IT Scripting and Automation

Managing Users and
Basic Network configuration
in PowerShell

Lecturer: Art Ó Coileáin

Account Creation

Introduction:

Account creation for users is one of the most frequent tasks asked of System Administrators. Accordingly, it makes sense to automate this task in as much as possible.

We will be looking at an example of creating local user accounts using PowerShell, (*however it can be applied in an Active Directory scenario with a few modifications*).

Account Creation

Command:

The cmdlet to create a new local user account is:

New-LocalUser.

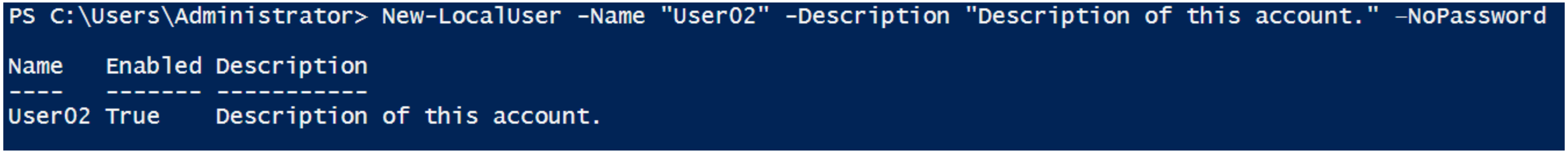
The cmdlet to create a new an Active Directory user account is:

New-ADUser.

Example:

To create a single local user account:

New-LocalUser -Name "User02" -Description "Description of this account." –NoPassword



```
PS C:\Users\Administrator> New-LocalUser -Name "User02" -Description "Description of this account." –NoPassword

Name   Enabled Description
----   ------- -----------
User02 True    Description of this account.
```

Account Creation

<u>…continuing Example</u>:

To create a single local user account:

```
New-LocalUser -Name "User02" -Description "Description of this account." –NoPassword
```

This command creates a local user account, without specifying the ***“AccountExpires”*** or ***“Password”*** parameters.

Therefore, the account doesn't expire or have a password by default.

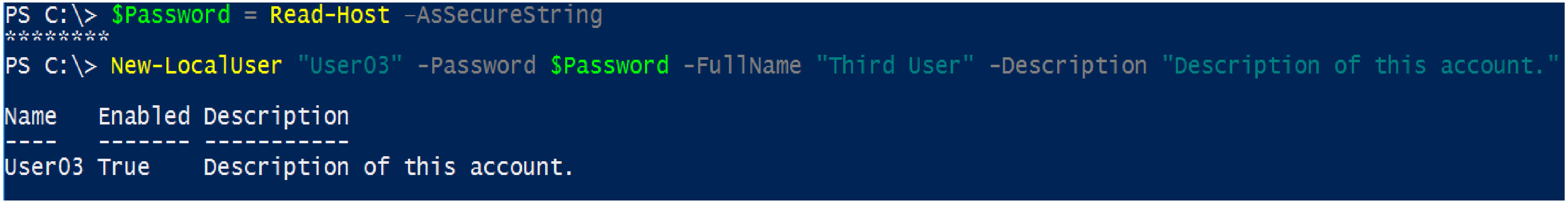
Account Creation

Example2:

To create a user account with a password:

PS C:\> $Password = Read-Host –AsSecureString

PS C:\>New-LocalUser "User03" -Password $Password -FullName "Third User" -Description "Description of this account.“



```
PS C:\> $Password = Read-Host -AsSecureString
********
PS C:\> New-LocalUser "User03" -Password $Password -FullName "Third User" -Description "Description of this account."

Name   Enabled Description
----   ------- -----------
User03 True    Description of this account.
```

The first command prompts you for a password by using the Read-Host cmdlet.

The command stores the password as a secure string in the $Password variable.

Continues….

