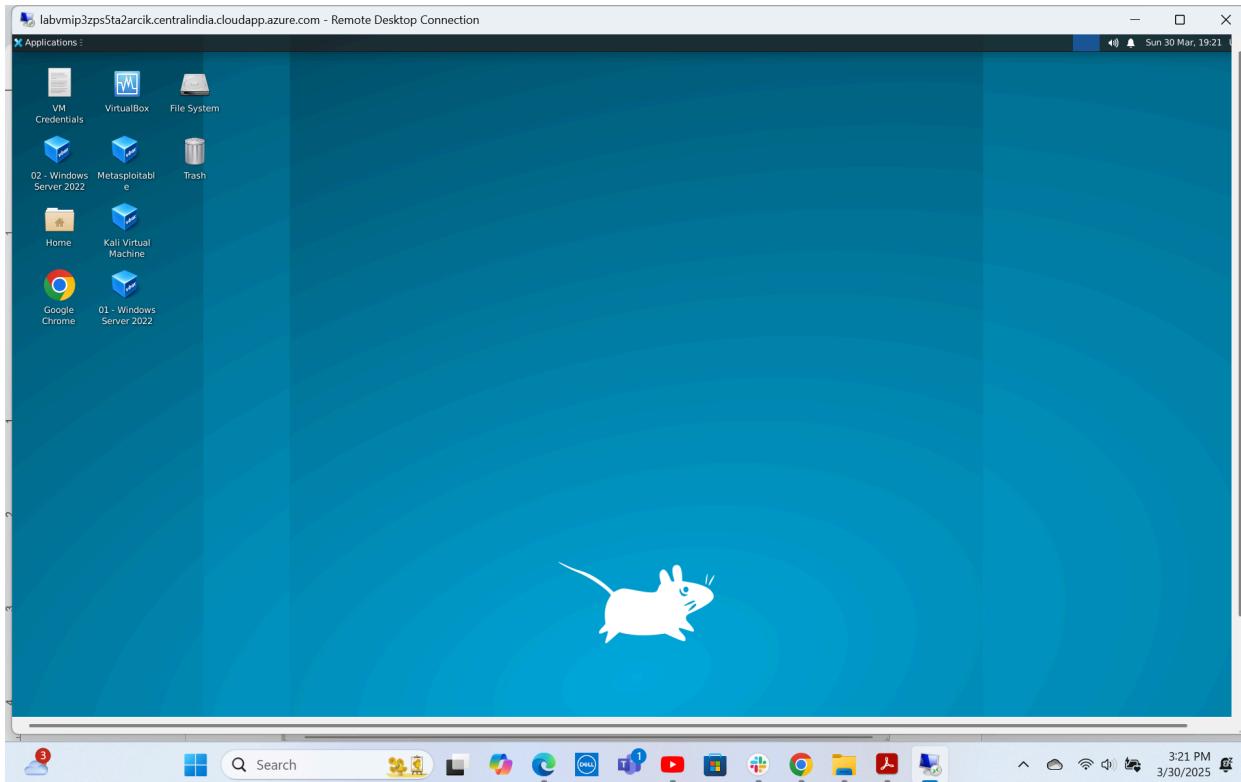


# **Project Topic: Automating Common IT Tasks By Janefrances Nwachukwu**

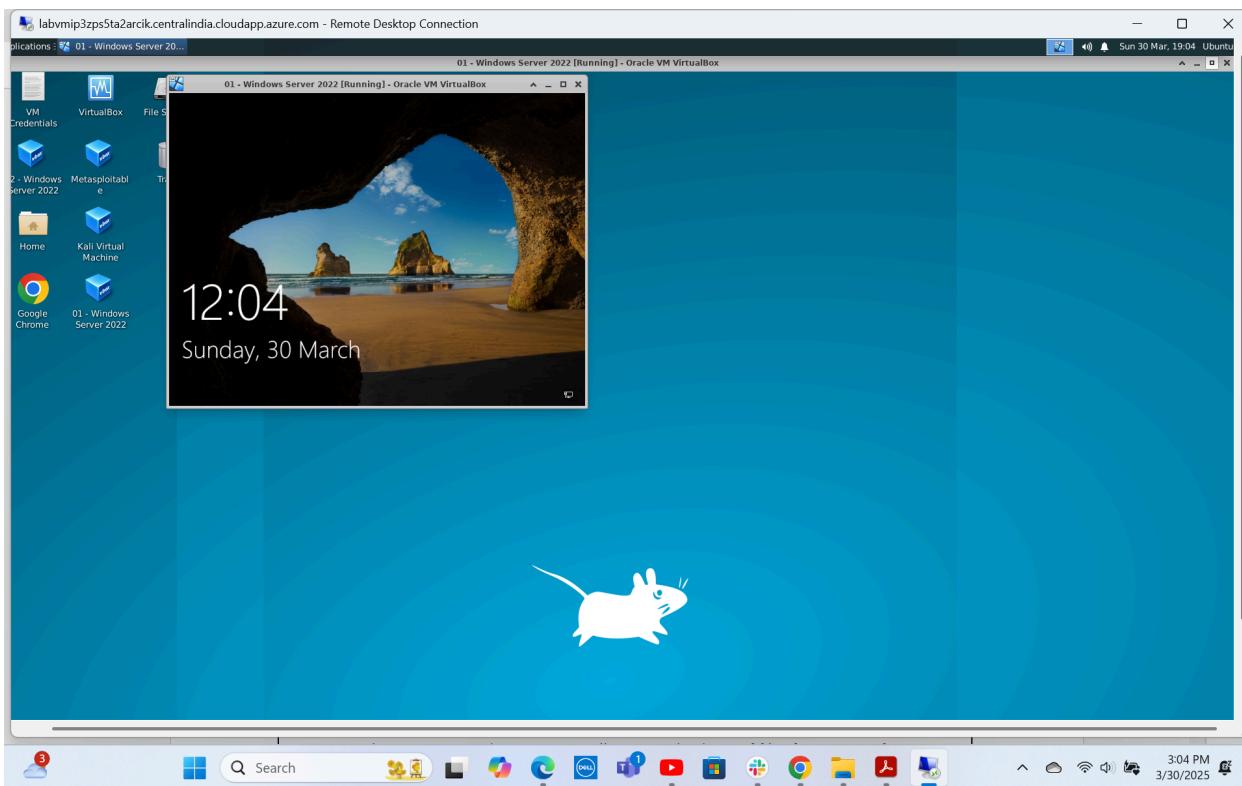
## **Output Screenshots**

## **Step 1: Evaluated strategies for automating system monitoring**

## **1.1 Opening Windows Server 22**



1.2



### 1.3 Logged in as an Administrator

CS - Operating Systems and Networking Essentials

Sessions attended : 60%

PG\_Cybersecurity\_Lab

This Lab will get reset on 30th March 2025, 10:32 PM

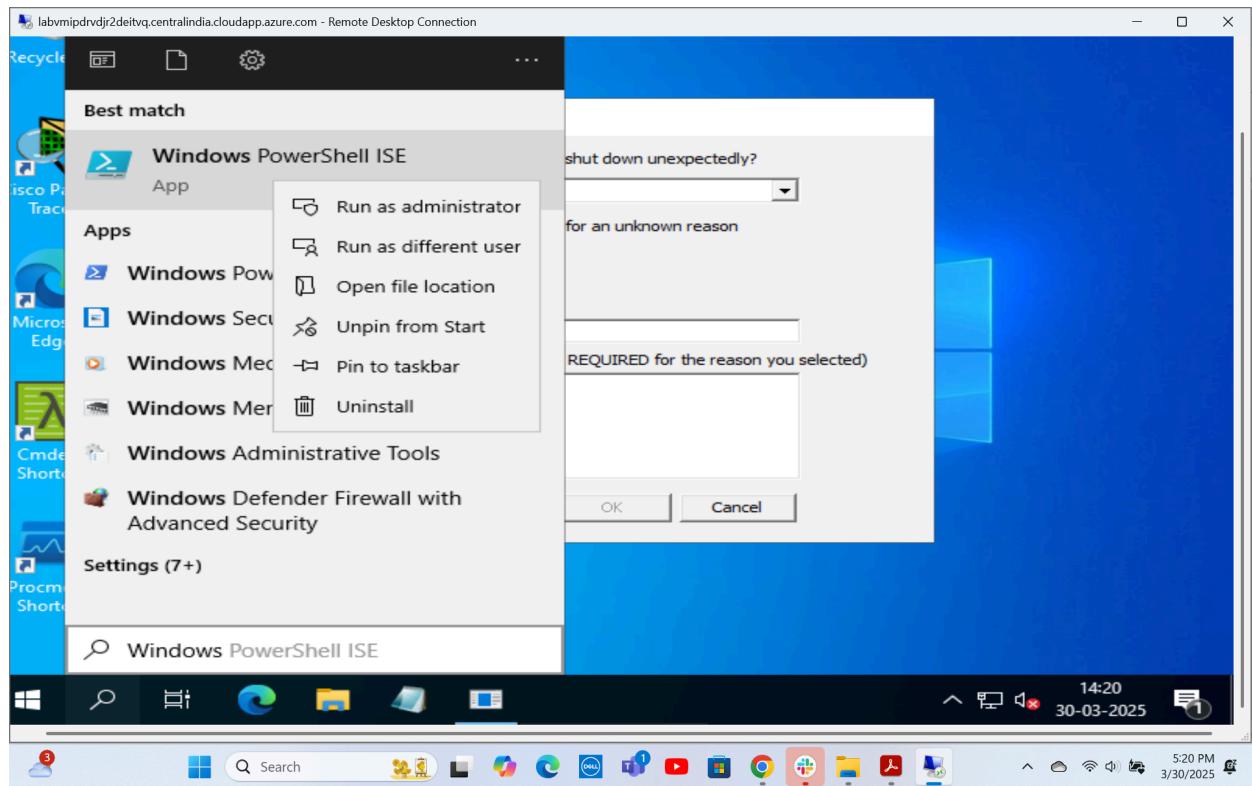
Administrator

Administrator

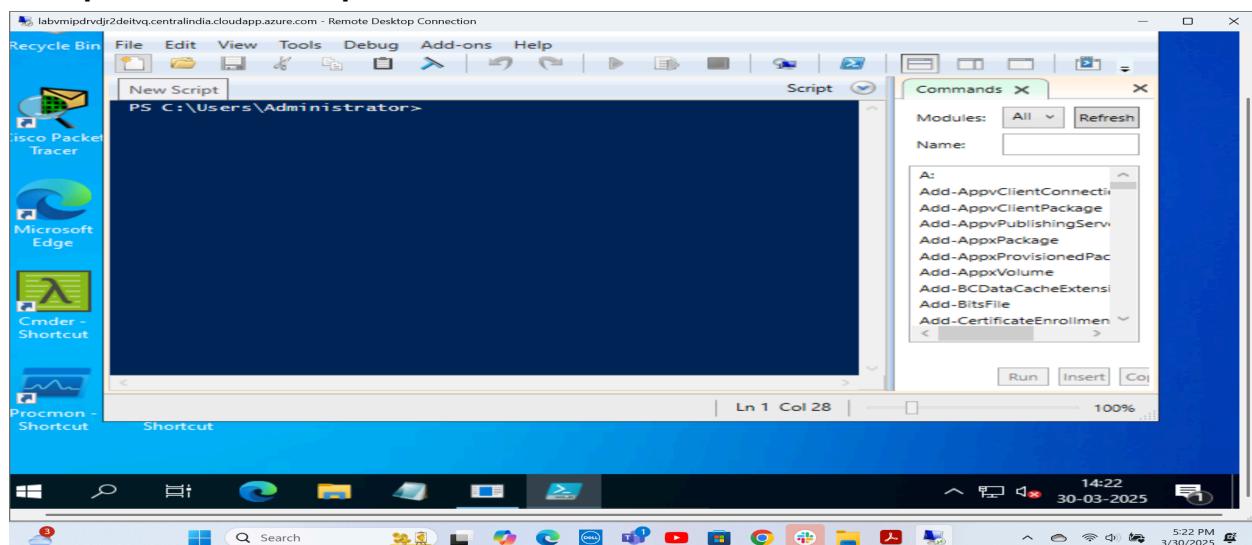
05h : 21m : 28s | Sun 30 Mar, 23:08 Ubuntu

