GET HANDS-ON & LEARN BEST PRACTICES FOR AWS DATA MIGRATIONS

OVERVIEW

The prospect of moving data workloads to the cloud can be daunting, so can trying to make sense of the array of tools, protocols, and mechanisms available to move data into AWS.

Objective of workshop - Get hands on experience in transferring data at scale using the available AWS online & hybrid services, where you will copy 10,000 local small files to Amazon S3, using AWS File Gateway.

CLIENT REQUIREMENTS

AWS account – you will need an AWS account to deploy & run this workshop

Browser – It is recommended that you use the latest version of Chrome or Firefox

Remote Desktop Client - You will need a RDP client to logon to the Windows EC2 instance (Windows RDP)

Key Pair – You will need a valid EC2 Key Pair in the AWS region you choose for your workshop (US-EAST-1 N.Virginia). Instructions are provided in this workshop on generating and downloading an EC2 Key Pair.

WORKSHOP MODULES

This workshop encompasses 2 modules

Module 1 - Deploy resources

Module 2 - AWS File Gateway

MODULE 2: AWS FILE GATEWAY

The AWS Storage Gateway service enables hybrid cloud storage access between an on-premises environment and AWS. The AWS Storage Gateway in File mode, enables you to store and retrieve objects in Amazon S3 using file protocols, such as NFS and SMB. Objects written through file gateway can be directly accessed in S3

Objective - In this module you will perform the following tasks

Create Amazon S3 buckets

Deploy an AWS storage gateway in “File” mode

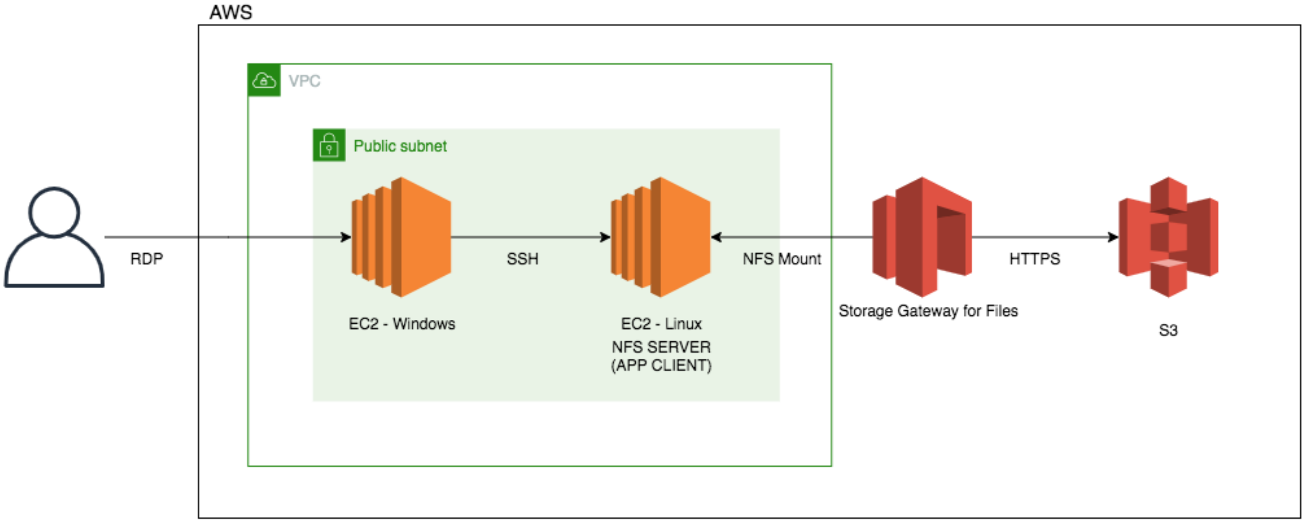
Create an NFS file share that is backed by an Amazon S3

bucket Mount the file gateway NFS share on a Linux host

Transfer data from the local Linux host to the NFS file gateway share

Verify data transferred to AWS

Architecture



Costing

Resource	Cost
1. EC2 - Windows – c5.xlarge	$0.354000 hourly (On – demand)
2. EC2 - Linux – m5.xlarge	$0.192000 hourly (On – demand)
3. Storage Gateway for Files	$0.01 per GB written to S3
4. S3 Standard	$0.023 per GB (first 50 TB / month)

