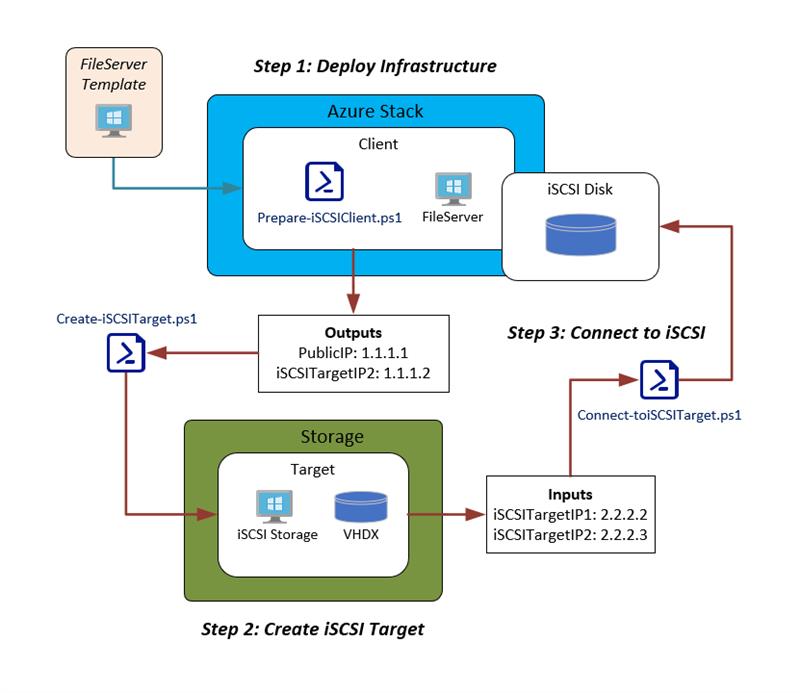
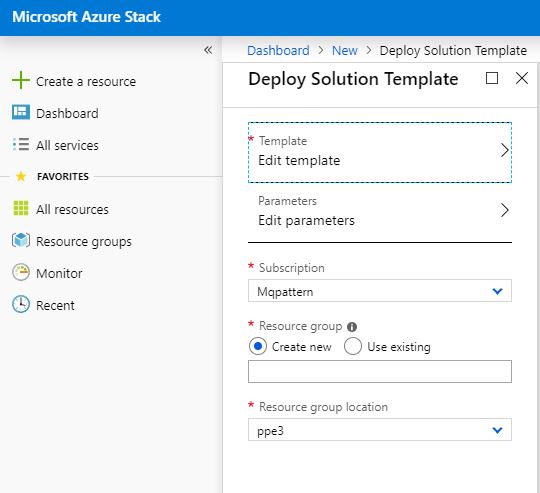
****

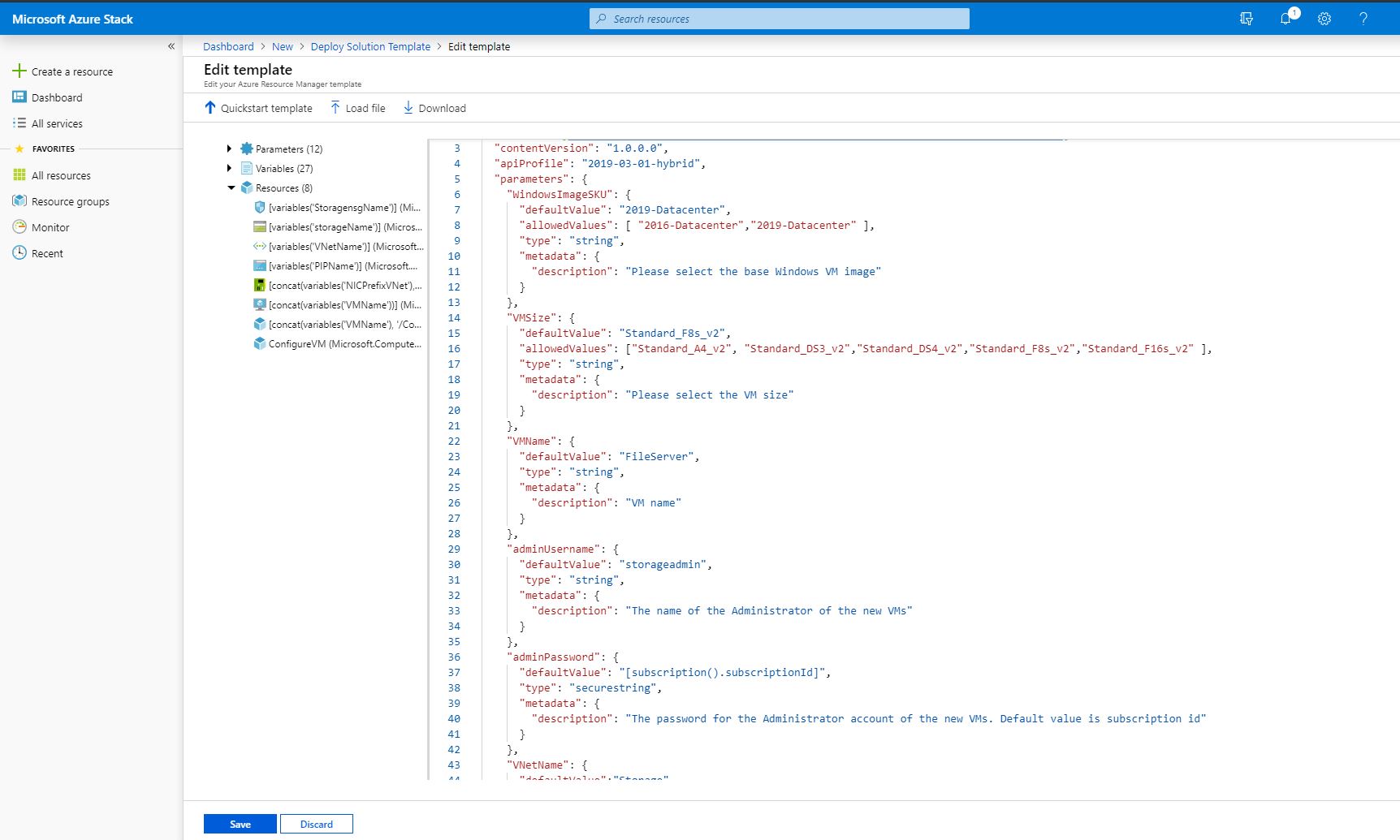
# Setting up iSCSI with a new vm using a template

**Deploying the Template:**

1. Go To <https://github.com/lucidqdreams/azure-intelligent-edge-patterns/tree/master/storage-iSCSI> and click download then Download ZIP to download the template and dsc files an save to the machine you will perform the install from.
2. Extract Files from downloaded .zip.
3. Open a browser to the Azure Stack Portal Site ( <https://portal.xxx.xxx.xx.xx>). Then login with an that access to your Azure Stack subscription.
4. Click on Create a resource then Template deployment.
5. Click on Edit template.



1. Click on Load Files and navigate to where your extracted template is located. Click on azuredeploy.json then click Save.



1. Click Parameters and you will see the parameters and will see the list of parameters below. Fill in the required information and click **OK**.

**Note:** If this is for a test environment you can just fill in the ADMINPASSWORD parameter and the rest of the defaults will work.

