Zerto

Deploy & Configure AWS Storage Gateway with Zerto LTR

VERSION 1
MAY 2020

Zerto

Table of Contents

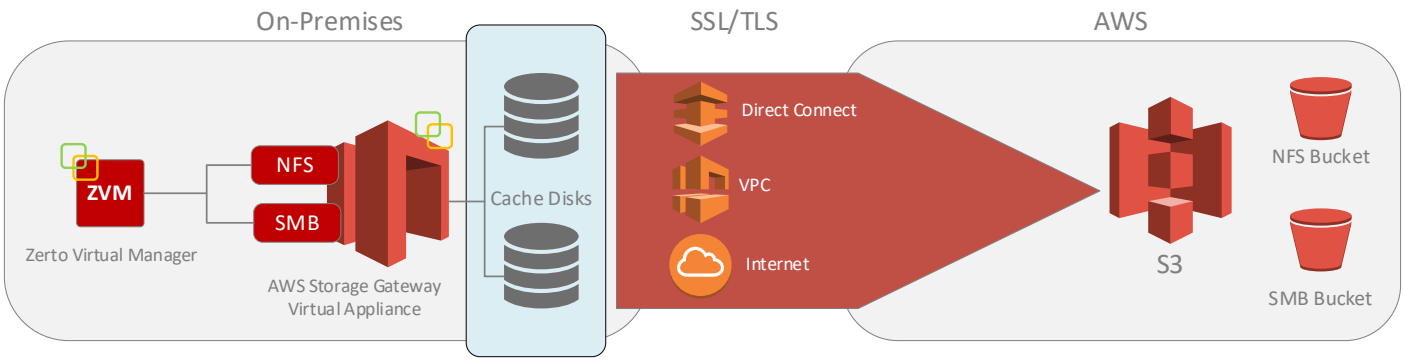
1. AWS Storage Gateway as a Zerto Long-Term Retention Repository

With the realease of Zerto 8.0 comes official support for the AWS Storage Gateway as an LTR repository. This allows for LTR on-premises as well as long-term storage of backup data to be offloaded to Amazon S3. Implementing the AWS Storage Gateway as a Zerto LTR repository is a quick and easy solution that allows long-term retention data to be stored in cost effective object storage in the public cloud.

Please refer to the AWS documentation to understand the costs associated with the use of the Storage Gateway service. For any issues with the AWS Storage Gateway, deployment, configuration, and service availability, you should contact AWS support for assistance.

1.1. How it Works

The AWS Storage Gateway (file gateway) works as an on-premises appliance (virtual or physical options available) that provides a local NFS/SMB repository backed by the Amazon Simple Storage Service (S3).



The file gateway integrates into Zerto to provide low-latency access to LTR data through transparent local caching, while managing data transfer to and from AWS. Transmission of the data takes place over an SSL/TLS connection. Additionally, the solution also optimizes and streams data in parallel and manages bandwidth consumption.

2. Pre-Requisites

This guide assumes you are familiar with deploying solutions within the AWS management console and in an on-premises vSphere environment.

This document only refers to deploying, configuring, and using the virtual AWS Storage Gateway in an on-premises vSphere Environment.

Documentation for the AWS Storage Gateway can be found at the following URL:
https://docs.aws.amazon.com/storagegateway/latest/userguide/WhatIsStorageGateway.html

2.1. AWS Pre-Requisites

- AWS account with access to create resources in the desired region.
- Designated AWS region where the AWS Storage Gateway will be deployed. You will also need to verify that the AWS Storage Gateway is available in your specified region:
  - Supported AWS Regions and list of AWS service endpoints for use with the Storage Gateway
  - Supported AWS Regions you can use with the AWS Storage Gateway Hardware Appliance
-

2.2. On-Premises Pre-Requisites

- In the on-premises vCenter, you will at a minimum need to have permissions to do the following:
  - Deploy OVF template
  - Administer/Edit VMs including adding virtual disks and assigning a network to a VM.
  - Ability to join a server to an Active Directory domain.
- In the on-premises Zerto Virtual Manager, you will need to be an administrator to perform the steps within this document.

3. Requirements for Virtual Appliance

You will need to meet the following requirements in your virtual environment in order to deploy the AWS Storage Gateway virtual appliance:

- 4 vCPUs
- 16 GB RAM
- 80 GB Disk space for installation of the OVF
- 150 GB (minimum) for on-prem cache disk (maximum supported size for cache disk is 16 TB)
- Optional: additional cache disk for the SMB mount, should you want to enable indexing for Zerto LTR – Indexing repository must be SMB.

For more information about AWS Storage Gateway on-premises requirements, including network and firewall requirements, refer to the URL below:

https://docs.aws.amazon.com/storagegateway/latest/userguide/Requirements.html

4. Sizing the Local Cache Disk

When deciding on the size of the local cache disk for the AWS Storage Gateway, it should be configured to match your use case appropriately. As this document specifically pertains to Zerto Long-Term Retention use of the file gateway, the following should be taken into consideration when planning your deployment.

4.1. Read Cache

The Storage Gateway can store data on the local cache disk to maximize read performance for frequently accessed data. If the read cache doesn't exist, then data is synchronously fetched from S3 each time it is requested, which will incur data transfer costs each time data is pulled back in from the cloud.

Given the nature of the Zerto short-term journal and it's capability to be retained on-premises for up to 30 days at a granular level for either VMs, application stacks, or files, this should always be the first resort to recover any data without the need to recover from the long-term retention repository (in this case Amazon S3).

In the instance data older than 30 days needs to be recovered from long-term retention, at that point is where Zerto will request the data from Amazon S3, and you must be aware of current data transfer rates and the time associated with recoverying that data back on-premises from the cloud. In the case of Zerto Long-Term Retention, the idea is that the data that is stored in S3 is primarly going to be used to meet compliance requirements on data retention, so recovering from LTR in S3 should be a rare, if ever, occurrence.

4.2. Write Cache

The AWS Storage Gateway uses a write-back mechanism to first persist the data being written by Zerto Long-Term Retention to the local cache disk. From there the data is asynchronously uploaded to Amazon S3. If the write cache is sized too small, this could result in failed LTR jobs, as there is no free cache to store the data locally pending upload to S3.

Starting in Zerto 8.0, Long-Term Retention data is written with zero elimination, which means that Zerto will no longer write empty blocks of data (zeroed data) to the recovery points. A retention set will only include data actually written to volumes on VMs. The result of zero elimination is that now retention set sizes are smaller, and less space on the repository is used, therefore, when uploading to Amazon S3, less data will be sent.

When sizing the write cache for the storage gateway, the size of the cache disk will need to be at least the same size as the working set of data (as recommended by AWS documentation), which translates to just the written blocks of the VM volumes, mentioned above as related to zero elimination.

Example:

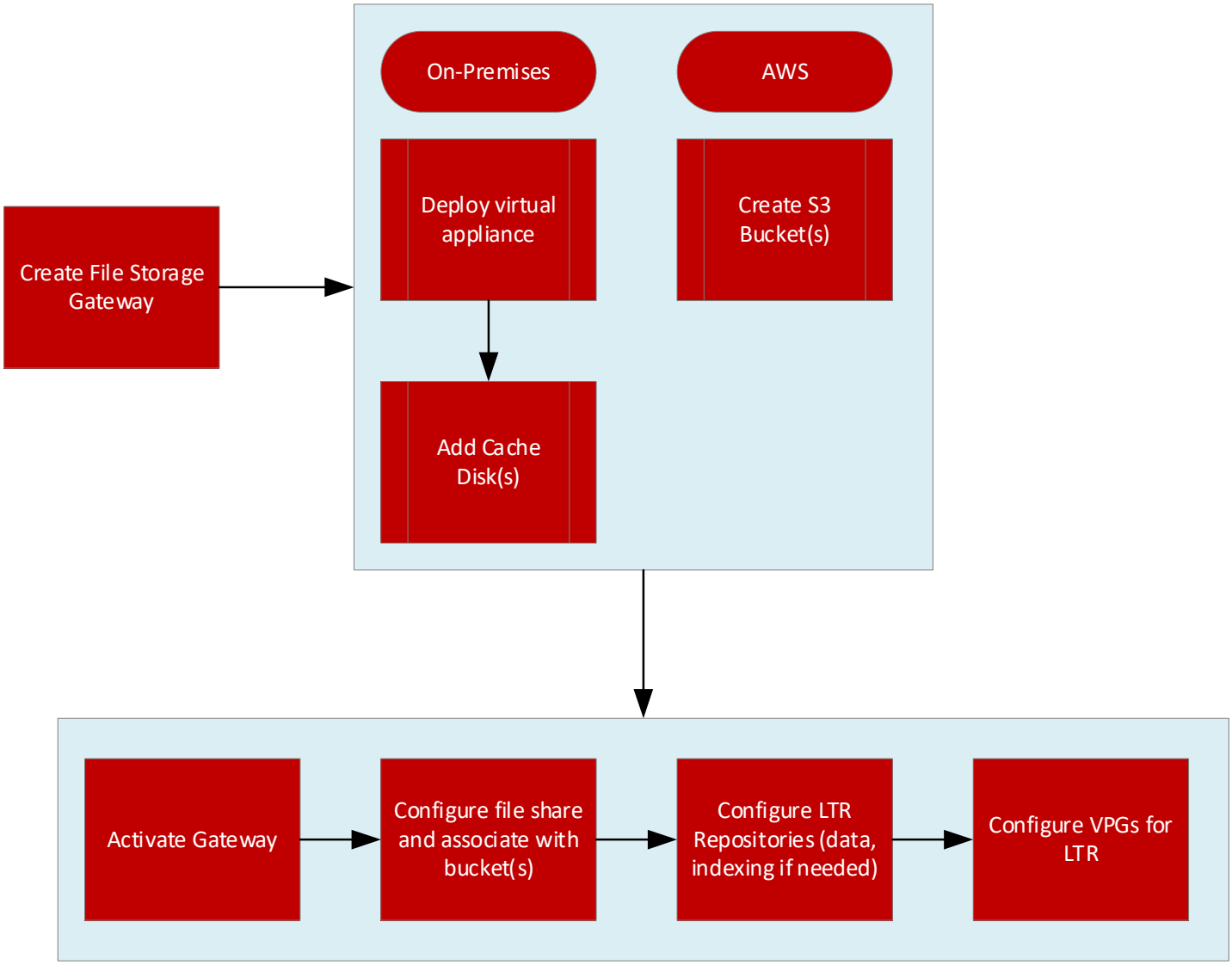
If an application made up of 2 VMs has an aggregated total provisioned size of 300 GB for all volumes, but only 150 GB is actually writen data, then given the AWS recommendation and zero elimination, the cache disk should be 150 GB in size at a minimum (see table).