Account Creation

...continuing Example:

To create a user account which has a password:

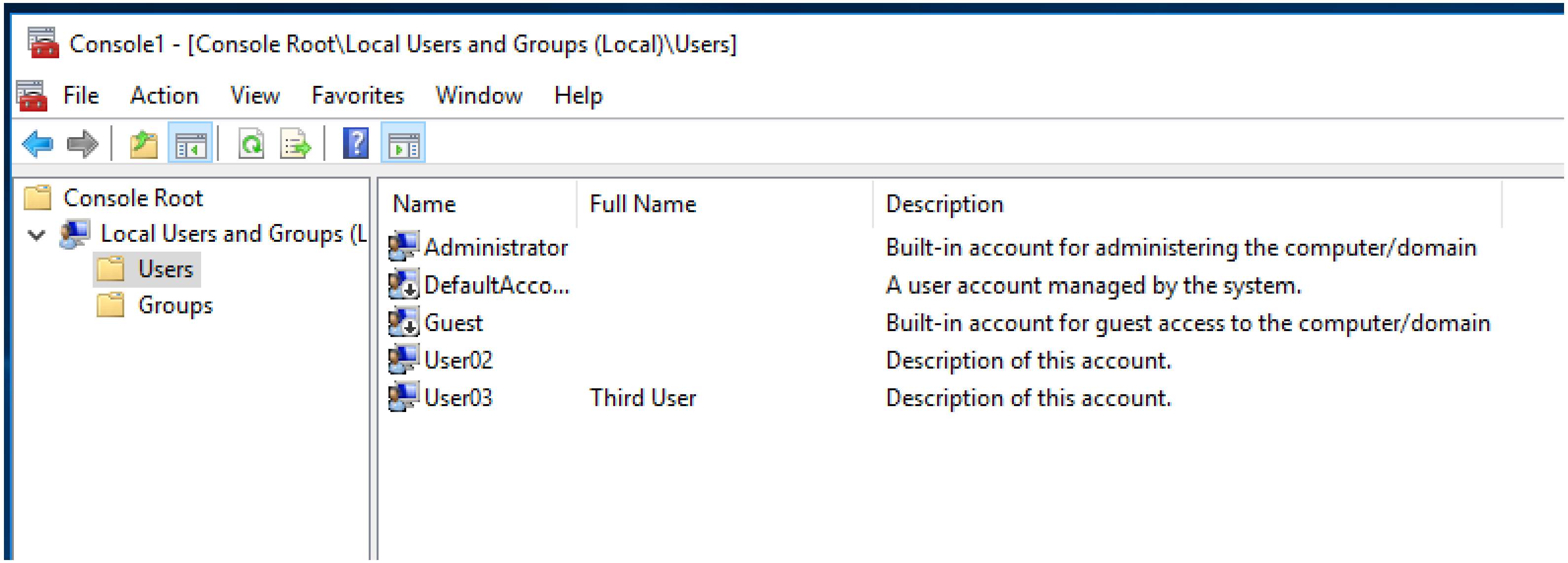
```
PS C:\> $Password = Read-Host –AsSecureString
PS C:\>New-LocalUser "User03" -Password $Password -FullName "Third User" -Description "Description of this account.“
```

The second command creates a local user account by using the password stored in $Password.

The command specifies a user name, full name, and description for the user account.

Account Creation

List of accounts on local machine using local administrative tools, after running both PowerShell **New-LocalUser** commands :



Console1 - [Console Root\Local Users and Groups (Local)\Users]

Name	Full Name	Description
Administrator		Built-in account for administering the computer/domain
DefaultAcco...		A user account managed by the system.
Guest		Built-in account for guest access to the computer/domain
User02		Description of this account.
User03	Third User	Description of this account.

Other cmdlets

Other Useful cmdlets:

Other useful cmdlets for local account and group management are:

