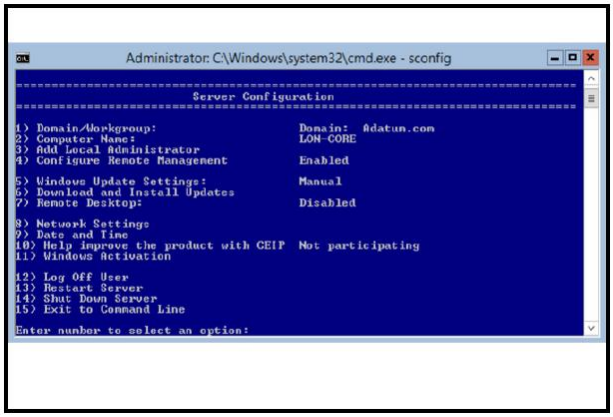
Configuring a Server Core Installation



Performing post installation on a computer running the Server Core operating system option can be daunting to administrators who have not performed the task before. Inst ead of having GUI-basedtools that simplify the post-installation configuration process, IT professionals must performing complex configur ation tasks from a command-line interface.

The good news is that you can perform the majority of post-installation configuration tasks by using the sconfig.cmd command-line tool. Using this tool minimizes the possibility of making syntaxerrors when you u se more complicated command-line tools.

You can use **sconfig.cmd** to perform the following tasks:

- Configure Domain and Workgroup information
- Configure the computer’s name
- Add local Administrator accounts
- Configure WinRM
- Enable Windows Update
- Download and install updates
- Enable Remote Desktop
- Configure Network Address information
- Set the date and time
- Perform Windows Activation
- Sign out
- Restart the server
- Shut down the server

Configure IP Address Information

You can configure the IP address and DNS information using **sconfig.cmd** or **netsh.exe**. To configure IP address information using **sconfig.cmd**, perform the following steps:

1. From a command-line command, run **sconfig.cmd**.
2. Choose **option 8** to configure Network Settings.
3. Choose the index number of the network adapter to which you want to assign an IP address.
4. In the Network Adapter Settings area, choose one of the following options:
    - Set Network Adapter Address
    - Set DNS Servers
    - Clear DNS Server Settings
    - Return to Main Menu

Change Server Name

You can change a server's name using the **netdom** command with the **renamecomputer** option.

For example, to rename a computer to Melbourne, type the following command:

```
Netdom renamecomputer %computername% /newname:Melbourne
```

You can change a server's name using **sconfig.cmd** by performing the following procedure:

1. From a command-line command, run **sconfig.cmd**.
2. Choose **option 2** to configure the new computer name.
3. Type the new computer name, and then press Enter.

You must restart the server for the configuration change to take effect.

Joining the Domain

You can join a Server Core computer to a domain using the **netdom** command with the **join** option.

For example, to join the adatum.com domain using the Administrator account, and to be prompted for a password, type the following command:

```
Netdom join %computername% /domain:adatum.com /UserD:Administrator
/PasswordD:*
```

Note: Prior to joining the domain, verify that you are able to ping the DNS server by hostname.

To join a Server Core computer to the domain using **sconfig.cmd**, perform the followi ng steps:

1. From a command-line command, run **sconfig.cmd**.
2. Choose **option 1** to configure Domain/Workgroup.
3. To choose the Domain option, type **D**, and then press Enter.
4. Type the name of the domain to which you want to join the computer.
5. Provide the details, in *domain\username* format, of an account that is authorized to join the dom
6. Type the password associated with that account.

To complete a domain join operation, it is necessary to restart the computer.

Adding Roles and Features

You can add and remove roles and features on a computer that is running the Server C ore installation option by using the Windows PowerShell cmdlets **Get-WindowsFeature**, **Install-WindowsFeature**, and **Remove-WindowsFeature**. These cmdlets are available after you load the ServerManager Win dows PowerShell module.

For example, you can view a list of roles and features that are installed by typing the f ollowing command:

```
Get-WindowsFeature | Where-Object {$_.InstallState -eq "Installed"}
```

You can also install a Windows role or feature using the **Install-WindowsFeature** cmdlet. For example, to install the NLB feature, execute the comm and:

```
Install-WindowsFeature NLB
```

Not all features are available directly for installation on a computer running the Server Core operating system. You can determine which features are not directly available f or installation by runningthe following command:

```
Get-WindowsFeature | Where-Object {$_.InstallState -eq "Removed"}
```

You can add a role or feature that is not directly available for installation by using the **-Source** parameter of the **Install-WindowsFeature** cmdlet. You must specify a source location that hosts amounted ins tallation image that includes the full version of Windows Server 2012. You can mount an installation image using the **DISM.exe** command-line tool. If you do not specify a source pathwhen you install a component that is not a

vailable and the server has Internet connectivity, **Install-WindowsFeature** will attempt to retrieve source files from Windows Update.

Add the GUI

You can configure a Server Core computer with the GUI using the **sconfig.cmd** command-line tool. To do this, choose **option 12** from within the sconfig.cmd Server Configuration menu.

Note: You can add or remove the graphical component of the Windows Server 2012 operating system by using the **Install-WindowsFeature** cmdlet.

You can also use the **dism.exe** command-line tool to add and remove Windows roles and features from a Server Core deployment, even though this tool is used primarily for managing image files.

Demonstration: Using DISM to Add Windows Features

Deployment Image Servicing and Management (DISM) is a command-line tool that you can use to service offline images or running operating systems. Use it to install, uninstall, configure, andupdate Windows features, packages, drivers and international settings.

After an image has been mounted to the file system, use DISM to service that image by specifying the path to the image and the servicing options in the command line. DISM can also service runningsystems when you specify the **/online** parameter and the servicing options in the command line.

In this demonstration, you will see how to use DISM to enable the Windows Server Backup feature for a running system. For example, if you were servicing an offline image, you would first use theDISM **/mount-image** parameter to mount the image to the file system. Then you would use the DISM **/image:**<*path to imagefile*> parameter and pass servicing commands to the image. To service a .vhd file, attach the virtual disk by using Windows PowerShell. Although it is not the preferred method, you also can use the DiskPart.exe command-line tool. The Windows PowerShell4.0 Mount-DiskImage cmdlet mounts an existing .vhd or .iso file and makes it appear as if it is a normal disk. For example, to mount a .vhd file named C:\BaseImage.vhd, you can perform thefollowing procedure.

To use Windows PowerShell to mount a virtual hard disk file named C:\BaseImage.vhd and assign the next available drive letter, start Windows PowerShell and run the following cmdlet:

```
Mount-DiskImage C:\BaseImage.vhd
```

After servicing the .vhd file using DISM, you use the Dismount-DiskImage cmdlet:

```
Dismount-DiskImage C:\BaseImage.vhd
```

To use DiskPart to attach a .vhd file and assign the drive letter V, at an elevated command prompt, run the following commands:

```
DiskPart
Select vdisk file C:\BaseImage.vhd
Attach vdisk
Assign letter=V
Exit
```

After you finish servicing the .vhd file using DISM, you can detach the .vhd file by using the following commands:

```
DiskPart
Select vdisk file C:\BaseImage.vhd
Detach vdisk
Exit
```