VM	Provisioned (GB)	Used (GB)	Cache Size (GB)
Database VM	200 GB	100 GB	**100 GB**
WebApp VM	100 GB	50 GB	**50 GB**

Once the total working data set for the target workloads has been determined, then the creation of the local cache disk can take place. It is also recommended to add at least 10-15% additional to the total for overhead to avoid filling the cache disk. Refer to the minimum and maximum cache disk sizes in section 3 above when planning the layout of the local cache disks during the AWS Storage Gateway deployment.

5. Workflow

At a high-level, the steps in this document will walk you through the following workflow:

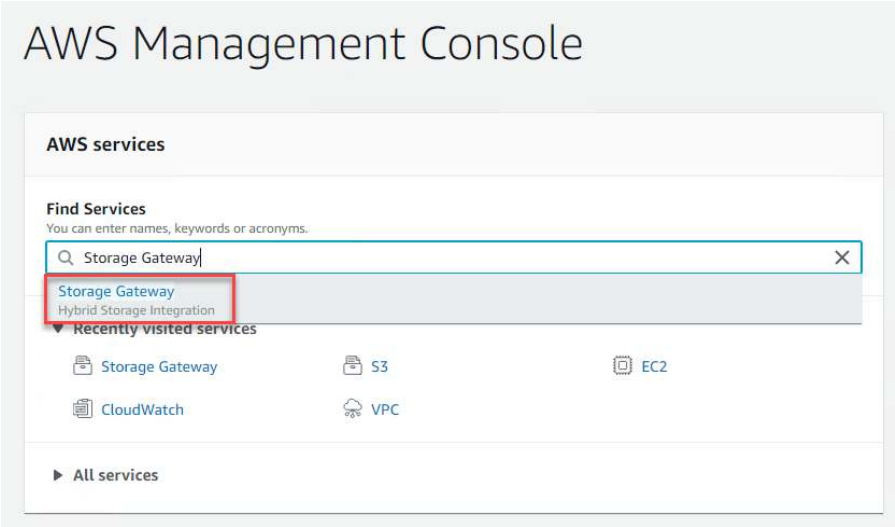


6. Deployment

6.1. Download and Deploy the Virtual Appliance

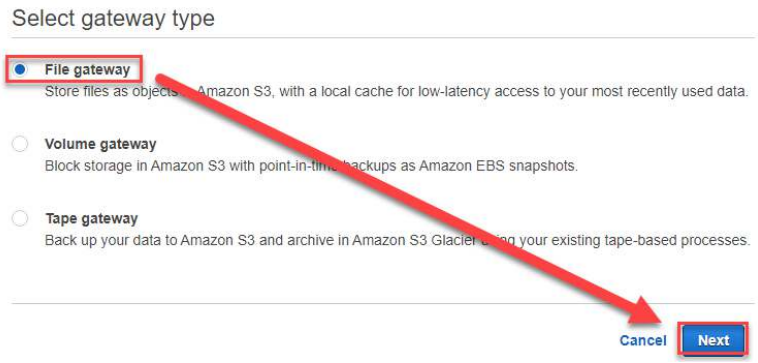
6.1.1. Download the AWS Storage Gateway

1. Log onto AWS the AWS Management Console
2. Under AWS Services, search for **Storage Gateway** and select it to be taken to the AWS Storage Gateway service.

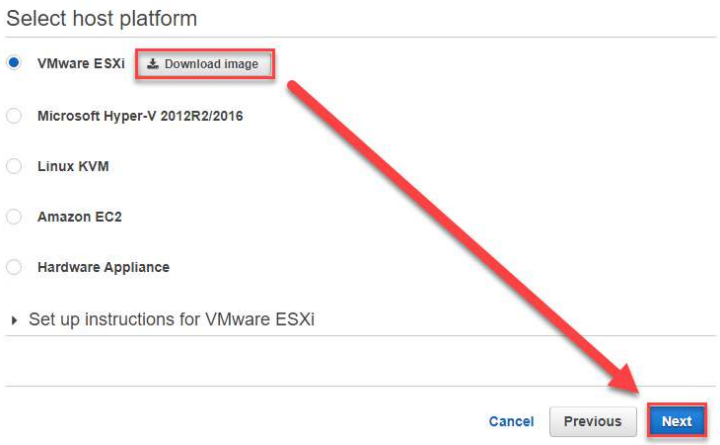


3. If it's the first time you're visiting the AWS Storage Gateway service, you'll get a page similar to the following. Click **Get Started** to begin. If you've deployed a gateway in the past, you should be taken to the service page where you can just click **Create Gateway**.

4. Select **File Gateway** as the gateway type and click **Next**.



5. Select **VMware ESXi** and click the button to download the image. **DO NOT** click next yet, we'll come back to that. First, we'll deploy the on-premises image.



6.1.2. Deploy the AWS Storage Gateway Virtual Appliance in vCenter

6. In your vCenter web client, right click on the cluster or folder where you want to deploy the downloaded virtual appliance OVF, and click **Deploy OVF**.
7. In the Deploy OVF Template wizard, select **Local File**, then browse and select the **aws-storage-gateway-latest.ova** file that you've downloaded, and click **Next**.
8. Enter a **virtual machine name** for the appliance**, select a location** to deploy it, and click **Next**.
9. Select a **Host**, **Cluster**, **or Resource Pool** to deploy the virtual appliance to, and click **Next** after compatibility checks complete.
10. On the **Review Details** screen, click **Next**.
11. Select a datastore to deploy the appliance to, **set** the **virtual disk format** to **Thick Provision** (either lazy or eager will work)**,** and click **Next**.
12. Choose the **destination network** (port group) to put the appliance on, then click **Next**.
13. Click **Finish** to deploy the virtual appliance.

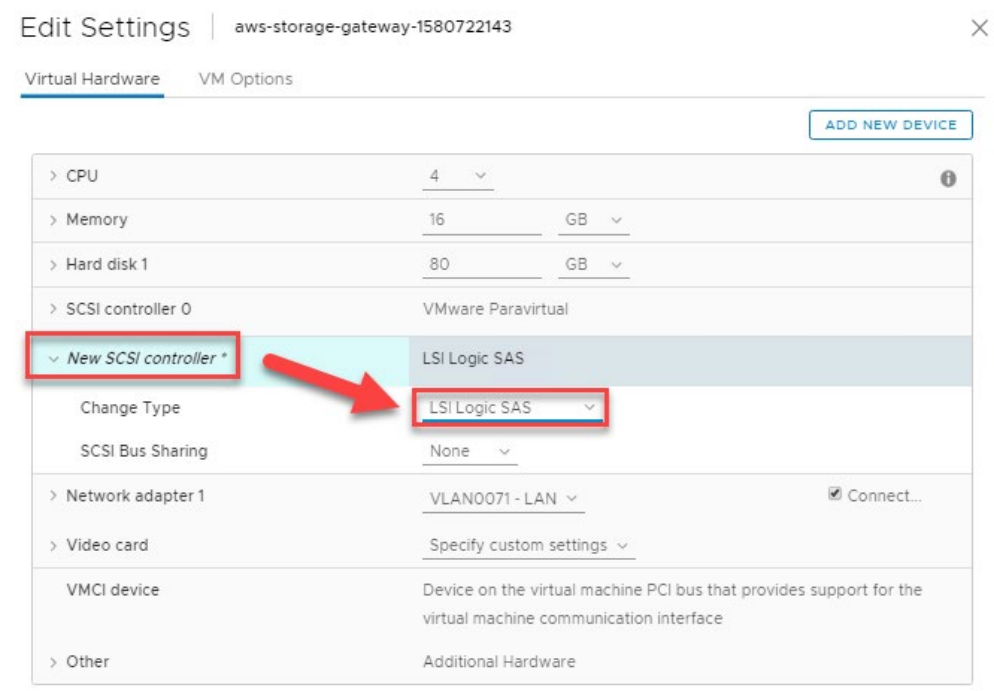
6.1.3. Add Cache Disk(s) to the AWS Storage Gateway Virtual Appliance

Now we will be adding the cache disk(s) to the virtual appliance that was just deployed. It's important to note the following:

If you've read the AWS Storage Gateway deployment document (written by AWS), they mention the disks as "paravirtualized" disks. There is a **VMware Paravirtualized SCSI** adapter type in vSphere. The mention of "paravirtualized" disk **does not** refer to the **VMware Paravirtualized SCSI** adapter type. If you use that adapter type for the cache disk(s), the gateway will not be able to see them.

For the above reason, you will need to choose **LSI Logic SAS** as the virtual SCSI adapter type when completing the next set of steps.

14. Right-click on the AWS Storage Gateway VM in vCenter, and click **Edit Settings**.
15. Click **Add New Device**, select **SCSI Controller**.
16. Expand **New SCSI controller*** and change the type to **LSI Logic SAS**.



17. Click **Add New Device** and select **Hard Disk**. For the size of the disk, at a minimum, set it to **150 GB**
18. If you are also going to be using the Zerto LTR Indexing feature, you will need to add another disk. Do this now as well by repeating the steps above. If you want to separate the virtual disks onto separate virtual SCSI adapters, just be sure to select **LSI Logic SAS** for the second adapter. Once you have added the necessary disk(s), click **OK** to save the VM settings.
19. Once you've completed the these steps, you can power on the AWS Storage Gateway appliance.