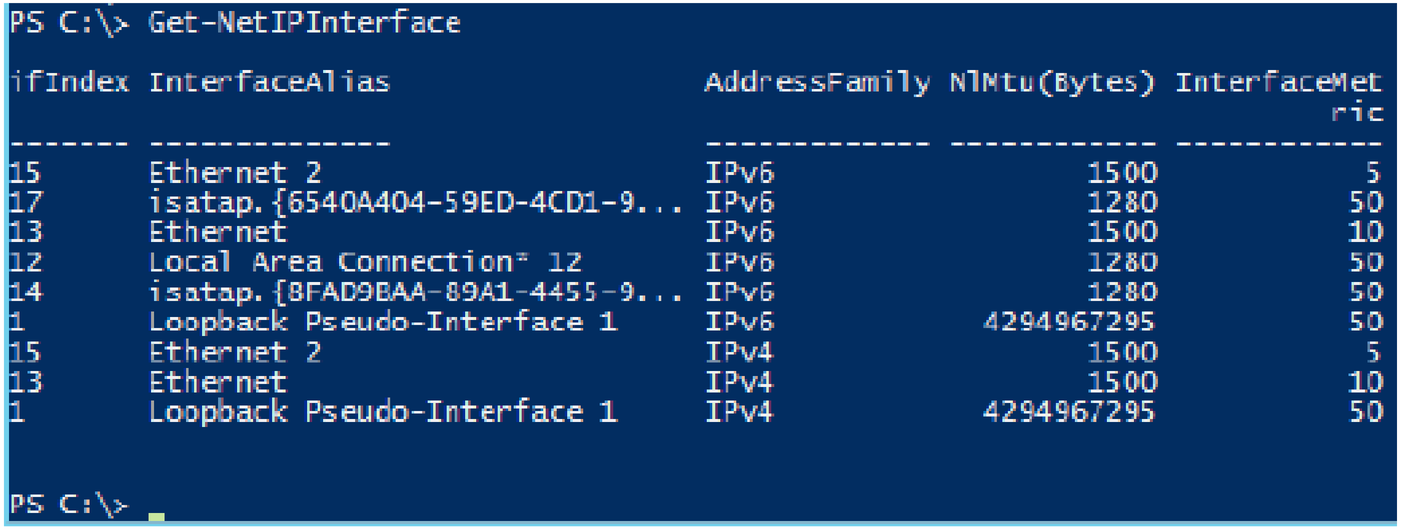
Cmdlet	Description
Add-LocalGroupMember	Adds members to a local group.
Disable-LocalUser	Disables a local user account.
Enable-LocalUser	Enables a local user account.
Get-LocalGroup	Gets the local security groups.
Get-LocalGroupMember	Gets members from a local group.
Get-LocalUser	Gets local user accounts.
New-LocalGroup	Creates a local security group.
Remove-LocalGroup	Deletes local security groups.
Remove-LocalGroupMember	Removes members from a local group.
Remove-LocalUser	Deletes local user accounts.
Rename-LocalGroup	Renames a local security group.
Rename-LocalUser	Renames a local user account.
Set-LocalGroup	Changes a local security group.
Set-LocalUser	Modifies a local user account.

Configuring Static Networking

Introduction:

When managing and configuring new PC’s/servers System Administrators often need to configure network interfaces. We can automate this task using PowerShell scripts via cmdlet’s .

To find the some basic information regarding which interface to set we can execute the **Get-NetIPInterface** cmdlet:



