perform the same function as the preceding procedure. Enter each cmdlet on a single line, even though they may appear word-wrapped across several lines here because of formatting constraints. Note that the first command results in a prompt to supply the user password.

```
New-ADUser -SamAccountName User1 -AccountPassword (read-host "Set user password" -
assecurestring) -name "User1" -enabled $true -PasswordNeverExpires $true -
ChangePasswordAtLogon $false

Add-ADPrincipalGroupMembership -Identity
"CN=User1,CN=Users,DC=corp,DC=contoso,DC=com" -MemberOf "CN=Enterprise
Admins,CN=Users,DC=corp,DC=contoso,DC=com","CN=Domain
Admins,CN=Users,DC=corp,DC=contoso,DC=com"
```

Step 2: Configure APP1

APP1 provides web and file sharing services. APP1 configuration consists of the following:

- Install the operating system.
- Configure TCP/IP.
- Join the computer to the domain.
- Install the Web Server (IIS) role.
- Create a shared folder.

Install the operating system on APP1

To install the operating system on APP1

1. Start the installation of Windows Server 2012 R2.
2. Follow the instructions to complete the installation, specifying a strong password for the local Administrator account. Log on using the local Administrator account.
3. Connect APP1 to a network that has Internet access and run Windows Update to install the latest updates for Windows Server 2012 R2.
4. Connect APP1 to the Corpnet subnet.

Configure TCP/IP properties on APP1

Do this step using Windows PowerShell

To configure TCP/IP properties on APP1

1. In Server Manager, click **Local Server** in the console tree. Click the link next to **Ethernet** in the Properties tile.
2. In **Network Connections**, right-click **Ethernet**, and then click **Properties**. Note that the "Ethernet" interface name may be different on your computer.
3. Click **Internet Protocol Version 4 (TCP/IPv4)**, and then click **Properties**.
4. Select **Use the following IP address**. In **IP address**, type **10.0.0.3**. In **Subnet mask**, type **255.255.255.0**.
5. Select **Use the following DNS server addresses**. In **Preferred DNS server**, type **10.0.0.1**.
6. Click **OK**, and then close the **Ethernet Properties** window.
7. Close the **Network Connections** window.
8. From the **Tools** menu in Server Manager, click **Windows PowerShell**.
9. To configure the firewall to allow ICMPv4 ping packets, type the following commands and press ENTER after each command.
   New-NetFirewallRule –DisplayName "Allow ICMPv4-In" –Protocol ICMPv4
   New-NetFirewallRule –DisplayName "Allow ICMPv4-Out" –Protocol ICMPv4 –Direction Outbound
10. To check name resolution and network communication between APP1 and DC1, type **ping dc1.corp.contoso.com** in the Windows PowerShell window and press ENTER.
11. Verify that there are four replies from 10.0.0.1.
12. Close the Windows PowerShell window.

Windows PowerShell equivalent commands

The following Windows PowerShell cmdlet or cmdlets perform the same function as the preceding procedure. Enter each cmdlet on a single line, even though they may appear word-wrapped across several lines here because of formatting constraints. Note that the "Ethernet" interface name may be different on your computer. Use **ipconfig /all** to list out the interfaces.

```
New-NetIPAddress -InterfaceAlias Ethernet -IPAddress 10.0.0.3 -AddressFamily IPv4 -
PrefixLength 24
Set-DnsClientServerAddress -InterfaceAlias Ethernet -ServerAddresses 10.0.0.1
New-NetFirewallRule –DisplayName "Allow ICMPv4-In" –Protocol ICMPv4
New-NetFirewallRule –DisplayName "Allow ICMPv4-Out" –Protocol ICMPv4 –Direction
Outbound
```

Join APP1 to the CORP domain

Do this step using Windows PowerShell

To join APP1 to the CORP domain

1. In Server Manager, click **Local Server** in the console tree. Click the link next to **Computer name** in the Properties tile.
2. In the **System Properties** dialog box, click the **Computer Name** tab. On the

Computer Name tab, click **Change**.

