**Project - Comprehensive IT Support Automation with PowerShell**

**Step 1: Define the Key IT Support Tasks**

Before writing the PowerShell script, we need to clearly define **what** we are automating and **why** each task is important. This step ensures the automation is structured, efficient, and meets real IT support needs.

**IT Support Tasks to Automate**

Based on common IT support responsibilities, we will focus on **four key areas**:

**User Management (Active Directory Automation)**

**What it does?**

* Creates new Active Directory (AD) users.
* Modifies existing user attributes (e.g., adding users to groups).
* Disables or removes inactive accounts automatically.

**Why automate it?**

* **Saves time** for IT admins when onboarding or managing users.
* **Reduces errors** by using standardized account creation and modification.
* **Improves security** by automatically disabling inactive accounts.

**System Health Monitoring**

**What it does?**

* Checks **CPU usage, memory usage, and disk space** on a system.
* Sends alerts if resources exceed a predefined threshold (e.g., **CPU > 80%** or **Disk usage > 90%**).
* Logs system performance for future analysis.

**Why automate it?**

* Helps **proactively detect issues** before they cause system slowdowns.
* Ensures **IT support teams are notified** about potential failures.
* Saves time compared to manually checking each system.

**Software Deployment**

**What it does?**

* Checks if a particular software is **already installed**.
* If not installed, **installs the software automatically** using MSI/EXE.
* Can be used for **mass software deployments** across multiple machines.

**Why automate it?**

* Ensures **consistent installations** across all workstations.
* **Reduces human intervention** in software setup.
* Prevents IT admins from manually installing software on multiple computers.

**Network Troubleshooting**

**What it does?**

* **Checks internet connectivity** by pinging external services (e.g., Google DNS 8.8.8.8).
* **Pings company servers** to ensure they are reachable.
* **Traces network routes** to detect connectivity issues.

**Why automate it?**

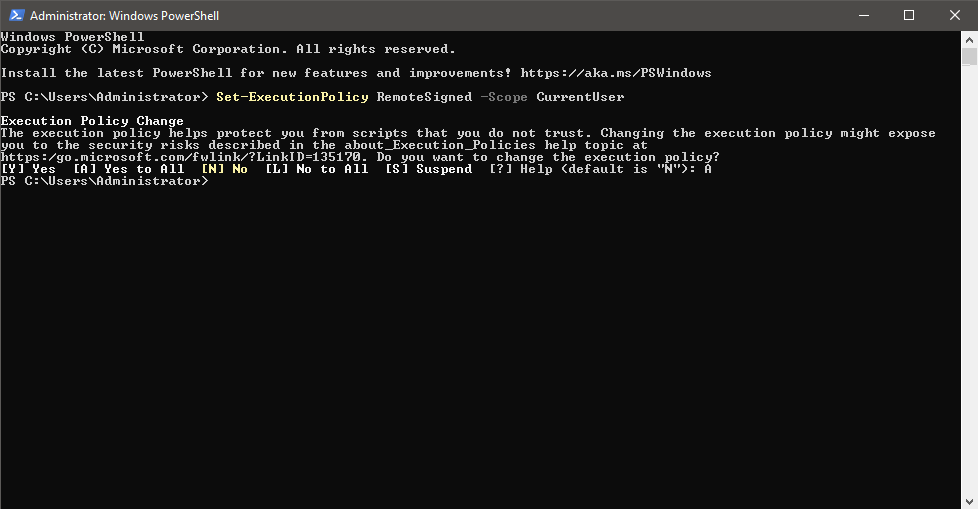
* Speeds up **network issue diagnosis** for IT support teams.
* Allows IT staff to **quickly detect and fix outages**.
* Provides useful insights for **resolving intermittent network issues**.

**Final Summary of Step 1**

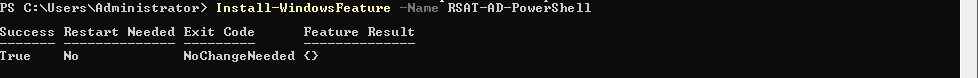
|  |  |  |
| --- | --- | --- |
| **Task** | **What It Does?** | **Why Automate?** |
| **User Management** | Automates user creation, modification, and deactivation in Active Directory | Saves time and ensures consistency |
| **System Health Monitoring** | Monitors CPU, memory, and disk usage and sends alerts | Proactive issue detection |
| **Software Deployment** | Installs software if not already installed | Ensures uniform installations |
| **Network Troubleshooting** | Pings servers, checks internet, and traces network routes | Speeds up troubleshooting |

**Step 2: Prepare PowerShell and System Permissions**

* Ensure you have **admin privileges** for running PowerShell scripts.
* Enable script execution using:



* Use **Active Directory Module** for user management (Install-WindowsFeature -Name RSAT-AD-PowerShell).