**WINDOWSIMAGESKU:** This is the SKU that will be used for the vm deployed on Azure stack as the iSCSI initiator.

**VMSIZE:** Select the size you want to use for the Azure stack iSCSI initiator vm.

**VMNAME:** Enter the name you want to use for the Azure stack iSCSI initiator vm.

**ADMINUSERNAME:** This will be the name of the Administrator for the Azure stack iSCSI initiator vm.

**ADMINSPASSWORD:** The password for the Administrator account of the new VM. Default value is subscription id.

**VNETNAME:** The name of VNet. This will be used to label the resources installed in VNet.

**VNETADDRESSSPACE:** Theaddress space that will be used for VNet.

**VNETINTERNALSUBNETNAME:** The name that will be uses for the internal Subnet.

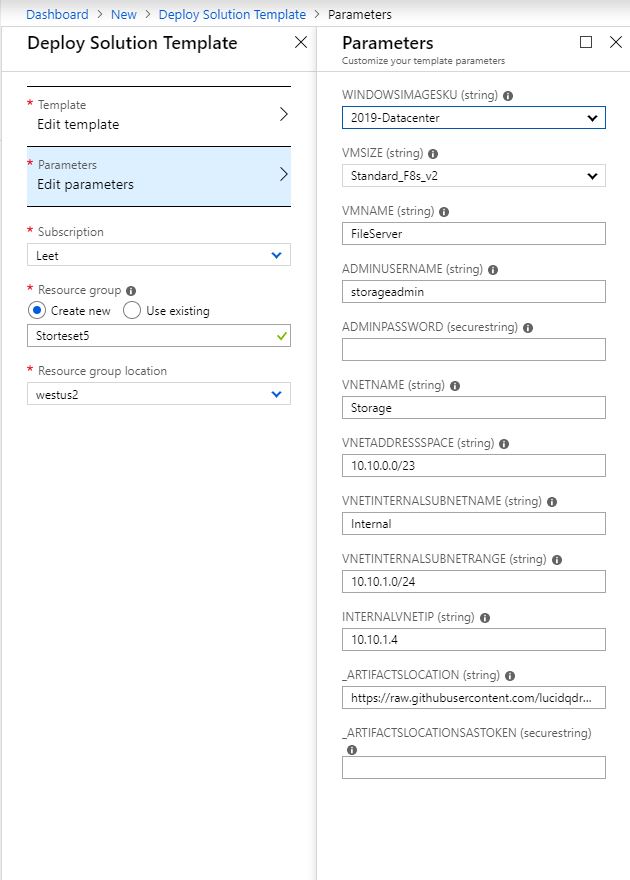
**VNETNTERNALSUBNETRANGE:** Address Range for Internal Subnet.

**INTERNALVNETIP:** Static Address for the internal IP of the File Server.

**ARTIFACTSLOCATION:** The location of resources, such as templates and DSC modules, that the template depends on.

**ARTIFACTSLOCATIONSASTOKEN:** Auto-generated token to access \_ARTIFACTSLOCATION. NOTE: This can be blank if no SAS token is required.

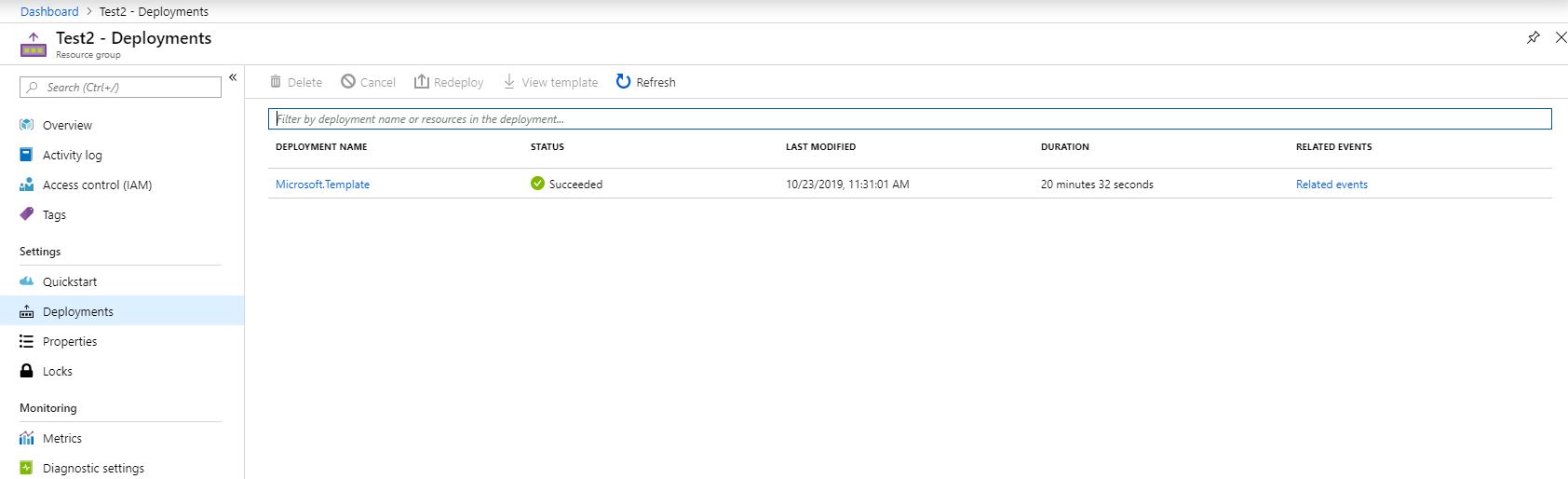
1. Next, select or create a resource group where the iSCSI initiator deployment will reside.
2. Once you are happy with your parameters and resource group click Create and the deployment will start.



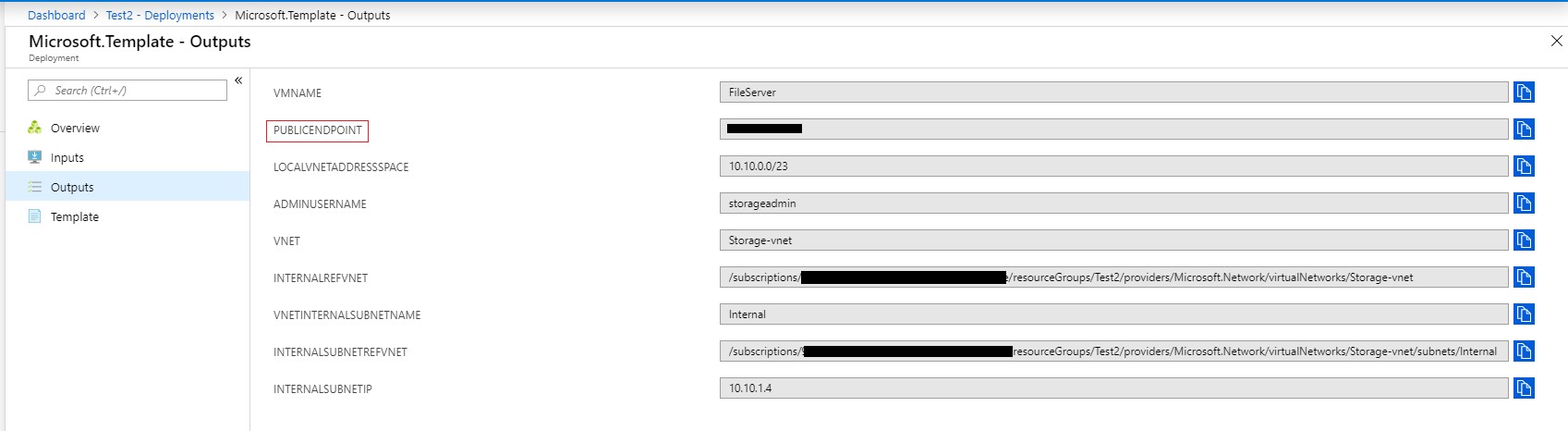
1. Once the Deployment has succeeded you can move on to preparing the iSCSI target vm.

