

## Windows PowerShell equivalents for common networking commands (IPCONFIG, PING, NSLOOKUP)

Network troubleshooting is part any System Administrator's life. Maybe you need to check the IP address of a machine or test if its networking connection is working. Maybe you need to see if DNS is properly configured or check the latency between two hosts.

If you have been in this field long enough, you probably have a few favorite commands that you learned years ago and use on a regular basis, like IPCONFIG or PING.

There are literally hundreds of networking-related PowerShell cmdlets in Windows these days. Just try out this command on your machine: `Get-Command -Module Net* | Group Module`

But more important than knowing every one of them, is to know the most useful cmdlets that have the potential to replace those old commands that you can't live without.

And it's when you combine the many networking cmdlets in ways that only PowerShell can do that you'll find amazing new troubleshooting abilities...

# IPCONFIG

Description: This command has many options, but the most common usage is just to show the IP address, subnet mask and default gateway for each network adapter in a machine.

PowerShell: Get-NetIPAddress

Sample command lines:

- Get-NetIPAddress | Sort InterfaceIndex | FT InterfaceIndex, InterfaceAlias, AddressFamily, IPAddress, PrefixLength -AutoSize
- Get-NetIPAddress | ? AddressFamily -eq IPv4 | FT -AutoSize
- Get-NetAdapter Wi-Fi | Get-NetIPAddress | FT -AutoSize

Sample output:

```
PS C:\> Get-NetIPAddress | Sort InterfaceIndex | FT InterfaceIndex, InterfaceAlias, AddressFamily, IPAddress, PrefixLength -AutoSize
```

InterfaceIndex	InterfaceAlias	AddressFamily	IPAddress	PrefixLength
1	Loopback Pseudo-Interface 1	IPv6	::1	128
1	Loopback Pseudo-Interface 1	IPv4	127.0.0.1	8
3	Wi-Fi	IPv6	fe80::88f2:9970:4082:4118%3	64
3	Wi-Fi	IPv6	fded:b22c:44c4:1:f188:1e45:58e3:9242	128
3	Wi-Fi	IPv6	fded:b22c:44c4:1:88f2:9970:4082:4118	64
3	Wi-Fi	IPv4	192.168.1.2	24
4	Ethernet	IPv6	fe80::ce6:97c9:ae58:b393%4	64
4	Ethernet	IPv4	169.254.179.147	16
6	Bluetooth Network Connection	IPv6	fe80::2884:6750:b46b:cec4%6	64
6	Bluetooth Network Connection	IPv4	169.254.206.196	16
7	Local Area Connection* 3	IPv6	fe80::f11f:1051:2f3d:882%7	64
7	Local Area Connection* 3	IPv4	169.254.8.130	16
8	Teredo Tunneling Pseudo-Interface	IPv6	2001:0:5ef5:79fd:1091:f90:e7e9:62f0	64
8	Teredo Tunneling Pseudo-Interface	IPv6	fe80::1091:f90:e7e9:62f0%8	64
9	isatap.{024820F0-C990-475F-890B-B42EA24003F1}	IPv6	fe80::5efe:192.168.1.2%9	128

```
PS C:\> Get-NetIPAddress | ? AddressFamily -eq IPv4 | FT -AutoSize
```

ifIndex	IPAddress	PrefixLength	PrefixOrigin	SuffixOrigin	AddressState	PolicyStore
7	169.254.8.130	16	WellKnown	Link	Tentative	ActiveStore
6	169.254.206.196	16	WellKnown	Link	Tentative	ActiveStore
3	192.168.1.2	24	Dhcp	Dhcp	Preferred	ActiveStore
1	127.0.0.1	8	WellKnown	WellKnown	Preferred	ActiveStore
4	169.254.179.147	16	WellKnown	Link	Tentative	ActiveStore

```
PS C:\> Get-NetAdapter Wi-Fi | Get-NetIPAddress | FT -AutoSize
```

ifIndex	IPAddress	PrefixLength	PrefixOrigin	SuffixOrigin	AddressState	PolicyStore
3	fe80::88f2:9970:4082:4118%3	64	WellKnown	Link	Preferred	ActiveStore
3	fded:b22c:44c4:1:f188:1e45:58e3:9242	128	RouterAdvertisement	Random	Preferred	ActiveStore
3	fded:b22c:44c4:1:88f2:9970:4082:4118	64	RouterAdvertisement	Link	Preferred	ActiveStore
3	192.168.1.2	24	Dhcp	Dhcp	Preferred	ActiveStore

# PING

Description: Checks connectivity to a specific host. Commonly used to check for liveness, but also used to measure network latency.