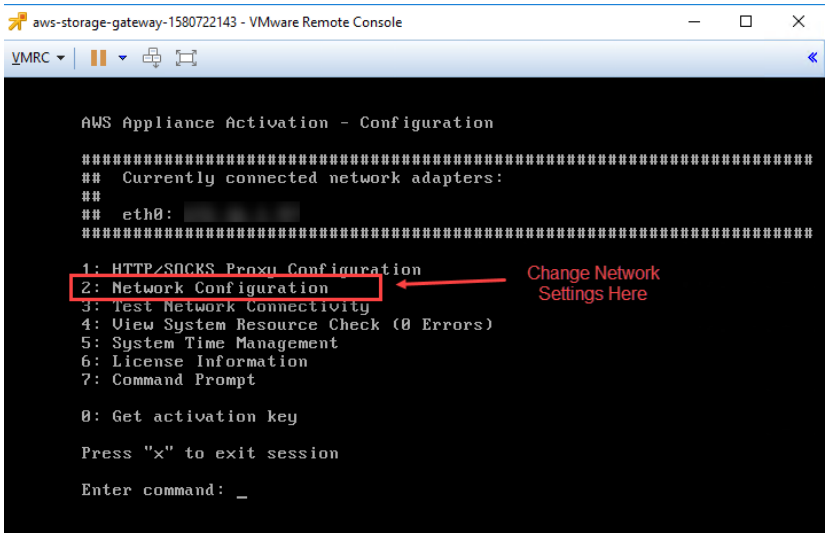
6.1.4. Provide Static IP and Validate Connectivity of the AWS Storage Gateway Appliance

By default, the AWS Storage Gateway virtual appliance will be set to DHCP. If you choose to use DHCP be sure to set a DHCP reservation and skip the static IP steps below and go straight to validation the connectivity of the appliance. If you need to apply a static IP address, follow the steps below.

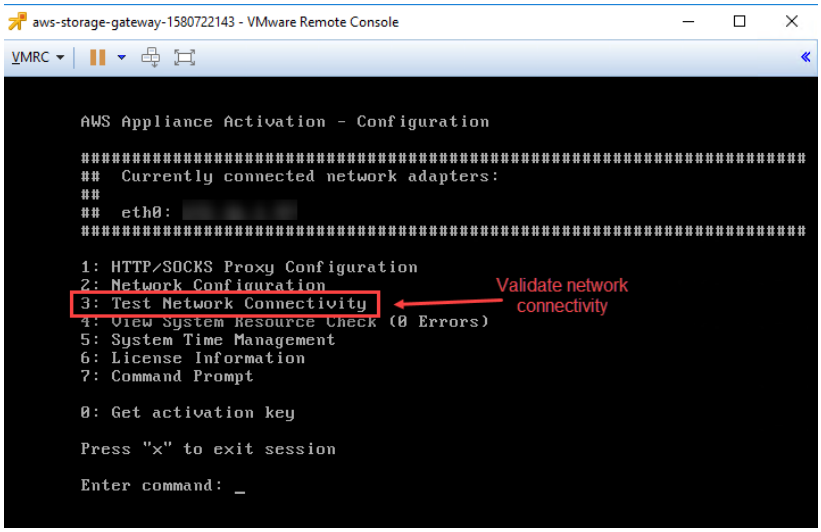
20. In the vSphere Web Client, open the remote console for the AWS Storage Gateway virtual appliance.
21. Log on using the default credentials for the **admin** user.

    Hint: See https://docs.aws.amazon.com/storagegateway/latest/userguide/manage-on-premises-common.html#LocalConsole-login-common for default credentials.

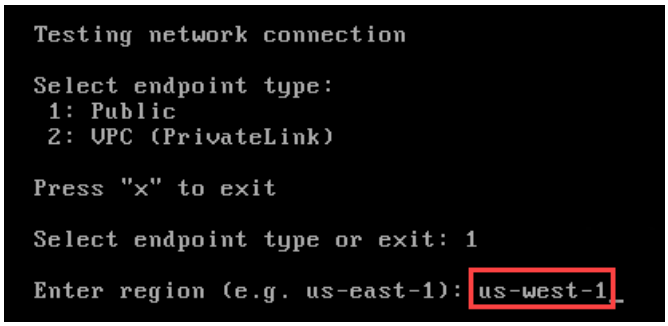
22. Once logged in, press **2** to configure the network settings



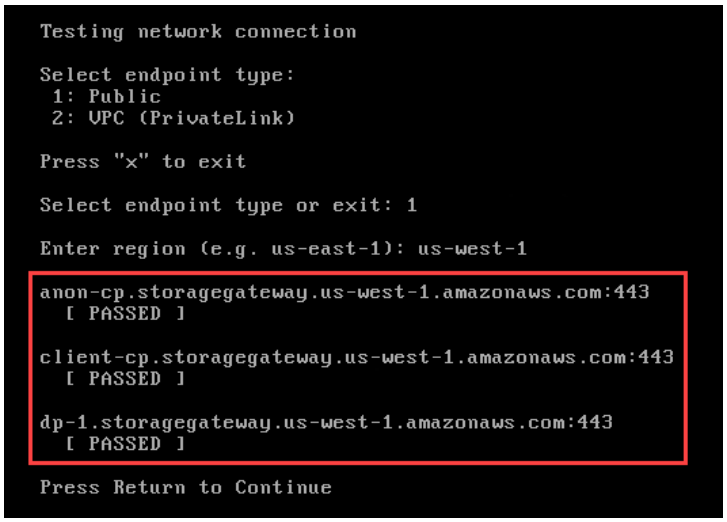
23. After you've configured the network settings, test connectivity to the AWS Storage Gateway public endpoints. Select option **3** at the selection screen, and press **enter**.



24. In the Testing network connection screen, select **1** for Public Endpoint and press **enter**.
25. Enter your region that you are configuring for the AWS Storage Gateway. In my example, I am deploying it in the US-West-1 region, so I will type **us-west-1** and press **enter**.



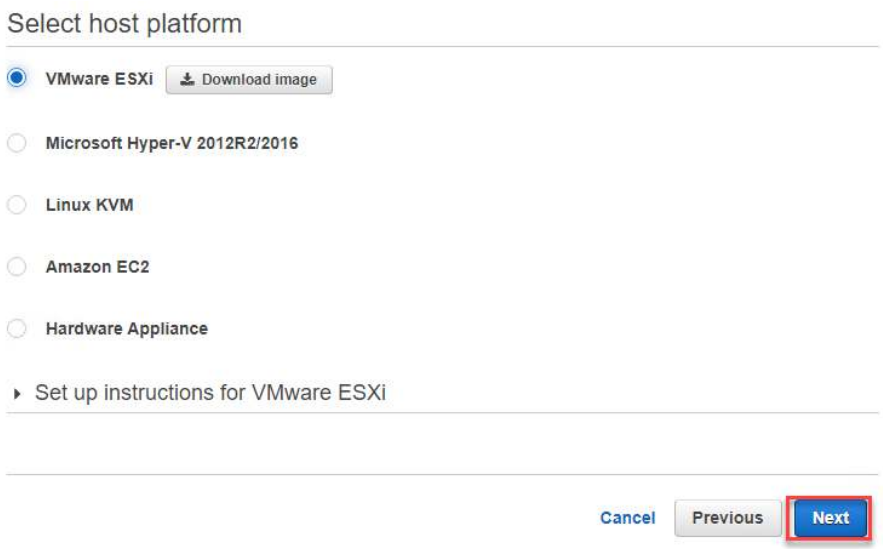
26. If the connection is successful, you will see the following screen. Press return to continue and exit the console window.



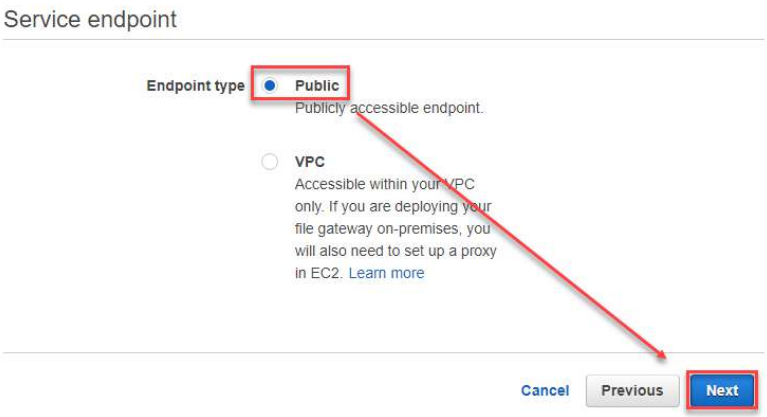
6.1.5. Complete the AWS Storage Gateway Deployment in the AWS Management Console

Once you've completed the steps above to configure the VM and have validated network settings and connectivity, you can continue with the deployment from the AWS Management Console.

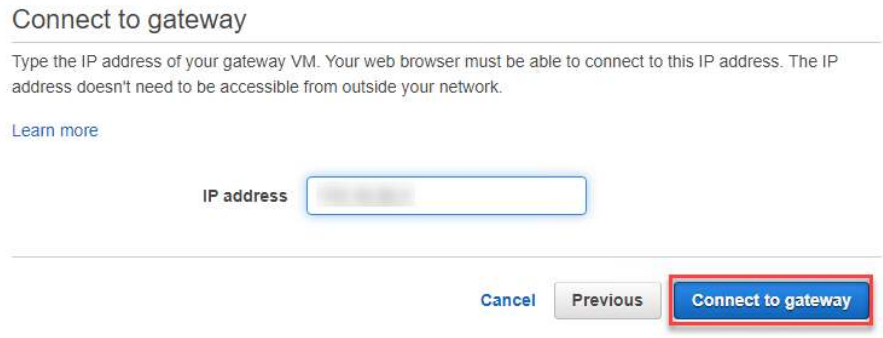
27. When you return to the AWS Management Console, you should still be at the screen where you downloaded the OVF file. From here, click **Next**.



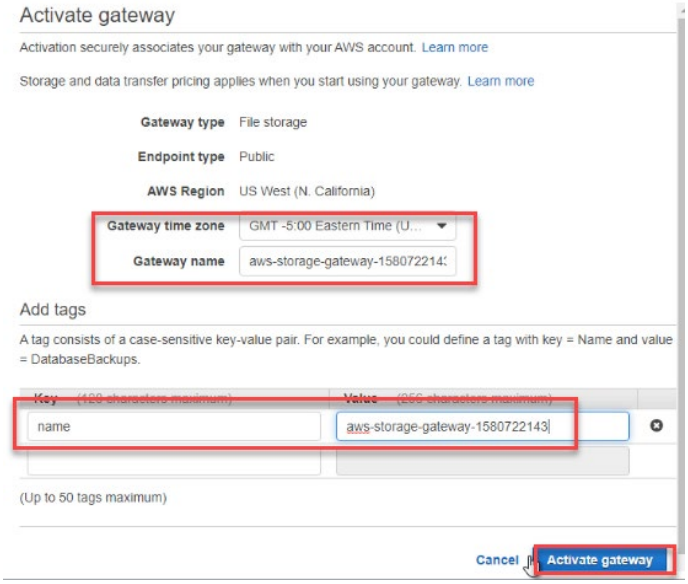
28. On the Service Endpoint selection screen, select **Public**, then click **Next**.



