POWERSHELL SCENARIO



Prepared By:

Name: Raja B

PS No: 10726001

TECHNICAL DOCUMENT

|  |  |  |
| --- | --- | --- |
| SI NO | Scenario No | Completion % |
| 1 | Scenario 1 | 100 |
| 2 | Scenario 2 | 100 |
| 3 | Scenario 3 | 100 |
| 4 | Scenario 4 | 100 |
| 5 | Scenario 5 | 100 |
| 6 | Scenario 6 | 100 |
| 7 | Scenario 7 | 100 |
| 8 | Scenario 8 | 100 |

TABLE OF CONTENTS

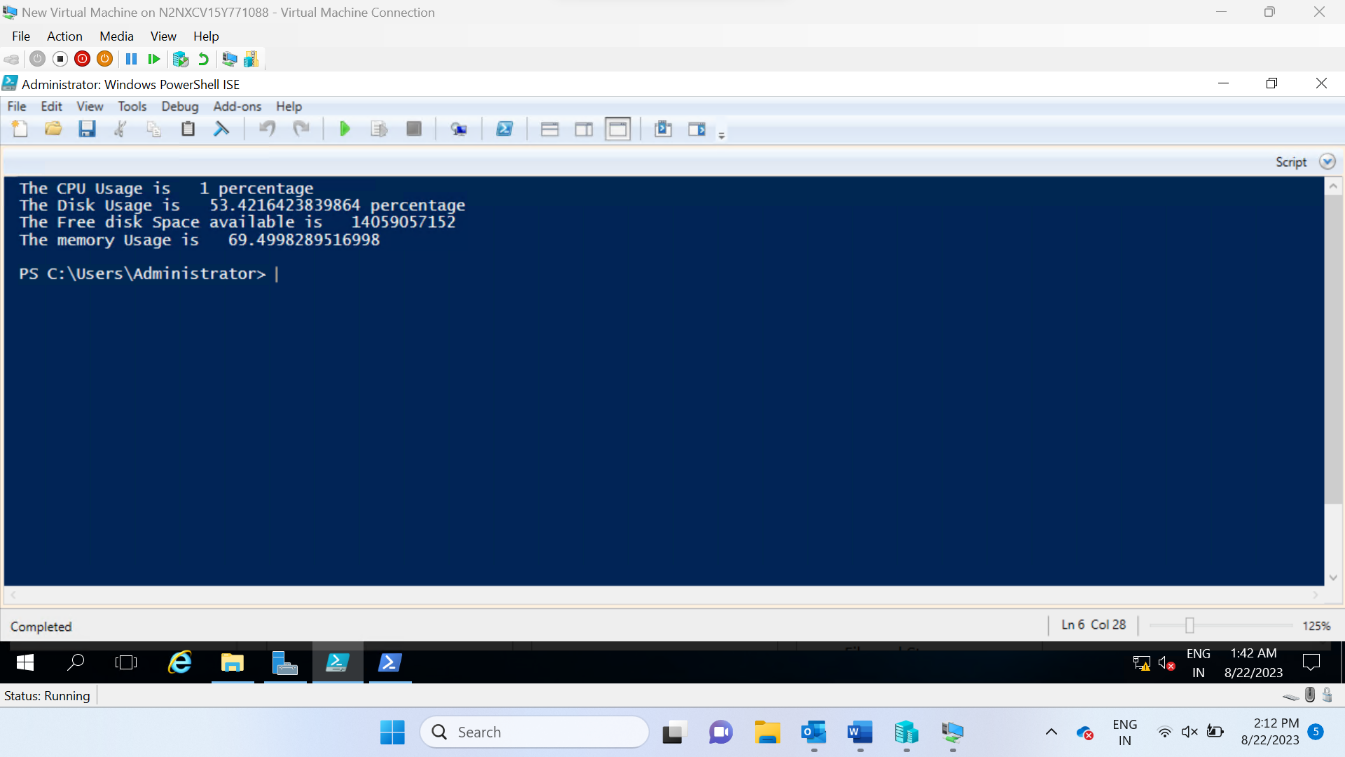
**Scenario 1**

1.Write a powershell to fetch memory usage, CPU usage, free disk space, used space of a system and display it in HTML format.