3. In **Computer Name**, type **APP1**. Under **Member of**, click **Domain**, and then type **corp.contoso.com**.
4. Click **OK**.
5. When you are prompted for a user name and password, type **User1** and its password, and then click **OK**.
6. When you see a dialog box welcoming you to the corp.contoso.com domain, click **OK**.
7. When you are prompted that you must restart the computer, click **OK**.
8. On the **System Properties** dialog box, click **Close**.
9. When you are prompted to restart the computer, click **Restart Now**.
10. After the computer restarts, click the **Switch User arrow icon**, then click **Other User** and log on to the CORP domain with the **User1** account.

Windows PowerShell equivalent commands

The following Windows PowerShell cmdlet or cmdlets perform the same function as the preceding procedure. Enter each cmdlet on a single line, even though they may appear word-wrapped across several lines here because of formatting constraints. Note that you must supply domain credentials after entering the Add-Computer command below.

```
Add-Computer -NewName APP1 -DomainName corp.contoso.com
Restart-Computer
```

Install the Web Server (IIS) role on APP1

Next, install the Web Server (IIS) role to make APP1 a web server.
Do this step using Windows PowerShell

To install the Web Server (IIS) server role

1. In the **Dashboard** console of Server Manager, click **Add roles and features**.
2. Click **Next** three times to get to the server role selection screen.
3. In the **Select Server Roles** dialog, select **Web Server (IIS)**, click **Add Features** when prompted, and then click **Next**.
4. Click **Next** three times to accept the default Web Server role settings, and then click **Install**.
5. Allow the installation to complete, and then click **Close**.

Windows PowerShell equivalent commands

The following Windows PowerShell cmdlet or cmdlets perform the same function as the preceding procedure.

Enter each cmdlet on a single line, even though they may appear word-wrapped across several lines here because of formatting constraints.

```
Install-WindowsFeature Web-WebServer -IncludeManagementTools
```

Create a shared folder on APP1

Next, create a shared folder and a text file within the folder.
Do this step using Windows PowerShell

To create a shared folder

1. From the desktop taskbar, click **File Explorer**.
2. Expand **This PC**, and then double-click **Local Disk (C:)**.
3. Right-click in the details pane, point to **New**, and then click **Folder**.
4. Type **Files**, and then press **ENTER**. Leave the **Local Disk** window open.
5. Click the Start icon, and then type **Notepad**. Under the Search results, right-click **Notepad**, and then click **Run as administrator**.
6. In the **Untitled – Notepad** window, type **This is a shared file**.
7. Click **File**, click **Save**, double-click **This PC**, double-click **Local Disk (C:)**, and then double-click the **Files** folder.
8. In **File name**, type **Example.txt**, and then click **Save**. Close the Notepad window.
9. In the **Local Disk** window, right-click the **Files** folder, point to **Share with**, and then click **Specific people**.
10. Click **Share**, and then click **Done**.
11. Close the **Local Disk** window.

Windows PowerShell equivalent commands

The following Windows PowerShell cmdlet or cmdlets perform the same function as the preceding procedure. Enter each cmdlet on a single line, even though they may appear word-wrapped across several lines here because of formatting constraints.

```
New-Item -path c:\Files -type directory
Write-Output "This is a shared file." | out-file c:\Files\example.txt
New-SmbShare -name files -path c:\Files -changeaccess CORP\User1
```

Step 3: Configure CLIENT1

CLIENT1 configuration consists of the following:

- Install the operating system
- Join CLIENT1 to the CORP domain
- Test access to intranet resources on the Corpnet subnet

Install the operating system on CLIENT1

To install the operating system on CLIENT1

1. Start the installation of Windows 8.1.
2. When you are prompted for a PC name, type **CLIENT1**.
3. When you are prompted by the Settings dialog, click **Use express settings**.
4. At the Log on prompt, click **Sign in without a Microsoft account.** Click **Local account**. If CLIENT1 does not have Internet access during setup, you will be prompted to **Create a local account**.
5. When you are prompted for a user name, type **User1**. Type a strong password twice, and type a password hint. Click **Finish**.
6. Connect CLIENT1 to a network that has Internet access and run Windows Update to install the latest updates for Windows 8.1.
7. Connect CLIENT1 to the Corpnet subnet. When prompted to automatically connect to devices on this network, click **Yes**.