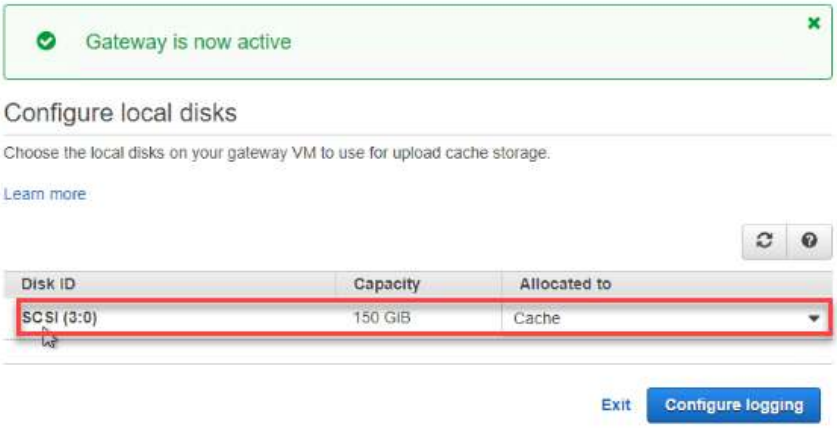
29. Click **Connect to gateway** to now link the AWS Storage Gateway service with the on-premises virtual appliance.



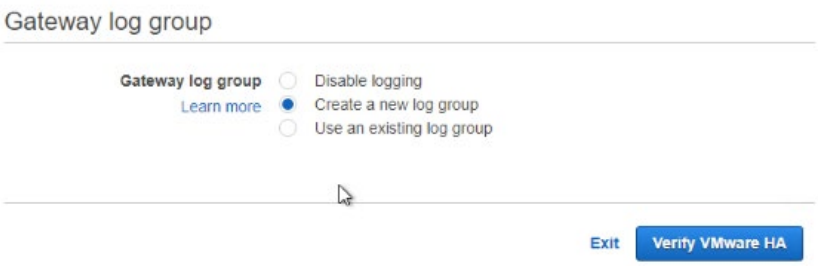
30. On the Activate Gateway screen, select the **time zone**, provide a **Gateway name**, and add tags if required (I usually add a key of “name” and a value of “theActualVMName”). Once you’ve entered this information, click **Activate Gateway**.



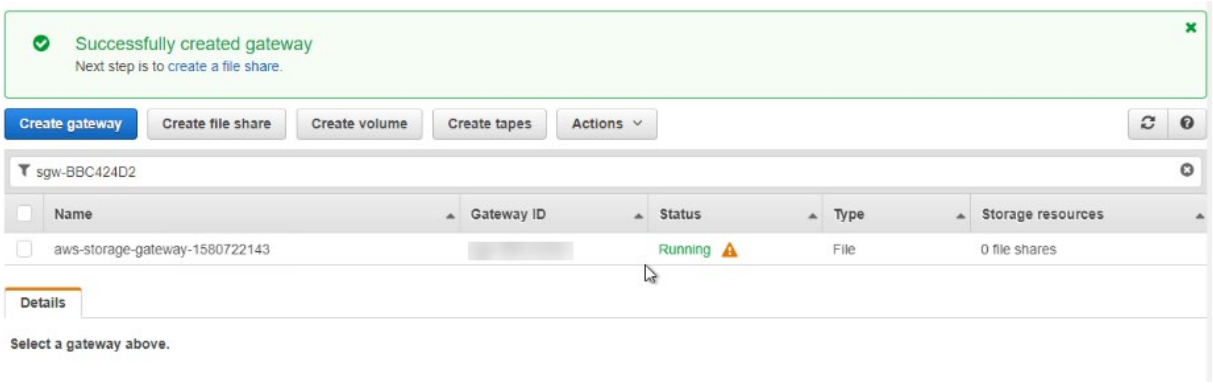
31. If you see the following screen, the gateway is active, and the disk is visible to the service. Click **Configure logging**.



32. For logging, you can choose whether or not to **disable logging**, or whether or not you want to **create a new log group** or **use an existing one**. Once you've made your selection, you now have two choices:
    a. **Exit** – Use this option if you either do not want to test VMware HA for the appliance, or if you do not have VMware HA enabled.
    b. **Verify VMware HA** – Use this option if you've deployed the AWS Storage Gateway virtual appliance onto a cluster that is configured for VMware HA.



33. Once the previous step has been completed, you will get a screen similar to the following, and you can now configure the file share(s).

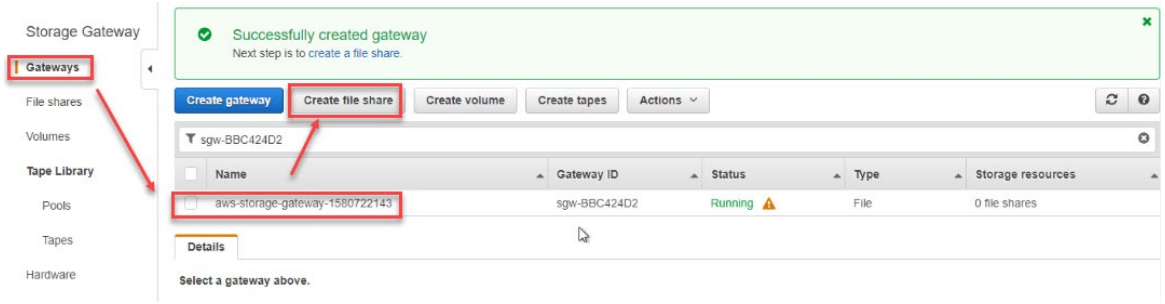


7. Create File Share for Use with the AWS Storage Gateway

Now that you've gotten through the deployment and configuration of the AWS Storage Gateway virtual appliance, you can add an NFS share to the gateway to enable use as a Zerto Long-Term Retention repository.

7.1. Configure a File Share for NFS

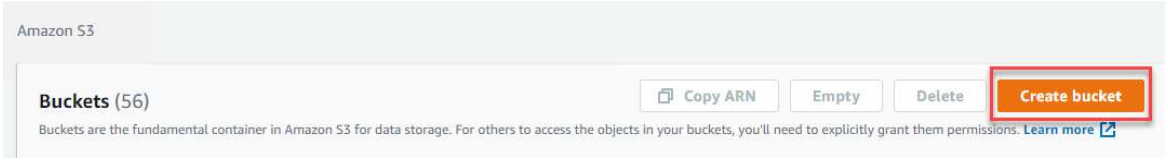
1. Go to Storage Gateway, click on **Gateways**, and select the AWS Storage Gateway you have deployed.
2. Click **Create file share**.



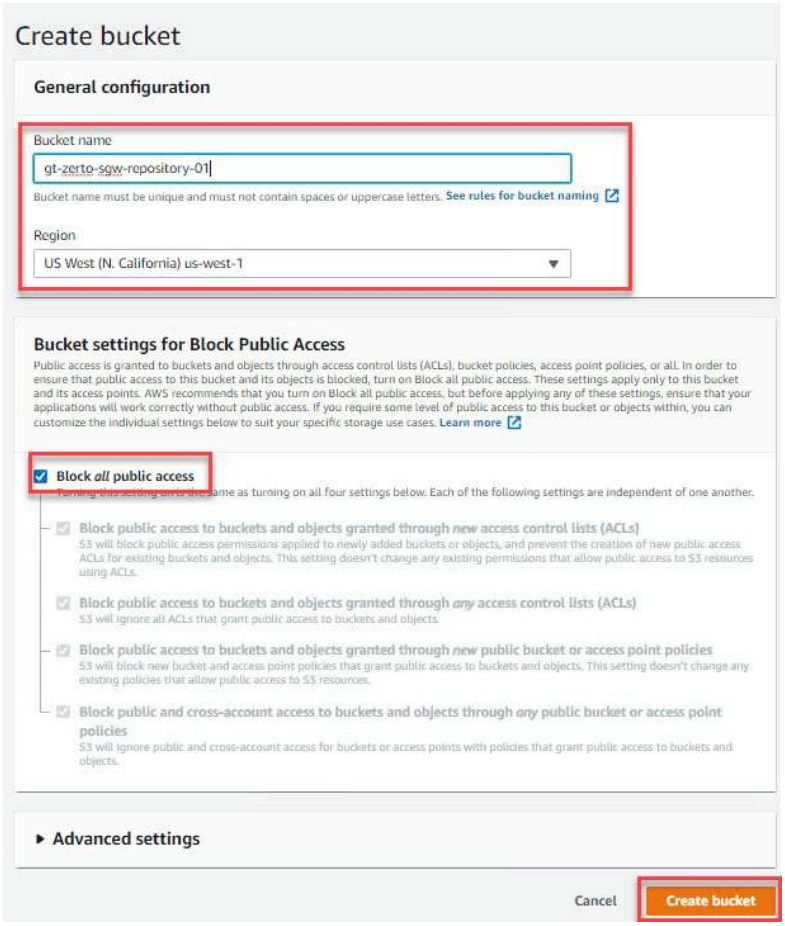
3. **If you have already created an S3 bucket** for use with the AWS Storage Gateway, skip to section **6.3.**

7.2. Create an AWS S3 Bucket to Use for the File Share

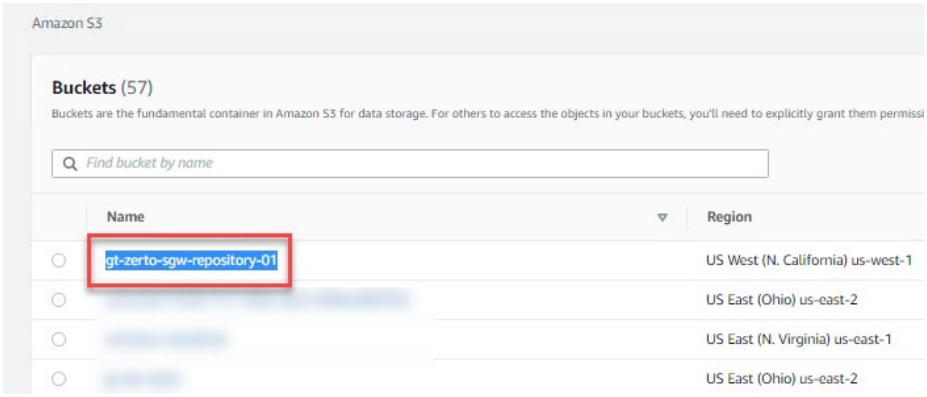
4. In the AWS Management Console, go to **S3**.
5. In the Buckets screen, click **Create bucket**.



