PowerShell Get-Help on command 'Stop-Computer'

PS C:\Users\wahid> Get-Help Stop-Computer

NAME

Stop-Computer

SYNOPSIS

Stops (shuts down) local and remote computers.

SYNTAX

Stop-Computer [[-ComputerName] <System.String[]>] [[-Credential] <System.Management.Automation.PSCredential>] [-AsJob] [-DcomAuthentication {Default | None | Connect | Call | Packet | PacketIntegrity | PacketPrivacy | Unchanged}] [-Force] [-Impersonation {Default | Anonymous | Identify | Impersonate | Delegate}] [-Protocol {DCOM | WSMan}] [-ThrottleLimit <System.Int32>] [-WsmanAuthentication {Default | Basic | Negotiate | CredSSP | Digest | Kerberos}] [-Confirm] [-WhatIf] [<CommonParameters>]

DESCRIPTION

The `Stop-Computer` cmdlet shuts down the local computer and remote computers.

You can use the parameters of `Stop-Computer` to run the shutdown operations as a background job, to specify the authentication levels and alternate credentials, to limit the concurrent connections that are created to run the command, and to force an immediate shut down.

This cmdlet doesn't require PowerShell remoting unless you use the AsJob parameter.

PARAMETERS

-AsJob <System.Management.Automation.SwitchParameter>

Indicates that this cmdlet runs as a background job.

To use this parameter, the local and remote computers must be configured for remoting and, on Windows Vista and later versions of the Windows operating system, you must open PowerShell by using the Run as administrator option. For more information, see about_Remote_Requirements (..//microsoft.powershell.core/about/about_remote_requirements.md).

When you specify the AsJob parameter, the command immediately returns an object that represents the background job. You can continue to work in the session while the job finishes. The job is created on the local computer and the results from remote computers are automatically returned to the local computer. To get the job results, use the `Receive-Job` cmdlet.

For more information about PowerShell background jobs, see about_Jobs (..//microsoft.powershell.core/about/about_jobs.md)and about_Remote_Jobs (../microsoft.powershell.core/about/about_remote_jobs.md).

-ComputerName <System.String[]>

Specifies the computers to stop. The default is the local computer.

Type the NETBIOS name, IP address, or fully qualified domain name of one or more computers in a comma-separated list. To specify the local computer, type the computer name or localhost.

This parameter doesn't rely on PowerShell remoting. You can use the ComputerName parameter even if your computer isn't configured to run remote commands.

-Credential <System.Management.Automation.PSCredential>

Specifies a user account that has permission to do this action. The default is the current user.

Type a user name, such as User01 or Domain01\User01 , or enter a PSCredential object generated by the `Get-Credential` cmdlet. If you type a user name, you're prompted to enter the password.

Credentials are stored in a PSCredential (/dotnet/api/system.management.automation.pscredential)object and the password is stored as a SecureString (/dotnet/api/system.security.securestring).

> [!NOTE] > For more information about SecureString data protection, see > How secure is SecureString? (/dotnet/api/system.security.securestring#how-secure-is-securestring).

-DcomAuthentication <System.Management.AuthenticationLevel>

Specifies the authentication level that this cmdlet uses with WMI. `Stop-Computer` uses WMI. The default value is Packet .

The acceptable values for this parameter are:

- Default : Windows Authentication. - None : No COM authentication. - Connect : Connect-level COM authentication. - Call : Call-level COM authentication. - Packet : Packet-level COM authentication. - PacketIntegrity : Packet Integrity-level COM authentication. -

PacketPrivacy : Packet Privacy-level COM authentication. - Unchanged : Same as the previous command.

For more information about the values of this parameter, see AuthenticationLevel (/dotnet/api/system.management.authenticationlevel).

-Force <System.Management.Automation.SwitchParameter>

Forces an immediate shut down of the computer.

-Impersonation <System.Management.ImpersonationLevel>

Specifies the impersonation level to use when this cmdlet calls WMI. The default value is Impersonate .

`Stop-Computer` uses WMI. The acceptable values for this parameter are:

- Default : Default impersonation. - Anonymous : Hides the identity of the caller. - Identify : Allows objects to query the credentials of the caller. - Impersonate : Allows objects to use the credentials of the caller.