Join CLIENT1 to the CORP domain

Do this step using Windows PowerShell

To join CLIENT1 to the CORP domain

1. From the Start screen or the desktop, click the **File Explorer** icon.
2. Right-click the **This PC** icon, and then click **Properties**.
3. On the **System** page, click **Advanced system settings**.
4. In the **System Properties** dialog box, click the **Computer Name** tab. On the **Computer Name** tab, click **Change**.
5. In the **Computer Name/Domain Changes** dialog box, under Member of, click **Domain**, type **corp.contoso.com**, and then click **OK**.
6. When you are prompted for a user name and password, type the user name and password for the User1 domain account, and then click **OK**.
7. When you see a dialog box that welcomes you to the corp.contoso.com domain, click **OK**.
8. When you see a dialog box that prompts you to restart the computer, click **OK**.
9. In the **System Properties** dialog box, click **Close**. Click **Restart Now** when prompted.
10. After the computer restarts, click the **Switch User arrow icon**, and then click **Other User**. Log on to the CORP domain with the **User1** account.

Windows PowerShell equivalent commands

The following Windows PowerShell cmdlet or cmdlets perform the same function as the preceding procedure. Enter each cmdlet on a single line, even though they may appear word-wrapped across several lines here because of formatting constraints. Note that you must supply domain credentials after entering the Add-Computer command below.

```
Add-Computer  -DomainName corp.contoso.com
Restart-Computer
```

Test access to resources from the Corpnet subnet

Next, verify that intranet web and file share resources on APP1 can be accessed by CLIENT1.

To test access to resources from CLIENT1

1. From the Start screen or the desktop taskbar, click the **Internet Explorer** icon.
2. In the Address bar, type **http://app1.corp.contoso.com/**, and then press **ENTER**. You should see the default Internet Information Services web page for APP1.
3. From the Start screen or the desktop taskbar, click the **File Explorer** icon.
4. In the address bar, type **\\app1\Files**, and then press **ENTER**.
5. You should see a folder window with the contents of the Files shared folder.
6. In the **Files** shared folder window, double-click the **Example.txt** file. You should see the contents of the Example.txt file.
7. Close the **Example - Notepad** and the **Files** shared folder windows.

Steps for Configuring the Internet Subnet

There are two steps to setting up the Internet subnet of the Windows Server 2012 R2 Base Configuration Test Lab.

1. Configure EDGE1.
2. Configure INET1.

Step 1: Configure EDGE1

EDGE1 configuration consists of the following:

- Install the operating system.
- Configure TCP/IP.
- Join the computer to the domain.

EDGE1 must have two network adapters installed. Connect one adapter to the **Corpnet** physical or virtual switch, and connect the second adapter to the **Internet** physical or virtual switch.

Install the operating system on EDGE1

First, install Windows Server 2012 R2 as a standalone server.

To install the operating system on EDGE1

1. Start the installation of Windows Server 2012 R2.
2. Follow the instructions to complete the installation, specifying a strong password for the local Administrator account. Log on using the local Administrator account.

3. Connect EDGE1 to a network that has Internet access and run Windows Update to install the latest updates for Windows Server 2012 R2.
4. Connect one network adapter to the Corpnet subnet and the other to the Internet subnet.

Configure TCP/IP properties on EDGE1

Configure the TCP/IP protocol with static IP addresses on both interfaces.
Do this step using Windows PowerShell