6. Provide a **bucket name**, select the same **region** that you've deployed the AWS Storage Gateway to, and click **Create bucket**. (Leave the default to **Block all public access**)



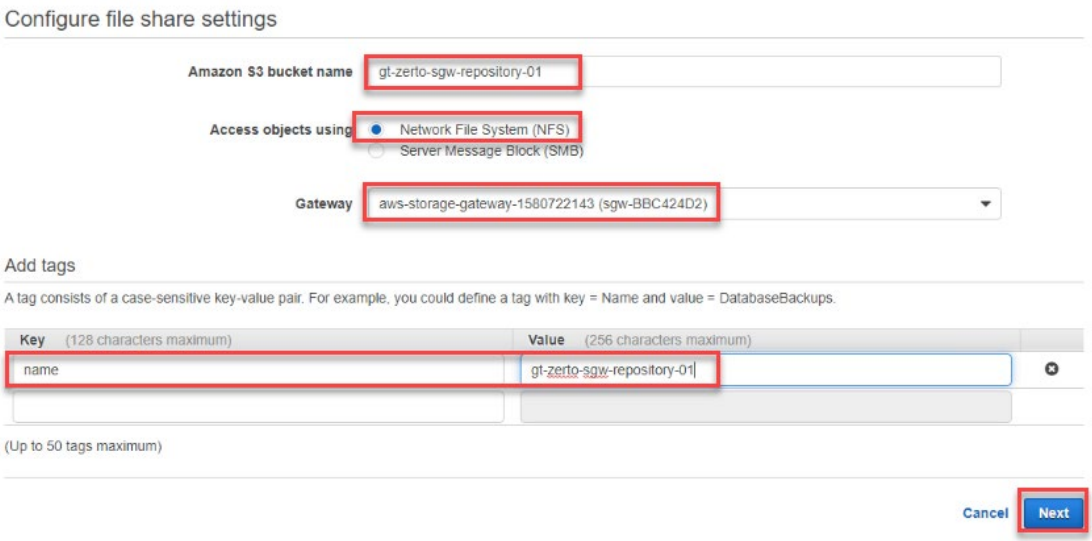
7. You will be returned to the **Buckets** page where you'll see a listing of all buckets. Select the **name** of your bucket, and copy it. You'll need it for the next step.



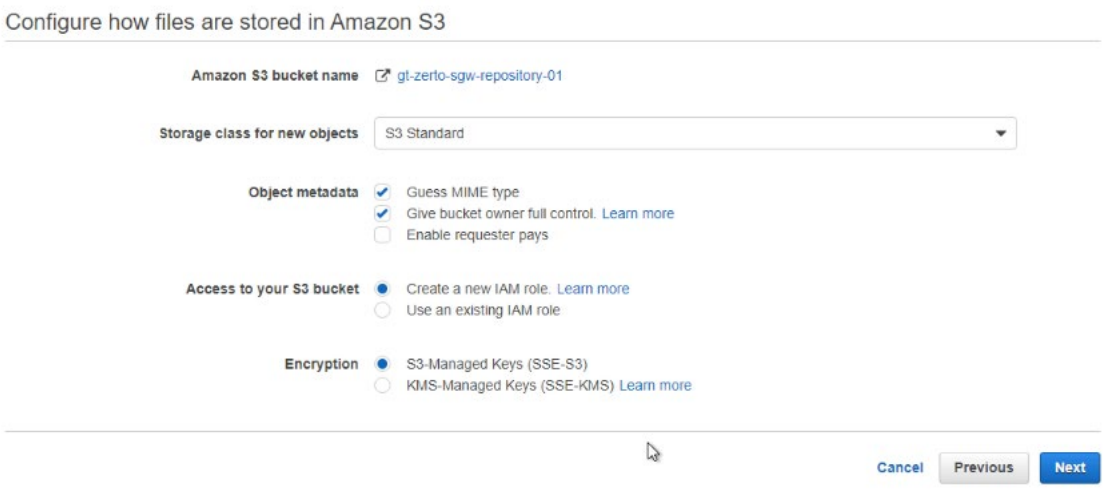
7.3. Configure the AWS Storage Gateway File Share

This is a continuation of the previous section if you needed to create an S3 bucket for use with the file share. If you skipped to here from section 6.1, you're going to need the name of the S3 bucket you're using for the NFS share.

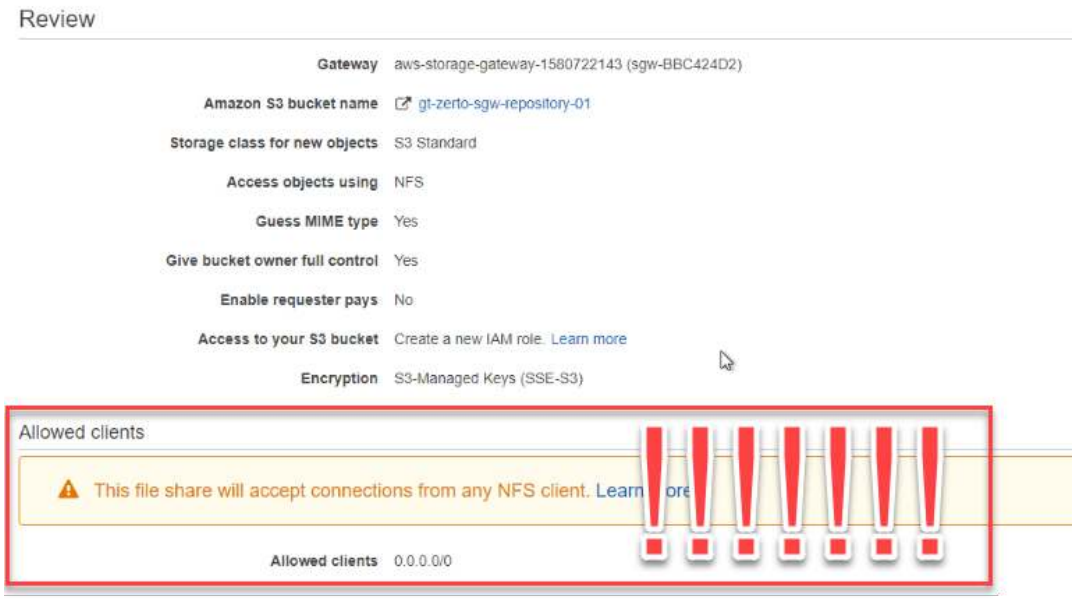
8. Enter the name of the S3 bucket you have configured for use with the AWS Storage Gateway.
9. Select **Network File System (NFS)**
10. Select the **AWS Storage Gateway** that you are using with this file share.
11. Add a tag if required, then click **Next**.



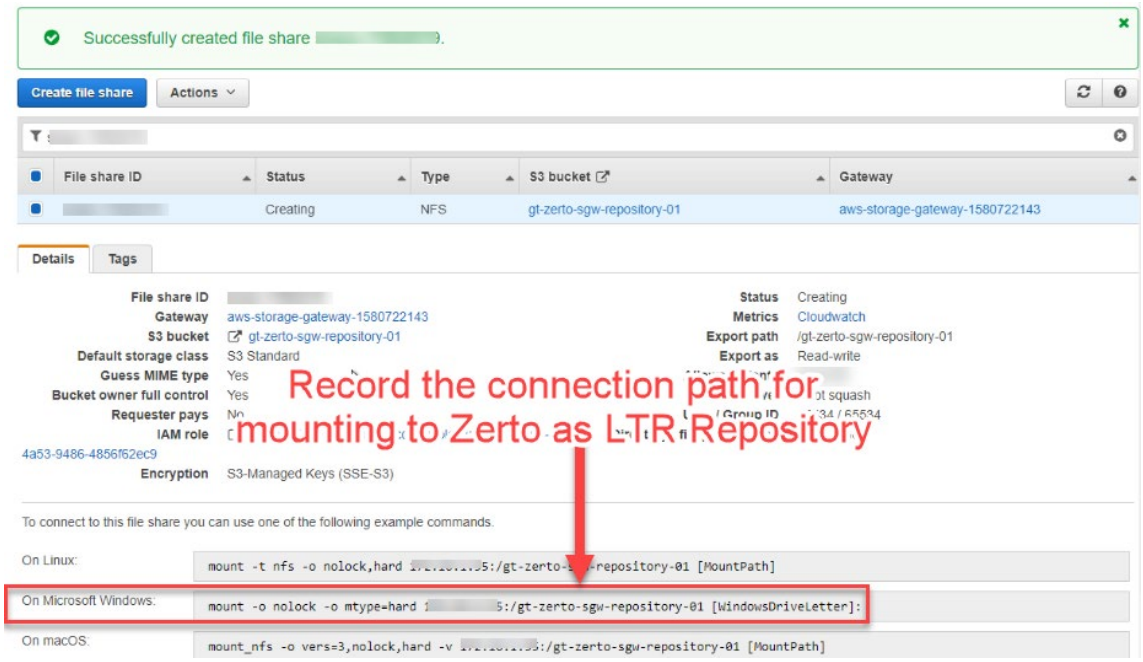
12. Configure how the files are stored in Amazon S3. I left the defaults for this, but if you require to make any changes, do that now, then click **Next**.



13. Review your settings, and be sure to pay special attention to the **Allowed Clients** section, and make the necessary changes to allow only specific clients with access to this NFS share. From a Zerto LTR perspective, you will need to allow in either IP Address or CIDR notation:
    a. **Zerto Virtual Manager (ZVM)** in the site where the AWS Storage Gateway is deployed
    b. **Each Virtual Replication Appliance (VRA)** in the site where the AWS Storage Gateway is deployed



14. When done, click **Create File Share**.
15. Once the file share has been created, select it to view it's properties, and be sure to save the connection information for Windows, as we will be using that to mount the repository to the Zerto Virtual Manager for use as a Long-Term Retention (LTR) repository.