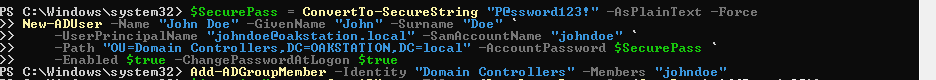
* Ensure **WinRM (Windows Remote Management) is enabled** (Enable-PSRemoting -Force).



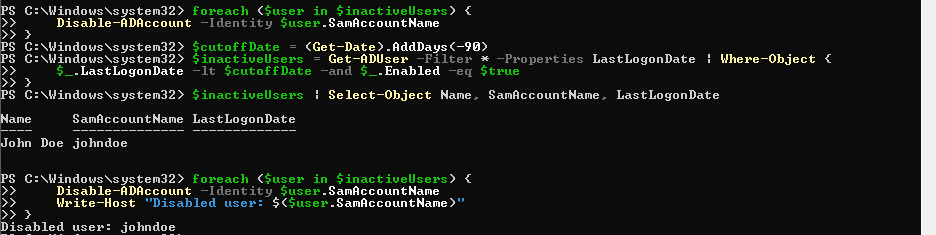
**Step 3: Write the PowerShell Scripts for Each Task**

**Automate User Management (Active Directory) -**

1. Creates a new user
2. Modifies existing users
3. Disables inactive users

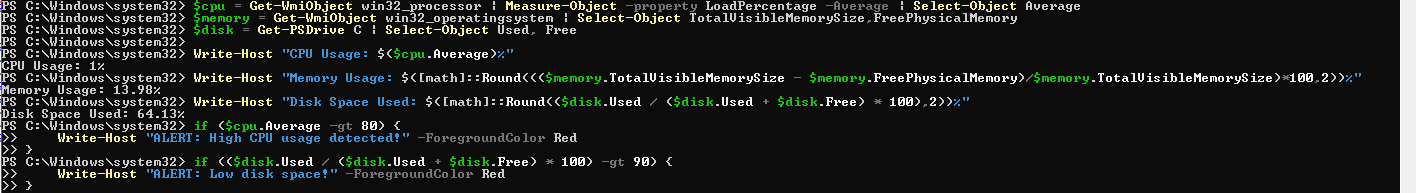






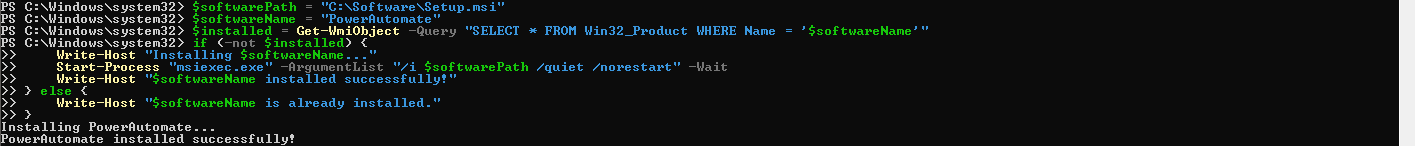
**Automate System Health Monitoring -**

1. Checks CPU, Memory, Disk Usage
