How to Get the Computer Name with PowerShell

lazyadmin.nl/powershell/get-the-computer-name

August 27, 2024

PowerShell has a lot of cmdlets, but strange enough not really one that returns only the computername. But that doesn't mean we can't get the name of the device, there are actually multiple options to return the computer name, even with a single command.

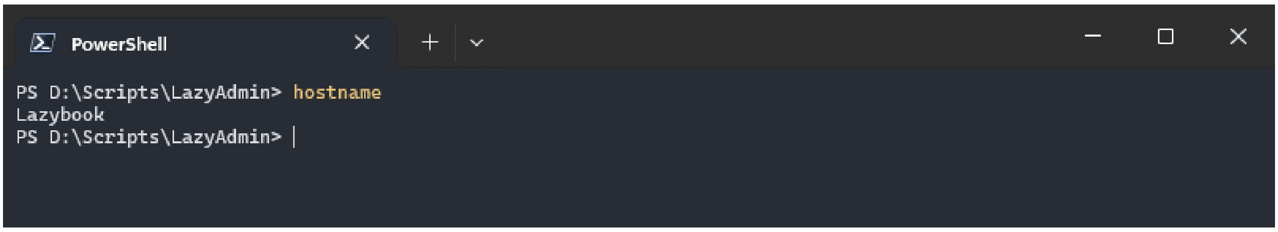
In case you are wondering why you want to look it up, I often use the computer name as part of a log file name for automated scripts. Or sometimes you just want to quickly see which machine you are working on.

In this article

In this article, I will show you the different methods to get the computer name from the local machine and remote computers.

Get Computer Name

The easiest method to get the name of the computer that you are currently working on is actually to use the old command `hostname`. This command is normally used in the command prompt, but it also works perfectly in PowerShell



```
PS D:\Scripts\LazyAdmin> hostname
Lazybook
PS D:\Scripts\LazyAdmin> |
```

We can simply store the hostname in a variable to use later in a script. It's also possible to pipe other cmdlets behind it, like clip to copy to the clipboard, or something else.

```
# Store hostname in variable
$hostname = hostname
# Copy hostname to clipboard
hostname | clip
```

Using the Environment Variable

A more PowerShell way to retrieve the hostname is to use the environment variable. This is a set of dynamic variables which is used by the operating system and other processes to store information. We can access the information stored in the environment variable by using the `$Env:` drive, followed by the variable name.

So to get the computer name, we can do the following:

```
$env:COMPUTERNAME
```

The problem however with this method, is it only works on Windows machines. In most cases that is good enough, but if you need to run your script on macOS as well, then a better option is to use the Environment class to get the machine name. This works on all machines:

```
[Environment]::MachineName
```

Using the Common Information Model (CIM)

Another fast and reliable method to retrieve the name of the device is to use the Common Information Model (CIM). This is the successor of the older, well-known, WMI object that we previously used to get information about the computer.

The advantage of using Get-CimInstance is that we can also use this method for remote computers.

```
(Get-CimInstance -ClassName Win32_ComputerSystem).Name
```

Get a Remote Computer Name

In some cases, you need to get the computer name of a remote machine. We could use any of the methods above and simply place them in the scriptblock of the invoke-command cmdlet, like this:

```
Invoke-Command -ComputerName "192.168.1.12", "192.168.1.14" -ScriptBlock { $env:COMPUTERNAME }
```

But in this case, a better option is to see the `Get-CimInstance` cmdlet. This is faster and allows you to get the computer name of multiple remote computers:

```
Get-CimInstance -ComputerName "192.168.1.12", "192.168.1.14" -ClassName Win32_ComputerSystem| Select PSComputerName,Name
```

Using Resolve DnsName

Another method to get the computer name based on the IP Address is to use the `Resolve-DnsName` cmdlet. This command will query the DNS server to resolve the IP Address of the computer.

```
Resolve-DnsName -Name 192.168.1.12 | Select-Object -ExpandProperty NameHost
```

Wrapping Up

There are more ways to look up the name of a device, but most of them are not really convenient, or even slower (like `Get-ComputerInfo`). On a local machine is the hostname command the easiest one to remember and use.

If you need to look up the name inside a script, then I would recommend using the environment variable or class depending on the operating systems in your environment.

If you have any questions, just drop a comment below.

Did you **Liked** this **Article**?
Get the latest articles like this **in your mailbox**
or share this article

I hate spam to, so you can unsubscribe at any time.