**Preparing iSCSI Target:**

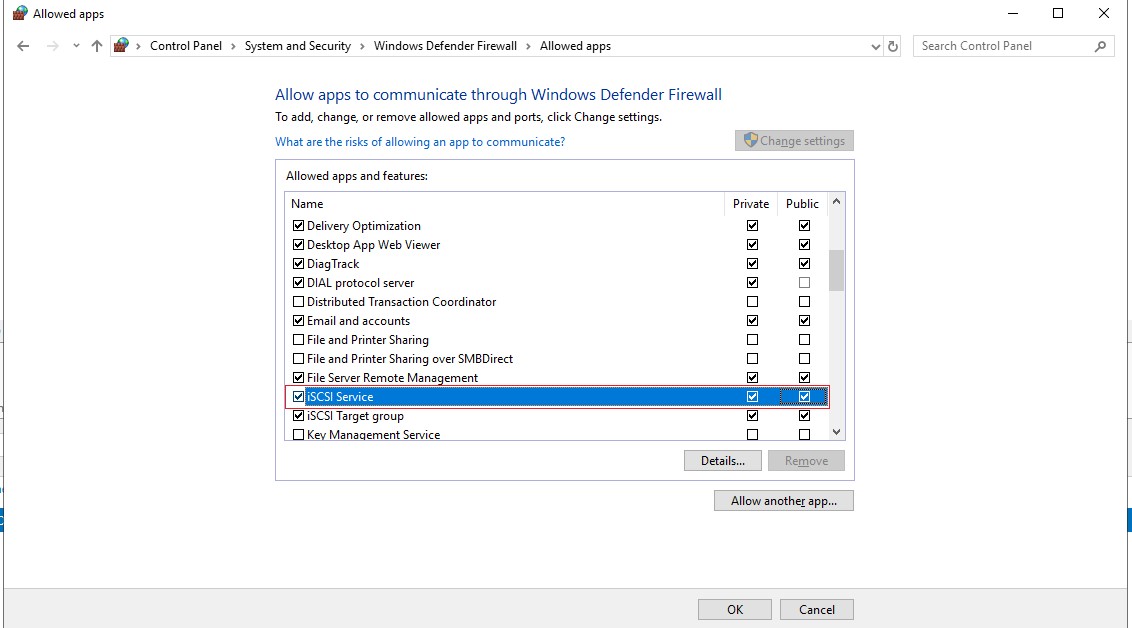
1. Open a browser to the Azure Stack Portal Site ( <https://portal.xxx.xxx.xx.xx>). Then login with an that access to your Azure Stack subscription.
2. Go to the resource group where installed the iSCSI initiator and click on **Deployments**.



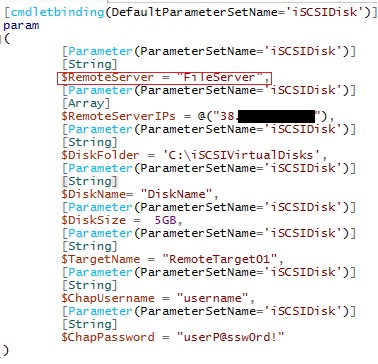
1. Click on the deployment then click on Outputs.
2. Take note of the PUBLICENDPOINT output. This will be the public IP address of the iSCSI initiator vm.



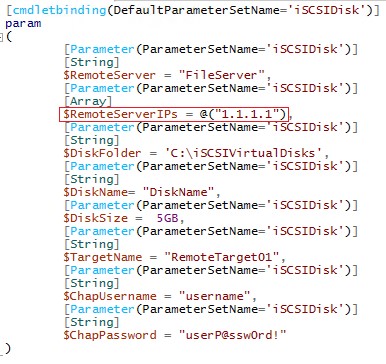
1. Login to the machine that you want to use as the iSCSI target.
2. Go to Start and open up **Windows Defender Firewall**.
3. Click on **Allow an app or feature through Windows Defender Firewall**.
4. Scroll down to iSCSI Service and check all three of the checkboxes for the App, Private and Public then click OK to apply.



1. Now open up the create-iSCSItarget.ps1 script in the PowerShell ISE.
2. Update the $RemoteServer parameter with the hostname for the new iSCSI client you just prepared.



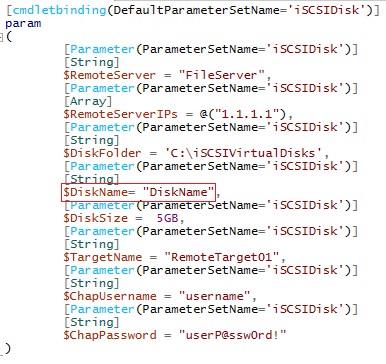
1. Go to the $DiskSize Parameter and adjust the disk size to the size you need. The default is 10GB.
2. Take the PublicENDPOINT ip you captured earlier and replace the IP in the $RemoteServerIPs Parameter in your script.



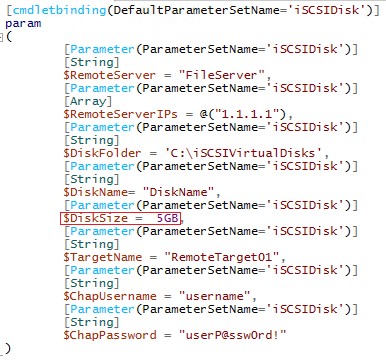
1. Update the $DiskFolder parameter with the path where you want the folder to live.



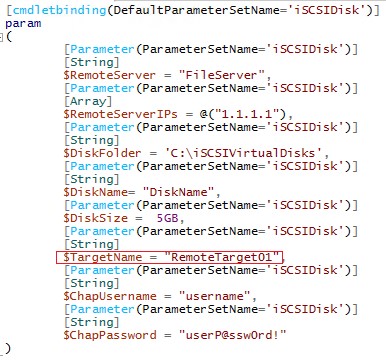
1. Update the $DiskName parameter with a meaningful name.



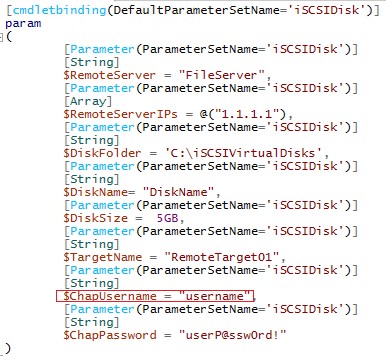
1. Update the $DiskSize parameter with the size that you would like to make the disk.



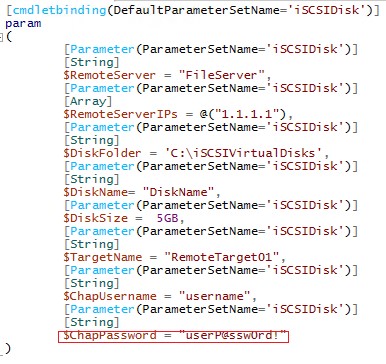
1. Update the $TargetName parameter to what you would like to name your iSCSI target.