2. Sends an alert if usage is high



**Automate Software Deployment -**

1. Installs software
2. Checks if software is already installed



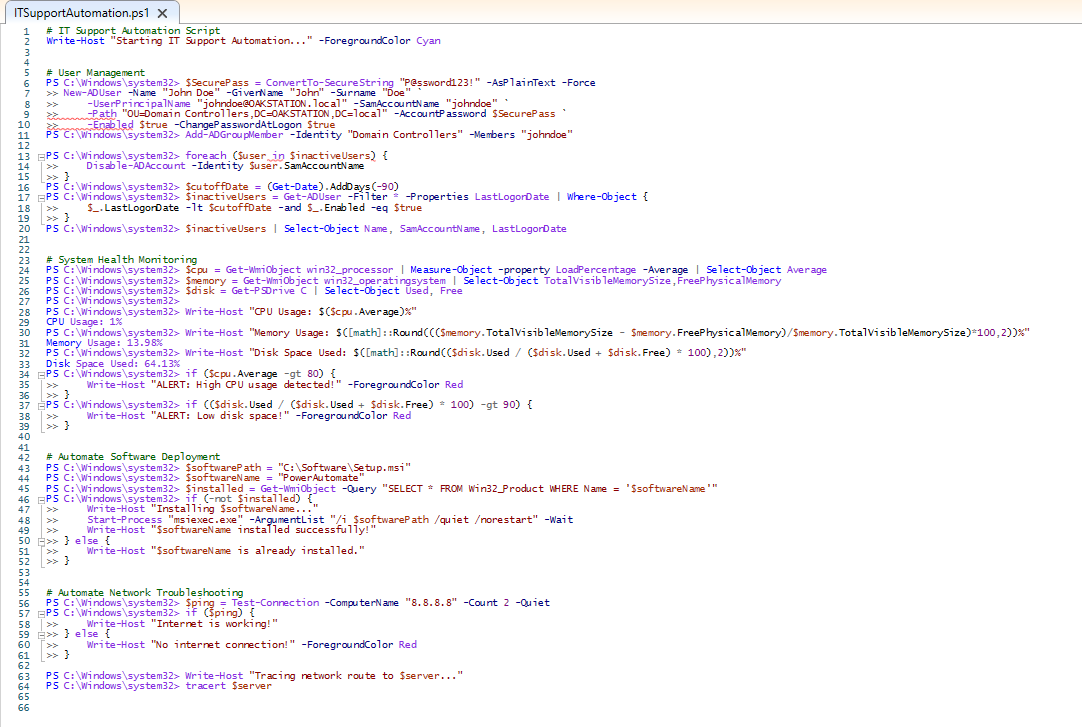
**Automate Network Troubleshooting**

1. Checks internet connection
2. Pings key network resources
3. Traces network route



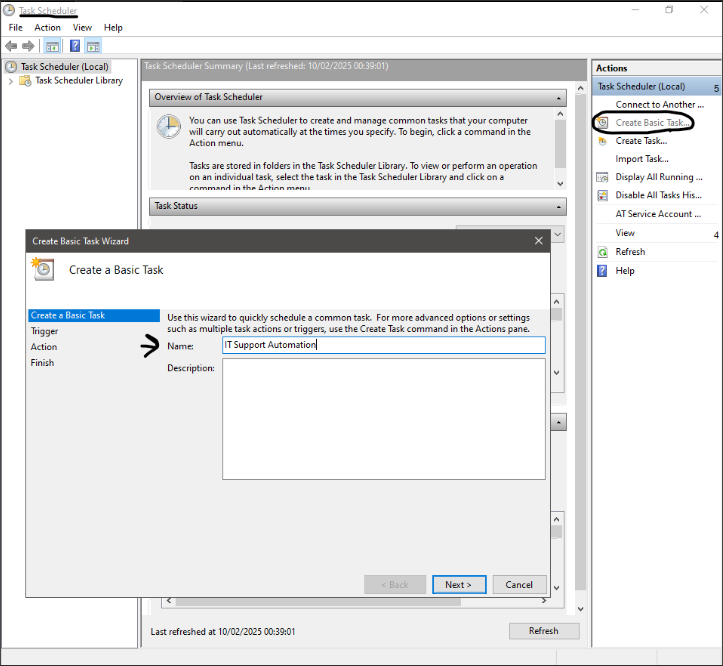
**Step 4: Combine Everything into One Automated Script**

Now, let's **combine** all tasks into a single script (ITSupportAutomation.ps1):