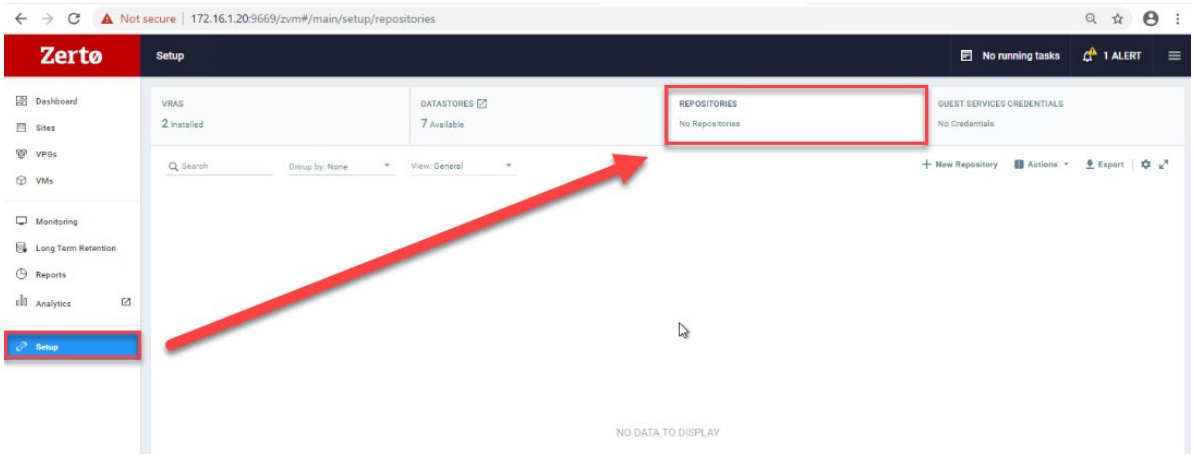


8. Mount the AWS Storage Gateway NFS File Share to Zerto

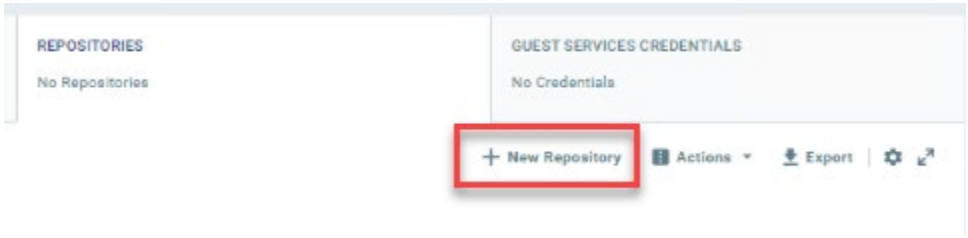
Once the Gateway has finished creating the NFS share we can add it as a Long Term Retention repository within Zerto.

8.1. Adding an NFS Mountpoint in Zerto for use as a Repository for LTR

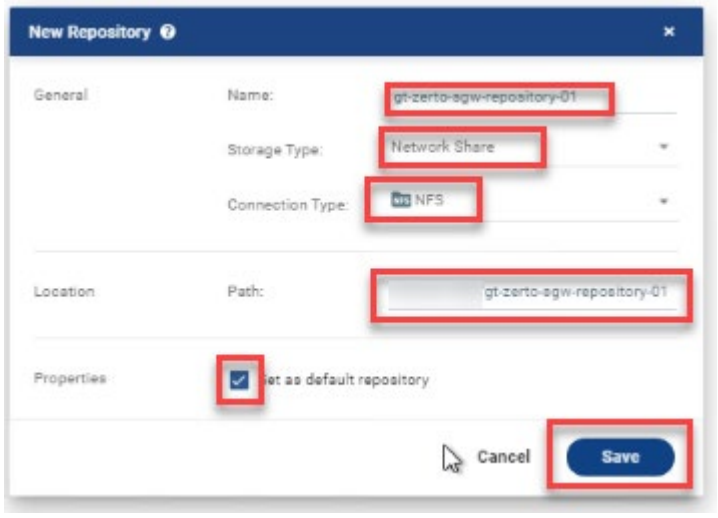
1. Log onto the Zerto Virtual Manager UI.
2. Click on **Setup**, on the left navigation bar, then click on the **Repositories** tab on the top right of the window.



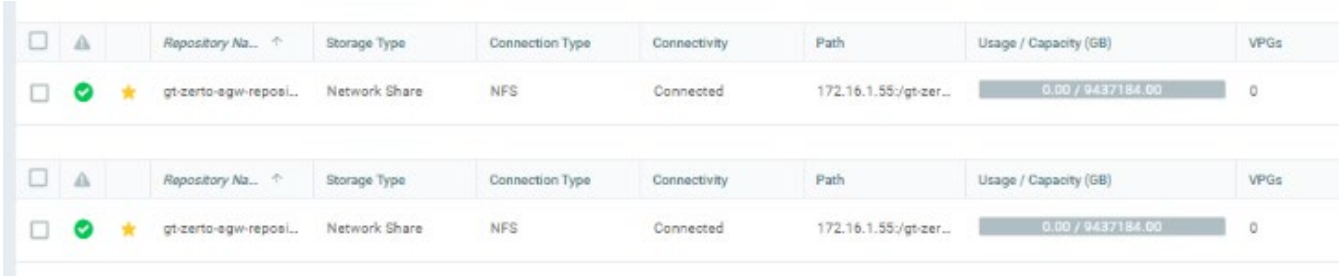
3. Click **+ New Repository**



4. Enter a **name** for the repository.
5. Storage Type: **Network Share**
6. Connection Type: **NFS**
7. Path: **enter the path you copied from the file share settings in AWS**
8. Optional: **Set as default repository**
9. Click **Save**.



10. If the mount was successful, you will now have a repository listed on the screen with a green checkmark. If it was not successful check your access permissions which were set earlier to allow all VRAs and ZVM to connect to the share.



Now you are ready to start configuring Virtual Protection Groups with Long-Term Retention!

9. Setting up the AWS Storage Gateway SMB Share for LTR Indexing

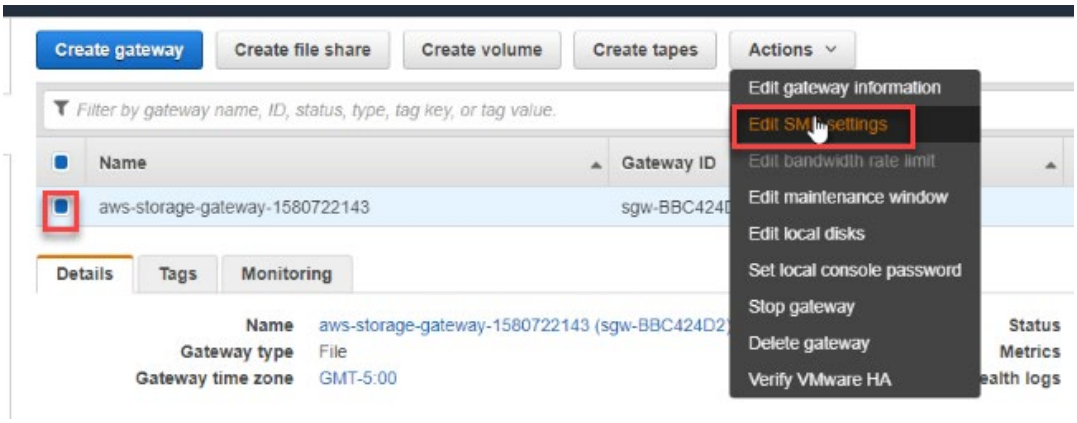
In this section, we will configure the SMB share that will be used as a Zerto LTR Repository for Indexing. Currently, the only supported method for indexing is via SMB share, so although the setup is similar to the NFS file share, there are additional considerations when setting up an SMB share with the AWS Storage Gateway.

If you need to configure a virtual disk for the AWS Storage Gateway to use for the SMB Cache, do so now, , as the following steps assume that disk has already been created. See section **5.1.3** in this document for how to add another disk to the AWS Storage Gateway.

9.1. Configure SMB Settings for the AWS Storage Gateway

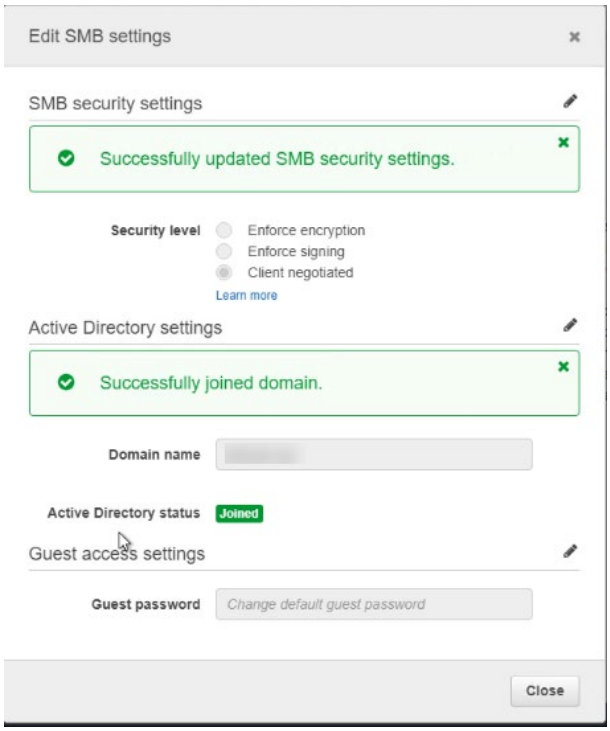
Before we configure the SMB share on the Storage Gateway, we will join the virtual appliance to the on-premises Active Directory domain.

1. In the AWS Management Console, navigate to your Storage Gateway.
2. Select the Storage Gateway, and go to **Actions → Edit SMB Settings**.