Step 1: Using Powershell Create variables that store memory usage, CPU usage, free disk space, used space of a system by using Command **Get-WMIObject**

**A screenshot of a computer

Description automatically generated**



Step 2: Store the output and convert the output into HTML Format using the command **Convert-to-HTML**

Scenario 2

Create 100 users in AD (user1, user .. user100). 50 users should be mapped to user OU and 50 users should be mapped to manager OU.

Create a CSV File which has the information about Name of the user and the OU the user belongs to

A screenshot of a computer

Description automatically generated

Create a script which read a csv file using import-csv and using foreach loop through each row of the csv file and by using information on each row Create Aduser by using Create-ADUser command and give the name and OU

A screenshot of a computer

Description automatically generated

Execute the command and See the results in the ADServer

A computer screen shot of a computer

Description automatically generated

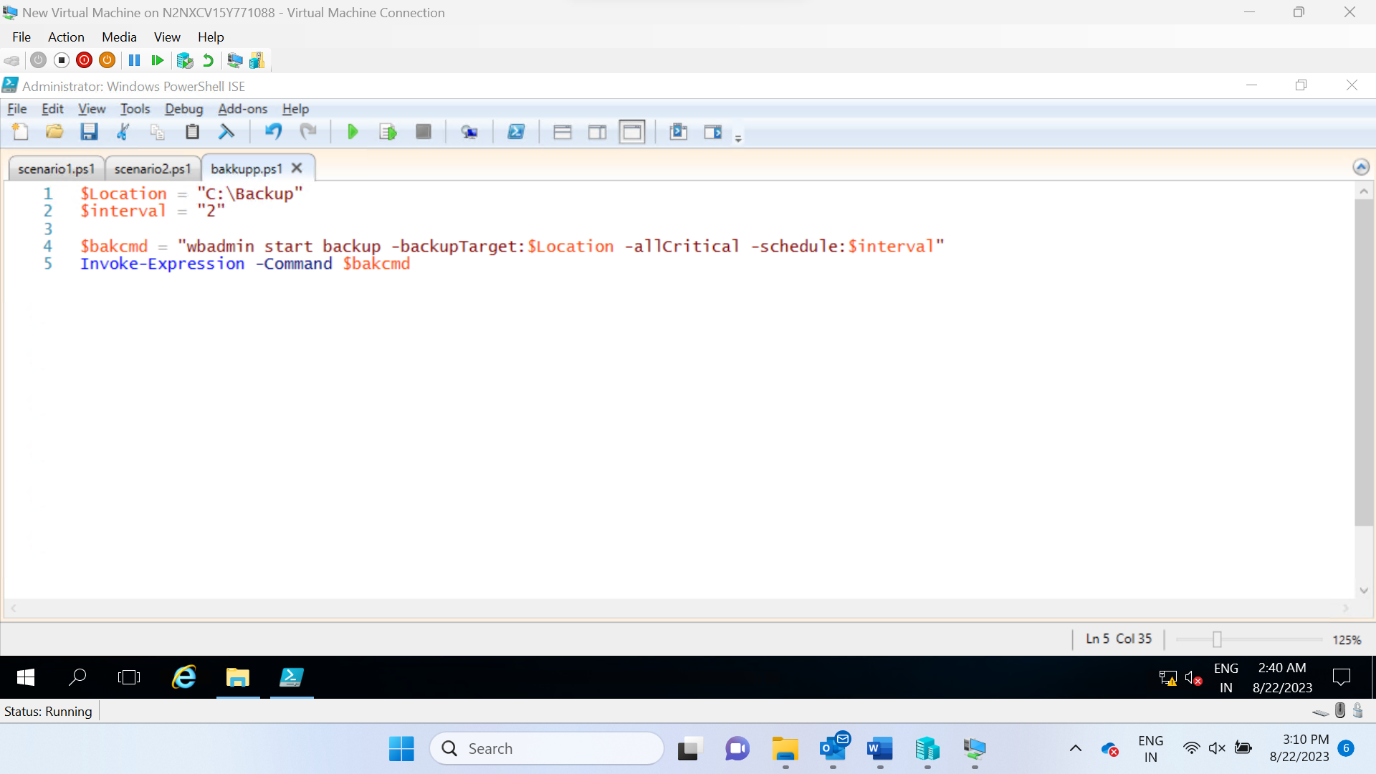
A computer screen shot of a computer

Description automatically generated

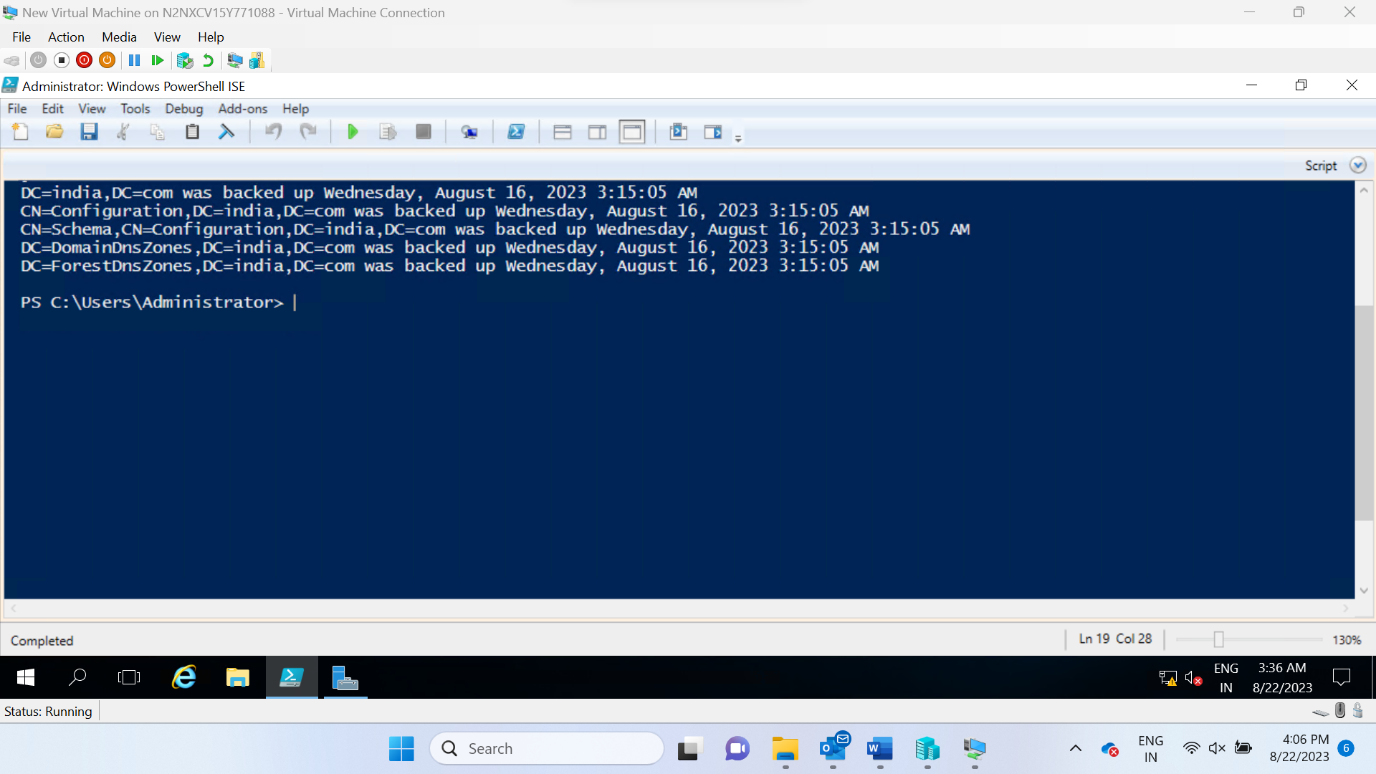
Scenario 3

Powershell script to backup AD server every 2 days.

Using Invoke-Expression Create a variable for storing backup location and variable for storing backup intervel and then create a variable which stores command and use the varible inside the invoke expression to execute the backup



After Running the script the backup taken will be shown in the output



Scenario 4

Powershell script to monitor critical services wrt to Active Directory. If any service is stopped, it should start the service.