CONNECT TO YOUR INSTANCE

Firstly let's retrieve the Windows administrator password from Secrets manager

1. From **your laptop** connect to the AWS console, click **Services** and type & select **Secrets Manager**
2. Click on the value shown under **Secret name** (i.e. AdminSecret-abczxy)
3. Scroll down the page and click on **Retrieve secret value**
4. Copy and paste the password value shown into a notepad file (i.e. zbf5%Uq*LAO!)

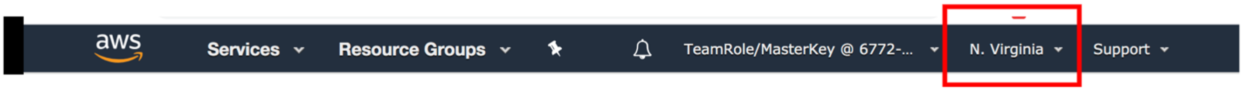
Next let's connect to your Windows Server EC2 instance

5. From **your laptop** connect to the AWS console, click **Services** and type & select **EC2**
6. From the left hand menu, select **Instances**
    - In the right hand pane, select the box next to "**Workshop Windows instance 1**", then right click and select **Connect**
    - Click on **Download Remote Desktop File**
    - Open the downloaded Remote Desktop File and select Connect at the prompt ○ Enter the credentials below and click on OK
        - username : Administrator
        - Password : the value you obtained from Secrets manager

7. When you have successfully logged into your Windows EC2 instance via the Remote Desktop Session, navigate back to your local workstation/laptop to where you stored the *.pem key file earlier. **Right click** & select **copy** on the *.pem file, and then go into your Remote Desktop Session (Windows EC2 instance) and **right click** on a free space on the desktop screen and select **paste** to copy the *.pem key file across.

Note: All remaining tasks for the workshop will be performed through the **Remote Desktop Session to the Windows EC2 Instance you just connected to in step 3**.

1. Open the Chrome icon located on the desktop of the **Windows EC2 instance** you are already logged into
2. Use Chrome to open the following URL https://dashboard.eventengine.run/login
   - *Follow the same steps used in Module 1*
   - From the AWS console, on the top left, click **Services** and type & select **EC2**
   - From the top right hand drop-down ensure your region is **us-east-1 (N. Virginia)**



   - From the left hand menu, select **Instances**
   - In the right hand pane, select the box next to “**Workshop Linux instance 1**”.
   - From the bottom window, select the **Description** tab, and write down the **private IP** address into your workshop.txt file where it states **Linux-Instance-Private-IP=**
   - You will use this **Linux-Instance-Private-IP** in the next module

CREATE S3 BUCKET- AWS FILE GATEWAY

Note: Ensure that you are logged into your Windows RDP session and to enter all required details into the workshop.txt file located on the desktop, as instructed.

This bucket will be used to back the AWS File Gateway that you will deploy

1. From the AWS console, click **Services** at the top of the screen and type & select **S3**
2. From the AWS S3 console select **+Create bucket**
3. Provide a unique bucket name for your **Source-S3-bucket**. Use the following naming convention “stg316-source-**xyz**” were **xyz** is combination your surname and first name (e.g. “**stg316-source-citizenj**”)
   - Take note of your **Source-S3-bucket** name in your workshop.txt file
4. Next select **US EAST (N.VIRGINIA)** as the region
5. Click **Next**
6. Click **Next**
7. Ensure the “**Block all public access**” check box is enabled, and select **Next**
8. On the final screen, select **Create bucket**