-Protocol <System.String>

Specifies which protocol to use to restart the computers. The acceptable values for this parameter are: WSMan and DCOM . The default value is DCOM .

This parameter was introduced in PowerShell 3.0.

-ThrottleLimit <System.Int32>

Specifies the maximum number of concurrent connections that can be established to run this command. If you omit this parameter or enter a value of 0, the default value, 32, is used.

The throttle limit applies only to the current command, not to the session or to the computer.

-WsmanAuthentication <System.String>

Specifies the mechanism that is used to authenticate the user credentials when this cmdlet uses the WSMan protocol. The default value is Default .

The acceptable values for this parameter are:

- Basic

- CredSSP

- Default

- Digest

- Kerberos

- Negotiate.

For more information about the values of this parameter, see AuthenticationMechanism (/dotnet/api/system.management.automation.runspaces .authenticationmechanism).

> [!CAUTION] > Credential Security Service Provider (CredSSP) authentication, in which the user credentials are > passed to a remote computer to be authenticated, is designed for commands that require > authentication on more than one resource, such as accessing a remote network share. This mechanism > increases the security risk of the remote operation. If the remote computer is compromised, the > credentials that are passed to it can be used to control the network session.

This parameter was introduced in PowerShell 3.0.

-Confirm <System.Management.Automation.SwitchParameter>

    Prompts you for confirmation before running the cmdlet.

-WhatIf <System.Management.Automation.SwitchParameter>

    Shows what would happen if the cmdlet runs. The cmdlet isn't run.

<CommonParameters>

    This cmdlet supports the common parameters: Verbose, Debug, ErrorAction, ErrorVariable, WarningAction, WarningVariable, OutBuffer, PipelineVariable, and OutVariable. For more information, see about_CommonParameters (https:/go.microsoft.com/fwlink/?LinkID=113216).

----------- Example 1: Shut down the local computer -----------

```
Stop-Computer -ComputerName localhost
```

Example 2: Shut down two remote computers and the local computer

```
Stop-Computer -ComputerName "Server01", "Server02", "localhost"
```

`Stop-Computer` uses the ComputerName parameter to specify two remote computers and the local computer. Each computer is shut down.

-- Example 3: Shut down remote computers as a background job --

```
$j = Stop-Computer -ComputerName "Server01", "Server02" -AsJob
$results = $j | Receive-Job
$results
```

`Stop-Computer` uses the ComputerName parameter to specify two remote computers. The AsJob parameter runs the command as a background job. The job objects are stored in the `$j` variable.

The job objects in the `$j` variable are sent down the pipeline to `Receive-Job`, which gets the job results. The objects are stored in the `$results` variable. The `$results` variable displays the job information in the PowerShell console.

Because AsJob creates the job on the local computer and automatically returns the results to the local computer, you can run `Receive-Job` as a local command.

------------ Example 4: Shut down a remote computer ------------

```
Stop-Computer -ComputerName "Server01" -Impersonation Anonymous
-DcomAuthentication PacketIntegrity
```

`Stop-Computer` uses the ComputerName parameter to specify the remote computer. The Impersonation parameter specifies a customized impersonation and the DcomAuthentication parameter specifies authentication-level settings.

---------- Example 5: Shut down computers in a domain ----------

```
$s = Get-Content -Path ./Domain01.txt
$c = Get-Credential -Credential Domain01\Admin01
Stop-Computer -ComputerName $s -Force -ThrottleLimit 10 -Credential $c
```

`Get-Content` uses the Path parameter to get a file in the current directory with the list of domain computers. The objects are stored in the `$s` variable.

`Get-Credential` uses the Credential parameter to specify the credentials of a domain administrator. The credentials are stored in the `$c` variable.

`Stop-Computer` shuts down the computers specified with the ComputerName parameter's list of computers in the `$s` variable. The Force parameter forces an immediate shutdown. The ThrottleLimit parameter limits the command to 10 concurrent connections. The Credential parameter submits the credentials saved in the `$c` variable.

REMARKS

To see the examples, type: "get-help Stop-Computer -examples".

For more information, type: "get-help Stop-Computer -detailed".

For technical information, type: "get-help Stop-Computer -full".

For online help, type: "get-help Stop-Computer -online"