Prerequisite: Determine the private IP of the specified servers

Windows Server 2022 Datacenter server

Hostname: EC2AMAZ-RFE9KT4

Public IP: 35.171.166.90

Private IP: 172.31.3.39

(obtained via ipconfig)

Windows Server 2022 Datacenter server

Hostname: EC2AMAZ-REFBCFS

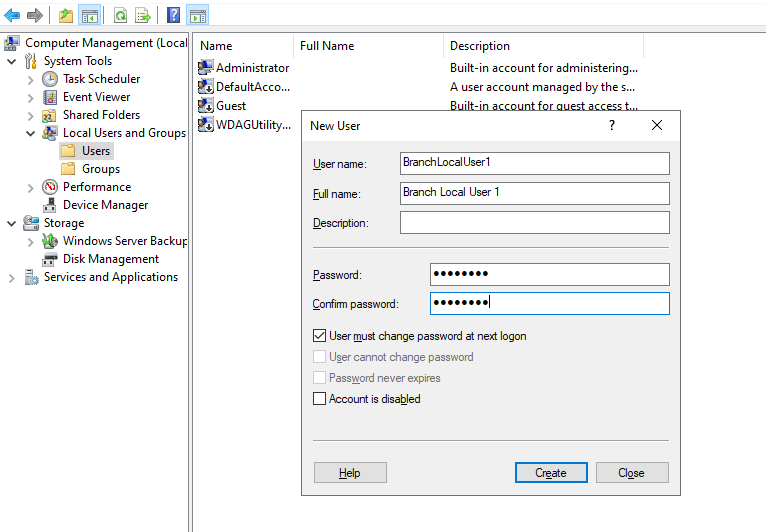
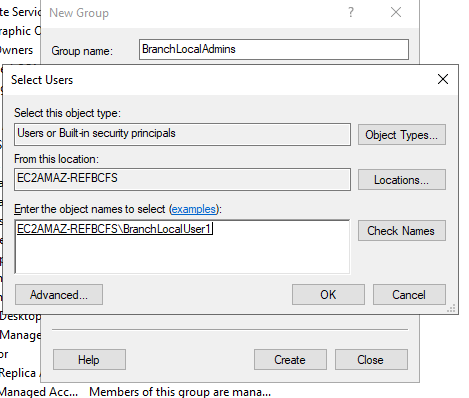
Public IP: 54.152.62.155

Private IP: 172.31.3.176

(obtained via ipconfig)

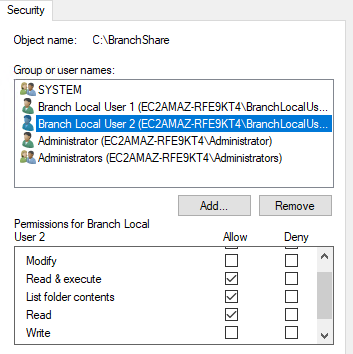
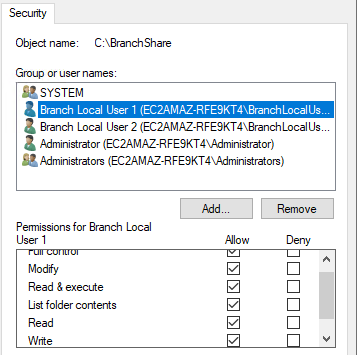
## Task 1:

Using the GUI - Create local user/groups for BranchLocalUser and BranchLocalAdmins group for each respective VM.

1. Right-click start menu - open Computer management
2. Open Local Users and Groups snap-in under System Tools
3. Right-click Users -> New User
4. Set pass to S@f3P@$$ and check ‘User must change password at next logon’.  
   
5. Repeat the steps for BranchLocalUser2 - but without the change password option (as that was not defined).
6. Right click on ‘Groups’ in the navigation pane, and click New Group.
7. Add the BranchLocalUser1 user to the group, and hit OK, then on the New Group window, click Create.  
   

NOTE: This could also be done with PS, but given there’s a separate task for this, I felt it made more sense to do it via GUI.

## Task 2 - Part 1

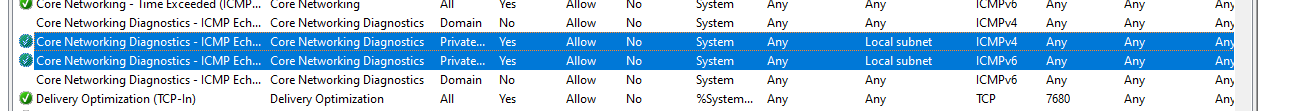
1. Create the folder in C:\ Directory.
2. Go into directory properties and under Share tab click Share
3. Select each user and grant respective Read or Read/Write permission.  
   \\EC2AMAZ-RFE9KT4\BranchShare is the UNC path
4. In the security tab, Click Edit then select Add to add users. Add both Users, and then configure respective permissions as below:  
      
     
   The Read/Write permissions could, and normally would be granted to the LocalBranchAdmins group as opposed to specific users.

Go on to Task 3 to allow file sharing.