DEPLOY FILEGATEWAY APPLIANCE

1. From the AWS console, at the top of the screen, click **Services** and type & select **Storage Gateway**
2. Click the **Get started** button (Appears if it’s the first time you have used the service). ○
   Select **File Gateway** from the list and select **Next**
   ○ Select **Amazon EC2** & Click on the **Launch Instance**
   Icon ○ On the next screen, select the following values
    - Select the box next to **c5.2xlarge**
    - Select **Next: Configure Instance Details**
    - In the **Network** value select the workshop VPC which has the label of “**STG316**”
    - In the **VPC** value select the one that has a label of **STG316**
    - Leave all other values as default
    - Click **Next: Add Storage**
    - Click on **Add New Volume** (to add a second volume to the File Gateway to use as your cache drive) with the following values
        - Size : **150GB**
        - Volume Type : **Provisioned IOPS SSD**
        - IOPS : **7500**
    - Click **Next: Add Tags**
    - Click on **Add Tag**
    - Enter the following values (case sensitive)
        - Key = **Name**
        - Value = **STG316-filegateway**
    - Click **Next: Configure Security Group**
        - Click on the “**Select an existing security group**” check box
        - Select the security group with the name of **STG316-FileGatewaySG**
    - Click **Review and Launch**
    - Click **Launch**

        - Select your **key pair** that you created previously, and acknowledge the checkbox and Click **Launch Instances**

3. From the AWS console, click **Services** and type & select **EC2**
    - From the left hand EC2 console menu, select **Instances**
        - In the right hand pane, select the box next to “**STG316-filegateway**”, ensure the “**Status Check**” column for this EC2 instance shows “**2/2 checks passed**“ before proceeding to the next step (this may take a few minutes)
        - In the bottom window pane, select the **Description** tab, and take note of the **private IP** Address for the File Gateway instance into your workshop.txt file for the value of **File-Gateway-Instance-Private-IP=**.

ACTIVATE FILE GATEWAY

1. From the AWS console, at the top of the screen, click **Services** and type & select **Storage Gateway**
    - Click the **Get started** button
    - Select **File Gateway** from the list, and select **Next**
    - Select **Amazon EC2**, and select **Next**

> **Do not click on the Launch Instance, you have already done that previously.**

- Select **Public** for endpoint type, click **Next**
- Enter the **private IP** address your File Gateway instance (value of **File-Gateway-Instance-Private-IP)**
- Select **Connect to Gateway**
- On the next screen, Leave the time zone unchanged
- Enter a desired **Gateway name** (i.e. STG316-filegateway) o Select **Activate gateway**
- On the next screen, from the “**Allocated to**” drop down, select “select “**Cache**”
- Click on **Configure logging** and the default settings

- Click on **Save and continue**

CREATE NFS SHARE

In the next steps you will create an NFS file share from your AWS File Gateway.

1. Following on from the previous steps you should still be located in the **AWS Storage gateway console**, if not, from the AWS console, at the top of the screen, click **Services** and type & select **Storage Gateway**
2. From the left hand pane of the AWS Storage Gateway console, select **File shares**
3. Select **Create file Share** from the top menu
4. Enter the name of your **Source-S3-bucket** in the **Amazon S3 bucket name** field.
5. Select **Network File System (NFS)**
6. Select the **File Gateway** you just deployed (STG316-filegateway)
7. Click **Next**
8. On the next page, leave all the defaults and select **Next**
9. On the next page, click the **Edit** value next to **Allowed clients**
   - Remove the existing **0.0.0.0/0** value and replace it with **192.168.0.0/16**
   - Then click the **Close** button to the on the right of the screen for Allowed clients
10. Click the **Edit** value next to **Mount options**
    - Select "**No root squash**" for Squash level
    - Leave export as read-write
    - Then click the **Close** button to the on the right of the screen for Mount options
11. Scroll to the bottom of the page and click **Create file share**
12. On the same File Share page, check the box next to the name of your **File share ID**
    - In the details pane below, copy the command for mounting "**On Linux**" in to your **workshop.txt** for the value of "**First-NFS-FileShare-mount-command***"