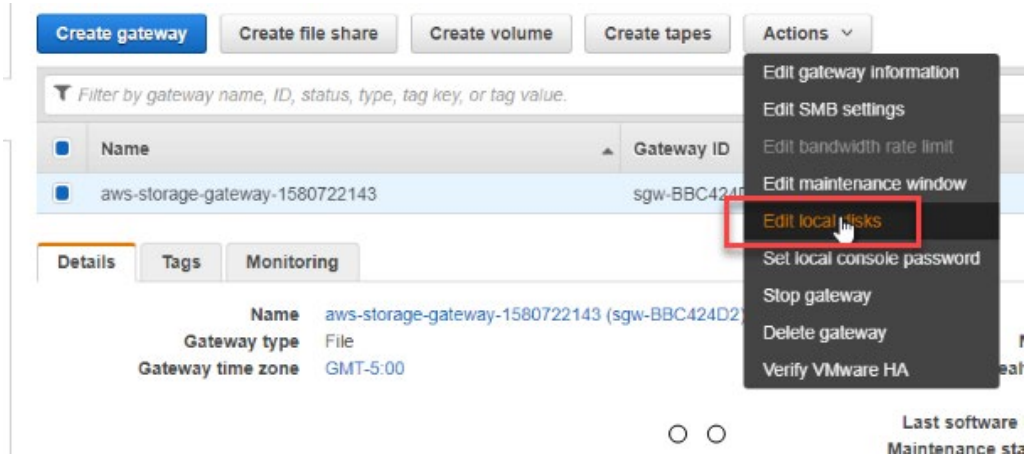
3. In the Edit SMB settings dialog box, set (be sure to click **Save** if editing multiple sections of this dialog box):
    a. Security level: **should adhere to your on-premises AD domain requirements**
    b. Click the edit (pencil) icon to the right of **Active Directory settings**, and enter the following:
        i. Domain name: **enter the domain name (FQDN – i.e. mylab.xyz)**
        ii. Domain user: **enter just the user name for the domain (needs to be able to join the domain)**

iii. Domain password: **the password for the user account used above**
iv. Organizational unit: **optional**
v. Domain controller(s): **optional** (If your DNS servers are also AD servers, this field is optional)
vi. Click **Save**, and the join process will immediately be attempted.
vii. Click **close** as soon as you've received validation that the appliance has joined the domain.

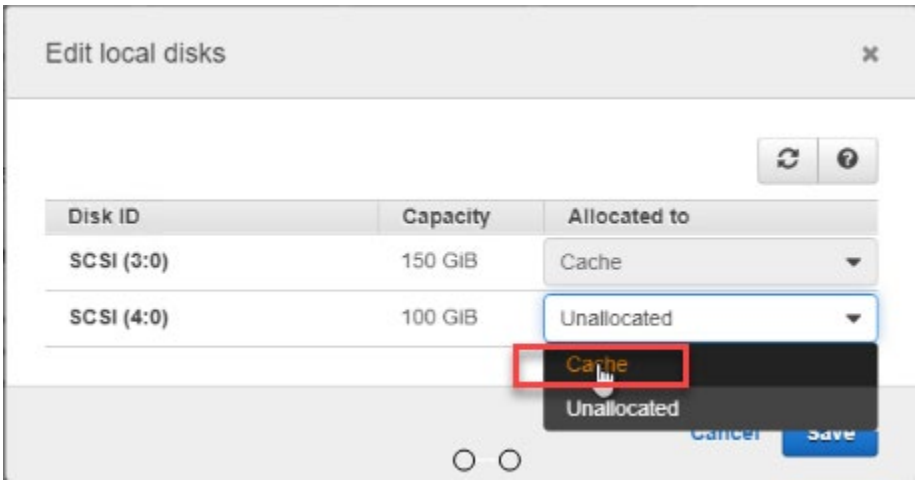


9.2. Verify Additional Disk is set to Cache Before Creating SMB Share

4. Go to **Actions → Edit local disks**



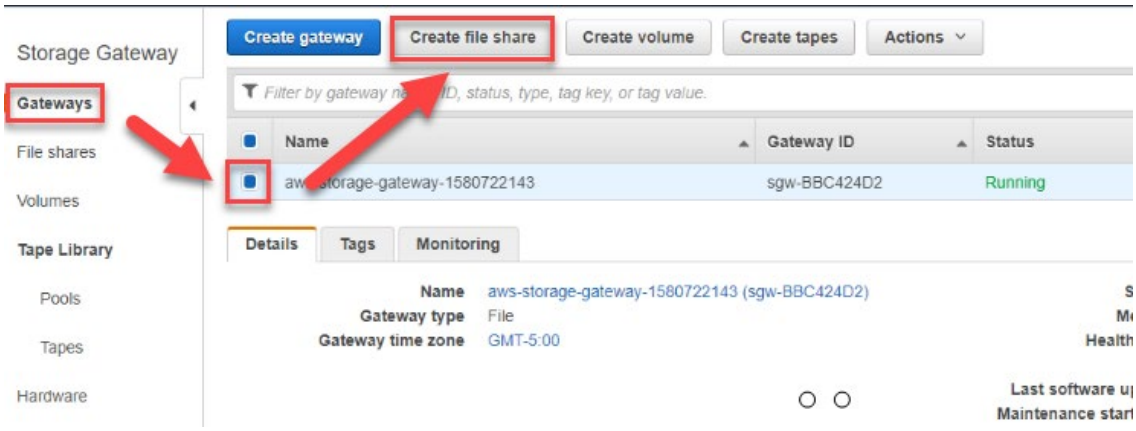
5. Verify that the disk you want to use for the SMB share has been allocated as a cache disk, and click **Save**



6. If you haven't created the S3 bucket yet, for the SMB file share, refer to section **6.2** of this document for the steps to create an S3 bucket. Once you have created one, copy the name, because you will need it when creating the SMB file share.

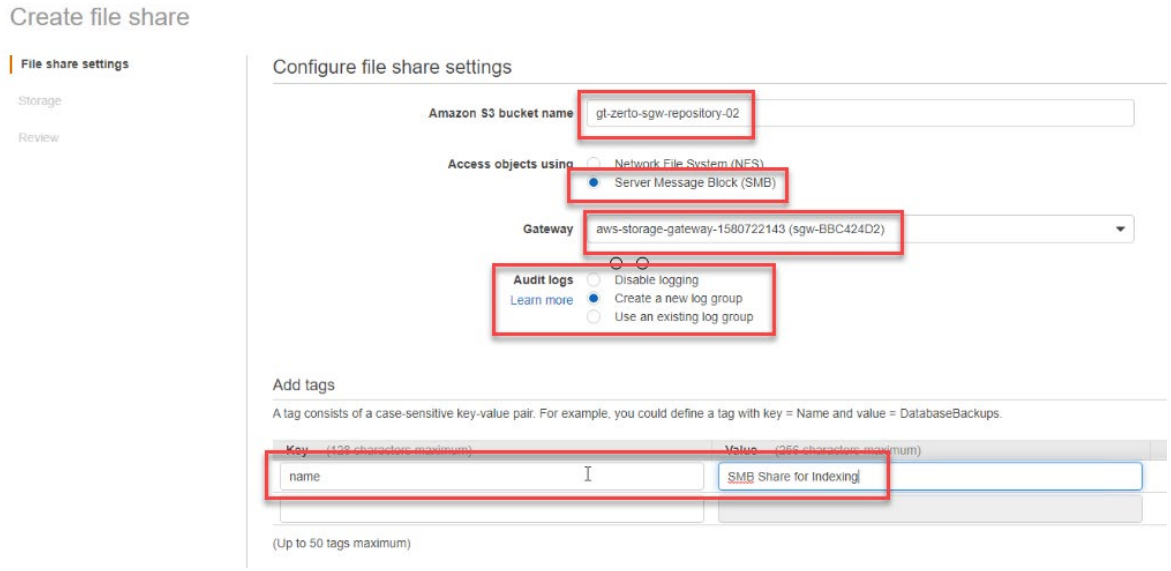
9.3. Create the SMB Share on the AWS Storage Gateway

7. Back on the Gateways page, select your AWS Storage Gateway, and click on **Create file share**.

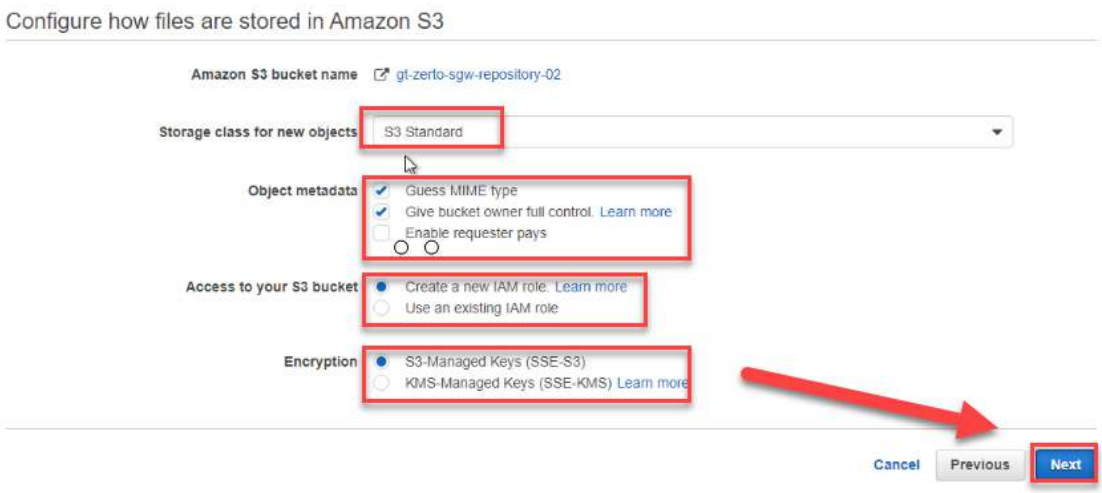


8. In the Configure file share settings:
    a. Amazon S3 bucket name: The **name of the S3 bucket** you've set up for the SMB share
    b. Access objects using: Select **Server Message Block (SMB)**
    c. Gateway: **make sure the correct AWS Storage Gateway has been selected**
    d. Audit logs: **Disable Logging, Create a new log group, or Use an existing log group** (you decide)
    e. Add tags: Key: **name** Value: **add a friendly name for this share**

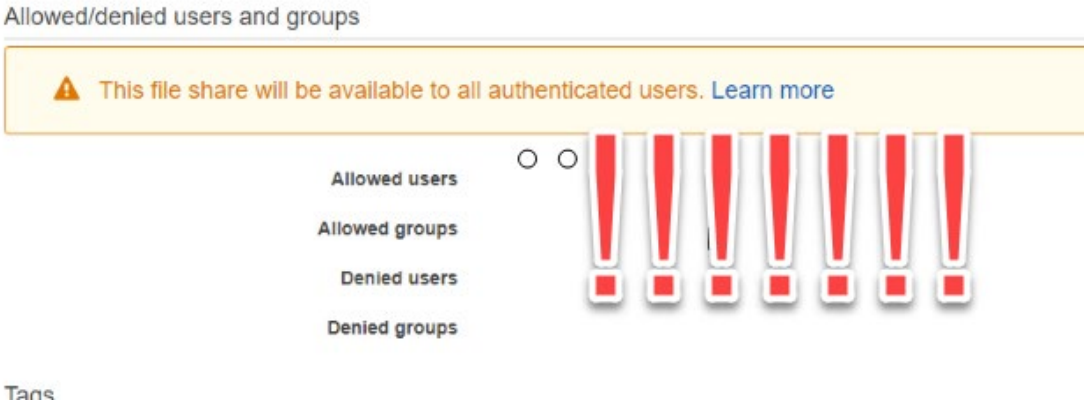
9. Click **Next**.