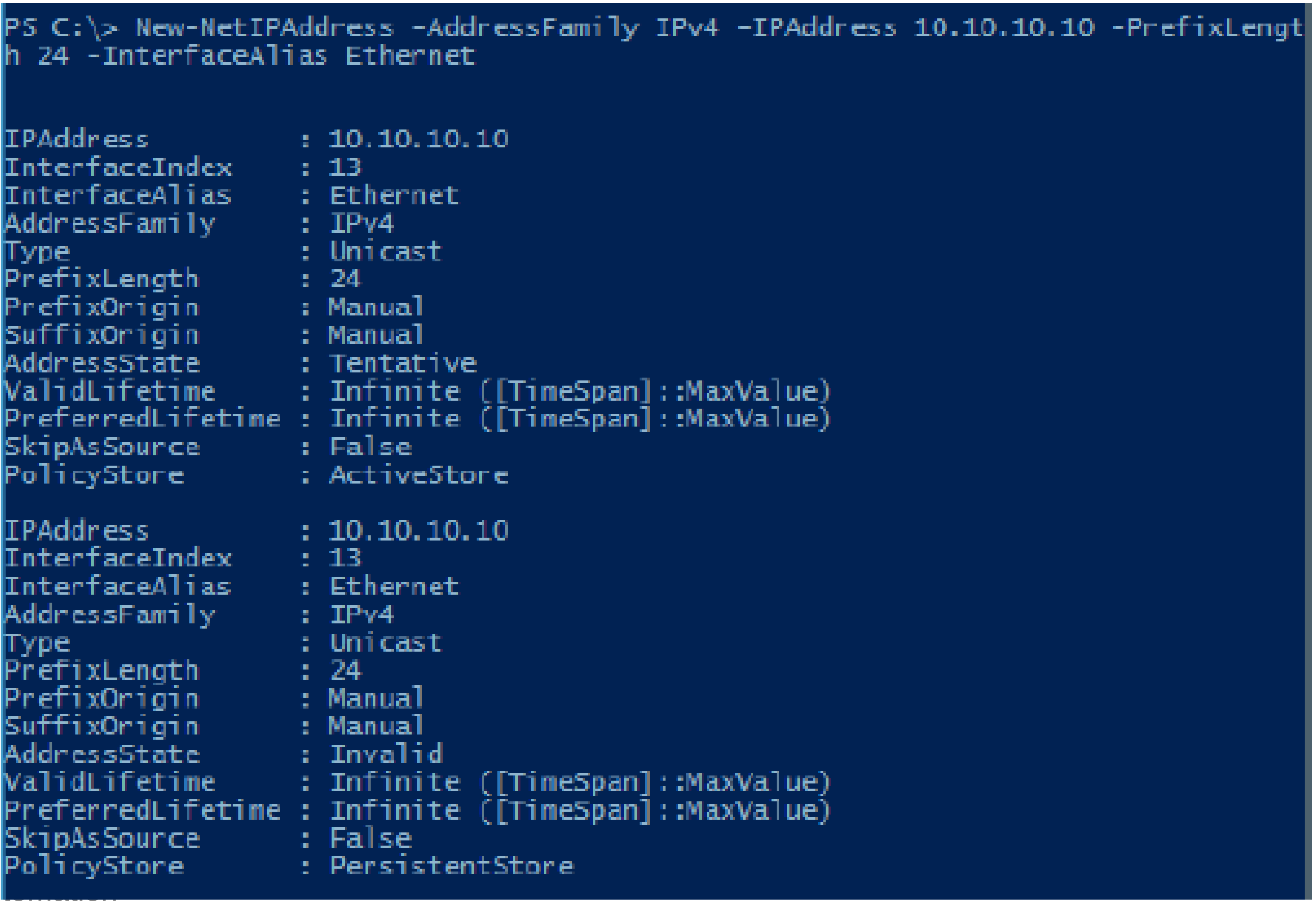
```
PS C:\> Get-NetIPInterface

ifIndex InterfaceAlias                  AddressFamily NlMtu(Bytes) InterfaceMet
                                                                            ric
------- --------------                  ------------- ------------ ------------
15      Ethernet 2                      IPv6                  1500            5
17      isatap.{6540A404-59ED-4CD1-9... IPv6                  1280           50
13      Ethernet                        IPv6                  1500           10
12      Local Area Connection* 12       IPv6                  1280           50
14      isatap.{8FAD9BAA-89A1-4455-9... IPv6                  1280           50
1       Loopback Pseudo-Interface 1     IPv6            4294967295           50
15      Ethernet 2                      IPv4                  1500            5
13      Ethernet                        IPv4                  1500           10
1       Loopback Pseudo-Interface 1     IPv4            4294967295           50


PS C:\> _
```

Configuring Static Networking

Setting the IP configuration:

To set the IP information one can use the **New-NetIPAddress** cmdlet as follows:

PS C:\> New-NetIPAddress -AddressFamily IPv4 -IPAddress 10.10.10.10 –PrefixLength 24 -InterfaceAlias Ethernet



```
PS C:\> New-NetIPAddress -AddressFamily IPv4 -IPAddress 10.10.10.10 -PrefixLengt
h 24 -InterfaceAlias Ethernet


IPAddress         : 10.10.10.10
InterfaceIndex    : 13
InterfaceAlias    : Ethernet
AddressFamily     : IPv4
Type              : Unicast
PrefixLength      : 24
PrefixOrigin      : Manual
SuffixOrigin      : Manual
AddressState      : Tentative
ValidLifetime     : Infinite ([TimeSpan]::MaxValue)
PreferredLifetime : Infinite ([TimeSpan]::MaxValue)
SkipAsSource      : False
PolicyStore       : ActiveStore

IPAddress         : 10.10.10.10
InterfaceIndex    : 13
InterfaceAlias    : Ethernet
AddressFamily     : IPv4
Type              : Unicast
PrefixLength      : 24
PrefixOrigin      : Manual
SuffixOrigin      : Manual
AddressState      : Invalid
ValidLifetime     : Infinite ([TimeSpan]::MaxValue)
PreferredLifetime : Infinite ([TimeSpan]::MaxValue)
SkipAsSource      : False
PolicyStore       : PersistentStore
```