CONNECT TO LINUX HOST

1. In your Remote Desktop session, click on Windows icon located at the bottom left of the screen
2. Type `CMD` and hit Enter to open a new command prompt
3. You should have stored your *.pem key file on the desktop as per the previous instructions. Enter the below commands in the command prompt
    - `cd  c:\users\administrator\desktop`

4. Next enter the below command to SSH into the Linux server, remember to replace the two values shown in **< >** with your values

    ssh -i ***<your-key-file-name>.****pem ec2-user@<****Linux-Instance-Private-IP>***

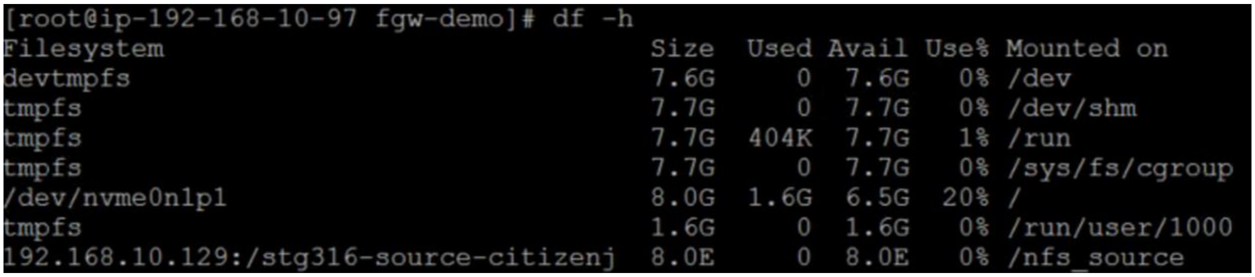
    i.e. `ssh –i stg316-key.pem ec2-user@192.168.10.102`

5. If this is the first time you have connected to this instance, a security alert dialog box that asks whether you trust the host to which you are connecting.
    - (Optional) Verify that the fingerprint in the security alert dialog box matches the fingerprint that you previously obtained in (Optional) Get the Instance Fingerprint (https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/connection-prereqs.html#connection-prereqs-fingerprint). If these fingerprints don’t match, someone might be attempting a “man-in-the-middle” attack. If they match, continue to the next step.
        - Choose **Yes** when you are ready to proceed**.**

    - A window opens and you are connected to your instance.

MOUNT NFSSHARE

1. In the open Putty SSH session type the following command
   - sudo su
2. Next, copy the NFS mount command you noted down in your workshop.txt for **First-NFS-FileShare-mount-command**, and simply replace the **[MountPath]** value at the end of the command with the value of "**/nfs_source"** and enter the entire command into the SSH session, and hit Enter
   - i.e. mount -t nfs -o nolock,hard 192.168.10.12:/stg316-source-citizenj /nfs_source
3. Run the below command to verify you have successfully mounted the NFS mount point of /**nfs_source**
   - df -h



```
[root@ip-192-168-10-97 fgw-demo]# df -h
Filesystem                            Size  Used Avail Use% Mounted on
devtmpfs                              7.6G     0  7.6G   0% /dev
tmpfs                                 7.7G     0  7.7G   0% /dev/shm
tmpfs                                 7.7G  404K  7.7G   1% /run
tmpfs                                 7.7G     0  7.7G   0% /sys/fs/cgroup
/dev/nvme0n1p1                        8.0G  1.6G  6.5G  20% /
tmpfs                                 1.6G     0  1.6G   0% /run/user/1000
192.168.10.129:/stg316-source-citizenj  8.0E     0  8.0E   0% /nfs_source
```

TRANSFER 10K FILES

Next we are going to copy 10,000 very small files from the local folder **/workshop_data** to the file gateway NFS share you created & mounted as **/nfs_source** , using a Linux copy script

1. First lets view the local data we are going to copy by running the following commands ○ `cd /workshop_data`
    - `find . -type f | wc -l`
        - This returns the number of files in the folder. How many files does it state?
2. Run the following commands to start the copy of 10,000 small files
    - `cd /scripts/fgw-demo`
    - `time ./copy_files_to_nfs.sh`

Note: Wait until you get the data transfer completed message before proceeding.

