How to Use Remove-NetworkSwitchEthernetPortIPAddress in PowerShell

In this article, we will explore the usage of the Remove-NetworkSwitchEthernetPortIPAddress cmdlet in PowerShell and its significance in the Windows environment. This cmdlet allows us to remove an IP address from an Ethernet port on a network switch. By understanding how to use this cmdlet, we can effectively manage and configure IP addresses on our Windows systems.

Examples:

1. To remove an IP address from an Ethernet port using Remove-NetworkSwitchEthernetPortIPAddress, follow these steps:

   ```
   PS C:\> $ethernetPort = Get-NetAdapter -Name "Ethernet Port"
   PS C:\> $ipAddress = "192.168.1.100"
   PS C:\> $subnetMask = "255.255.255.0"
   PS C:\> $gateway = "192.168.1.1"

   PS C:\> $ethernetPort | Remove-NetworkSwitchEthernetPortIPAddress -IP
   Address $ipAddress -SubnetMask $subnetMask -DefaultGateway $gateway
   ```

   This example retrieves the Ethernet port named "Ethernet Port" using the Get-NetAdapter cmdlet. Then, it removes the specified IP address, subnet mask, and default gateway from the Ethernet port using the Remove-NetworkSwitchEthernetPortIPAddress cmdlet.

2. If you want to remove multiple IP addresses from an Ethernet port, you can use the following example:

   ```
   PS C:\> $ethernetPort = Get-NetAdapter -Name "Ethernet Port"
   PS C:\> $ipAddresses = "192.168.1.100", "192.168.1.101", "192.168.1.1
   02"
   PS C:\> $subnetMask = "255.255.255.0"
   PS C:\> $gateway = "192.168.1.1"

   PS C:\> $ethernetPort | Remove-NetworkSwitchEthernetPortIPAddress -IP
   Address $ipAddresses -SubnetMask $subnetMask -DefaultGateway $gateway
   ```

   In this example, we specify an array of IP addresses to remove from the Ethernet port. The cmdlet will remove all the specified IP addresses, along with the subnet mask and default gateway.