

## Understanding Network Connections in Windows

In this article, we will explore the topic of network connections and their importance in the Windows environment. Network connections play a crucial role in enabling communication between devices and allowing users to access resources on a network. Understanding how network connections work in Windows can help users troubleshoot connectivity issues, optimize performance, and ensure secure connections.

To align this topic with the Windows environment, we will focus on the network connection management tools and techniques available in Windows operating systems. These include the Command Prompt (CMD) and the PowerShell, which are commonly used in Windows for network configuration and troubleshooting.

### Examples:

#### 1. Checking Network Connection Status:

- CMD: Use the command `ipconfig` to display the IP configuration and network interfaces of the device.
- PowerShell: Execute the command `Get-NetAdapter` to retrieve information about network adapters and their status.

#### 2. Configuring Network Settings:

- CMD: Utilize the `netsh` command to configure network settings, such as setting a static IP address or modifying DNS settings.
- PowerShell: Use the `Set-NetIPAddress` cmdlet to configure IP addresses or the `Set-DnsClientServerAddress` cmdlet to modify DNS server addresses.

#### 3. Troubleshooting Network Connectivity:

- CMD: Employ the `ping` command to test network connectivity to a specific IP address or domain name.
- PowerShell: Execute the `Test-NetConnection` cmdlet to diagnose network connectivity issues, including testing TCP/IP connectivity, port availability, and routing.

#### 4. Creating Virtual Private Network (VPN) Connections:

- CMD: Use the `rasdial` command to establish a VPN connection from the Command Prompt.
- PowerShell: Utilize the `Add-VpnConnection` cmdlet to create a VPN connection