To configure TCP/IP properties on the Corpnet adapter

1. In Server Manager, click **Local Server** in the console tree. Click the link next to **Ethernet** in the Properties tile.
2. In **Network Connections**, right-click the network connection that is connected to the Corpnet subnet, and then click **Rename**.
3. Type **Corpnet**, and then press **ENTER**.
4. Right-click **Corpnet**, and then click **Properties**.
5. Click **Internet Protocol Version 4 (TCP/IPv4)**, and then click **Properties**.
6. Select **Use the following IP address**. In **IP address**, type **10.0.0.2**. In **Subnet mask**, type **255.255.255.0**.
7. Select **Use the following DNS server addresses**. In **Preferred DNS server**, type **10.0.0.1**.
8. Click **Advanced**, and then the **DNS** tab.
9. In **DNS suffix for this connection**, type **corp.contoso.com**, and then click **OK** three times to close the network properties dialog.
10. In the **Network Connections** window, right-click the network connection that is connected to the Internet subnet, and then click **Rename**.
11. Type **Internet**, and then press **ENTER**.
12. Right-click **Internet**, and then click **Properties**.
13. Click **Internet Protocol Version 4 (TCP/IPv4)**, and then click **Properties**.
14. Select **Use the following IP address**. In **IP address**, type **131.107.0.2**. In **Subnet mask**, type **255.255.255.0**.
15. Select **Use the following DNS server addresses**. In **Preferred DNS server**, type **131.107.0.1**.
16. Click **Advanced**. On the **IP Settings** tab, click **Add** under **IP Addresses**. In the **TCP/IP Address** section, type **131.107.0.3** in **IP address**, type **255.255.255.0** in **Subnet mask**, and then click **Add**.
17. Click the **DNS** tab.
18. In **DNS suffix for this connection**, type **isp.example.com**, and then click **OK** three times to close the network properties dialog.
19. Close the **Network Connections** window.
20. From the **Tools** menu in Server Manager, click **Windows PowerShell**.
21. To configure the firewall to allow ICMPv4 ping packets, type the following commands

and press ENTER after each command.
New-NetFirewallRule –DisplayName "Allow ICMPv4-In" –Protocol ICMPv4
New-NetFirewallRule –DisplayName "Allow ICMPv4-Out" –Protocol ICMPv4 –Direction Outbound

22. To check name resolution and network communication between EDGE1 and DC1, type **ping dc1.corp.contoso.com** in the command prompt window and press ENTER.
23. Verify that there are four responses from 10.0.0.1.
24. Close the Windows PowerShell window.

Windows PowerShell equivalent commands

The following Windows PowerShell cmdlet or cmdlets perform the same function as the preceding procedure. Enter each cmdlet on a single line, even though they may appear word-wrapped across several lines here because of formatting constraints.

Note: Prior to executing these commands, rename the network connections to **Corpnet** and **Internet** according to their associated subnets.

```
New-NetIPAddress -InterfaceAlias "Corpnet" -IPAddress 10.0.0.2 -AddressFamily IPv4 -PrefixLength 24
Set-DnsClientServerAddress -InterfaceAlias "Corpnet" -ServerAddresses 10.0.0.1
Set-DnsClient -InterfaceAlias "Corpnet" -ConnectionSpecificSuffix corp.contoso.com
New-NetIPAddress -InterfaceAlias "Internet" -IPAddress 131.107.0.2 -AddressFamily IPv4 -PrefixLength 24
New-NetIPAddress -InterfaceAlias "Internet" -IPAddress 131.107.0.3 -AddressFamily IPv4 -PrefixLength 24
Set-DnsClientServerAddress -InterfaceAlias "Internet" -ServerAddresses 131.107.0.1
Set-DnsClient -InterfaceAlias "Internet" -ConnectionSpecificSuffix isp.example.com
New-NetFirewallRule –DisplayName "Allow ICMPv4-In" –Protocol ICMPv4
New-NetFirewallRule –DisplayName "Allow ICMPv4-Out" –Protocol ICMPv4 –Direction Outbound
```

Join EDGE1 to the CORP domain

Do this step using Windows PowerShell

To join EDGE1 to the CORP domain

1. In Server Manager, click **Local Server** in the console tree. Click the link next to **Computer name** in the Properties tile.
2. In the **System Properties** dialog box, click the **Computer Name** tab. On the **Computer Name** tab, click **Change**.
3. In **Computer Name**, type **EDGE1**. Under **Member of**, click **Domain**, and then type **corp.contoso.com**.
4. Click **OK**.
5. When you are prompted for a user name and password, type **User1** and its password, and then click **OK**.