7:08 PM 3/30/2025

## 1.4 Running PowerShell as an Administrator



## 1.5 Opened a new script



## 1.6 Created a new script

labvmipdrvdjr2deitvq.centralindia.cloudapp.azure.com - Remote Desktop Connection

01 - Windows Server 2022 [Running] - Oracle VM VirtualBox

Administrator: Windows PowerShell ISE

```
MonitorSystemEvents.ps1
1 # Script to monitor system events for security logs
2
3 $BeginTime = (Get-Date).AddHours(-24)
4
5 $Events = Get-WinEvent -LogName Security | Where-Object { $_.TimeCreated -ge $BeginTime }
6
7 $Events | Select-Object TimeCreated, Id, LevelDisplayName, Message | Format-Table
8
9 Write-Host "Security Log Report generated at $env:USERPROFILE\Desktop\SecurityLogReport.txt"
```

PS C:\Users\Administrator\Desktop>

Completed Shortcut Ln 7 Col 36 100%

This screenshot shows a Windows Server 2022 desktop environment. A PowerShell ISE window is open, displaying a script named 'MonitorSystemEvents.ps1'. The script uses the Get-WinEvent cmdlet to retrieve security log events from the last 24 hours and formats them into a table. The output is directed to a file named 'SecurityLogReport.txt' located in the user's desktop folder. The desktop background is blue, and the taskbar at the bottom includes icons for various Microsoft applications like Edge, File Explorer, and Mail.

labvmipdrvdjr2deitvq.centralindia.cloudapp.azure.com - Remote Desktop Connection

01 - Windows Server 2022 [Running] - Oracle VM VirtualBox

Administrator: Windows PowerShell ISE

File Edit View Tools Debug Add-ons Help

New Ctrl+N

Open... Ctrl+O

Save Ctrl+S

Save As... (highlighted)

Run F5

Run Selection F8

Stop Operation Ctrl+Break

Close Ctrl+F4

New PowerShell Tab Ctrl+T

Close PowerShell Tab Ctrl+W

New Remote PowerShell Tab... Ctrl+Shift+R

Start PowerShell.exe Ctrl+Shift+P

C:\Users\Administrator\Desktop\MonitorSystemEvents.ps1

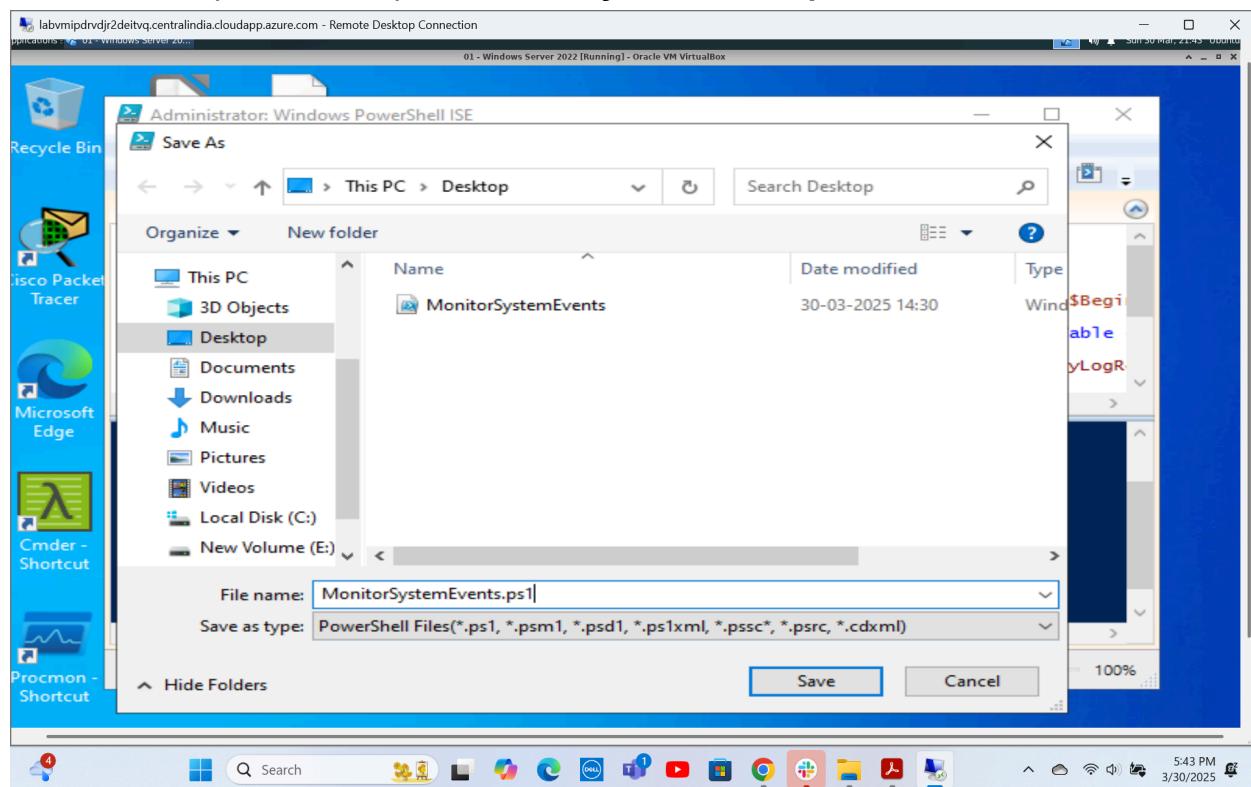
Exit Alt+F4

```
{ $_.TimeCreated -ge $BeginTime } | Select-Object TimeCreated, Id, LevelDisplayName, Message | Format-Table > $env:USERPROFILE\Desktop\SecurityLogReport.txt
```

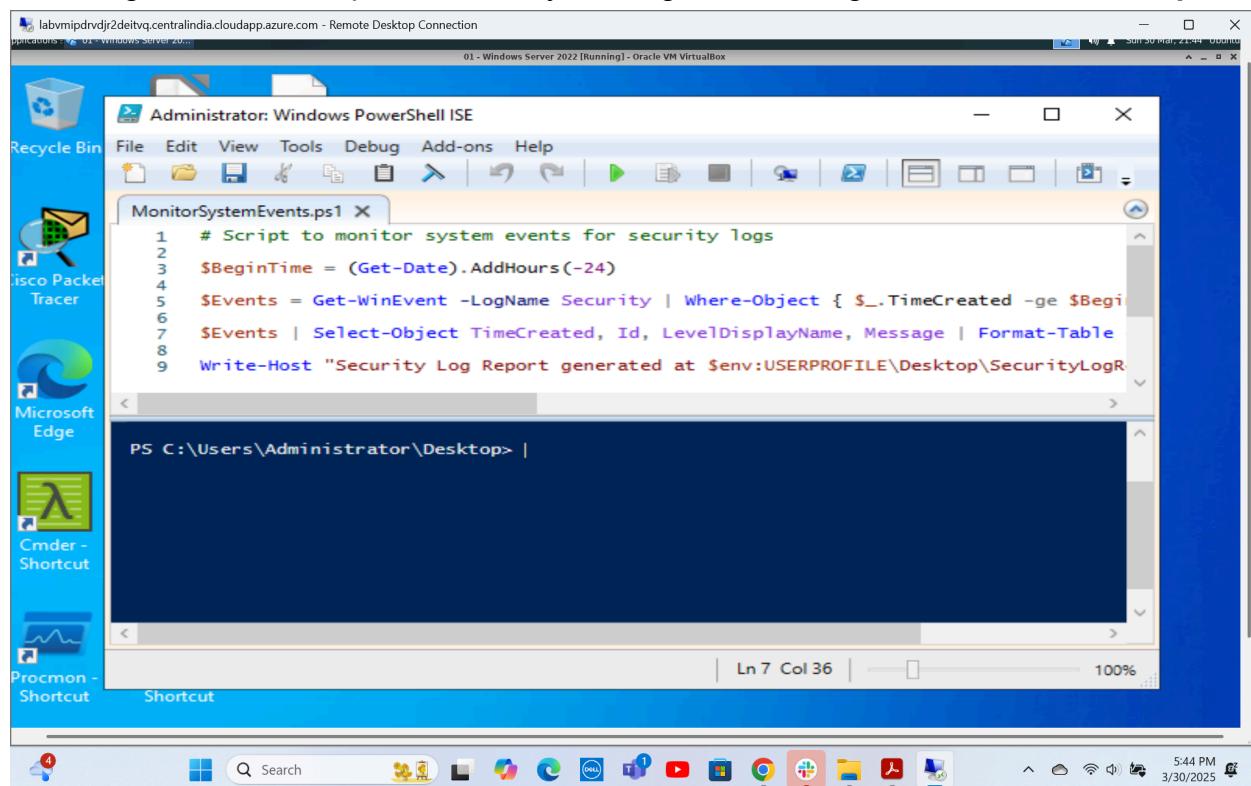
Completed Shortcut Ln 7 Col 36 100%

This screenshot shows the same Windows Server 2022 desktop environment as the previous one. However, the 'File' menu in the PowerShell ISE window is now open, displaying various options like 'New', 'Open...', 'Save', and 'Save As...'. The 'Save As...' option is currently selected. The rest of the interface and the running script are identical to the first screenshot.