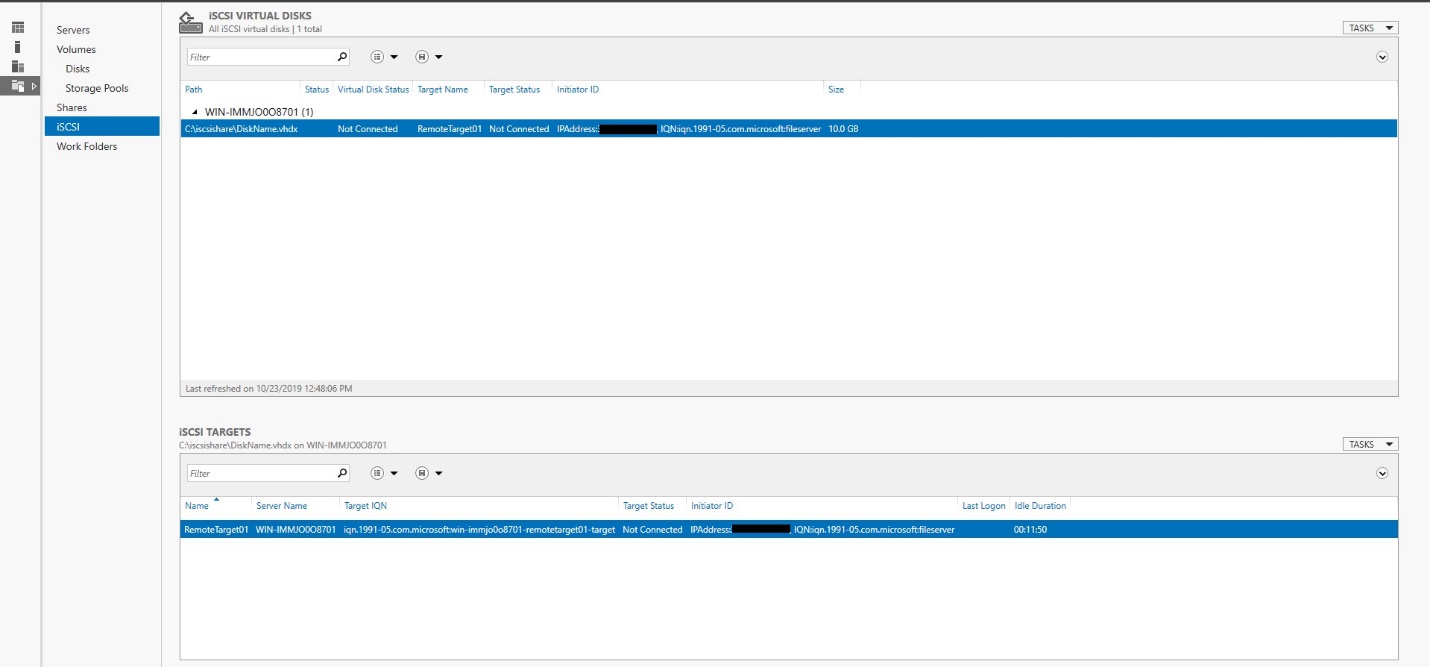
1. Update the $ChapUserName parameter to to something meaningful.



1. Update the $ChapPassword parameter to something that fits your password policy.



1. Once you are happy with your parameters press F5 and run the script. **NOTE:** When the script is complete it will reboot the machine.
2. Once the machine has rebooted you can take a look to make sure everything has been deployed. You should see an iSCSI virtual disk and an iSCSI target.



**Connecting to iSCSI client:**

1. Open a browser to the Azure Stack Portal Site ( <https://portal.xxx.xxx.xx.xx>). Then login with an that access to your Azure Stack subscription.
2. Go to the resource group you deployed earlier, click on the virtual machine and click on **Connect,** then Download RDP File. Then login to your virtual machine.
3. Once your RDP file has downloaded click Connect and login.
4. Copy **the Connect-toiSCSITarget.ps1** file over to your virtual machine and open it in PowerShell.
5. Go to the param block in the script and change the $TargetiSCSiAddresses to be the IP address or IP addresses of the iSCSI target you created earlier. **NOTE:** It is recommended that there be 2 nics for the iSCSI target for redundancy.



1. Then go the $LocalIPAddresses and enter the IP of the iSCSI initiator vm you are logged into now. **NOTE**: If you used the defaults in the template this does not need to be changed.



1. You can also change the name Load Balance Policy name by changing $LoadBalancePolicy to whatever name makes sense for you.



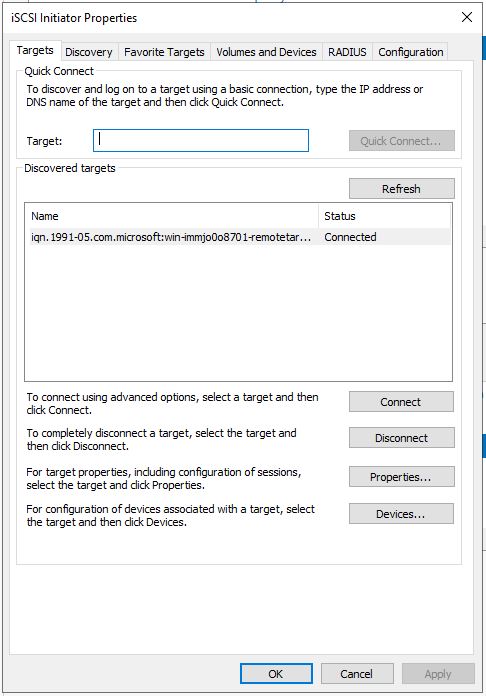
1. Edit the $ChapUserName to a meaningful user name.



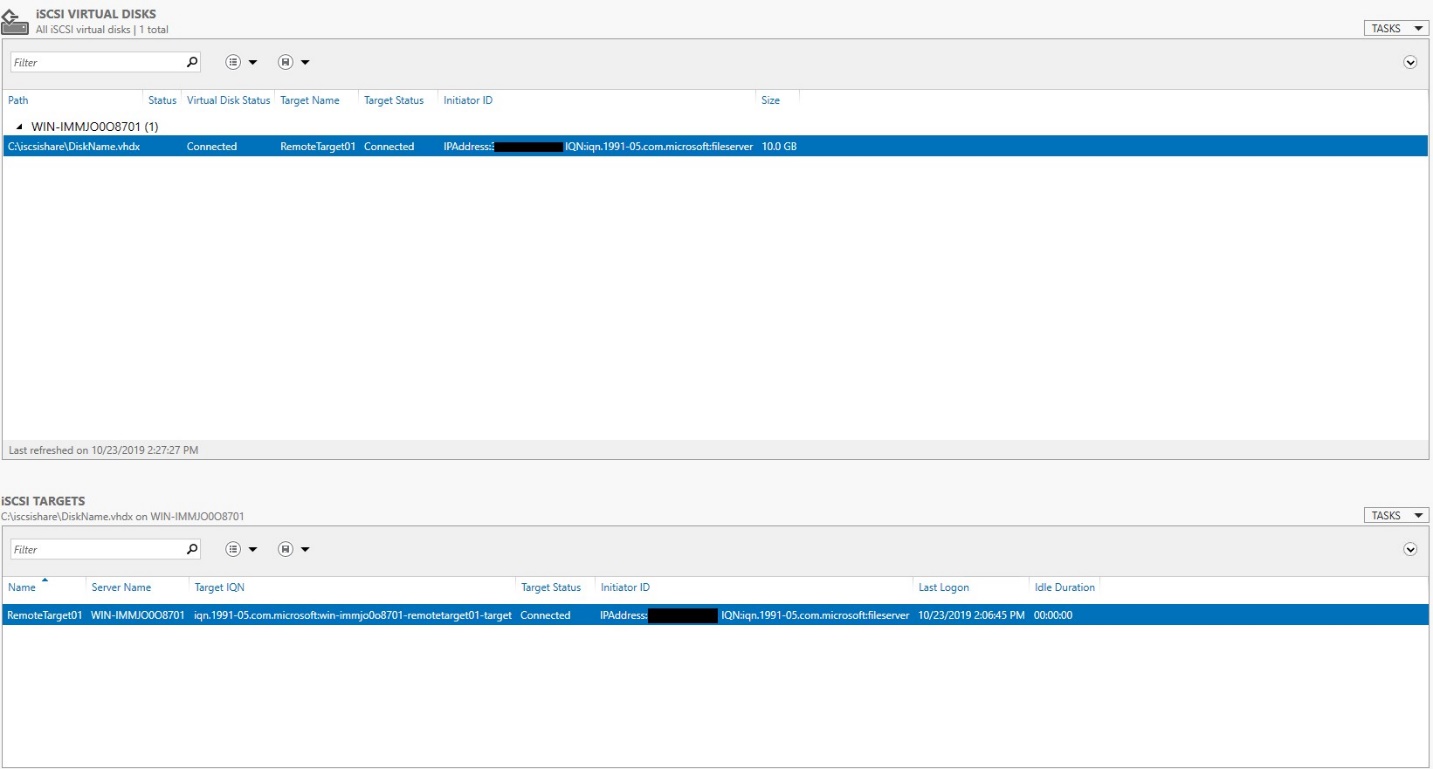
1. Edit the $ChapPassword to something that fits your password policy.