PowerShell: Test-NetConnection

Sample command lines:

- Test-NetConnection [www.microsoft.com](http://www.microsoft.com)
- Test-NetConnection -ComputerName www.microsoft.com -InformationLevel Detailed
- Test-NetConnection -ComputerName www.microsoft.com | Select -ExpandProperty PingReplyDetails | FT Address, Status, RoundTripTime
- 1..10 | % { Test-NetConnection -ComputerName www.microsoft.com -RemotePort 80 } | FT -AutoSize

Sample output

```
PS C:\> Test-NetConnection www.microsoft.com
```

```
ComputerName           : www.microsoft.com
RemoteAddress          : 104.66.197.237
InterfaceAlias         : Wi-Fi
SourceAddress          : 192.168.1.2
PingSucceeded          : True
PingReplyDetails (RTT) : 22 ms
```

```
PS C:\> Test-NetConnection -ComputerName www.microsoft.com -InformationLevel Detailed
```

```
ComputerName           : www.microsoft.com
RemoteAddress          : 104.66.197.237
AllNameResolutionResults : 104.66.197.237
                        2600:1409:a:396::2768
                        2600:1409:a:39b::2768
InterfaceAlias         : Wi-Fi
SourceAddress          : 192.168.1.2
NetRoute (NextHop)     : 192.168.1.1
PingSucceeded          : True
PingReplyDetails (RTT) : 14 ms
```

```
PS C:\> Test-NetConnection -ComputerName www.microsoft.com | Select -ExpandProperty PingReplyDetails | FT  
Address, Status, RoundTripTime -AutoSize
```

Address	Status	RoundtripTime
-----	-----	-----
104.66.197.237	Success	22

```
PS C:\> 1..10 | % { Test-NetConnection -ComputerName www.microsoft.com -RemotePort 80 } | FT -AutoSize
```

ComputerName	RemotePort	RemoteAddress	PingSucceeded	PingReplyDetails (RTT)	TcpTestSucceeded
-----	-----	-----	-----	-----	-----
www.microsoft.com	80	104.66.197.237	True	17 ms	True
www.microsoft.com	80	104.66.197.237	True	16 ms	True
www.microsoft.com	80	104.66.197.237	True	15 ms	True
www.microsoft.com	80	104.66.197.237	True	18 ms	True
www.microsoft.com	80	104.66.197.237	True	20 ms	True
www.microsoft.com	80	104.66.197.237	True	20 ms	True
www.microsoft.com	80	104.66.197.237	True	20 ms	True
www.microsoft.com	80	104.66.197.237	True	20 ms	True
www.microsoft.com	80	104.66.197.237	True	15 ms	True
www.microsoft.com	80	104.66.197.237	True	13 ms	True

## NSLOOKUP

Description: Name server lookup. Mostly used to find the IP address for a given DNS name (or vice-versa). Has many, many options.

PowerShell: `Resolve-DnsName`

Sample command lines:

- `Resolve-DnsName www.microsoft.com`
- `Resolve-DnsName microsoft.com -type SOA`
- `Resolve-DnsName microsoft.com -Server 8.8.8.8 -Type A`

Sample output

```
PS C:\> Resolve-DnsName www.microsoft.com
```

Name	Type	TTL	Section	NameHost
-----	----	---	-----	-----
<a href="http://www.microsoft.com">www.microsoft.com</a>	CNAME	6	Answer	toggle.www.ms.akadns.net
toggle.www.ms.akadns.net	CNAME	6	Answer	<a href="http://www.microsoft.com-c.edgekey.net">www.microsoft.com-c.edgekey.net</a>
<a href="http://www.microsoft.com-c.edgekey.net">www.microsoft.com-c.edgekey.net</a>	CNAME	6	Answer	<a href="http://www.microsoft.com-c.edgekey.net.globalredir.akadns.net">www.microsoft.com-c.edgekey.net.globalredir.akadns.net</a>
t				
<a href="http://www.microsoft.com-c.edgekey.net">www.microsoft.com-c.edgekey.net</a>	CNAME	6	Answer	e10088.dspb.akamaiedge.net
t.globalredir.akadns.net				

```
Name      : e10088.dspb.akamaiedge.net
QueryType : AAAA
TTL       : 6
Section   : Answer
IP6Address : 2600:1409:a:39b::2768
```

```
Name      : e10088.dspb.akamaiedge.net
QueryType : AAAA
TTL       : 6
Section   : Answer
IP6Address : 2600:1409:a:396::2768
```