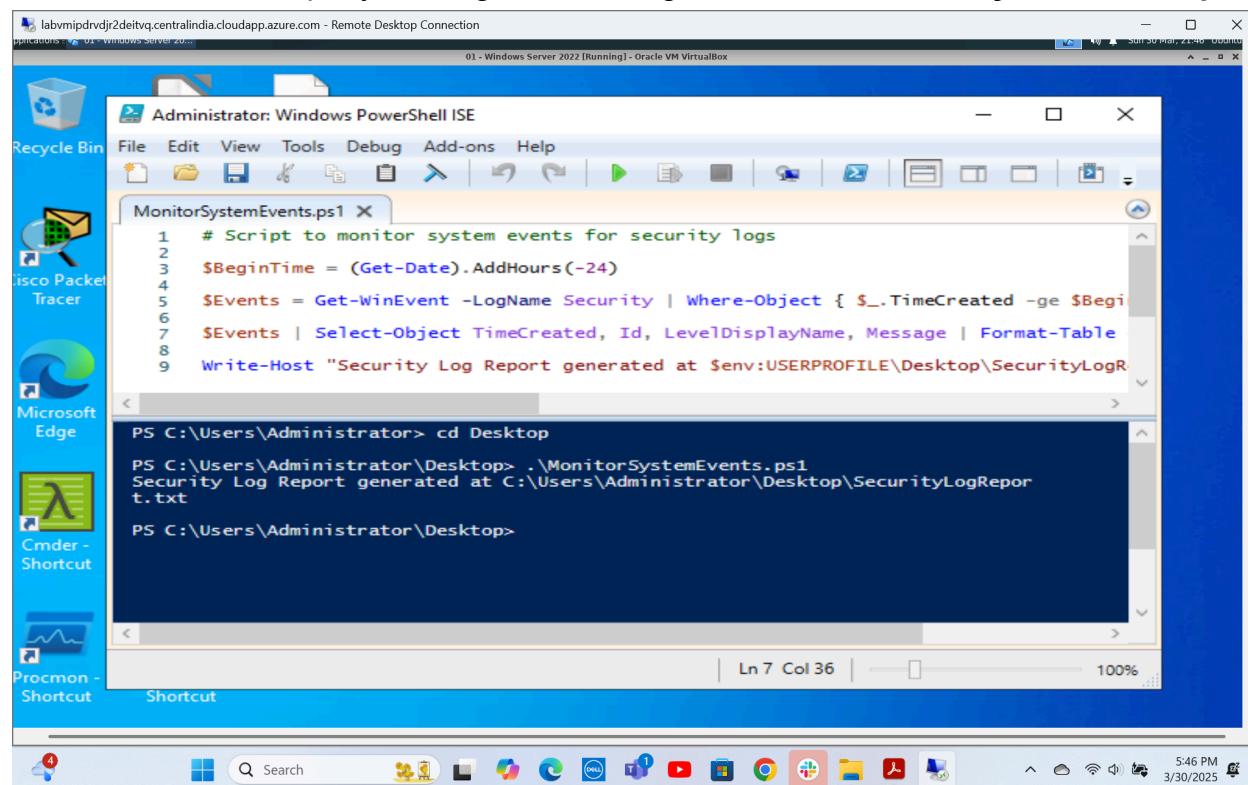
## 1.7 Saved Script on Desktop as **MonitorSystemEvents.ps1**



## 1.8 Navigated to the script's location by running the following command **cd Desktop**



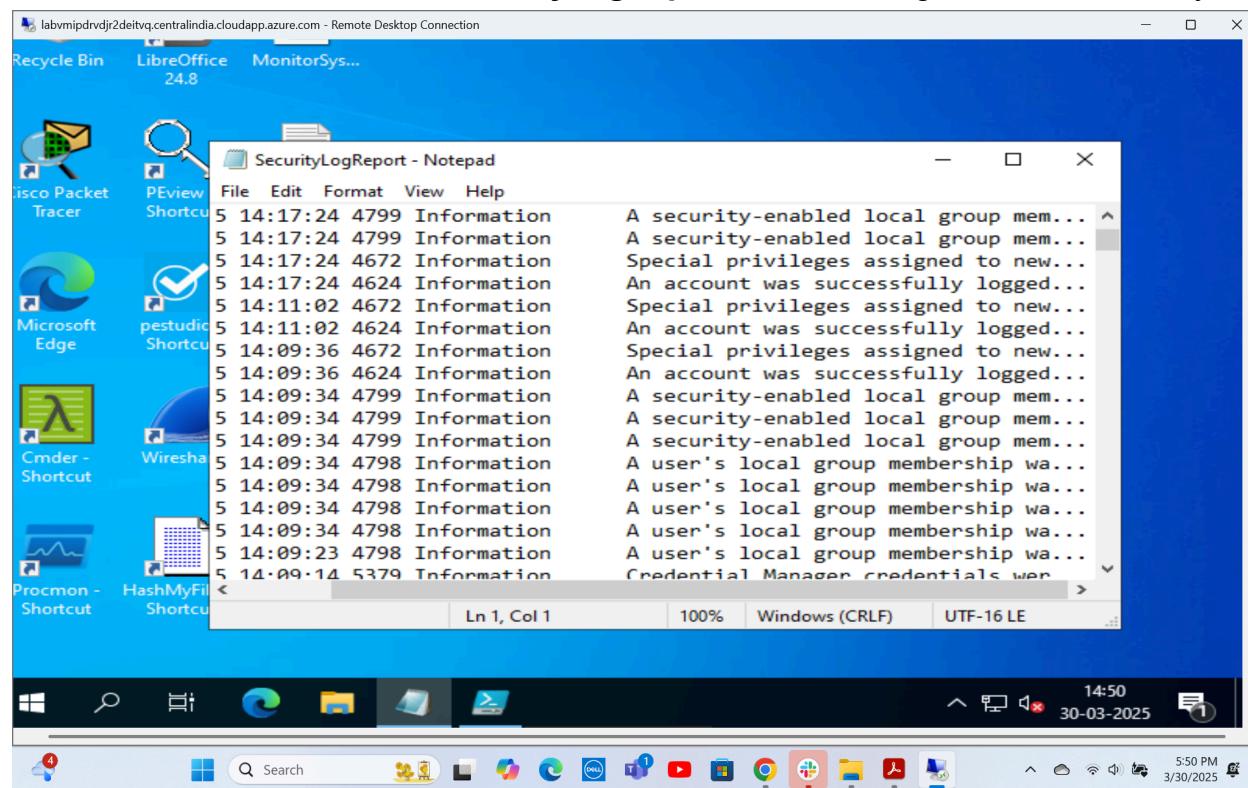
## 1.9 Executed the script by running the following command .\MonitorSystemEvents.ps1



```
# Script to monitor system events for security logs
$BeginTime = (Get-Date).AddHours(-24)
$Events = Get-WinEvent -LogName Security | Where-Object { $_.TimeCreated -ge $BeginTime }
$Events | Select-Object TimeCreated, Id, LevelDisplayName, Message | Format-Table
Write-Host "Security Log Report generated at $env:USERPROFILE\Desktop\SecurityLogReport.txt"

PS C:\Users\Administrator> cd Desktop
PS C:\Users\Administrator\Desktop> .\MonitorSystemEvents.ps1
Security Log Report generated at C:\Users\Administrator\Desktop\SecurityLogReport.txt
PS C:\Users\Administrator\Desktop>
```

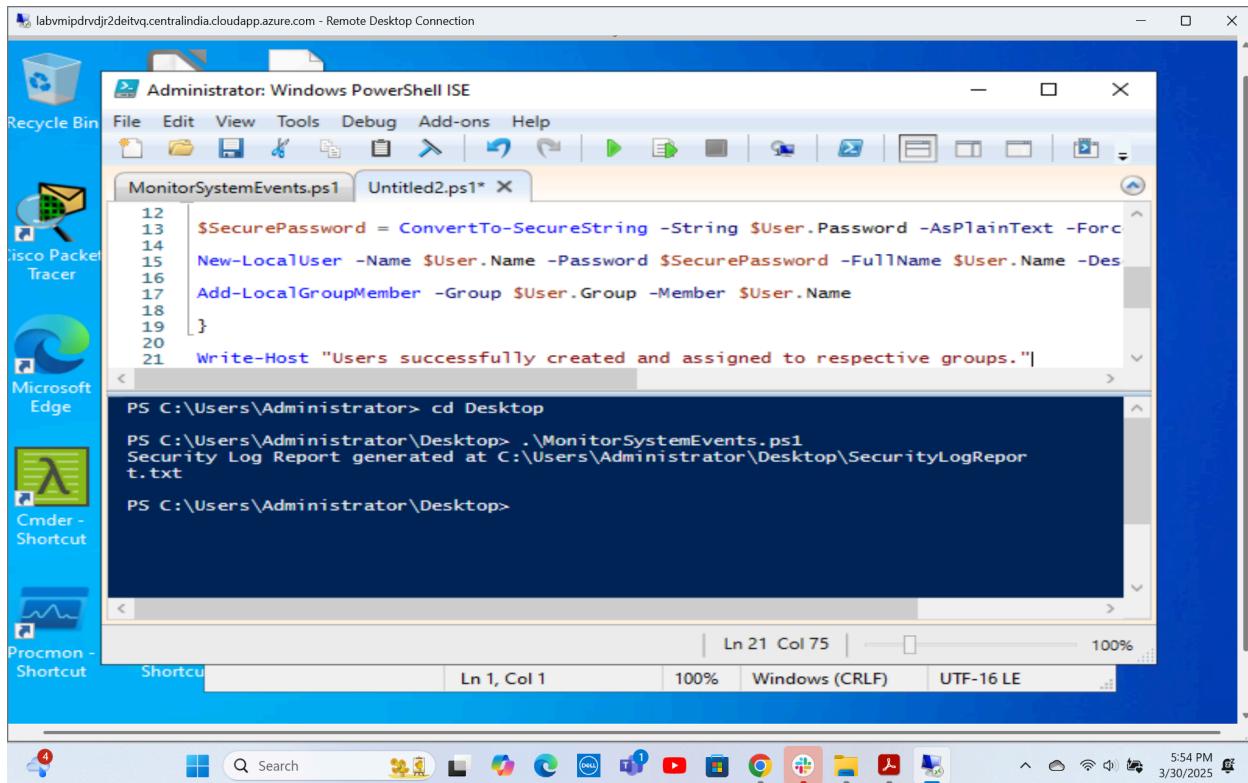
## 1.10 Browsed for a file named **SecurityLogReport.txt** containing the filtered security events



```
File Edit Format View Help
5 14:17:24 4799 Information A security-enabled local group mem...
5 14:17:24 4799 Information A security-enabled local group mem...
5 14:17:24 4672 Information Special privileges assigned to new...
5 14:17:24 4624 Information An account was successfully logged...
5 14:11:02 4672 Information Special privileges assigned to new...
5 14:11:02 4624 Information An account was successfully logged...
5 14:09:36 4672 Information Special privileges assigned to new...
5 14:09:36 4624 Information An account was successfully logged...
5 14:09:34 4799 Information A security-enabled local group mem...
5 14:09:34 4799 Information A security-enabled local group mem...
5 14:09:34 4799 Information A security-enabled local group mem...
5 14:09:34 4798 Information A user's local group membership wa...
5 14:09:34 4798 Information A user's local group membership wa...
5 14:09:34 4798 Information A user's local group membership wa...
5 14:09:23 4798 Information A user's local group membership wa...
5 14:09:14 5379 Information Credential Manager credentials were
```