1. Once these changes are made you can hit F5 to run the script.
2. After the script completes go to Server Manager\Tools\iSCSI Initiator.
3. On the Targets screen under Discovered Targets you should have one remote target with a status of Connected.



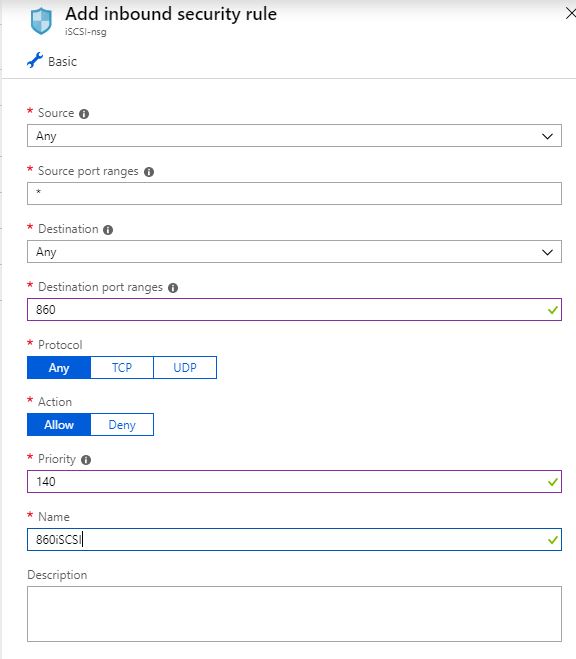
1. You can verify this from the Target side by logging in to your target server. Then go to Server Manager and click iSCSI.
2. The status for the iSCSI virtual Disks and iSCSI targets should now show as Connected. NOTE: you may need to refresh for these to show up if you previously had this view open.



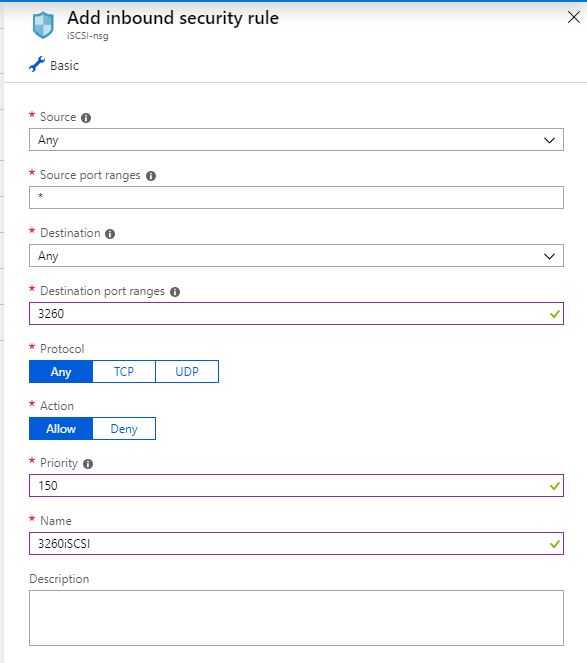
# Deploying Additional Disks to existing Target vm:

**Prepare Existing vm:**

1. Open a browser to the Azure Stack Portal Site ( <https://portal.xxx.xxx.xx.xx>). Then login with an that access to your Azure Stack subscription.
2. Go to the resource group where your new client vm is deployed and go to the overview of the vm and make note of the **Public IP Address** and **vm name**.
3. Next go to the NSG attached to the nic that you will be using for you iSCSI client.
4. Click inbound security rules and then click add.
5. For destination port ranges enter **860** for priority select **140** and for name **860iSCSI**.



1. For destination port ranges enter **3260** for priority select **140** and for name **3260iSCSI**.



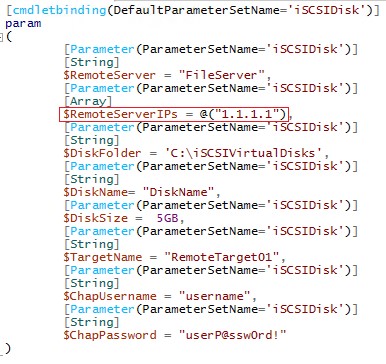
1. Login to your existing client vm.
2. Copy the Prepare-iSCSIClient.ps1 to your new client vm
3. Run **Prepare-iSCSIClient.ps1**. No inputs are required as this script only installs prerequisites for the iSCSI client.
4. Once the install has completed reboot your client vm.

**Connect to existing Target vm (using ISE):**

1. Login to your existing iSCSI target vm.
2. Open up the Create-iSCSITarget.ps1 script in the PowerShell ISE.
3. Update the $RemoteServer parameter with the hostname for the new iSCSI client you just prepared.



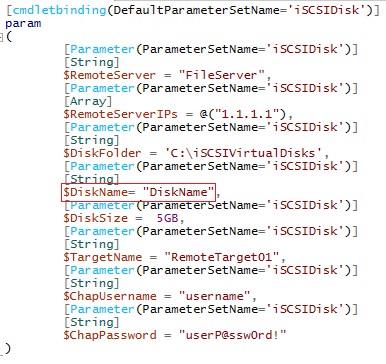
1. Update the $RemoteServerIPs parameter with Public IP address for the new iSCSI client you just prepared.



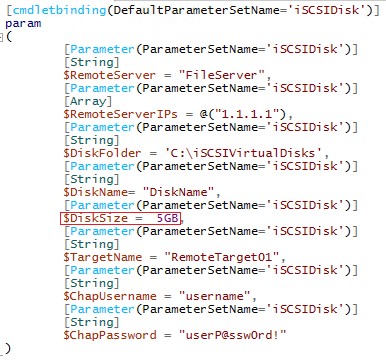
1. Update the $DiskFolder parameter with the path where you want the folder to live.



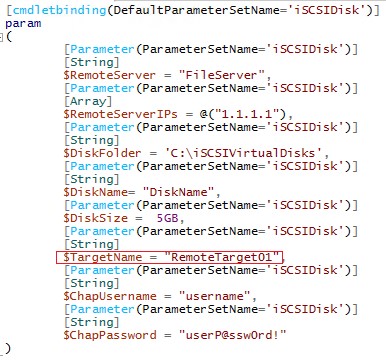
1. Update the $DiskName parameter with a meaningful name.



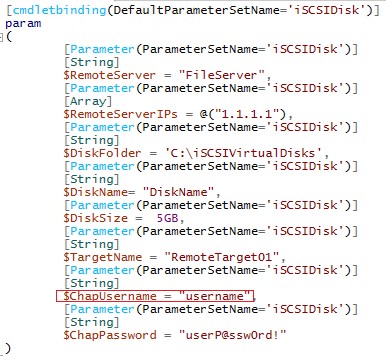
1. Update the $DiskSize parameter with the size that you would like to make the disk.



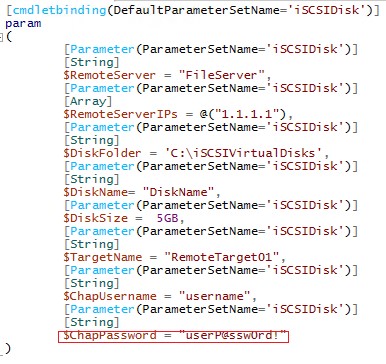
1. Update the $TargetName parameter to what you would like to name your iSCSI target.