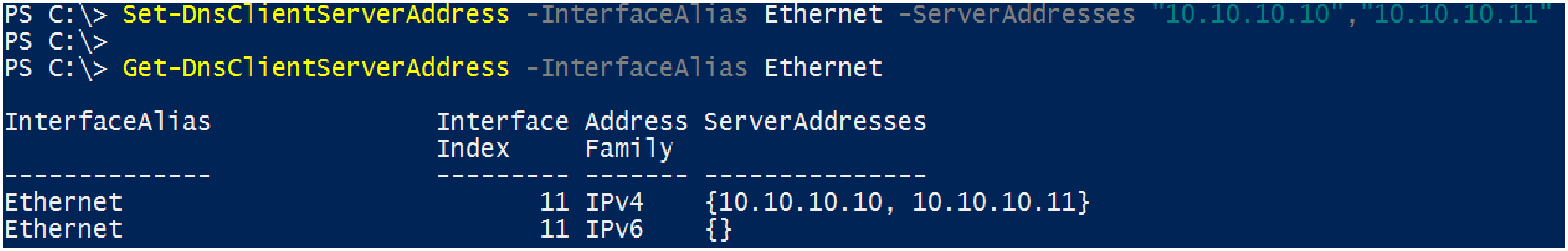
Setting DNS Servers

Setting DNS Servers:

To set the DNS Servers one can use the Set-DnsClientServerAddress cmdlet as follows:

Example:

PS C:\> Set-DnsClientServerAddress –InterfaceAlias Ethernet -ServerAddresses "10.10.10.10", "10.10.10.11“



```
PS C:\> Set-DnsClientServerAddress -InterfaceAlias Ethernet -ServerAddresses "10.10.10.10","10.10.10.11"
PS C:\>
PS C:\> Get-DnsClientServerAddress -InterfaceAlias Ethernet

InterfaceAlias               Interface Address ServerAddresses
                             Index     Family
--------------               --------- ------- ---------------
Ethernet                            11 IPv4    {10.10.10.10, 10.10.10.11}
Ethernet                            11 IPv6    {}
```

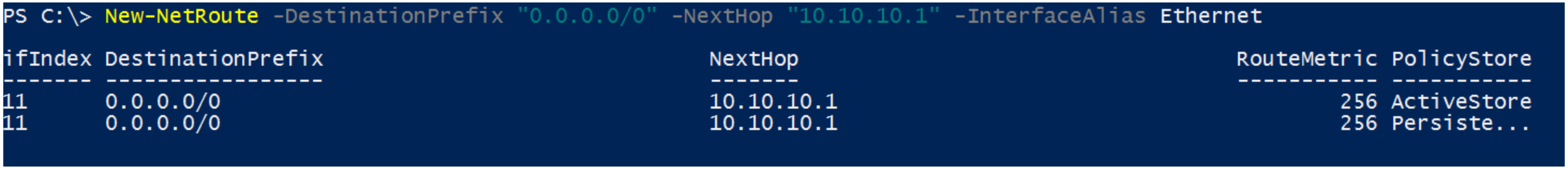
Configuring Static Networking

Setting the default gateway/route :

To Set the default gateway/route of the interface one can use the **New-NetRoute** cmdlet as follows:

Example:

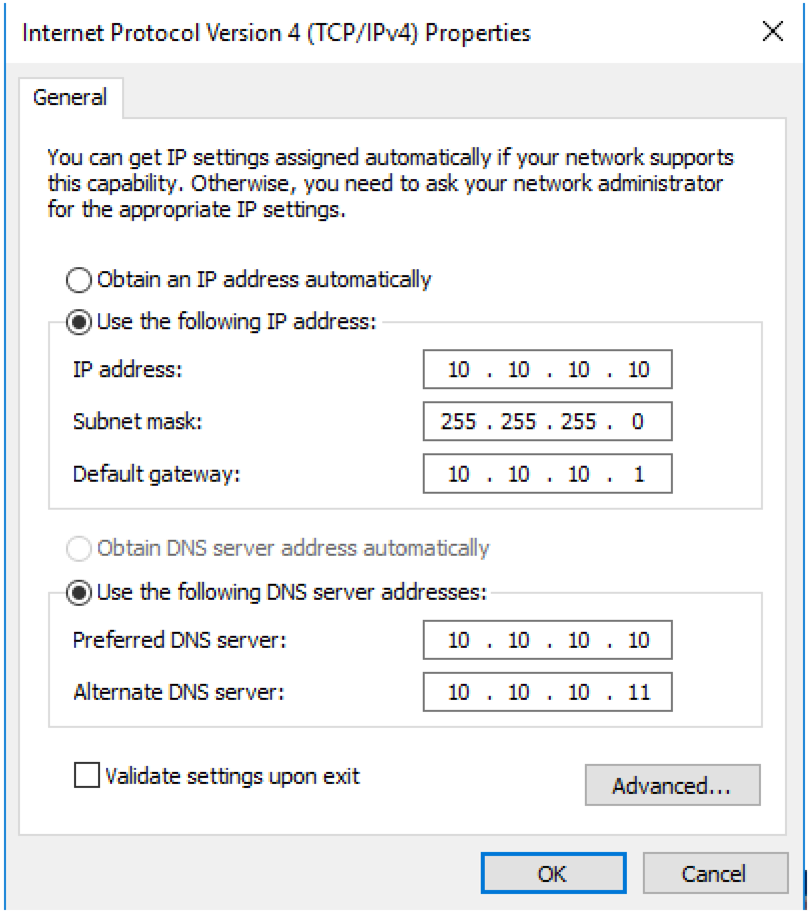
PS C:\> New-NetRoute -DestinationPrefix "0.0.0.0/0" -NextHop "10.10.10.1" -InterfaceAlias Ethernet



```
PS C:\> New-NetRoute -DestinationPrefix "0.0.0.0/0" -NextHop "10.10.10.1" -InterfaceAlias Ethernet

ifIndex DestinationPrefix                              NextHop                                  RouteMetric PolicyStore
------- -----------------                              -------                                  ----------- -----------
11      0.0.0.0/0                                      10.10.10.1                                       256 ActiveStore
11      0.0.0.0/0                                      10.10.10.1                                       256 Persiste...
```

Basic Network Config

The result of the applying of the previous cmdlet's to our IPv4 Ethernet interface out windows network interface will look like the following:



Internet Protocol Version 4 (TCP/IPv4) Properties

General

You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

○ Obtain an IP address automatically

● Use the following IP address:

IP address:	10 . 10 . 10 . 10
Subnet mask:	255 . 255 . 255 . 0
Default gateway:	10 . 10 . 10 . 1

○ Obtain DNS server address automatically

● Use the following DNS server addresses:

Preferred DNS server:	10 . 10 . 10 . 10
Alternate DNS server:	10 . 10 . 10 . 11

□ Validate settings upon exit

IPv6 Addressing

IPv6:

In addition to configuring IPv4, PowerShell can also configure IPv6addresses. The process for configuring static IPv6 addressing is exactly the same as IPv4, the only change is the addresses themselves:

```
New-NetIPAddress -AddressFamily IPv6 -IPAddress 2001:db8:1::10 `
-PrefixLength 64 -InterfaceAlias Ethernet
```

```
New-NetRoute -DestinationPrefix ::/0 -NextHop 2001:db8:1::1 `
-InterfaceAlias Ethernet
```

```
Set-DnsClientServerAddress -InterfaceAlias Ethernet `
-ServerAddresses "2001:db8:1::10", "2001:db8:1::11"
```

Other cmdlets

Other Useful cmdlets:

Other useful cmdlets for network management are:

Cmdlet	Description
Find-NetRoute	Finds the best local IP address and the best route to reach a remote address.
Get-NetIPAddress	Gets the IP address configuration.
Get-NetIPConfiguration	Gets IP network configuration.
Get-NetIPInterface	Gets an IP interface.
Get-NetRoute	Gets the IP route information from the IP routing table.
Get-NetTCPConnection	Gets TCP connections.
Get-NetTCPSetting	Gets information about TCP settings and configuration.
New-NetIPAddress	Creates and configures an IP address.
New-NetRoute	Creates a route in the IP routing table.
Remove-NetIPAddress	Removes an IP address and its configuration.
Remove-NetRoute	Removes IP routes from the IP routing table.
Set-NetIPAddress	Modifies the configuration of an IP address.
Set-NetIPInterface	Modifies an IP interface.
Set-NetIPv4Protocol	Modifies information about the IPv4 Protocol configuration.
Set-NetIPv6Protocol	Modifies the IPv6 protocol configuration.
Set-NetRoute	Modifies an entry or entries in the IP routing table.
Set-NetTCPSetting	Modifies a TCP setting.
Test-NetConnection	Displays diagnostic information for a connection.