

How to Use Invoke-Command in PowerShell

The Invoke-Command cmdlet is a powerful feature in Windows PowerShell that allows you to run commands or scripts on local or remote computers. This is particularly useful for system administrators who need to manage multiple machines simultaneously. With Invoke-Command, you can execute a script block on one or more remote computers, retrieve the output, and handle it as needed. This cmdlet is part of the Windows PowerShell remoting feature, which must be enabled on the target machines.

Examples:

1. Running a Command on a Local Machine:

To run a command on your local machine using Invoke-Command, you can simply use the -ScriptBlock parameter. Here's an example:

```
Invoke-Command -ScriptBlock { Get-Process }
```

This command will list all the processes running on your local machine.

2. Running a Command on a Remote Machine:

To run a command on a remote machine, you need to specify the -ComputerName parameter along with the target machine's name or IP address. Ensure that PowerShell Remoting is enabled on the remote machine.

```
Invoke-Command -ComputerName "RemotePC" -ScriptBlock { Get-Process }
```

This command will list all the processes running on the remote machine named "RemotePC".

3. Running a Script on Multiple Remote Machines:

You can also run a script on multiple remote machines by providing an array of computer names.

```
$computers = @("RemotePC1", "RemotePC2", "RemotePC3")  
Invoke-Command -ComputerName $computers -ScriptBlock { Get-Process }
```

This command will list all the processes running on the specified remote machines.

4. Passing Arguments to the Script Block:

Sometimes, you may need to pass arguments to the script block. You can do this using the `-ArgumentList` parameter.

```
$scriptBlock = {  
    param ($processName)  
    Get-Process -Name $processName  
}  
Invoke-Command -ComputerName "RemotePC" -ScriptBlock $scriptBlock -ArgumentList "notepad"
```

This command will get the process named "notepad" on the remote machine "RemotePC".

5. Using Credentials for Remote Execution:

If you need to provide credentials for the remote machine, you can use the `-Credential` parameter.

```
$credential = Get-Credential  
Invoke-Command -ComputerName "RemotePC" -ScriptBlock { Get-Process } -Credential $credential
```

This command will prompt you for credentials and then list all the processes running on the remote machine "RemotePC".