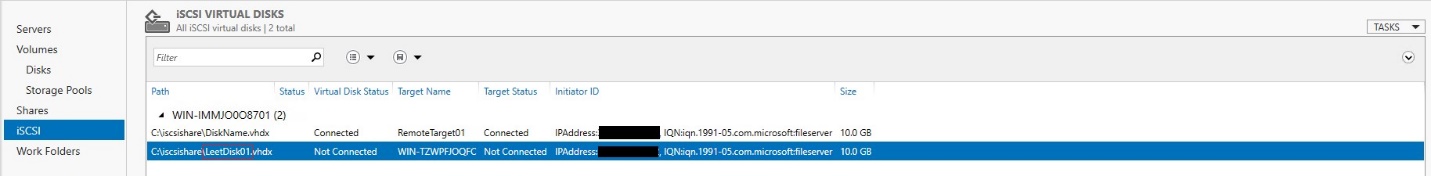
1. Update the $ChapUserName parameter to to something meaningful.



1. Update the $ChapPassword parameter to something that fits your password policy.

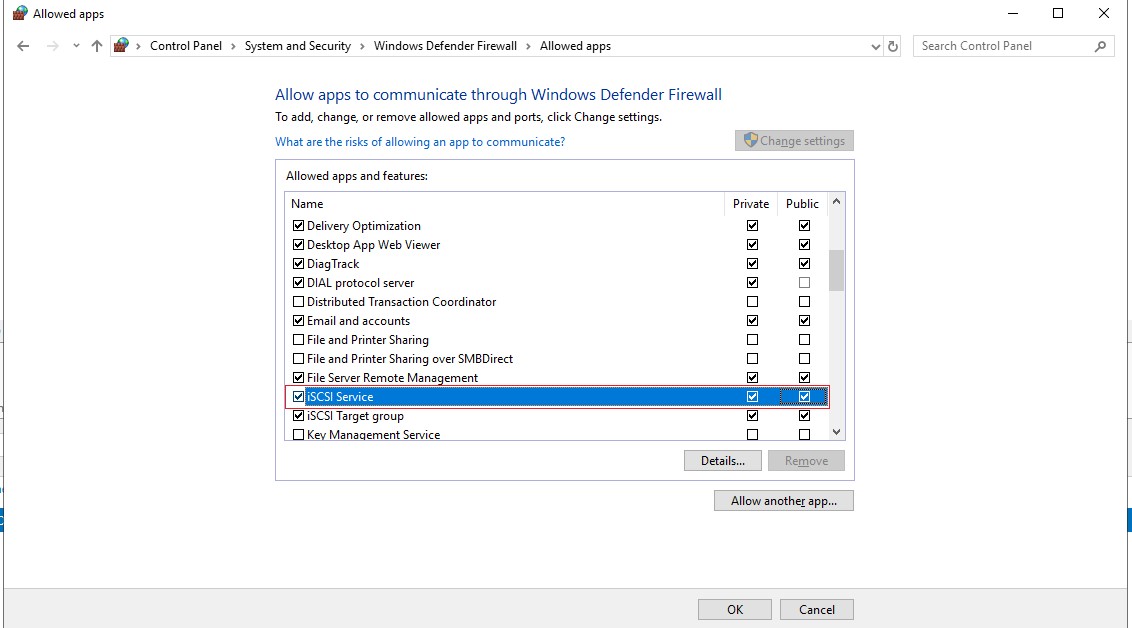


1. Once you are happy with your parameters press F5 and run the script. **NOTE:** When the script is complete it will reboot the machine.
2. Once the machine has rebooted you can take a look to make sure everything has been deployed. You should see a new iSCSI virtual disk.



**Connecting iSCSI client:**

1. Open a browser to the Azure Stack Portal Site ( <https://portal.xxx.xxx.xx.xx>). Then login with an that access to your Azure Stack subscription.
2. Go to the resource group you deployed earlier, click on the virtual machine and click on **Connect,** then Download RDP File. Then login to your virtual machine.
3. Once your RDP file has downloaded click Connect and login.
4. Go to Start and open up **Windows Defender Firewall**.
5. Click on **Allow an app or feature through Windows Defender Firewall**.
6. Scroll down to iSCSI Service and check all three of the checkboxes for the App, Private and Public then click OK to apply.



1. Copy **the Connect-toiSCSITarget.ps1** file over to your virtual machine and open it in PowerShell.
2. Go to the param block in the script and change the $TargetiSCSIAddresses to be the IP address or addresses of the iSCSI target you created earlier.



1. Then go the $LocalIPAddresses and enter the IP of the iSCSI initiator vm you are logged into now. **NOTE**: If you used the defaults in the template this does not need to be changed.



1. You can also change the name Load Balance Policy name by changing $LoadBalancePolicy to whatever name makes sense for you.



1. Edit the $ChapUserName to a meaningful user name.



1. Edit the $ChapPassword to something that fits your password policy.



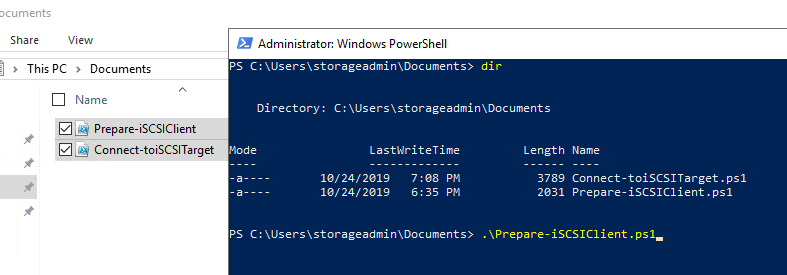
1. Once these changes are made you can hit F5 to run the script.
2. After the script completes go to Server Manager\Tools\iSCSI Initiator.
3. On the Targets screen under Discovered Targets you should have one remote target with a status of Connected.

# Connect 2 disks to an existing VM

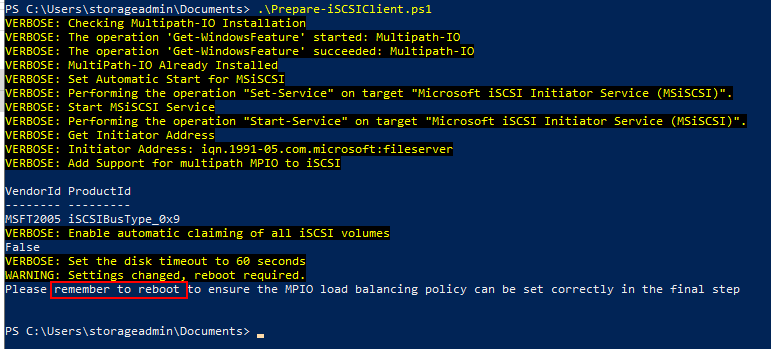
This walkthrough will cover using PowerShell to create and connect two disks from your iSCSI Target to an existing VM with multipathing to the target.