## Step 2: Analyzed methods to optimize user management

2.1 Repeated step 1.5 to open a new script and added multiple users, set their passwords, and assigned them to groups



The screenshot shows a Windows PowerShell ISE window titled "Administrator: Windows PowerShell ISE". The left pane displays icons for various desktop applications like Recycle Bin, Cisco Packet Tracer, Microsoft Edge, and Cmder. The main pane contains two tabs: "MonitorSystemEvents.ps1" and "Untitled2.ps1\*". The "MonitorSystemEvents.ps1" tab shows a PowerShell script with the following code:

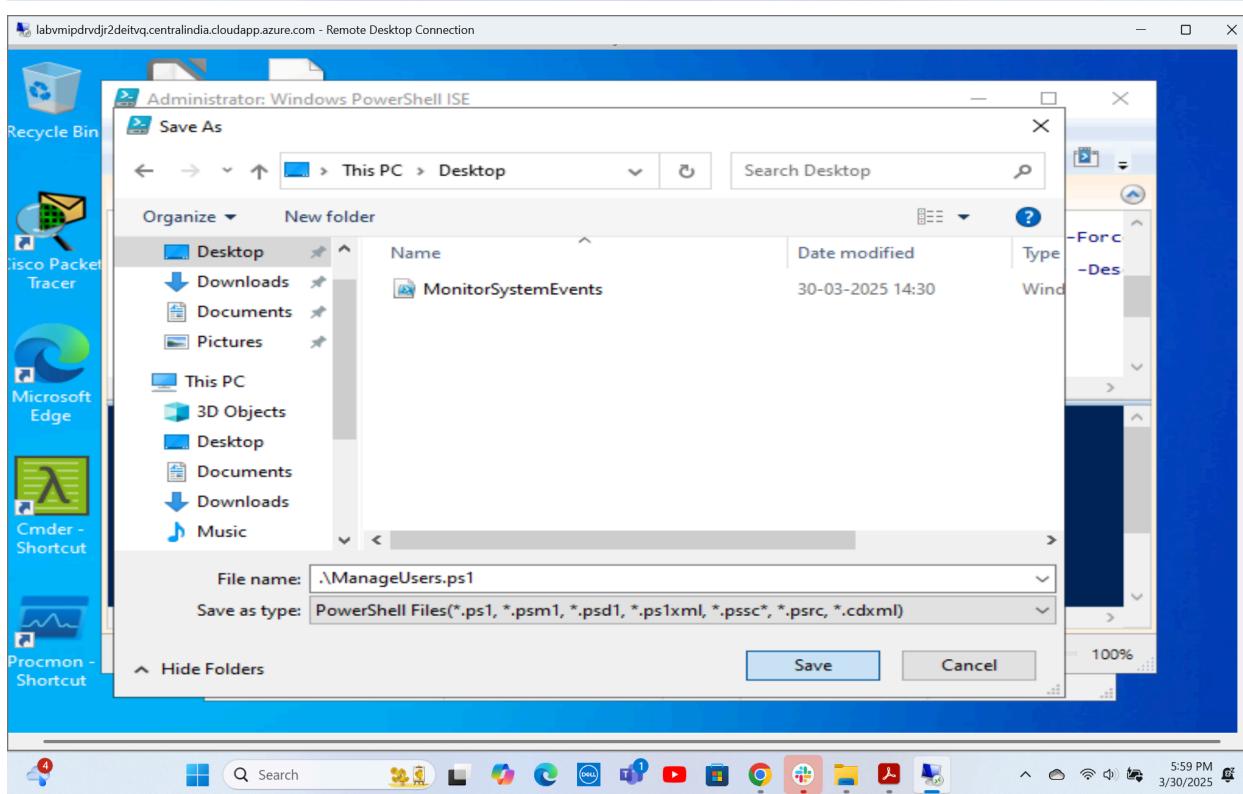
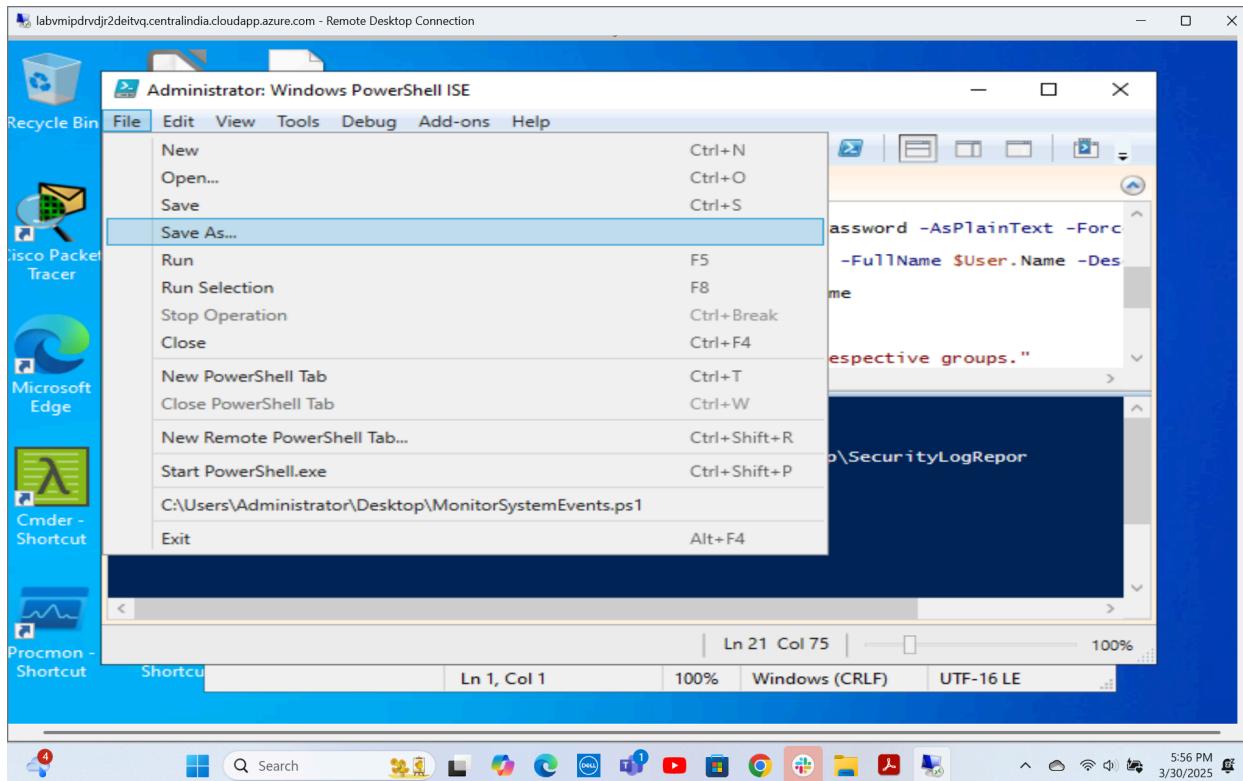
```
12 $SecurePassword = ConvertTo-SecureString -String $User.Password -AsPlainText -Force
13 New-LocalUser -Name $User.Name -Password $SecurePassword -FullName $User.Name -Descripti
14 on $User.Description
15 Add-LocalGroupMember -Group $User.Group -Member $User.Name
16
17 }
18
19 }
20
21 Write-Host "Users successfully created and assigned to respective groups."|
```

The "Untitled2.ps1\*" tab is currently empty. Below the tabs, the PowerShell command history shows:

```
PS C:\Users\Administrator> cd Desktop
PS C:\Users\Administrator\Desktop> .\MonitorSystemEvents.ps1
Security Log Report generated at C:\Users\Administrator\Desktop\SecurityLogReport.txt
PS C:\Users\Administrator\Desktop>
```

The status bar at the bottom indicates the current line is "Ln 1, Col 1" and the encoding is "UTF-16 LE". The taskbar at the bottom of the screen shows various pinned icons.

## 2.2 Saved the script on the Desktop as **ManageUsers.ps1**



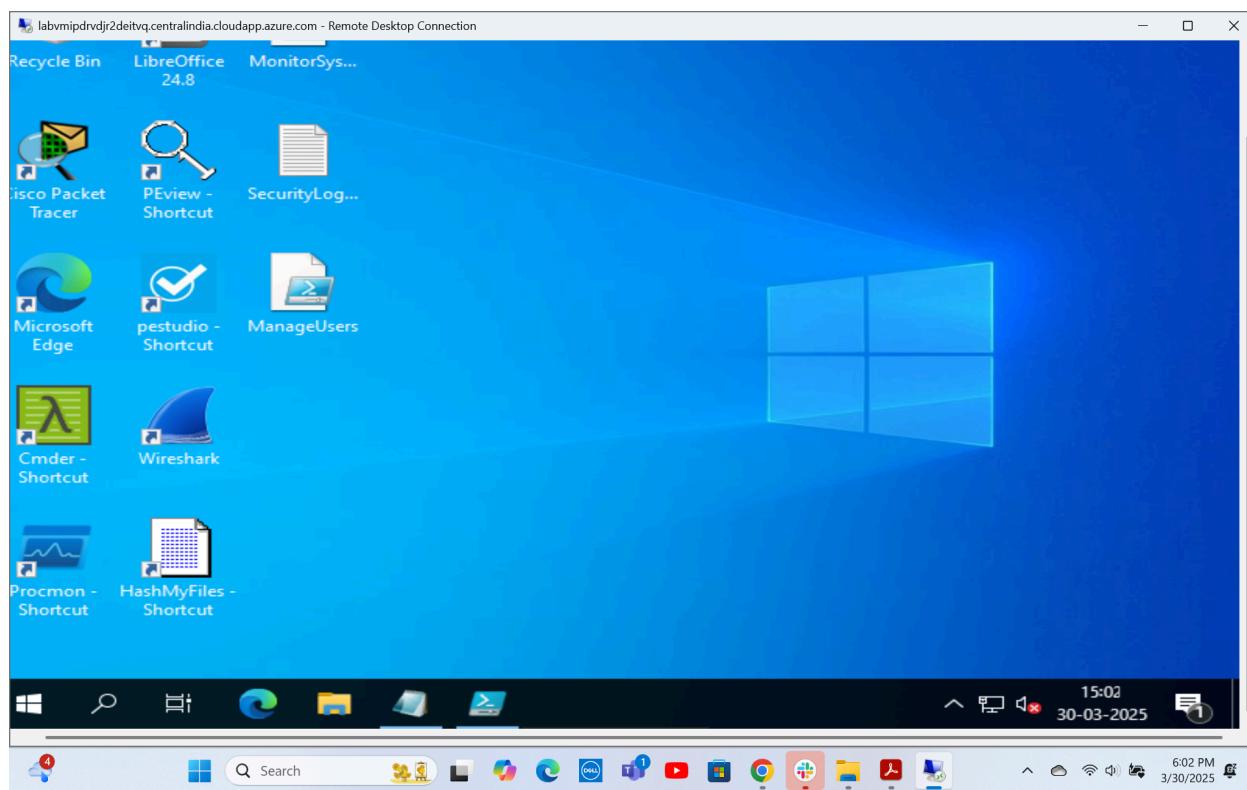
## 2.3 Executed the script by running the program using .\ManageUsers.ps1

The screenshot shows a Windows Server 2022 desktop environment with a Remote Desktop Connection window titled "Administrator: Windows PowerShell ISE". The taskbar at the bottom includes icons for File Explorer, Task View, Start, Search, Taskbar settings, and several pinned applications: Recycle Bin, Cisco Packet Tracer, Microsoft Edge, Cmder - Shortcut, and Procmon - Shortcut. The PowerShell ISE window displays two tabs: "BackupFiles.ps1" and "ConfigureNetwork.ps1", with "ManageUsers.ps1" currently active. The code in "ManageUsers.ps1" creates two users, JohnDoe and JaneSmith, and adds them to their respective groups. The output window shows the command ".\ManageUsers.ps1" being run and the resulting message: "Users successfully created and assigned to respective groups." followed by a table of user details.