6. When you see a dialog box welcoming you to the corp.contoso.com domain, click **OK**.
7. When you are prompted that you must restart the computer, click **OK**.
8. On the **System Properties** dialog box, click **Close**.
9. When you are prompted to restart the computer, click **Restart Now**.
10. After the computer restarts, click the **Switch User arrow icon**, then click **Other User** and log on to the CORP domain with the **User1** account.

Windows PowerShell equivalent commands

The following Windows PowerShell cmdlet or cmdlets perform the same function as the preceding procedure. Enter each cmdlet on a single line, even though they may appear word-wrapped across several lines here because of formatting constraints. Note that you must supply domain credentials after entering the Add-Computer command below.

```
Add-Computer -NewName EDGE1 -DomainName corp.contoso.com
Restart-Computer
```

Step 2: Configure INET1

INET1 configuration consists of the following:

- Install the operating system
- Configure TCP/IP
- Rename the computer
- Install the Web Server (IIS) and DNS server roles
- Create DNS records
- Install DHCP
- Configure the NCSI web site
- Test CLIENT1 access to Internet resources from the Internet subnet

Install the operating system on INET1

To install the operating system on INET1

1. Start the installation of Windows Server 2012 R2.
2. Follow the instructions to complete the installation, specifying a strong password for the local Administrator account. Log on using the local Administrator account.
3. Connect INET1 to a network that has Internet access and run Windows Update to install the latest updates for Windows Server 2012 R2.
4. Connect INET1 to the Internet subnet.

Configure TCP/IP properties on INET1

Do this step using Windows PowerShell

To configure TCP/IP properties on INET1

1. In Server Manager, click **Local Server** in the console tree. Click the link next to **Ethernet** in the Properties tile.
2. In the **Network Connections** window, right-click **Ethernet**, and then click **Properties**.
3. Click **Internet Protocol Version 4 (TCP/IPv4)**, and then click **Properties**.
4. Select **Use the following IP address**. In **IP address**, type **131.107.0.1**. In **Subnet mask**, type **255.255.255.0**. In **Preferred DNS server**, type **127.0.0.1.**
5. Click **Advanced**, and then click the **DNS** tab.
6. In **DNS suffix for this connection**, type **isp.example.com**, and then click **OK**.
7. Click **OK** twice and then close the **Ethernet Properties** dialog box.
8. Close the **Network Connections** window.
9. From the **Tools** menu in Server Manager, click **Windows PowerShell**.
10. To configure the firewall to allow ICMPv4 ping packets, type the following commands and press ENTER after each command.
    New-NetFirewallRule –DisplayName “Allow ICMPv4-In” –Protocol ICMPv4
    New-NetFirewallRule –DisplayName “Allow ICMPv4-Out” –Protocol ICMPv4 –Direction Outbound
11. To verify network connectivity between INET1 and EDGE1, type **ping 131.107.0.2** in the Windows PowerShell window and press ENTER.
12. Verify that there are four replies from 131.107.0.2.
13. Close the Windows PowerShell window.

Windows PowerShell equivalent commands

The following Windows PowerShell cmdlet or cmdlets perform the same function as the preceding procedure. Enter each cmdlet on a single line, even though they may appear word-wrapped across several lines here because of formatting constraints. Note that the "Ethernet" interface name may be different on your computer. Use **ipconfig /all** to list out the interfaces.

```
New-NetIPAddress -InterfaceAlias Ethernet -IPAddress 131.107.0.1 -AddressFamily IPv4 -PrefixLength 24
Set-DnsClientServerAddress -InterfaceAlias Ethernet -ServerAddresses 127.0.0.1
Set-DnsClient -InterfaceAlias Ethernet -ConnectionSpecificSuffix isp.example.com
New-NetFirewallRule –DisplayName “Allow ICMPv4-In” –Protocol ICMPv4
New-NetFirewallRule –DisplayName “Allow ICMPv4-Out” –Protocol ICMPv4 –Direction Outbound
```

Rename the computer to INET1

Do this step using Windows PowerShell

To rename the computer to INET1

1. In Server Manager, click **Local Server** in the console tree. Click the link next to **Computer name** in the Properties tile.
2. In the **System Properties** dialog box, click the **Computer Name** tab. On the **Computer Name** tab, click **Change**.
3. In **Computer Name**, type **INET1**. Click **OK**.
4. When you are prompted that you must restart the computer, click **OK**.
5. On the **System Properties** dialog box, click **Close**.
6. When you are prompted to restart the computer, click **Restart Now**.
7. After the computer restarts, log on with the local administrator account.