Name : e10088.dspb.akamaiedge.net  
QueryType : A  
TTL : 6  
Section : Answer  
IP4Address : 104.66.197.237

PS C:\> **Resolve-DnsName microsoft.com -type SOA**

Name	Section	PrimaryServer	Type	NameAdministrator	SerialNumber
-----			----	-----	-----
-----					
microsoft.com			SOA 2976 Answer	ns1.msft.net	msnhst.microsoft.com
801					2015041

PS C:\> **Resolve-DnsName microsoft.com -Server 8.8.8.8 -Type A**

Name	Type	TTL	Section	IPAddress
-----	----	---	-----	-----
microsoft.com	A	1244	Answer	134.170.188.221
microsoft.com	A	1244	Answer	134.170.185.46

# ROUTE

Description: Shows the IP routes in a given system (also used to add and delete routes)

PowerShell: Get-NetRoute (also New-NetRoute and Remove-NetRoute)

Sample command lines:

- Get-NetRoute -Protocol Local -DestinationPrefix 192.168\*
- Get-NetAdapter Wi-Fi | Get-NetRoute

Sample output:

```
PS C:\WINDOWS\system32> Get-NetRoute -Protocol Local -DestinationPrefix 192.168*
```

ifIndex	DestinationPrefix	NextHop	RouteMetric	PolicyStore
2	192.168.1.255/32	0.0.0.0	256	ActiveStore
2	192.168.1.5/32	0.0.0.0	256	ActiveStore
2	192.168.1.0/24	0.0.0.0	256	ActiveStore

```
PS C:\WINDOWS\system32> Get-NetAdapter Wi-Fi | Get-NetRoute
```

ifIndex	DestinationPrefix	NextHop	RouteMetric	PolicyStore
2	255.255.255.255/32	0.0.0.0	256	ActiveStore
2	224.0.0.0/4	0.0.0.0	256	ActiveStore
2	192.168.1.255/32	0.0.0.0	256	ActiveStore
2	192.168.1.5/32	0.0.0.0	256	ActiveStore
2	192.168.1.0/24	0.0.0.0	256	ActiveStore
2	0.0.0.0/0	192.168.1.1	0	ActiveStore
2	ff00::/8	::	256	ActiveStore
2	fe80::d1b9:9258:1fa:33e9/128	::	256	ActiveStore
2	fe80::/64	::	256	ActiveStore
2	fded:b22c:44c4:1:d1b9:9258:1fa:33e9/128	::	256	ActiveStore
2	fded:b22c:44c4:1:c025:aa72:9331:442/128	::	256	ActiveStore
2	fded:b22c:44c4:1::/64	::	256	ActiveStore



## TRACERT

Description: Trace route. Shows the IP route to a host, including all the hops between your computer and that host.

PowerShell: Test-NetConnection -TraceRoute

Sample command lines:

- Test-NetConnection www.microsoft.com -TraceRoute
- Test-NetConnection outlook.com -TraceRoute | Select -ExpandProperty TraceRoute | %{ Resolve-DnsName \$\_ -type PTR -ErrorAction SilentlyContinue }

Sample output:

```
PS C:\> Test-NetConnection www.microsoft.com -TraceRoute
```

```
ComputerName           : www.microsoft.com
RemoteAddress          : 104.66.197.237
InterfaceAlias         : Wi-Fi
SourceAddress          : 192.168.1.2
PingSucceeded          : True
PingReplyDetails (RTT) : 16 ms
TraceRoute              : 192.168.1.1
                        10.0.0.1
                        TimedOut
                        68.86.113.181
                        69.139.164.2
                        68.85.240.94
                        68.86.93.165
                        68.86.83.126
                        104.66.197.237
```

```
PS C:\> Test-NetConnection outlook.com -TraceRoute | Select -ExpandProperty TraceRoute | % { Resolve-DnsName $_ -  
type PTR -ErrorAction SilentlyContinue }
```

Name	Type	TTL	Section	NameHost
----	----	---	-----	-----
125.144.85.68.in-addr.arpa	PTR	7200	Answer	te-0-1-0-10-sur02.bellevue.wa.seattle.comcast.net
142.96.86.68.in-addr.arpa	PTR	4164	Answer	be-1-sur03.bellevue.wa.seattle.comcast.net
6.164.139.69.in-addr.arpa	PTR	2469	Answer	be-40-ar01.seattle.wa.seattle.comcast.net
165.93.86.68.in-addr.arpa	PTR	4505	Answer	be-33650-cr02.seattle.wa.ibone.comcast.net
178.56.167.173.in-addr.arpa	PTR	7200	Answer	as8075-1-c.seattle.wa.ibone.comcast.net
248.82.234.191.in-addr.arpa	PTR	3600	Answer	ae11-0.co2-96c-1a.ntwk.msn.net

## NETSTAT

Description: Shows current TCP/IP network connections.