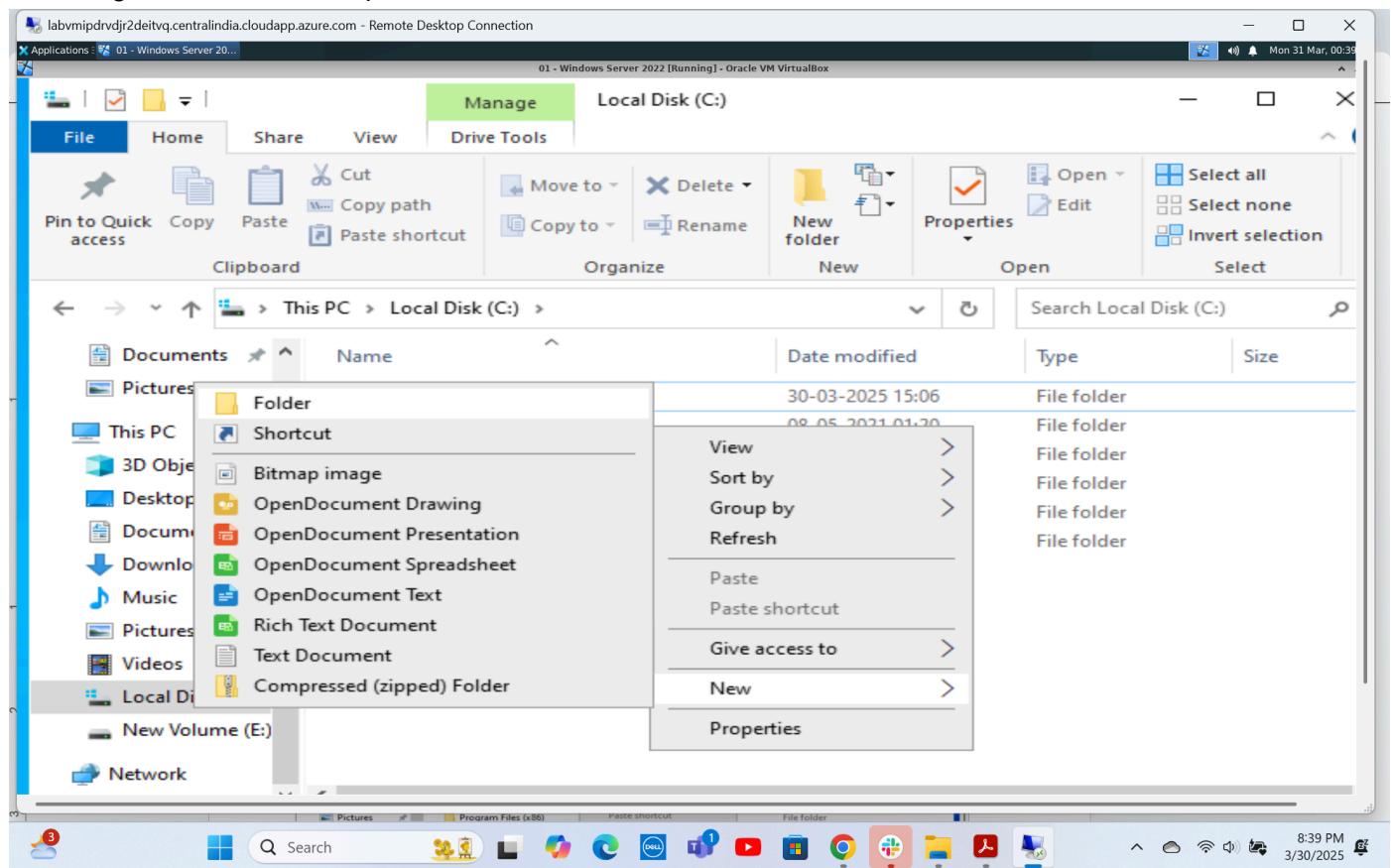
```
1 # Script to add multiple users and assign them to groups
2
3 $Users = @(
4     @{ Name = "JohnDoe"; Password = "P@ssw0rd1"; Group = "Administrators" }
5     @{ Name = "JaneSmith"; Password = "P@ssw0rd2"; Group = "Users" }
)
6
7 foreach ($User in $Users) {
8     New-LocalUser -Name $User.Name -Password $User.Password -Enabled $User.Enabled -Description $User.Description
9     Add-LocalGroupMember -Group $User.Group -Member $User.Name
10 }
11
12
PS C:\Users\Administrator\Desktop> .\ManageUsers.ps1
13
14
Users successfully created and assigned to respective groups.
15
16
Name      Enabled Description
17 ----  -----  -----
18 JohnDoe   True    Added by script
19 JaneSmith True    Added by script
20
21
PS C:\Users\Administrator\Desktop>
```

## Step 3: Create a streamlined process for file management

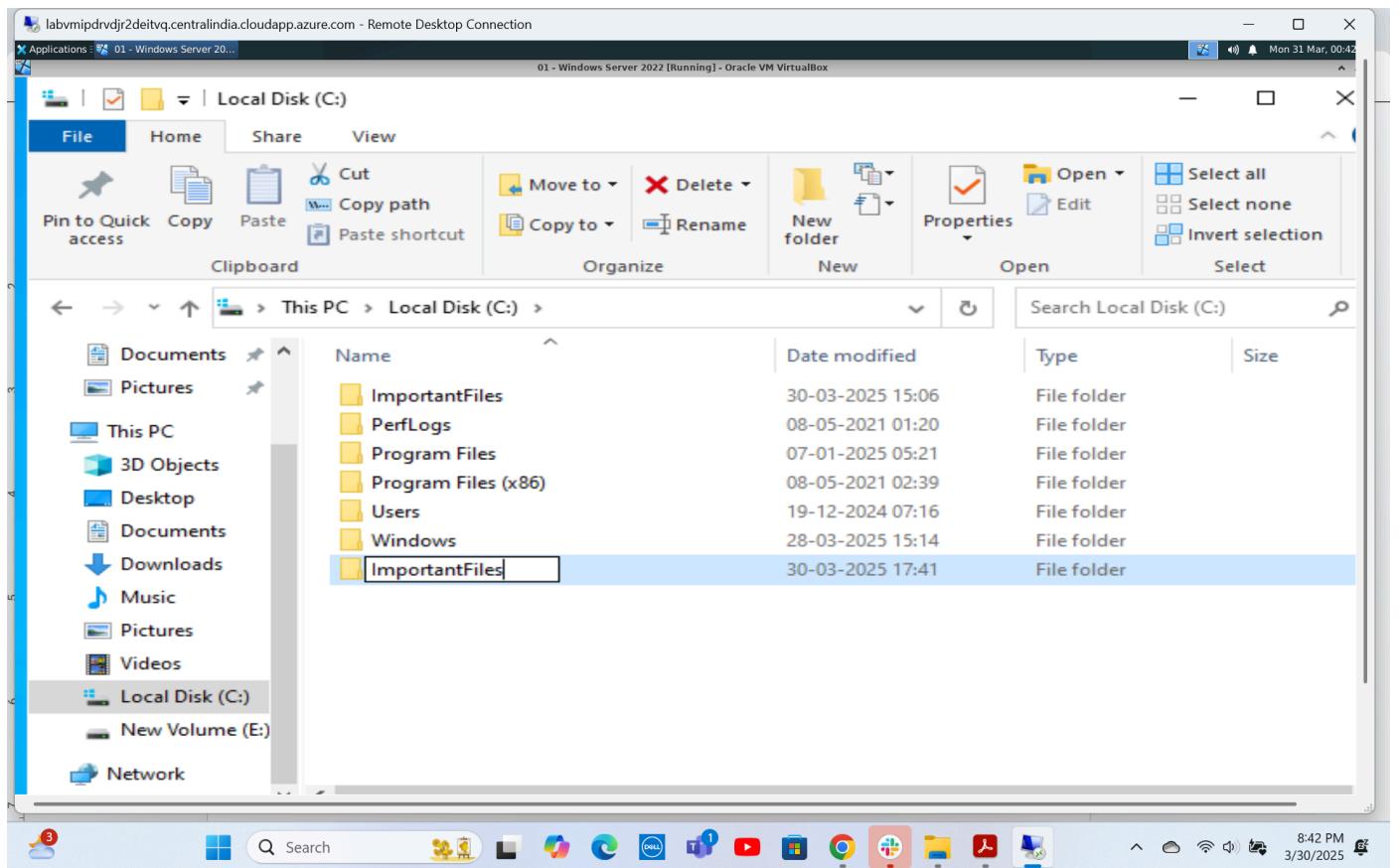
3.1 Click on File Explorer in the menu tab to create a dummy data folder