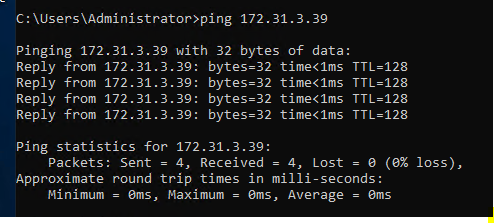
## Task 3

Go into Firewall Advanced Settings and Inbound rules for each server and setup the following:

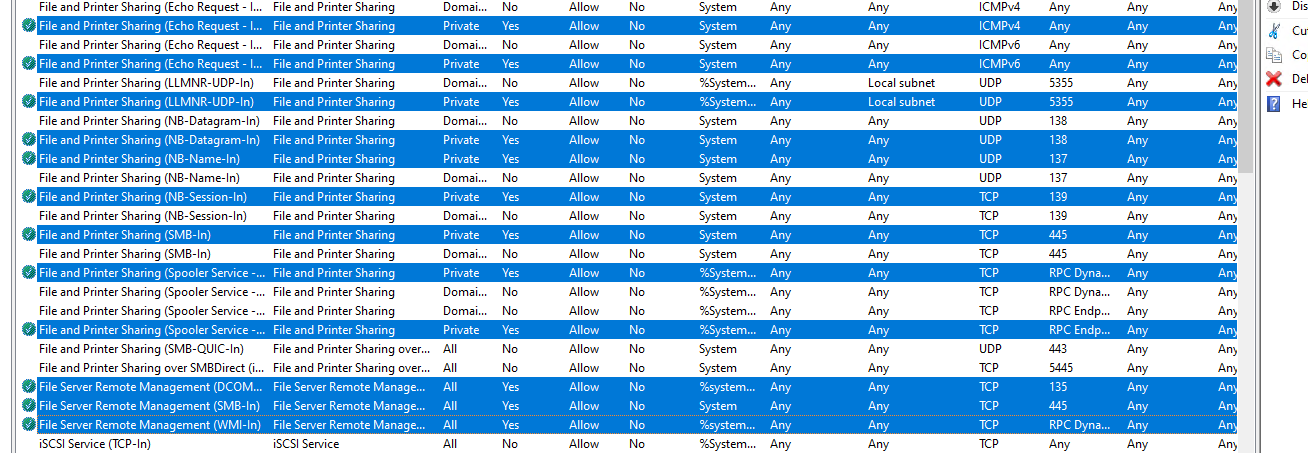
ICMP (protocol used for ping) on the private network to enabled



Validated ping successful both ways.



Test will be validated in TASK 2 part 2



I checked VM2 and the network was set as Public, so I changed it to Private, as this was a private network. Done in Local Security Policy

## Task 2 - Part 2

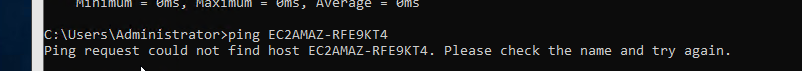
During trial I noticed the VM2 cannot access the network share. See next Task 5.

## Task 5

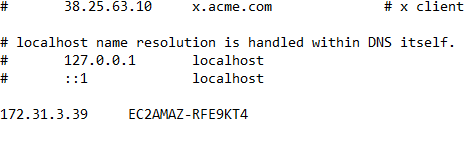
Network share was not working when using the UNC Address.

Ping from VM2 to VM1 using the IP Private Address was successful.

Ping the VM2 to VM1 user host name was not successful.



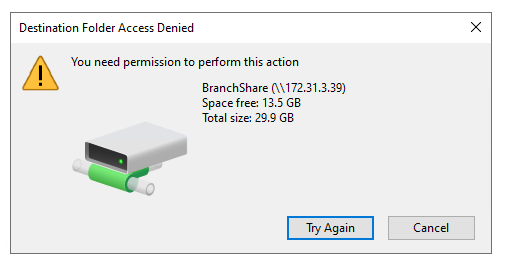
As I do not have access to DNS server at 172.31.0.2 - I created a host file entry that pointed to the correct IP.



Testing again I was able to make it work and ping the host name and connect to the shared directory using UNC path. Normally, I would make update in DNS though and then ipconfig /flushdns.

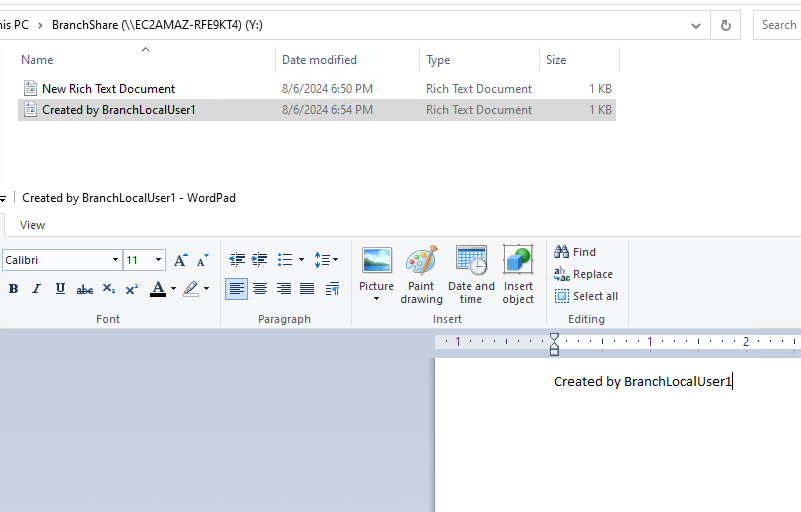
## Task 2 - Part 3

As BranchLocalUser2 I attempted to modify/create document and was successfully denied. I could open the existing test document I created.