## Prepare iSCSI Client

1. On the existing VM copy these two scripts over and run the prepare-iSCSIClient.ps1

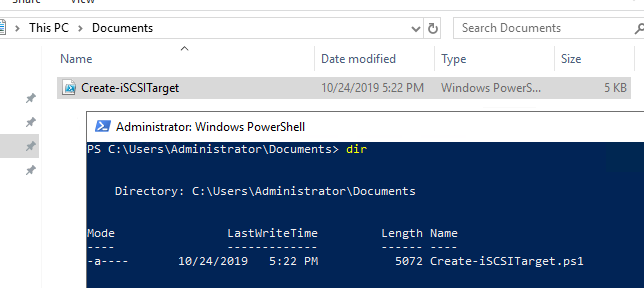


1. The execution will look like this. Remember to reboot the iSCSI client after executing this script

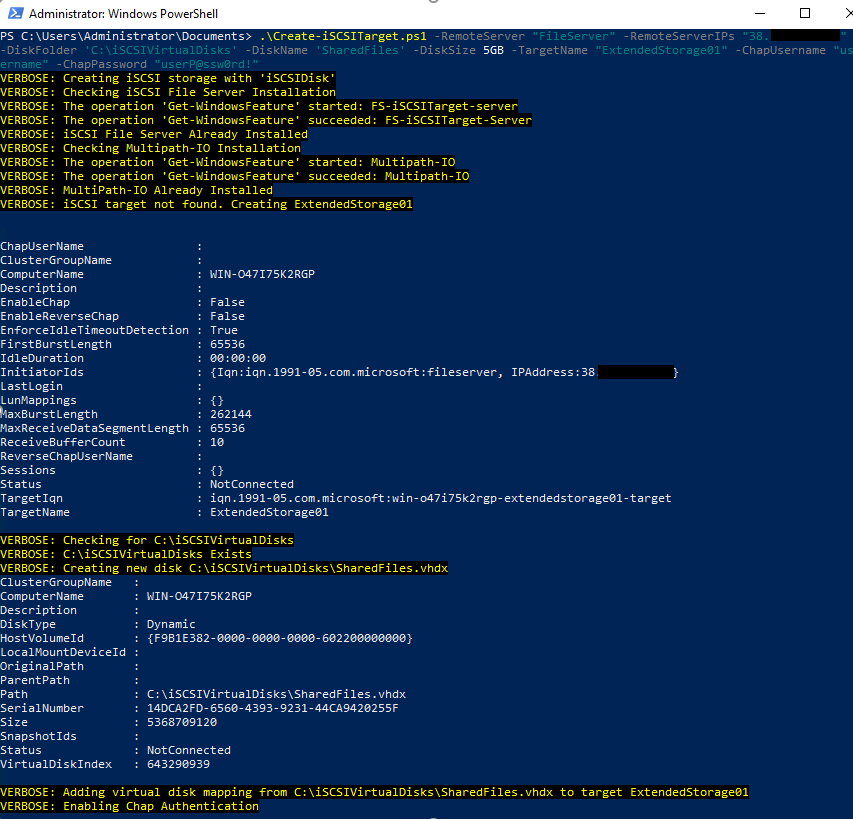


1. **Take note** of the Servers actual hostname and public IP address

## Create iSCSI Target

1. Copy over the create-iSCSITarget.ps1
2. 
3. Let’s create the first disk using this command

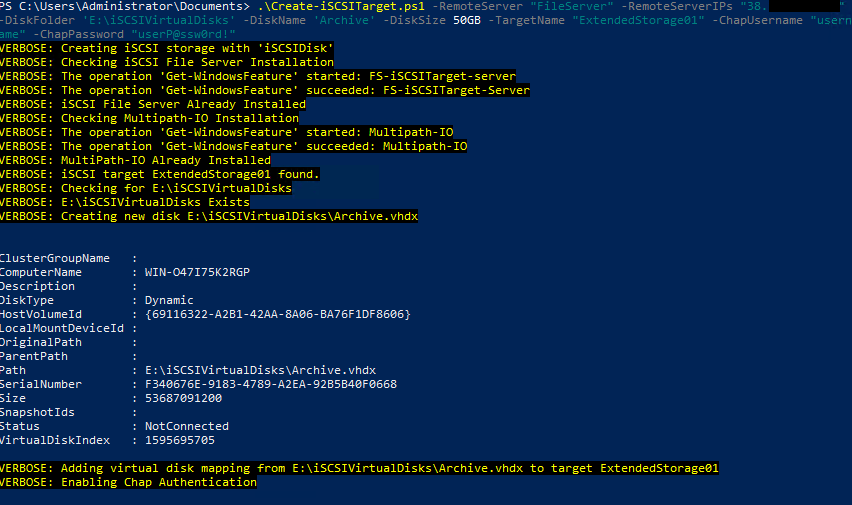
.\Create-iSCSITarget.ps1 -RemoteServer "FileServer" -RemoteServerIPs "38.x.x.x" -DiskFolder 'C:\iSCSIVirtualDisks' -DiskName 'SharedFiles' -DiskSize 5GB -TargetName "ExtendedStorage01" -ChapUsername "username" -ChapPassword "userP@ssw0rd!"



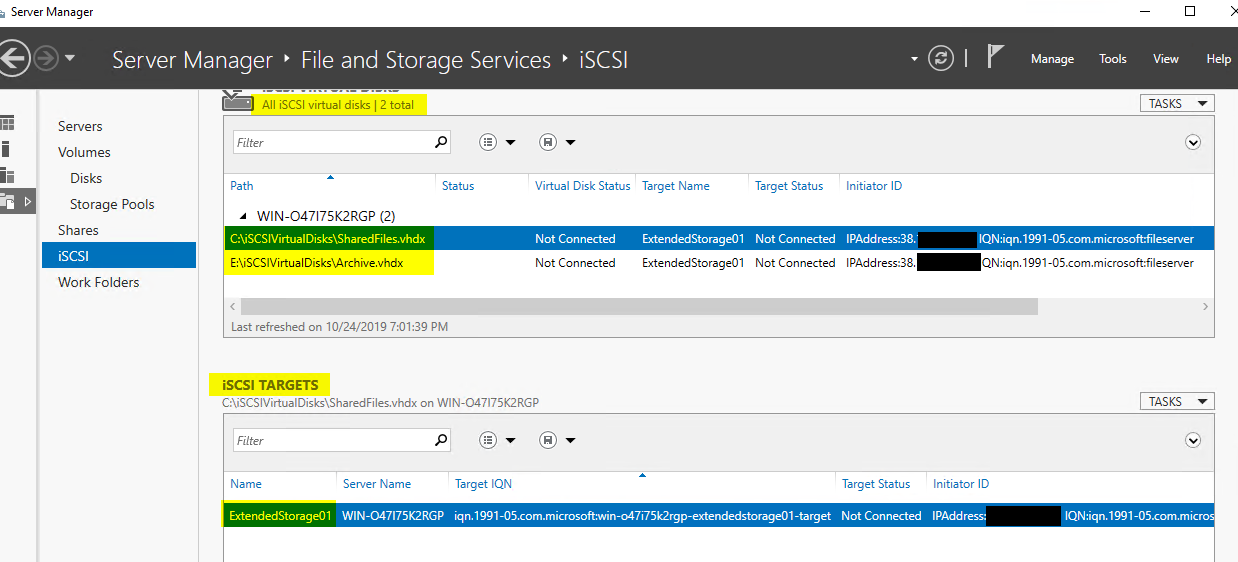
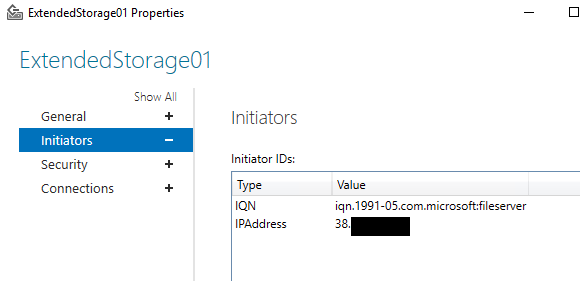
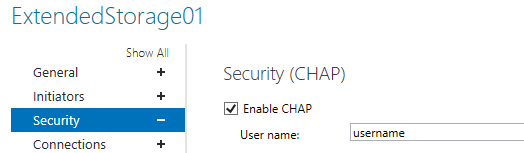
We can see the iSCSI target is created along with the virtual disk which is attached to that target.