PowerShell: Get-NetTCPConnection

Sample command lines:

- `Get-NetTCPConnection | Group State, RemotePort | Sort Count | FT Count, Name -AutoSize`
- `Get-NetTCPConnection | ? State -eq Established | FT -AutoSize`
- `Get-NetTCPConnection | ? State -eq Established | ? RemoteAddress -notlike 127* | % { $_; Resolve-DnsName $_.RemoteAddress -type PTR -ErrorAction SilentlyContinue }`

Sample output:

```
PS C:\> Get-NetTCPConnection | Group State, RemotePort | Sort Count | FT Count, Name -AutoSize
```

```
Count Name
-----
1 SynSent, 9100
1 Established, 40028
1 Established, 65001
1 Established, 27015
1 Established, 5223
1 Established, 49227
1 Established, 49157
1 Established, 49156
1 Established, 12350
1 Established, 49200
2 Established, 5354
2 TimeWait, 5357
2 Established, 80
3 Established, 443
36 Listen, 0
```

```
PS C:\> Get-NetTCPConnection | ? State -eq Established | FT -AutoSize
```

LocalAddress	LocalPort	RemoteAddress	RemotePort	State	AppliedSetting
127.0.0.1	65001	127.0.0.1	49200	Established	Internet
192.168.1.2	59619	91.190.218.57	12350	Established	Internet
192.168.1.2	57993	213.199.179.175	40028	Established	Internet
192.168.1.2	54334	17.158.28.49	443	Established	Internet
192.168.1.2	54320	96.17.8.170	80	Established	Internet
192.168.1.2	54319	23.3.105.144	80	Established	Internet
192.168.1.2	54147	65.55.68.119	443	Established	Internet
192.168.1.2	49257	17.143.162.214	5223	Established	Internet
127.0.0.1	49227	127.0.0.1	27015	Established	Internet
127.0.0.1	49200	127.0.0.1	65001	Established	Internet
192.168.1.2	49197	157.56.98.92	443	Established	Internet
127.0.0.1	49157	127.0.0.1	5354	Established	Internet
127.0.0.1	49156	127.0.0.1	5354	Established	Internet
127.0.0.1	27015	127.0.0.1	49227	Established	Internet
127.0.0.1	5354	127.0.0.1	49157	Established	Internet
127.0.0.1	5354	127.0.0.1	49156	Established	Internet

```
PS C:\> Get-NetTCPConnection | ? State -eq Established | ? RemoteAddress -notlike 127* | % { $_; Resolve-DnsName
$_RemoteAddress -type PTR -ErrorAction SilentlyContinue }
```

LocalAddress	LocalPort	RemoteAddress	RemotePort	State	AppliedSetting
192.168.1.2	59619	91.190.218.57	12350	Established	Internet
192.168.1.2	57993	213.199.179.175	40028	Established	Internet
192.168.1.2	54334	17.158.28.49	443	Established	Internet
192.168.1.2	54320	96.17.8.170	80	Established	Internet

```
Name      : 170.8.17.96.in-addr.arpa
QueryType : PTR
TTL       : 86377
Section   : Answer
NameHost  : a96-17-8-170.deploy.akamaitechnologies.com
```

192.168.1.2	54319	23.3.105.144	80	Established	Internet
-------------	-------	--------------	----	-------------	----------

```
Name      : 144.105.3.23.in-addr.arpa
QueryType : PTR
TTL       : 7
Section   : Answer
NameHost  : a23-3-105-144.deploy.static.akamaitechnologies.com
```

192.168.1.2	54147	65.55.68.119	443	Established	Internet
-------------	-------	--------------	-----	-------------	----------

```
Name      : 119.68.55.65.in-addr.arpa
QueryType : PTR
TTL       : 850
Section   : Answer
NameHost  : snt404-m.hotmail.com
```

192.168.1.2	49257	17.143.162.214	5223	Established	Internet
-------------	-------	----------------	------	-------------	----------

192.168.1.2	49197	157.56.98.92	443	Established	Internet
-------------	-------	--------------	-----	-------------	----------

```
Name      : 92.98.56.157.in-addr.arpa
QueryType : PTR
TTL       : 3600
Section   : Answer
NameHost  : bn1wns1011516.wns.windows.com
```