To test file sharing, I had to set new password first - set to N3wP@SS

I was able to create new document as BranchLocalUser1



## Task 4

Creating the PowerShell script was a bit tricky due to the fact that the servers aren’t on the same domain and there is not one user on both servers that could run the commands with Admin Privileges. Nevertheless, I don’t believe it sensible to run different scripts or re-write them.

If this was an admin user account with admin access to both computers, the following script and adjoining text document would do the job for both at once.

<https://github.com/bshaulov85/GoTropical-PS-Test>

Here is how it works:

The beginning script block is the script that will run when called. It contains variables for:

* PWD - Password
* Name - Username
* Description
* AcctExpiration

Then it runs the command, leaving off the -ComputerName portion for the remote user account to be created on. That will occur in the loop later.

The first three variables are self explanatory.

The AcctExpiration uses the Get-Date cmdlet then uses .AddDays to add 30 days, this way you don’t have to enter the exact date every time, if you re-use this script, you can simply know it will always use Account Expiration 30 days from the day the script is run.

Then the actual command is there, using New-LocalUser Cmdlet.

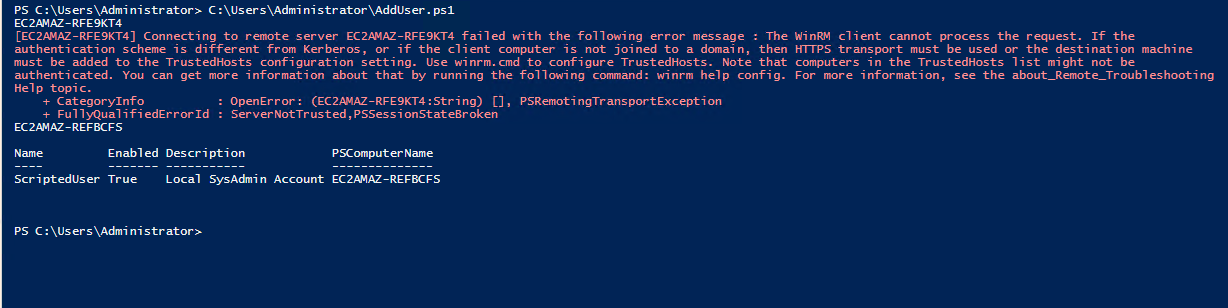
End Script Block

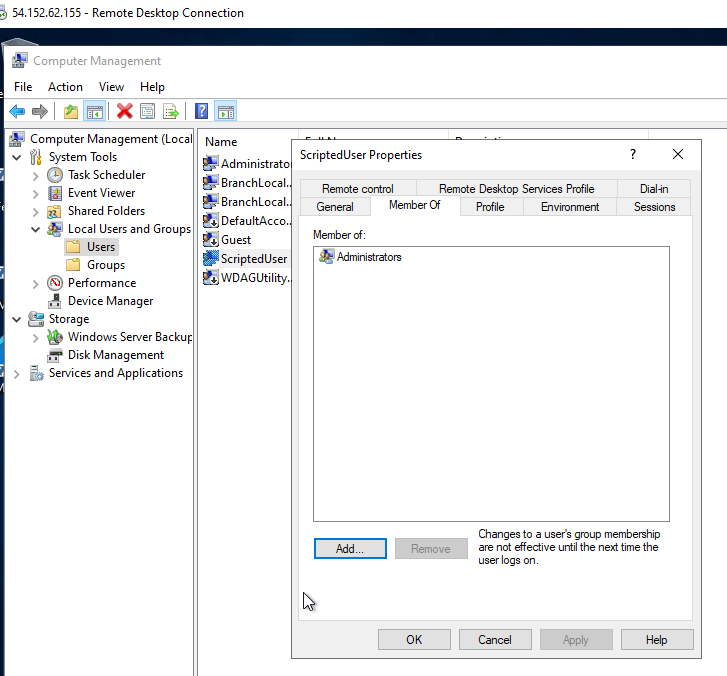
Next we import the servers.txt (the file location must be in C:\Users\Administrator\ directory, but this could easily be modified to utilize an environmental variable within Windows).

Finally we loop through each of the servers in the Servers.txt and run the associated script block along with the -ComputerName parameter using Invoke-Command

Testing:

As predicted, it did not work on the second VM, but the script worked on the local machine. My expectation is that if this was a domain account with admin access to both, it would have worked on both.





I then used the network share I created to copy the script, put into the correct directory and ran the same script in elevated PS terminal successfully.