Using **Get-service** and **where-object** we can extract the critical services of Active directory

Create a variable and save the output to the variable

Use if to check whether the service status is running if it is not running Using command called **start-service** we Can restart the service

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Scenario 5

Write a powershell script for employee management (Add employee, delete employee, salary calculation)

Create an array to store the employee information

Create a function using function keyword to add employee

A screenshot of a computer

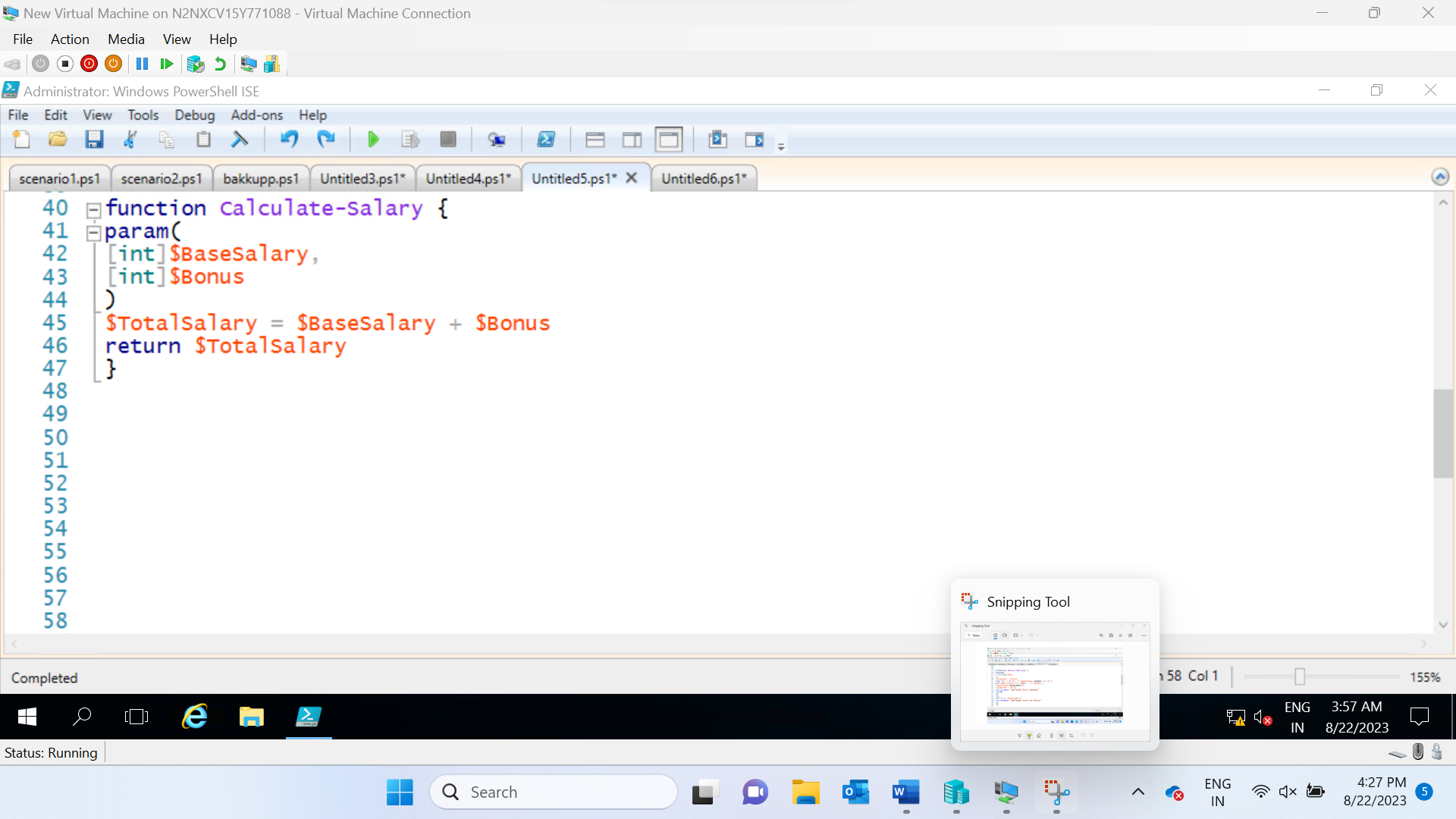
Description automatically generated

Create a function to remove employee which search for the employee name and remove all the details of the emoployee in the array

A screenshot of a computer

Description automatically generated

Create a function to calculate salary which takes bonus and base salary as an input and calculate the salary



After that call the function to get the output

A screenshot of a computer

Description automatically generated

Scenario 6

Write a powershell script for creating a calculator program (add, subtract, multiply, divide, square, square root). User should be able to give any number of inputs.