1. Next, we want to create a second larger disk on the E drive called Archive.

.\Create-iSCSITarget.ps1 -RemoteServer "FileServer" -RemoteServerIPs "38.x.x.x" -DiskFolder 'E:\iSCSIVirtualDisks' -DiskName 'Archive' -DiskSize 50GB -TargetName "ExtendedStorage01" -ChapUsername "username" -ChapPassword "userP@ssw0rd!"



We can see that only the disk is created as the target already exists

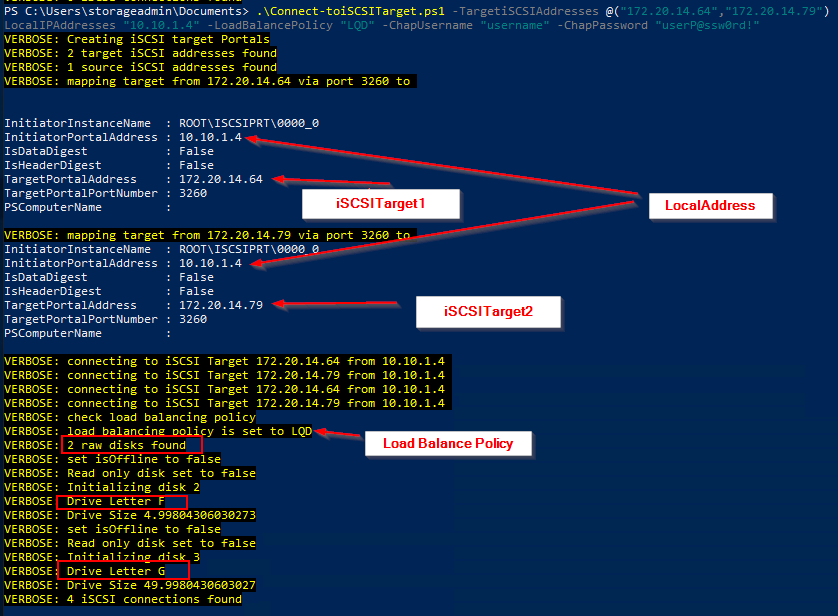
1. If you open Server Manager you can see the results
2. 
3. You can check the properties of the iSCSI target and review the initiators you requested in the command line.
4. 
5. 

### Connect iSCSI Client to iSCSI Target

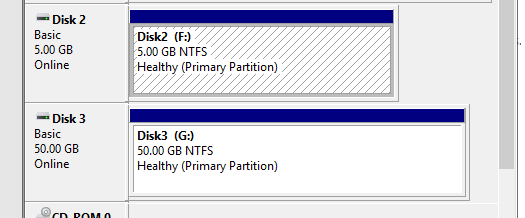
1. If you return to the iSCSI client machine we can now run the command to connect the disks
2. We can run one command to connect to both disks

.\Connect-toiSCSITarget.ps1 -TargetiSCSIAddresses @("172.20.14.64","172.20.14.79") -LocalIPAddresses "10.10.1.4" -LoadBalancePolicy "LQD" -ChapUsername "username" -ChapPassword "userP@ssw0rd!"

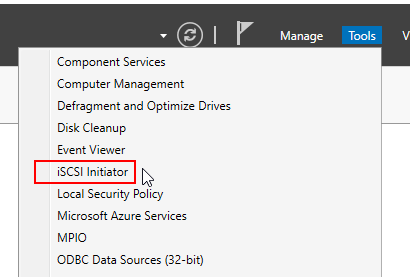
**Note** we have specified two TargetiSCSIAddresses as an **array** but using @(“”,””)



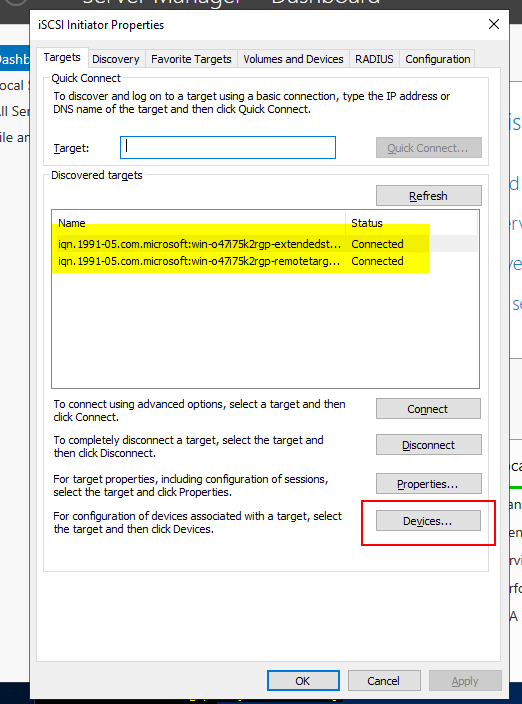
1. We can see the process and the 2 disks have been found, initialized and mapped to a drive letter



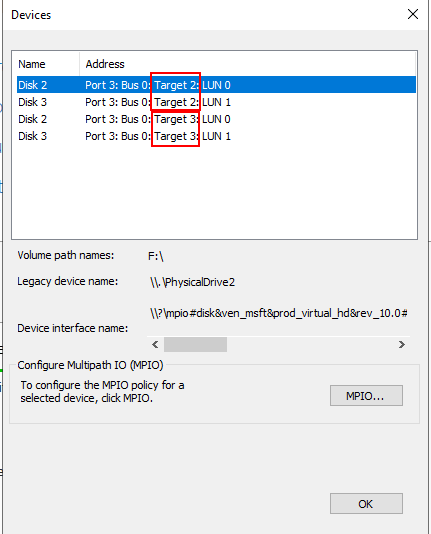
1. You can view the configuration from the iSCSI initiator



1. We can see the two devices (disks) connected

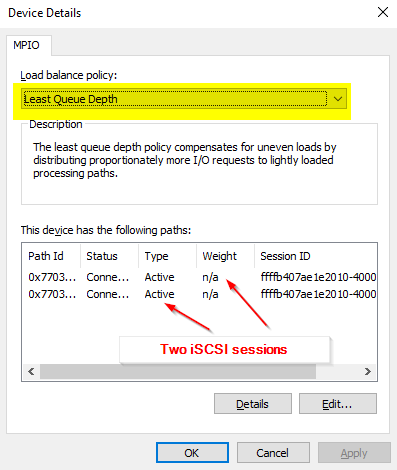


1. If you select one of the discovered targets and click ‘Devices’ we can see the multiple paths to the device.



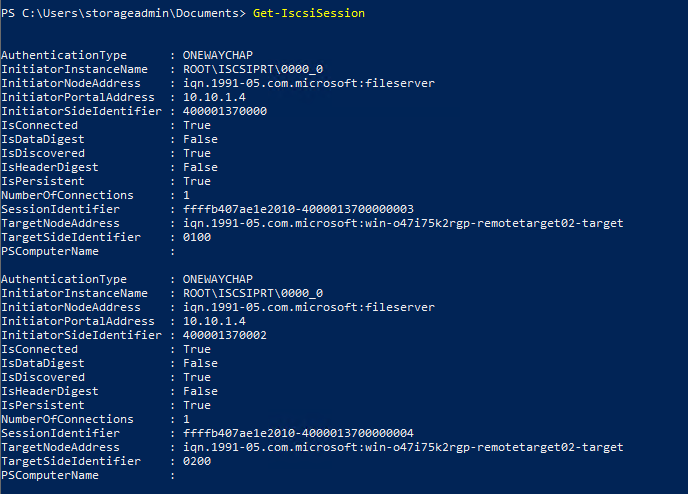
In this case we can see Disk 2 and Disk 3 have two target paths to the same device

1. If you click you a MPIO you can see your selected load balanced policy and the iSCSI sessions



1. There are also PowerShell commands to help review these sessions as well

Get-IscsiSession



Get-IscsiConnection