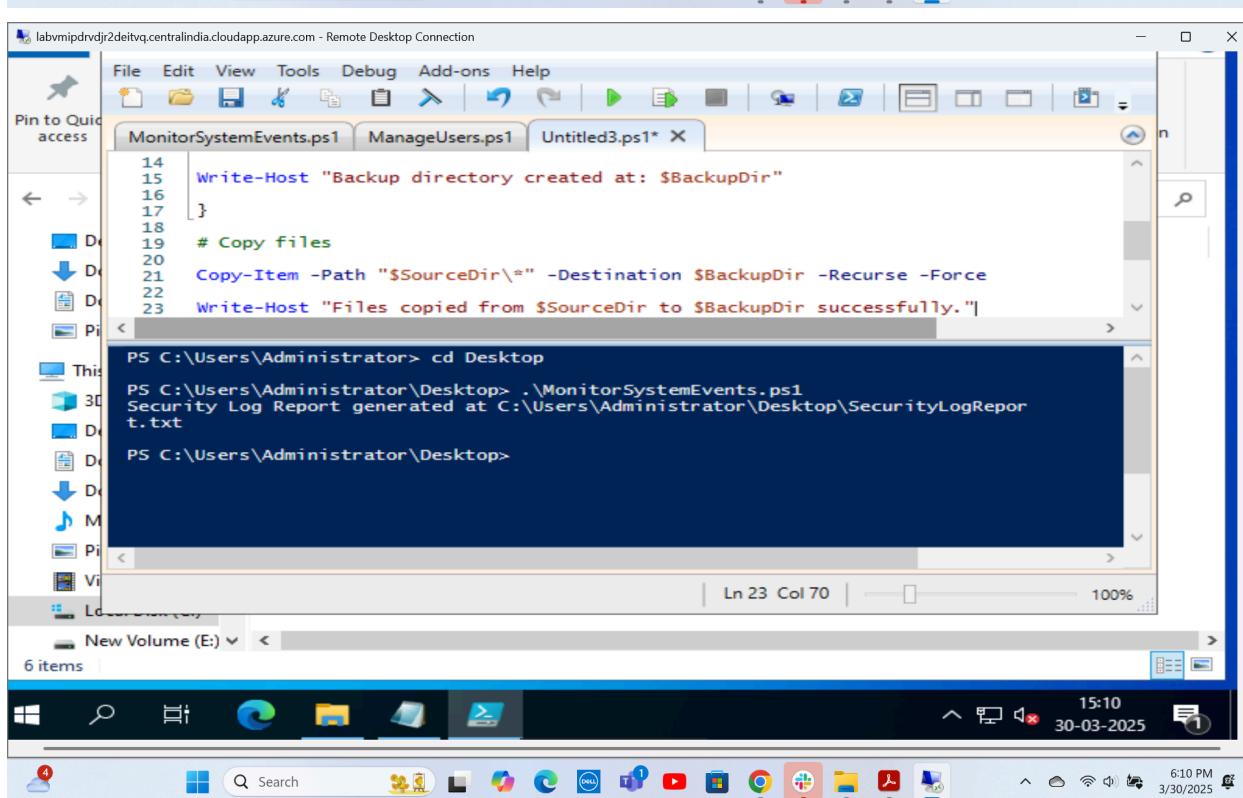
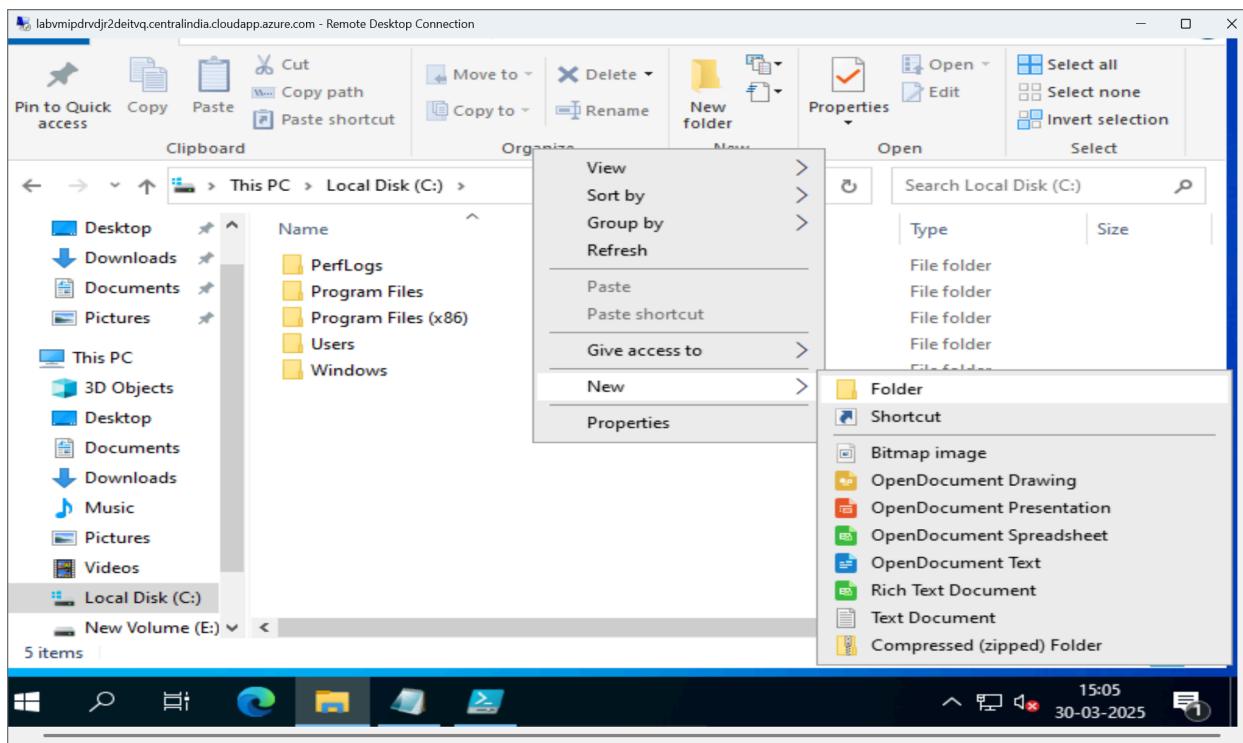
### 3.2 Navigated to **This PC**, opened **C Drive** and created a new folder



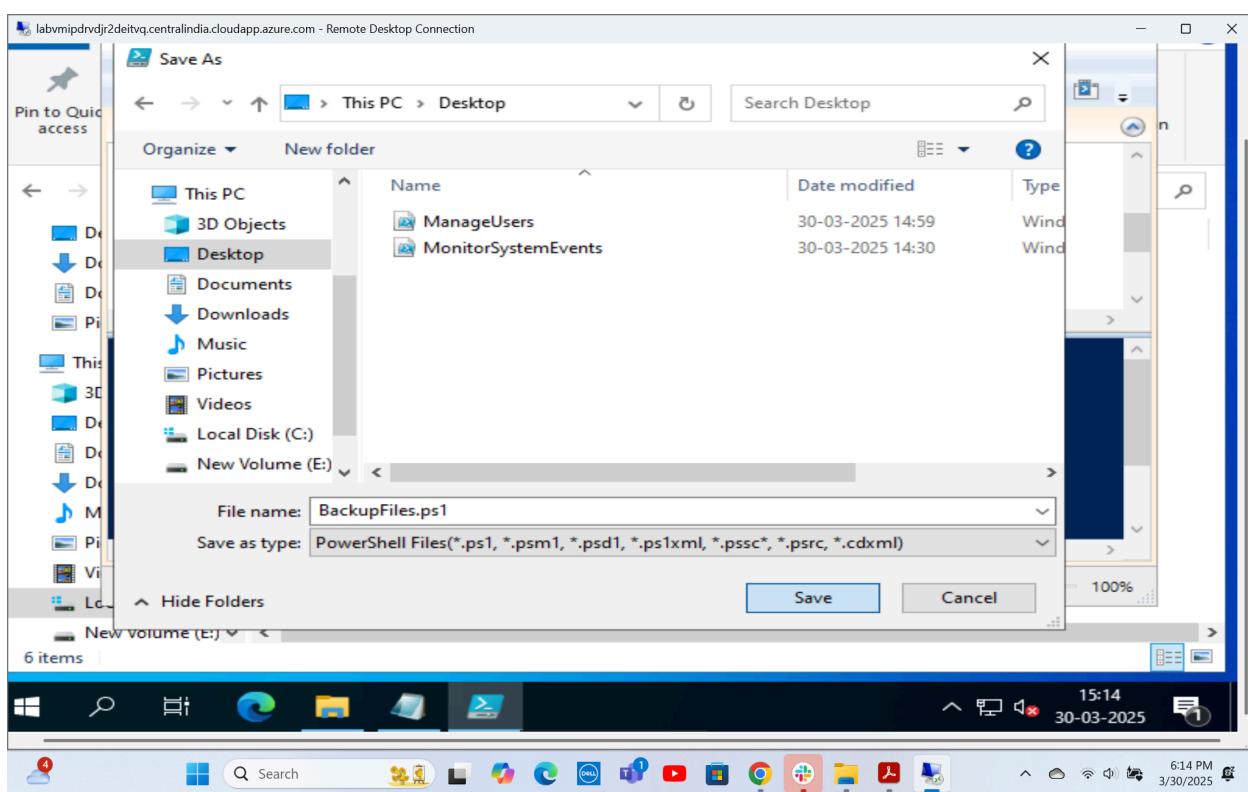
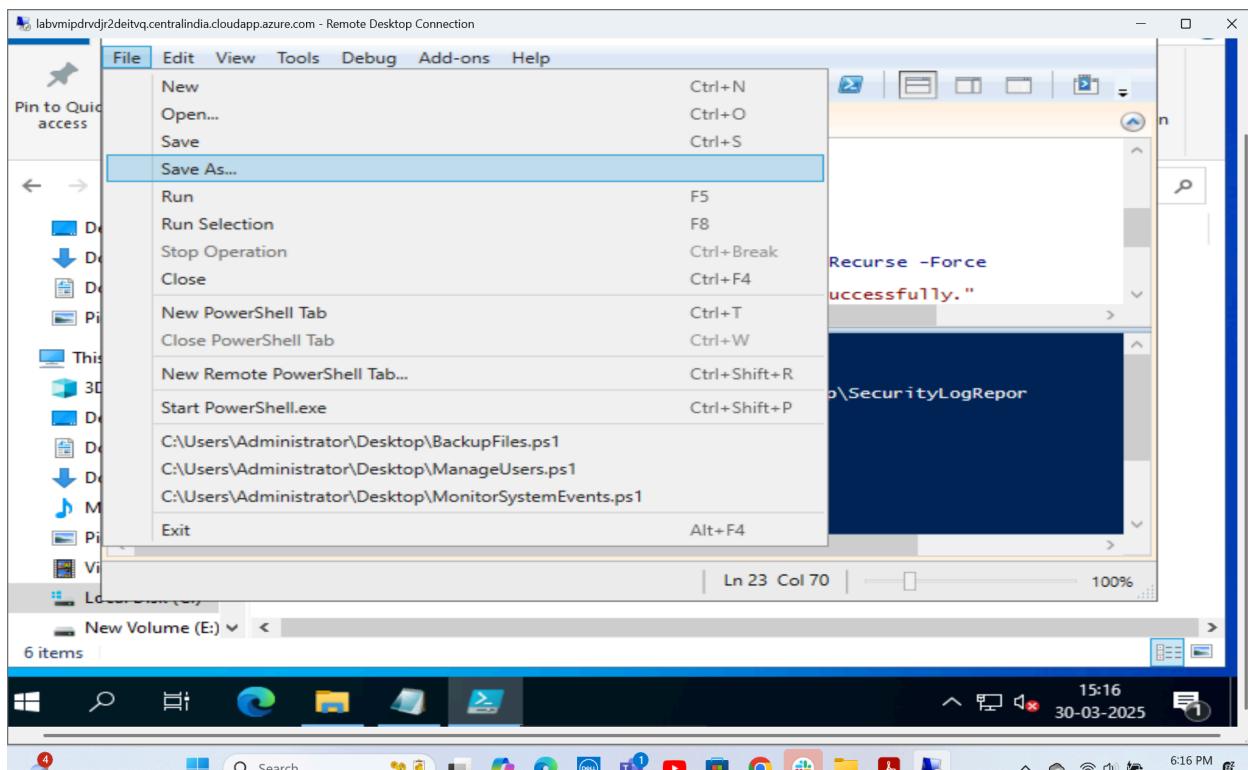
### 3.3 Named the folder as **ImportantFiles** and saved it