Create a calculator by using while to get input continously and if else and else if operations to check various conditions and select thee operators according to the input of the user

do

{

Write-Host "`t`t`t Arithmetic Operations `t`t`t`n" -BackgroundColor Darkred

Write-Host "1.Addition `n"

Write-Host "2.Subtraction `n"

Write-Host "3.Multiplication `n"

Write-Host "4.Division `n"

Write-Host "5. Squareroot of first number`n"

Write-Host "6. Squareroot of second number`n"

Write-Host "7. Square of first number`n"

Write-Host "8. Square of second number`n"

$choice=Read-Host "Enter the choice (1,2,3,4,5,6,7,8)?"

[decimal]$num1=Read-Host "Enter a number"

[decimal]$num2=Read-Host "Enter a number"

if($choice -eq 1)

{

$num=$num1+$num2

Write-Host "Addition is $num"

}

elseif($choice -eq 2)

{

$num=$num1-$num2

Write-Host "Subtraction is $num"

}

elseif($choice -eq 3)

{

$num=$num1\*$num2

Write-Host "Multiplication is $num"

}

elseif($choice -eq 4)

{

$num=$num1/$num2

Write-Host "Division is $num"

}

elseif($choice -eq 5)

{

$num = [math]::sqrt($num1)

Write-Host "The squareroot is $num"

}

elseif($choice -eq 6)

{

$num = [math]::sqrt($num2)

Write-Host "The squareroot is $num"

}

elseif($choice -eq 7)

{

$num=$num1\*$num1

Write-Host "The square is $num"

}

elseif($choice -eq 8)

{

$num=$num2\*$num2

Write-Host "The square is $num"

}

else

{

Write-Host "Invalid Input"

}

}

while ($choice -in 1..8)

Here is the user entered value and the output of the input

A screenshot of a computer

Description automatically generated

Scenario 7

Script to add,delete, change the user attributes of users in Active directory.(In menu type)

function Show-Menu {

Clear-Host

Write-Host "Active Directory User Attributes Management"

Write-Host "1. Add User"

Write-Host "2. Delete User"

Write-Host "3. Change User Attributes"

Write-Host "4. Exit"

}

Create a loop and show menu opitons as add,delete and modify users in active directory

Create a function to add users to active directory users

function Add-User {

$username = Read-Host "Enter username"

$givenName = Read-Host "Enter first name"

$surname = Read-Host "Enter last name"

New-ADUser -Name $username -GivenName $givenName -Surname $surname

Write-Host "User $username added."

}

Create a function to remove users to active directory users

function Delete-User {

$username = Read-Host "Enter username to delete"

Remove-ADuser $username -Confirm -ErrorAction SilentlyContinue

Write-Host "User $username deleted."

}

Create a function to change attributes to active directory users

function Change-UserAttributes {

$username = Read-Host "Enter username to modify"

$attributeName = Read-Host "Enter attribute name"

$attributeValue = Read-Host "Enter new value"

Set-ADUser $username -GivenName -Replace @{$attributeName = $attributeValue}

Write-Host "Attribute $attributeName updated for user $username."

}

Create a loop to do this action again and again

while ($loop) {

Show-Menu

$choice = Read-Host "Enter your choice"

switch ($choice) {

"1" { Add-User }

"2" { Delete-User }

"3" { Change-UserAttributes }

"4" { $loop = $false }

default { Write-Host "Invalid choice. Please select a valid option." }

}

}

Here is the screenshot of the above functions

A screenshot of a computer

Description automatically generated

Scenario 8

Host a website using powershell script.

Create a HTML page for our website using powershell we are hosting a website First create a index.html file

A screenshot of a computer

Description automatically generated

Give the filepath to the Script in the directory path and give the do website name with url and try to run the script

$website = "C:\Users\Administrator\Desktop\index.html"

New-website -name "www.google.com" -port 80 -physicalpath $website

start-website -name "Google"

Open the browser and type the url of your website and check whether the website is loading correctly or not

A screenshot of a computer

Description automatically generated

**------------------------------------------END------------------------------------------------**