Windows PowerShell equivalent commands

The following Windows PowerShell cmdlet or cmdlets perform the same function as the preceding procedure. Enter each cmdlet on a single line, even though they may appear word-wrapped across several lines here because of formatting constraints. Note that the "Ethernet" interface may be different on your computer. Use ipconfig /all to list out the interfaces.

```
Rename-Computer -NewName INET1
Restart-Computer
```

Install the DNS Server and Web Server (IIS) server roles on INET1

Next, install role services for INET1, which will act as an Internet web and DNS server for computers that are connected to the Internet subnet.

Do this step using Windows PowerShell

To install the IIS and DNS server roles

1. On the Server Manager **Dashboard** screen, under **Configure this local server**, click **Add roles and features**.
2. Click **Next** three times to get to the server role selection screen.
3. On the **Select Server Roles** page, select **DNS Server** and click **Add Features** when prompted**.**
4. Select **Web Server (IIS)**, click **Add Features** when prompted, and then click **Next**.
5. Click **Next** four times to accept the default DNS server and web server settings, and then click **Install**.
6. Verify that the installations were successful, and then click **Close**.

Windows PowerShell equivalent commands

The following Windows PowerShell cmdlet or cmdlets perform the same function as the preceding procedure.

Enter each cmdlet on a single line, even though they may appear word-wrapped across several lines here because of formatting constraints.

```
Install-WindowsFeature DNS -IncludeManagementTools
Install-WindowsFeature Web-WebServer -IncludeManagementTools
```

Create DNS records on INET1

Next, create DNS records for the INET1 and EDGE1 IPv4 addresses on the Internet subnet and for the Network Connectivity Status Indicator (NCSI).

Do this step using Windows PowerShell

To create A records

1. From the **Tools** menu in Server Manager, click **DNS**.
2. In the console tree of DNS Manager, expand **INET1**, and click **Forward Lookup Zones**.
3. Right-click **Forward Lookup Zones**, click **New Zone**, and then click **Next**.
4. On the **Zone Type** page, click **Next**.
5. On the **Zone Name** page, type **isp.example.com**, and then click **Next**.
6. Click **Next** twice to accept defaults for zone file and dynamic update, and then click **Finish**.
7. In the console tree, expand **Forward Lookup Zones**, right click **isp.example.com**, and then click **New Host (A or AAAA)**.
8. In **Name**, type **INET1**. In **IP address**, type **131.107.0.1**. Click **Add Host**.
9. Click **OK**, and then click **Done**.
10. In the console tree, right-click **Forward Lookup Zones**, click **New Zone**, and then click **Next**.
11. On the **Zone Type** page, click **Next**.
12. On the **Zone Name** page, type **contoso.com**, and then click **Next**.
13. Click **Next** twice to accept defaults for zone file and dynamic update, and then click **Finish**.
14. In the console tree, right click **contoso.com**, and then click **New Host (A or AAAA)**.
15. In **Name**, type **EDGE1**. In **IP address**, type **131.107.0.2**.
16. Click **Add Host**. Click **OK**, and then click **Done**.
17. In the console tree, right-click **Forward Lookup Zones**, click **New Zone**, and then click **Next**.
18. On the **Zone Type** page, click **Next**.
19. On the **Zone Name** page, type **msftncsi.com**, and then click **Next**.
20. Click **Next** twice to accept defaults for zone file and dynamic update, and then click **Finish**.
21. In the console tree, right click **msftncsi.com**, and then click **New Host (A or AAAA)**.
22. In **Name**, type **www**. In **IP address**, type **131.107.0.1**.
23. Click **Add Host**. Click **OK**.

23. In **Name**, type **dns**. In **IP address**, type **131.107.255.255**. Click **Add Host**. Click **OK**. Click **Done**.
24. Close the DNS Manager console.