- How long did it take to copy 10,000 small files?

Note: The output of the script will return a **time** value, where the value for “**real**” tells you how long the copy operation to the file gateway local cache took

3. Run the below commands to verify the 10,000 files were copied to the File Gateway NFS share
    - `cd /nfs_source`
    - `ls -ltr`
    - `find . -type f | wc -l`
        - How many files does it show that you copied?

4. Next we are going to change the permissions & ownership of a file which will be a reference point in module 3, used to verify metadata being copied across. Run the following commands:

   ○ cd /nfs_source/appdata ○
   chmod 444 saturn.gif
   ○ chown -R user9:appadmin saturn.gif ○
   ls -ltr

5. Lastly, lets verify that the data from the local Linux server has been copied through your File Gateway NFS share to your **Source-S3-Bucket**
   - Return to your Chrome session and from the AWS console, at the top of the screen, click **Services** type & select **S3**
   - Select your **Source-S3-Bucket** name from the list
   - Check the box next to **Name** to select all objects
   - Click on **Actions** → **Get total size**
     - Note the total object stored in your S3 bucket via File Gateway (10901 = 10845 files + 56 folders)
   - Click **Cancel** when done viewing.

SUMMARY

In this module you have obtained hands on experience on how simple and seamless it is to leverage AWS File Gateway as a file transfer mechanism to Amazon S3 (in this case 10,000 small files), and also how the AWS File Gateway can enable hybrid cloud file storage architectures, where you can access your hot data via the local file gateway cache, where all your data is backed in an Amazon S3 bucket.

END OF MODULE 2

CLEAN UP WORKSHOP

Note: You will need to perform all the cleanup workshop from a browser session from **YOUR LAPTOP/WORKSTATION** and not from the Remote Desktop Session.

DELETETHE TWO S3 BUCKETS YOU CREATED

1. Navigate to the AWS console, at the top of the screen, click **Services** and type & select **S3**
2. Locate the bucket you created for **Source-S3-Bucket & Target-S3-Bucket**
3. Click on the check box next to the name
4. Select Delete from the top options
5. Follow the prompts
6. Repeat step 2-5 for **Target-S3-Bucket**

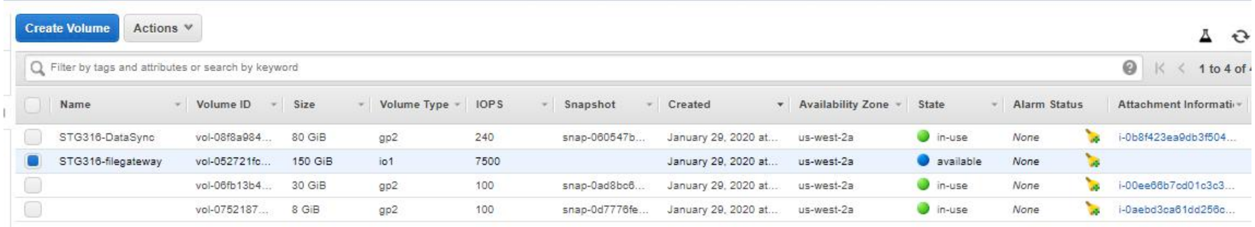
DELETE FILE GATEWAY NFS SHARES & GATEWAY

1. Navigate to the AWS console, at the top of the screen, click **Services** and type & select **Storage Gateway**
2. From the left hand menu select **File shares**
3. Select the two NFS shares you created (check the box next to them)
4. **Click on Actions → Delete File share**
5. Confirm deletion of resources and select **Delete**
6. From the left hand menu select **Gateways**
7. Select the File Gateway you deployed (STG316-filegateway)
8. Click on **Actions** → **Delete gateway**
9. Confirm deletion of resources and select **Delete**