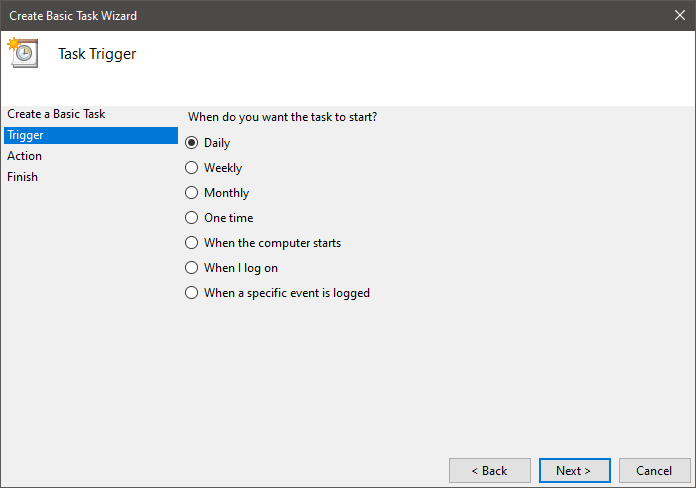


**Step 5: Automate Execution (Task Scheduler)**

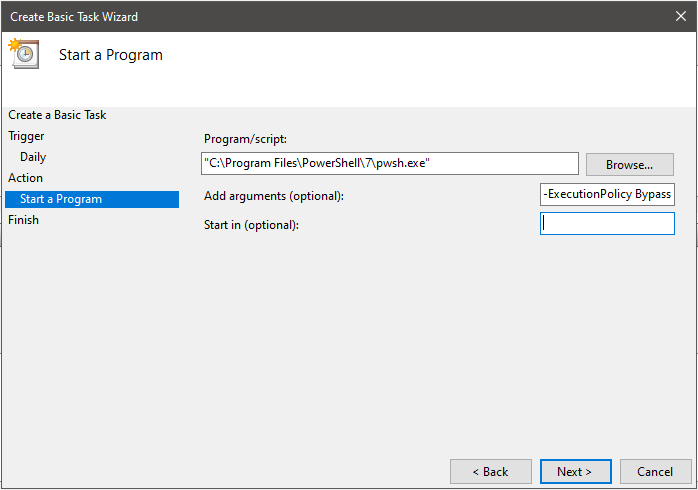
* **Schedule the script** to run automatically using Task Scheduler:
  1. Open **Task Scheduler** (taskschd.msc).
  2. Click **Create Basic Task**.
  3. Name: **IT Support Automation**.



* 1. Trigger: **Daily** or **At Logon**.



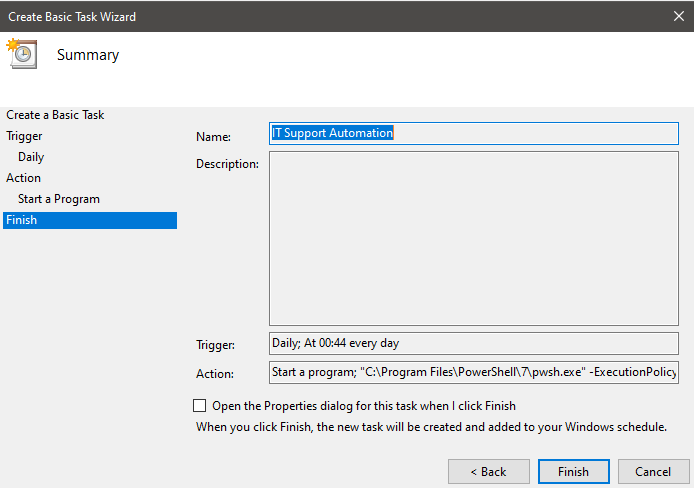
* 1. Action: **Start a Program** → Browse for PowerShell.exe.

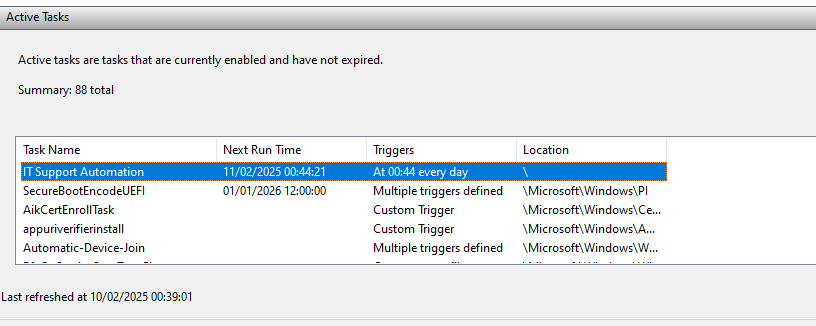


* 1. Add arguments:

-ExecutionPolicy Bypass -File "C:\Scripts\ITSupportAutomation.ps1"

* 1. Click **Finish**.





**You now have a fully automated IT support script!**  
This script improves **efficiency**, **reduces errors**, and **saves time**.