Windows PowerShell equivalent commands

The following Windows PowerShell cmdlet or cmdlets perform the same function as the preceding procedure. Enter each cmdlet on a single line, even though they may appear word-wrapped across several lines here because of formatting constraints.

```
Add-DnsServerPrimaryZone -Name isp.example.com -ZoneFile isp.example.com.dns
Add-DnsServerResourceRecordA -ZoneName isp.example.com -Name inet1 -IPv4Address
131.107.0.1
Add-DnsServerPrimaryZone -Name contoso.com -ZoneFile contoso.com.dns
Add-DnsServerResourceRecordA -ZoneName contoso.com -Name edge1 -IPv4Address
131.107.0.2
Add-DnsServerPrimaryZone -Name msftncsi.com -ZoneFile msftncsi.com.dns
Add-DnsServerResourceRecordA -ZoneName msftncsi.com -Name www -IPv4Address
131.107.0.1
Add-DnsServerResourceRecordA -ZoneName msftncsi.com -Name dns -IPv4Address
131.107.255.255
```

Install and configure DHCP on INET1

Next, configure INET1 as a DHCP server so that CLIENT1 can automatically configure itself when connecting to the Internet subnet.

Do this step using Windows PowerShell

To install and configure the DHCP server role on INET1

1. On the Server Manager **Dashboard** screen, under Configure this local server, click **Add roles and features**.
2. Click **Next** three times to get to the server role selection screen.
3. In the Select Server Roles dialog, select **DHCP Server**, click **Add Features** when prompted, and then click **Next**.
4. In the Select features dialog, click **Next**.
5. Click **Next** on the Introduction screen, and then click **Install**.
6. Allow the installation to complete, and then in the Installation progress window, click the link for **Complete DHCP configuration**.
7. In the DHCP Post-Install configuration wizard, click **Commit**, and then click **Close**.
8. In the Installation progress window, click **Close**.
9. From the **Tools** menu in Server Manager, click **DHCP**.
10. In the DHCP console tree, expand **INET1**. Right-click **IPv4**, and click **New Scope**.
11. Click **Next** in the New Scope Wizard.
12. Type **Internet** for scope name, and then click **Next**.
13. Next to **Start IP Address**, type **131.107.0.100**, next to **End IP Address**, type

131.107.0.150, and next to **Subnet Mask**, type **255.255.255.0**.

14. Click **Next** four times to accept default settings for exclusions, delay and lease duration.
15. On the **Router (Default Gateway)** dialog, type **131.107.0.1**. Click **Add**, and then click **Next**.
16. On the **Domain Name and DNS Servers** page, next to **Parent domain**, type **isp.example.com**. Under IP address, type **131.107.0.1**. Click **Add**, and then click **Next**.
17. On the WINS Servers page, click **Next**.
18. On the Activate Scope page, click **Next**, and then click **Finish**.
19. Close the DHCP Manager console.

Windows PowerShell equivalent commands

The following Windows PowerShell cmdlet or cmdlets perform the same function as the preceding procedure. Enter each cmdlet on a single line, even though they may appear word-wrapped across several lines here because of formatting constraints.

```
Install-WindowsFeature DHCP -IncludeManagementTools
Add-DhcpServerv4Scope -name "Internet" -StartRange 131.107.0.100  -EndRange
131.107.0.150 -SubnetMask 255.255.255.0
Set-DhcpServerv4OptionValue -DnsDomain isp.example.com -DnsServer 131.107.0.1 -
Router 131.107.0.1
```

Configure the NCSI web site on INET1

Windows clients attempt to connect to the URL http://www.msftncsi.com/ncsi.txt and resolve the name dns.msftncsi.com to determine if they have Internet connectivity. In the following procedure, you create the ncsi.txt file and place it in the WWWROOT directory on INET1.
Do this step using Windows PowerShell

To configure the NCSI web site on INET1

1. On INET1, launch **File Explorer**, and then navigate to **C:\inetpub\wwwroot**.
2. In the details pane, right click an empty area, point to **New**, and then click **Text Document**.
3. Rename the document to **ncsi**.
4. Double-click on **ncsi**.
5. In the **Notepad** window, type **Microsoft NCSI** and do *not* press ENTER to add a new line.
6. Click **File**, and then click **Exit**. In the **Notepad** dialog box, click **Save**.
7. Close the File Explorer window.