DELETE EC2 RESOURCES DEPLOYED OUTSIDE OF CLOUDFORMATION

1. Navigate to the AWS console, at the top of the screen, click **Services** and type & select **EC2**
2. From the left hand menu select **STG316-FileGateway**
3. Click on **Actions → Instance state → Terminate**
4. Confirm deletion of resources and select **Delete**
5. From the left hand window pane select **Volumes**
6. Click on the **refresh** button on the top right hand corner until your **150GB io1** volume is showing its state as **available.**

7. Verify that the **Attachment information** is blank for this volume (not attached to any host)

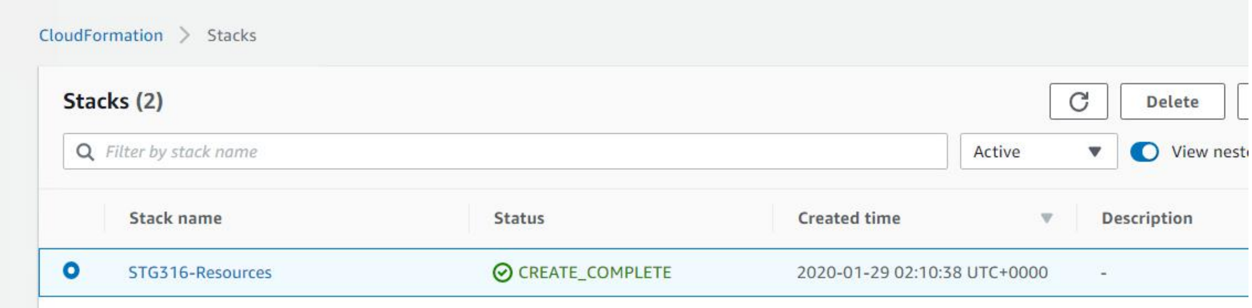


	Name	Volume ID	Size	Volume Type	IOPS	Snapshot	Created	Availability Zone	State	Alarm Status	Attachment Informati
□	STG316-DataSync	vol-08f8a984...	80 GiB	gp2	240	snap-060547b...	January 29, 2020 at...	us-west-2a	● in-use	None	i-0b8f423ea9db3f504...
■	STG316-filegateway	vol-052721fc...	150 GiB	io1	7500		January 29, 2020 at...	us-west-2a	● available	None	
□		vol-06fb13b4...	30 GiB	gp2	100	snap-0ad8bc6...	January 29, 2020 at...	us-west-2a	● in-use	None	i-00ee66b7cd01c3c3...
□		vol-0752187...	8 GiB	gp2	100	snap-0d7776fe...	January 29, 2020 at...	us-west-2a	● in-use	None	i-0aebd3ca61dd256c...

8. Select your **150GB io1** volume and click **on Actions → Delete Volume**
9. Confirm deletion of resources and select **Delete**

DELETE CLOUD FORMATIONSTACK – STG316-RESOURCES

1. Navigate to the AWS console, at the top of the screen, click **Services** and type & select **CLOUDFORMATION**
2. Select the **STG316-Resources** stack you deployed



	Stack name	Status	Created time	Description
●	STG316-Resources	CREATE_COMPLETE	2020-01-29 02:10:38 UTC+0000	-

3. Click on **Delete**
4. Confirm deletion of resources and select **Delete stack**
5. Click on the refresh icon until the stack disappears from the list
6. Next click on the dropdown and change it from Active to Deleted
7. Verify **STG316-Resources** shows a status of **DELETE_COMPLETE**
8. Next click on the dropdown and change it from Deleted to Active
9. From the same Cloudformation window select the **STG316-VPC** stack
10. Repeat steps 3-7

Cleanup tasks complete