3.4 Navigated back to PowerShell and **repeated step 1.5**. Typed the script to implement file backup in the **E Drive**:



3.5 Repeated step 2.3 and saved the file with the name **BackupFiles.ps1** on the desktop



3.6 Executed the script of backing up files of the C drive to the D drive having the current date by running the following command: .\BackupFiles.ps1

The screenshot shows a Windows Remote Desktop Connection window titled "labvmipdrvdjr2deltvq.centralindia.cloudapp.azure.com - Remote Desktop Connection". The main area displays a PowerShell session with the following content:

```
PS C:\Users\Administrator\Desktop> .\BackupFiles.ps1
14 Write-Host "Backup directory created at: $BackupDir"
15
16 }
17 }
18 #
19 # Copy files
20 Copy-Item -Path "$SourceDir\*" -Destination $BackupDir -Recurse -Force
21
22 Write-Host "Files copied from $SourceDir to $BackupDir successfully."
23
```

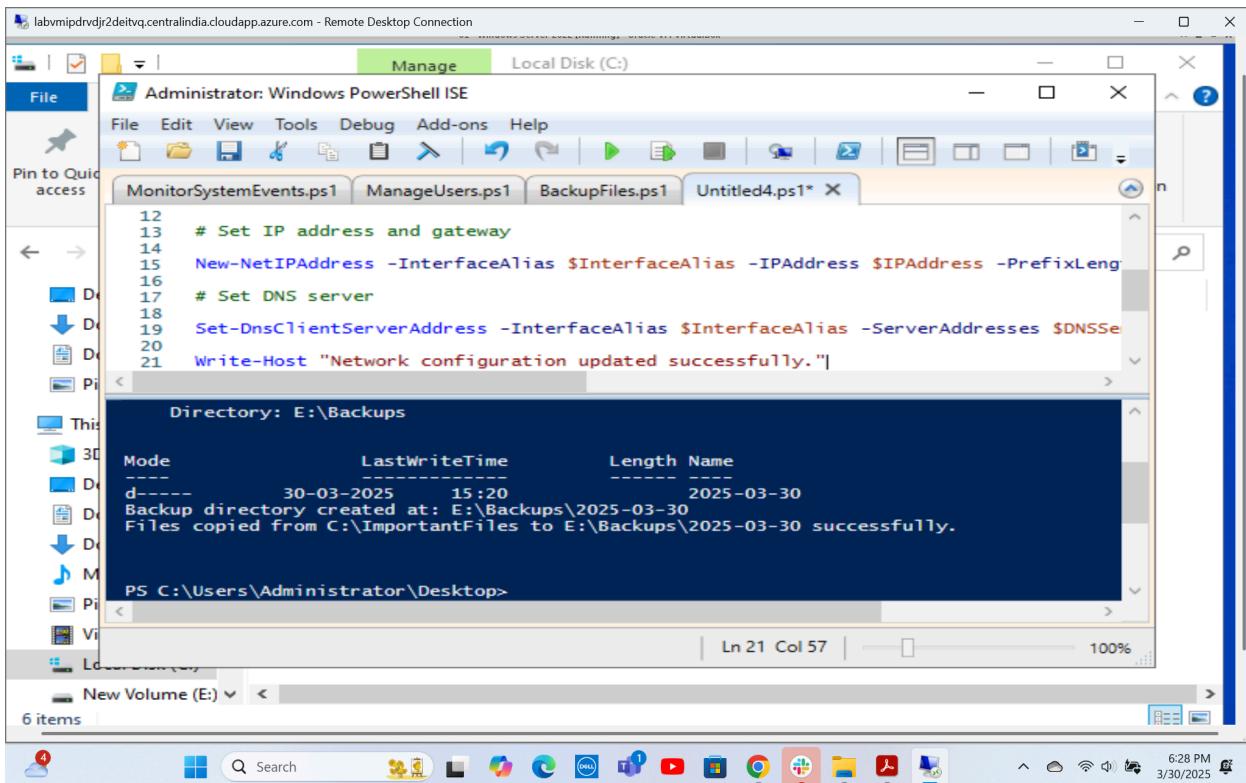
Below the code, the PowerShell prompt shows the output of the script execution:

```
Directory: E:\Backups
Mode                LastWriteTime        Length Name
----                -              -          -
d----- 30-03-2025 15:20                 2025-03-30
Backup directory created at: E:\Backups\2025-03-30
Files copied from C:\ImportantFiles to E:\Backups\2025-03-30 successfully.
```

The taskbar at the bottom of the screen shows several pinned icons, including File Explorer, Edge, and File History. The system tray indicates the date as 30-03-2025 and the time as 6:21 PM.

## Step 4: Designed an approach for network configuration

### 4.1 Repeated step 2.1 to write the script for performing network configuration



The screenshot shows a Windows desktop environment with a PowerShell ISE window open. The title bar of the window reads "Administrator: Windows PowerShell ISE". The window displays a PowerShell script with the following content:

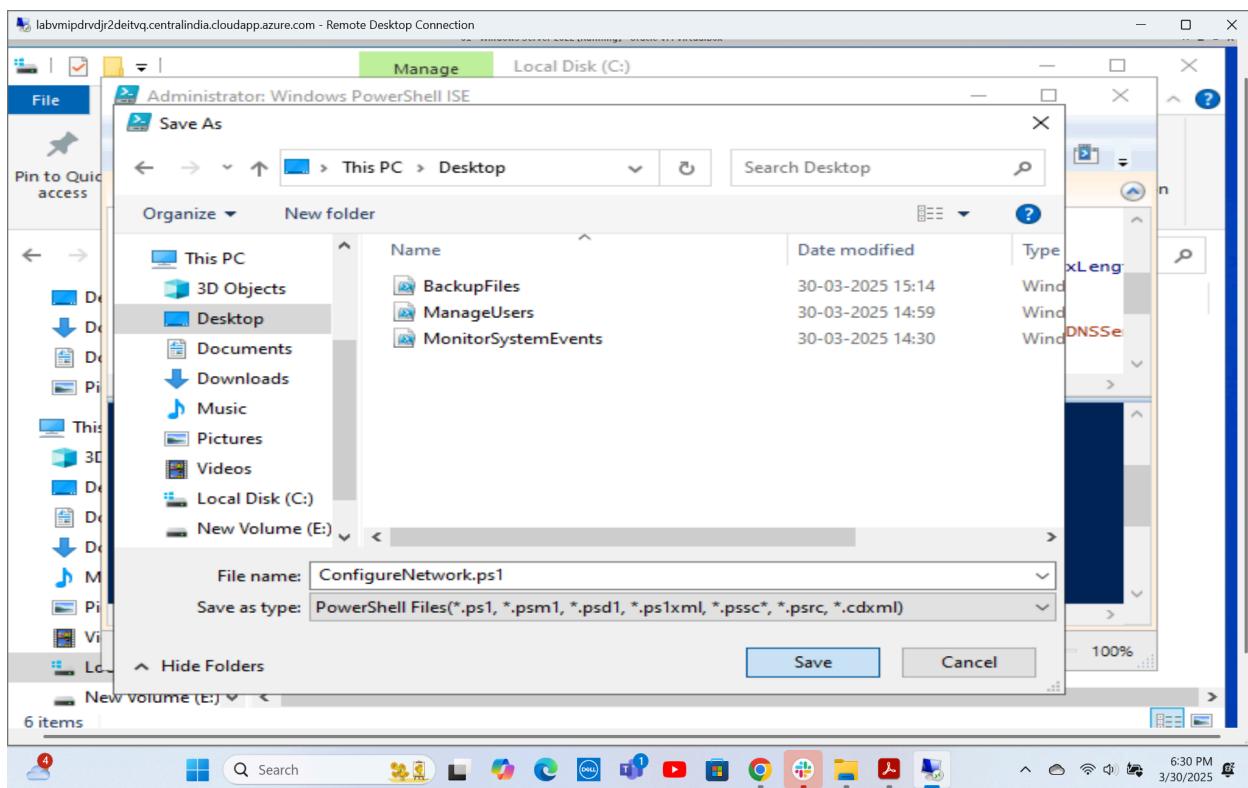
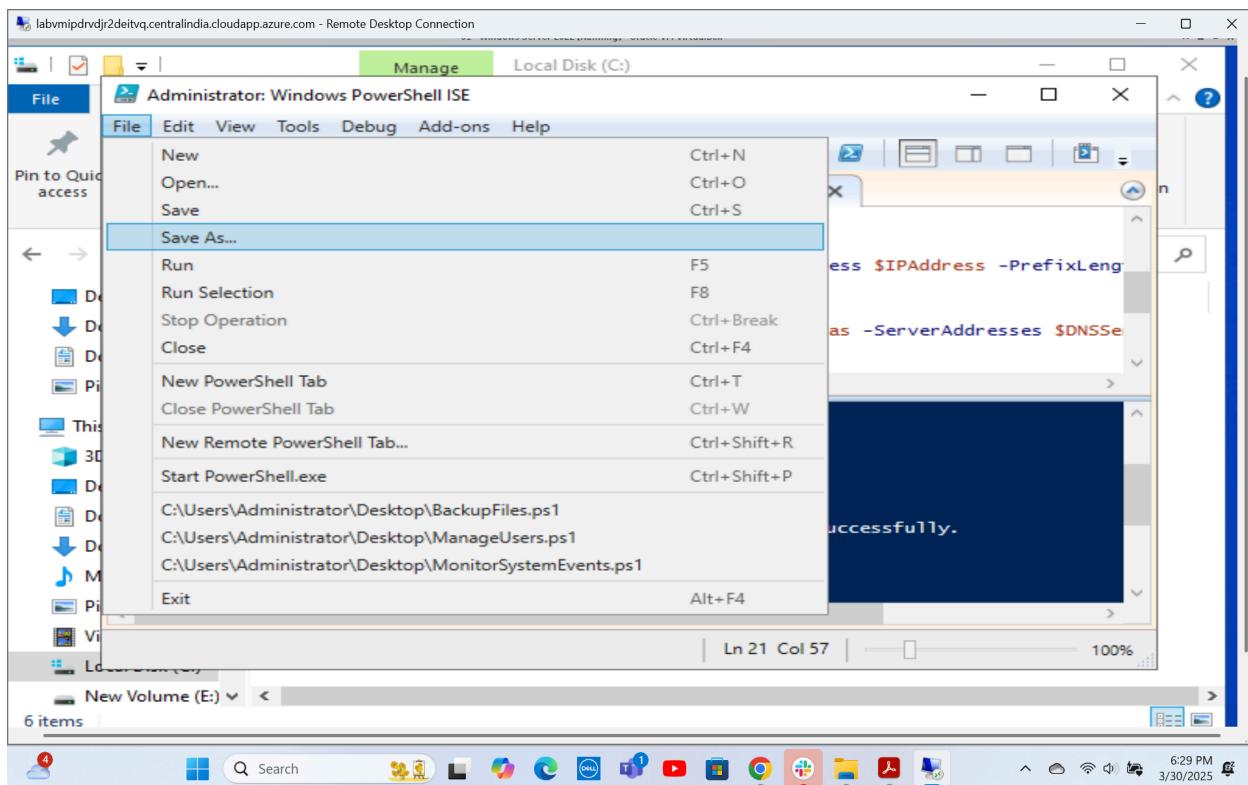
```
12
13 # Set IP address and gateway
14
15 New-NetIPAddress -InterfaceAlias $InterfaceAlias -IPAddress $IPAddress -PrefixLength
16
17 # Set DNS server
18
19 Set-DnsClientServerAddress -InterfaceAlias $InterfaceAlias -ServerAddresses $DNSSe
20
21 Write-Host "Network configuration updated successfully."
```

Below the script, the PowerShell prompt shows the execution of a command to backup files to a new volume (E:). The output indicates that a directory was created at E:\Backups\2025-03-30 and files were copied from C:\ImportantFiles to this location.