Windows PowerShell equivalent commands

The following PowerShell commands perform the same steps to write the Ncsi.txt file without a new line after the "Microsoft NCSI" string:

```
$filename = "C:\inetpub\wwwroot\ncsi.txt"
$text = "Microsoft NCSI"
[System.IO.File]::WriteAllText($fileName, $text)
```

Test access to Internet resources from the Internet subnet

Next, connect CLIENT1 to the Internet subnet and test connectivity to resources on INET1.

To test access to Internet resources from CLIENT1 when connected to the Internet subnet

1. Move CLIENT1 from the Corpnet subnet to the Internet subnet. Note that after network detection is complete, the warning symbol on the network icon in the system notification area no longer appears. Hover over the network icon in the system notification area and notice that it indicates *Internet access*. When prompted to automatically connect to devices on this network, click **Yes**.
2. Click the Internet Explorer icon.
3. In the **Address** bar, type **http://inet1.isp.example.com/**, and then press **ENTER**. You should see the default Internet Information Services web page.
4. Close the Internet Explorer window.
5. From the Start screen, type **command**, and then under the Search results click **Command Prompt**.
6. Type **ping inet1.isp.example.com** and press **ENTER**. You should see four responses from **131.107.0.1**. Type **ping edge1.contoso.com** and press **ENTER**. You should see four responses from **131.107.0.2**.
7. Move CLIENT1 from the Internet subnet to the Corpnet subnet.
8. From the Windows PowerShell window, type **ping inet1.isp.example.com**, and then press ENTER. You should see a “Ping request could not find host inet1.isp.example.com” message and no responses. Type **ping 131.107.0.1**, and then press ENTER. You should see “transmit failed” messages and no responses. This indicates that there is no connectivity between the Corpnet subnet and the Internet subnet.
   Although EDGE1 is connected to both the Internet and Corpnet subnets, it is not providing any routing, address translation, or proxying services to allow computers on the Corpnet subnet to access resources on the Internet subnet. Additional test lab guides may configure Internet subnet access from the Corpnet subnet as needed.

Snapshot the Configuration

This completes the Base Configuration test lab. To save this configuration for additional test labs, do the following:

1. On all physical computers or virtual machines in the test lab, close all windows and then perform a graceful shutdown.
2. If your lab is based on virtual machines, save a snapshot of each virtual machine and name the snapshots **Windows Server 2012 R2 Base Configuration**. If your lab uses physical computers, create disk images to save the Base Configuration.

Additional Resources

We strongly encourage you to develop and publish your own TLG content for Windows Server 2012 R2, either in the TechNet Wiki (example: Test Lab Guide: Demonstrate Remote Access VPNs) or in your own publishing forum (example: Test Lab Guide (Part 1) - Demonstrate TMG PPTP, L2TP/IPsec and SSTP Remote Access VPN Server). If you want to publish your TLG content in the TechNet wiki, see the How to contribute series of TLG blog posts for information about the types of content you can create and for links to templates, guidance, and examples.
For a list of additional Microsoft TLGs, see Test Lab Guides in the TechNet Wiki.

Appendix

This appendix describes how to change the default User Account Control (UAC) behavior.

Set UAC behavior of the elevation prompt for administrators

By default, UAC is enabled in Windows Server 2012 R2 and Windows 8.1. This service will prompt for permission to continue during several of the configuration tasks described in this guide. In all cases, you can click **Continue** in the UAC dialog box to grant this permission, or you can use the following procedure to change the UAC behavior of the elevation prompt for administrators.

To set UAC behavior of the elevation prompt for administrators

1. From the Start screen, type **secpol.msc**, and press ENTER.
2. In the console tree, open **Local Policies**, and then click **Security Options**.
3. In the contents pane, double-click **User Account Control: Behavior of the elevation prompt for administrators in Admin Approval Mode**.

4. Select **Elevate without prompting** in the list, and then click **OK**.
6. Close the **Local Security Policy** window.