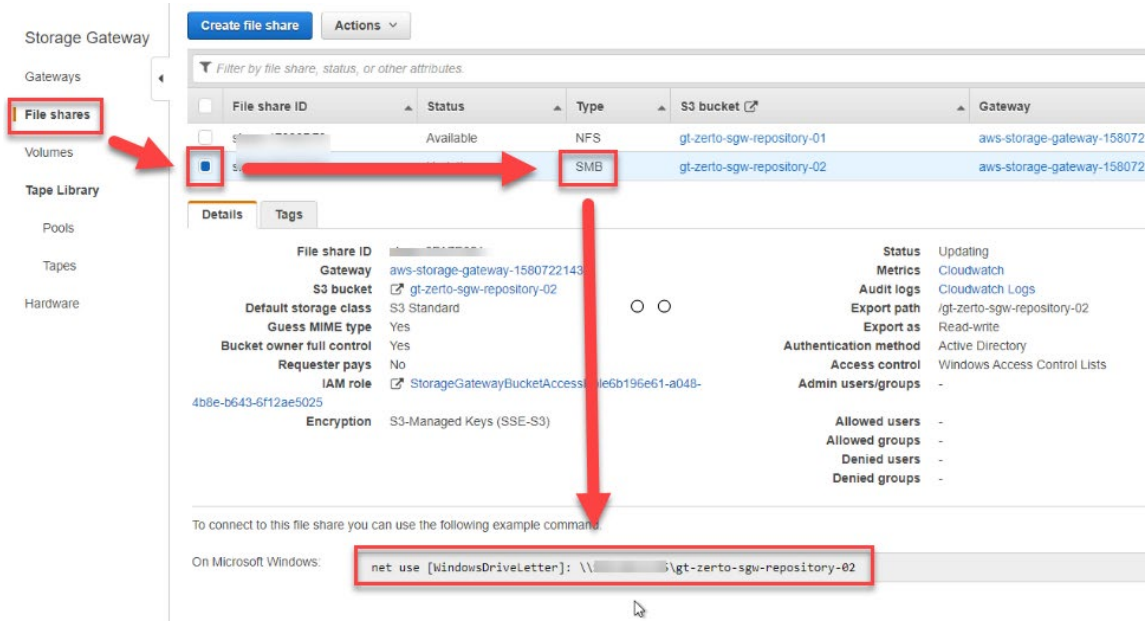
10. In the Configure how files are stored in Amazon S3 section, make the necessary changes required, then click **Next**. In my example, I left all the defaults:
    a. Storage class for new objects: **S3 Standard**
    b. Object metadata:
        i. Enabled **guess MIME type**
        ii. Enabled **Give bucket owner full control**
        iii. Disabled **enable requester pays**
    c. Access to your S3 bucket: **Create a new IAM role**
    d. Encryption: **S3-Managed Keys (SSE-S3)**



11. Important: On the SMB Share settings page, be sure to configure the **Allowed/denied users and groups** and restrict access as needed!

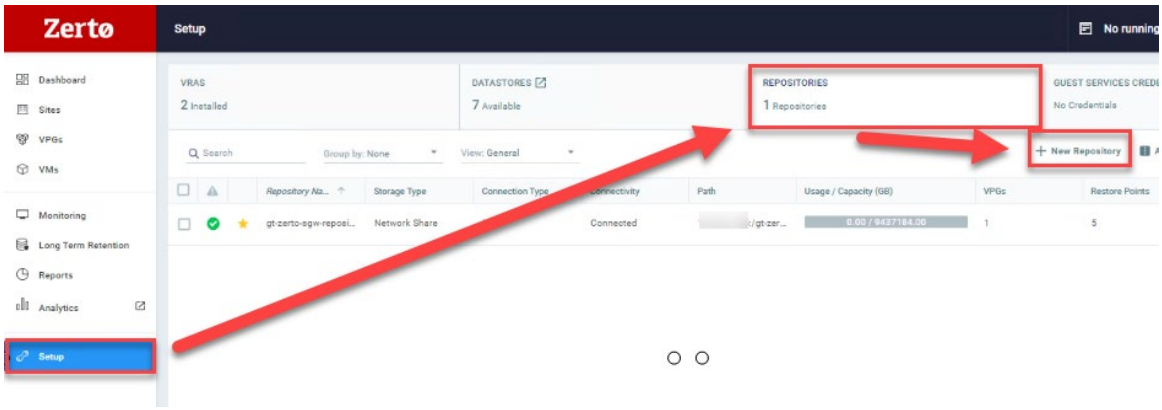


12. When done, click **Create file share**.
13. Get the path to the file share, in preparation for configuring the Index Repository in Zerto

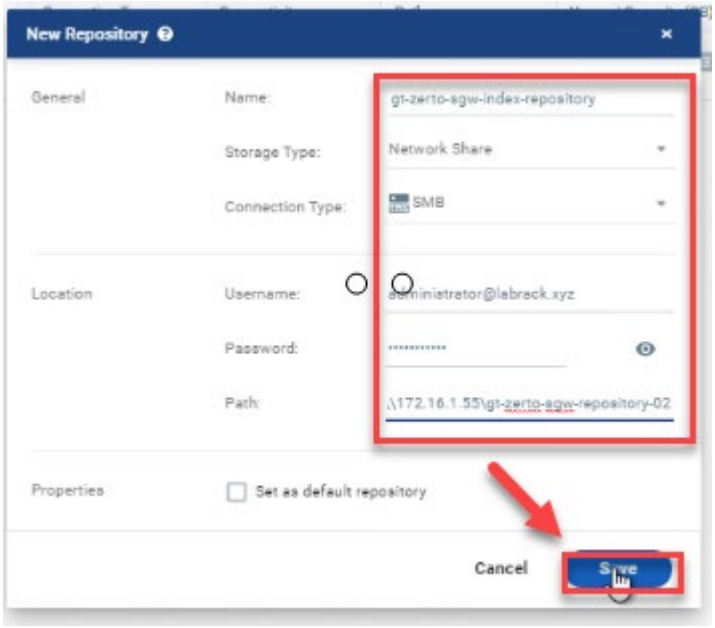


10. Adding the SMB Share as a Repository in Zerto for LTR Indexing

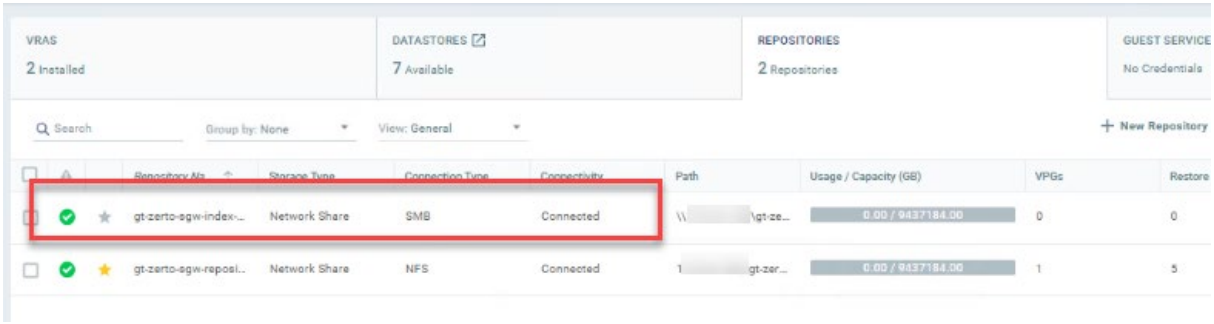
1. Log onto the Zerto UI in the same site you added the NFS LTR Repository to.
2. Click on **Setup** in the left navigation area, the click on **Repositories** on the right to go to the repositories view.
3. Click **+ New Repository**



4. Enter the repository information:
    a. Name: **enter the name of the repository**
    b. Storage Type: **Network Share**
    c. Connection Type: **SMB**
    d. Username: **domain username in the following format: user@mylab.xyz**
    e. Password: **password for this user**
    f. Path: **enter the UNC path you copied from the AWS Management Console for the SMB File Share**
5. Click **Save**.

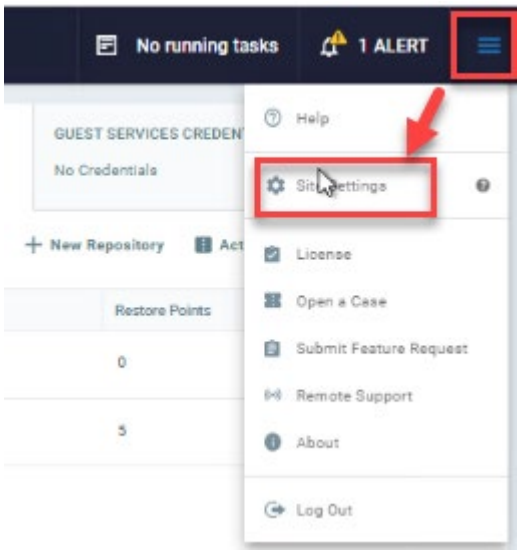


6. Verify the connection was successful



10.1. Configure the SMB Repository as the LTR Index Repository

7. Go to the Zerto Configuration menu (some call it the hamburger icon) at the top right of the user interface and click on **Site Settings**.



8. Click on **LTR Settings**, and select the **indexing repository** you just added, then click **Save**.