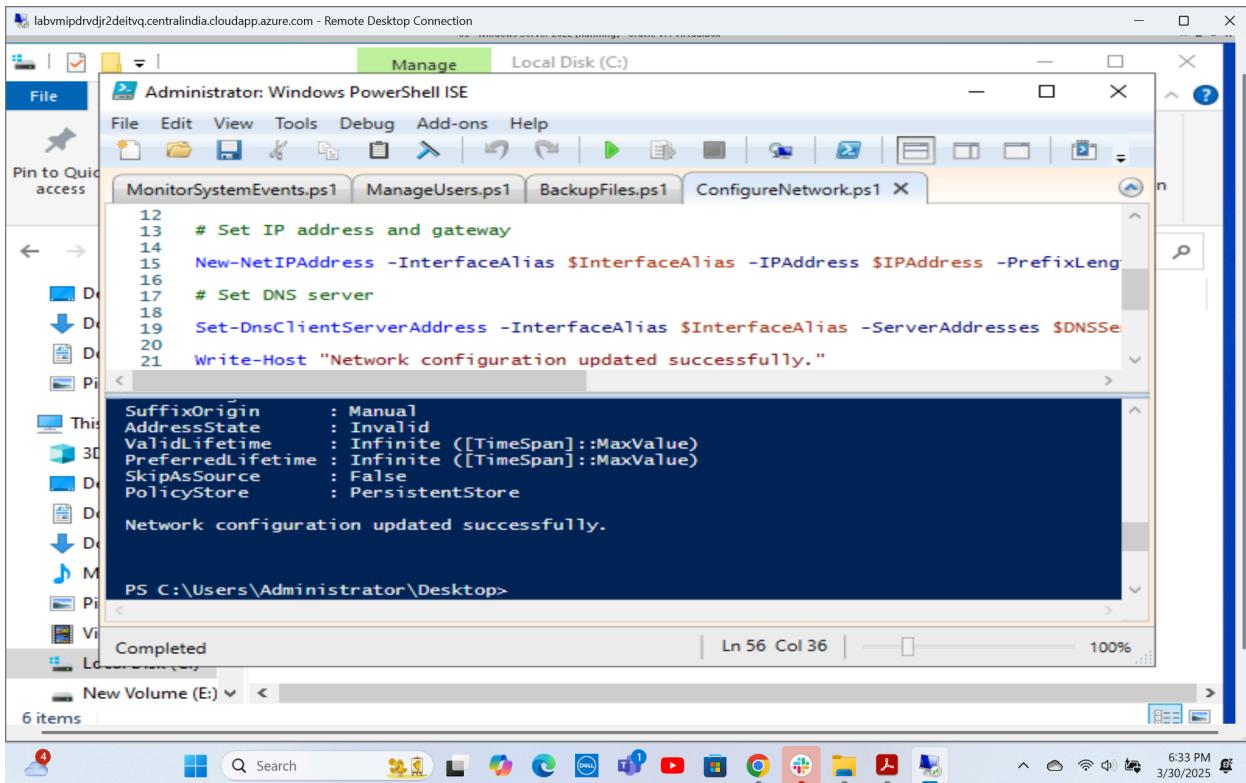
```
PS C:\Users\Administrator\Desktop> Directory: E:\Backups
Mode                LastWriteTime         Length Name
----                -----        ----- 
d----- 30-03-2025      15:20            2025-03-30
Backup directory created at: E:\Backups\2025-03-30
Files copied from C:\ImportantFiles to E:\Backups\2025-03-30 successfully.
```

The taskbar at the bottom of the screen shows various pinned icons, including File Explorer, Microsoft Edge, and File History. The system tray indicates the date and time as 6:28 PM on 3/30/2025.

#### 4.2 Repeated step 2.3 and saved the file with the name **ConfigureNetwork.ps1** on the desktop



4.3 Executed the script that updates the network adapter settings with the new IP configuration by running the following command: **.\ConfigureNetwork.ps1**



The screenshot shows a Windows PowerShell ISE window titled "Administrator: Windows PowerShell ISE". The script tab "ConfigureNetwork.ps1" is selected. The code in the editor is as follows:

```
12
13 # Set IP address and gateway
14
15 New-NetIPAddress -InterfaceAlias $InterfaceAlias -IPAddress $IPAddress -PrefixLength $PrefixLength
16
17 # Set DNS server
18
19 Set-DnsClientServerAddress -InterfaceAlias $InterfaceAlias -ServerAddresses $DNSServer
20
21 Write-Host "Network configuration updated successfully."
```

Below the code, the output pane shows the results of the command execution:

```
SuffixOrigin      : Manual
AddressState     : Invalid
ValidLifetime   : Infinite ([TimeSpan]::MaxValue)
PreferredLifetime : Infinite ([TimeSpan]::MaxValue)
SkipAsSource    : False
PolicyStore      : PersistentStore

Network configuration updated successfully.
```

The status bar at the bottom indicates "Completed" and "Ln 56 Col 36". The taskbar at the bottom of the screen shows various icons for Microsoft applications like File Explorer, Edge, and Mail.

A screenshot of a Windows Remote Desktop Connection window titled "labvmipdrvdjr2deitvq.centralindia.cloudapp.azure.com - Remote Desktop Connection". Inside the window, there is a "Administrator: Windows PowerShell ISE" window. The PowerShell window shows the output of a PowerShell script named "ConfigureNetwork.ps1". The script contains the following code:

```
12
13 # Set IP address and gateway
14
15 New-NetIPAddress -InterfaceAlias $InterfaceAlias -IPAddress $IPAddress -PrefixLength
16
17 # Set DNS server
18
19 Set-DnsClientServerAddress -InterfaceAlias $InterfaceAlias -ServerAddresses $DNSSE
20
21 Write-Host "Network configuration updated successfully."
```

The output of the command `Get-NetIPAddress` is displayed, showing the configuration for the "Ethernet adapter Ethernet 2":

```
Default Gateway . . . . . : 192.168.1.1
Ethernet adapter Ethernet 2:
  Connection-specific DNS Suffix . : yobe2id3hxxutce12odf4frueh.rx.internal.cloudapp.net
    Link-local IPv6 Address . . . . . : fe80::79dc:503d:2db7:bbec%17
      IPv4 Address . . . . . : 10.0.2.6
        Subnet Mask . . . . . : 255.255.255.0
          Default Gateway . . . . . : 10.0.2.1
PS C:\Users\Administrator\Desktop>
```

4.4 Verified the new changes by running the following commands: **ipconfig**

**Get-NetIPAddress**

A screenshot of a Windows Remote Desktop Connection window titled "labvmipdrvdjr2deitvq.centralindia.cloudapp.azure.com - Remote Desktop Connection". Inside the window, there is a "Administrator: Windows PowerShell ISE" window. The PowerShell window shows the output of the "ipconfig" command followed by the "Get-NetIPAddress" command.

The output of the "ipconfig" command is:

```
Windows IP Configuration

Ethernet adapter Ethernet:
  Connection-specific DNS Suffix . :
  Link-local IPv6 Address . . . . . : fe80::d815:b01a:105d:495d%3
  IPv4 Address . . . . . : 192.168.1.100
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.1.1
PS C:\Users\Administrator\Desktop>
```

The output of the "Get-NetIPAddress" command is:

```
PS C:\Users\Administrator\Desktop> Get-NetIPAddress
Windows IP Configuration

Ethernet adapter Ethernet:
  Connection-specific DNS Suffix . :
  Link-local IPv6 Address . . . . . : fe80::d815:b01a:105d:495d%3
  IPv4 Address . . . . . : 192.168.1.100
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.1.1
PS C:\Users\Administrator\Desktop>
```

labvmipdrvdrj2deitvq.centralindia.cloudapp.azure.com - Remote Desktop Connection

Administrator: Windows PowerShell ISE

File Edit View Tools Debug Add-ons Help

MonitorSystemEvents.ps1 ManageUsers.ps1 BackupFiles.ps1 ConfigureNetwork.ps1

```
12
13 # Set IP address and gateway
14
15 New-NetIPAddress -InterfaceAlias $InterfaceAlias -IPAddress $IPAddress -PrefixLength
16
17 # Set DNS server
18
19 Set-DnsClientServerAddress -InterfaceAlias $InterfaceAlias -ServerAddresses $DNSSE
20
21 Write-Host "Network configuration updated successfully."
```

PS C:\Users\Administrator\Desktop> Get-NetIPAddress

PropertyName	Value
IPAddress	fe80::79dc:503d:2db7:bbec%17
InterfaceIndex	17
InterfaceAlias	Ethernet 2
AddressFamily	IPv6
Type	Unicast
PrefixLength	64
PrefixOrigin	WellKnown
SuffixOrigin	Link
AddressState	Preferred

Completed Ln 168 Col 36 100%

New Volume (E:) < >

6 items

The screenshot shows a Windows Remote Desktop session with a PowerShell ISE window open. The window title is 'Administrator: Windows PowerShell ISE'. The code in the editor is for configuring a network interface, specifically setting an IPv6 address and a DNS server. After running the script, the output shows the current state of the network interface, including its IPv6 address (fe80::79dc:503d:2db7:bbec%17), interface index (17), alias (Ethernet 2), family (IPv6), type (Unicast), prefix length (64), origin (WellKnown), suffix (Link), and state (Preferred). The PowerShell window is integrated with a file explorer sidebar and a taskbar at the bottom.