DevOps

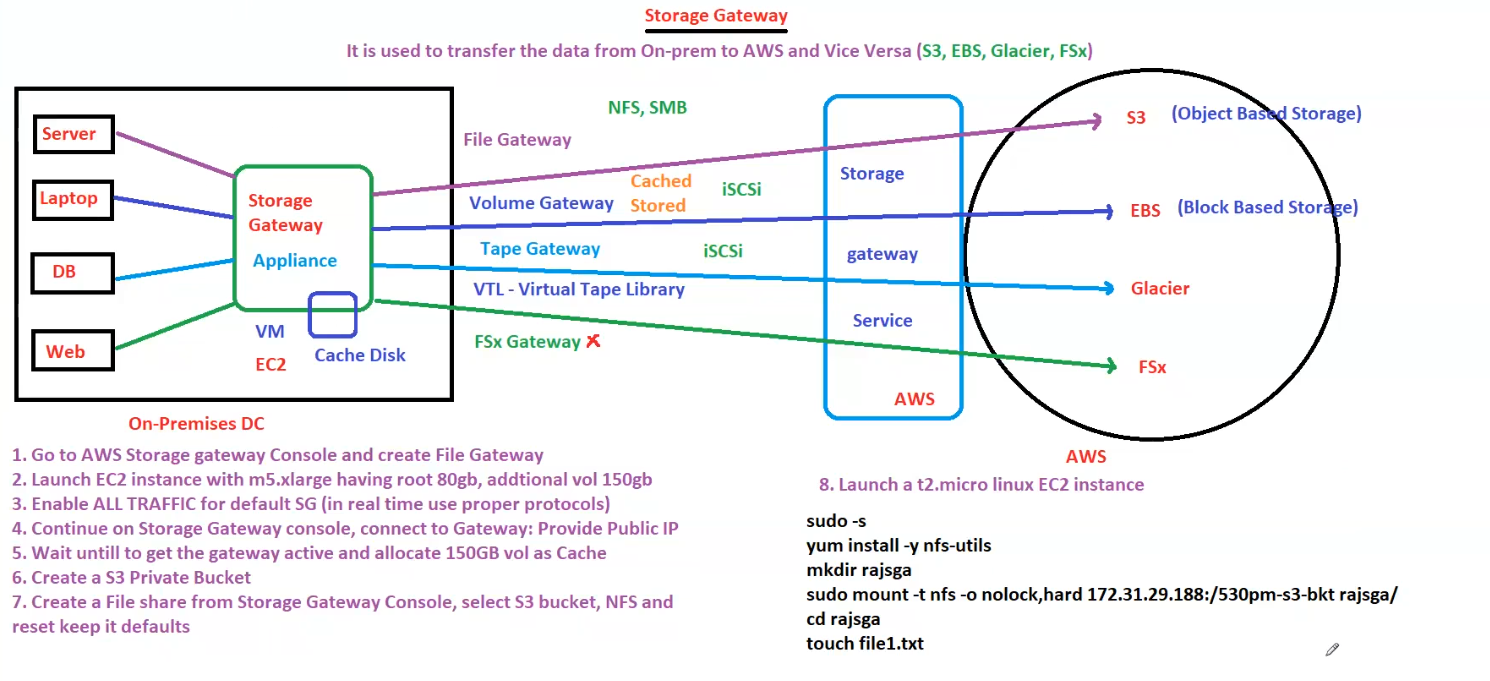
[Stroage Gateway 2](#_Toc193686953)

[Overview 2](#_Toc193686954)

[File Gateway Architecture 5](#_Toc193686955)

# Stroage Gateway

## Overview

****

**Here are the structured notes based on the image:**

**AWS Storage Gateway Overview**

**AWS Storage Gateway is used to transfer data between On-Premises (DC) and AWS in both directions. It integrates on-prem infrastructure with AWS services like S3, EBS, Glacier, and FSx.**

**Types of Storage Gateway**

1. **File Gateway (NFS, SMB)**
   * **Enables file-based storage.**
   * **Transfers files to Amazon S3 (Object Storage).**
   * **Used for backup and cloud-based file sharing.**
2. **Volume Gateway (iSCSI)**
   * **Provides block storage access.**
   * **Two modes:** 
     + **Cached Mode: Stores frequently accessed data on-prem, while full data resides in AWS.**
     + **Stored Mode: Entire dataset is stored locally, with asynchronous backup to AWS.**
3. **Tape Gateway (iSCSI)**
   * **Creates a Virtual Tape Library (VTL).**
   * **Used for backup and archival to Amazon Glacier.**
4. **FSx Gateway ❌ *(Not supported in this setup)***
   * **Used to integrate Windows file servers with Amazon FSx.**

**Components Involved**

1. **On-Premises Infrastructure**
   * **Server**
   * **Laptop**
   * **Database**
   * **Web Application**
2. **Storage Gateway Appliance**
   * **Acts as an intermediary to connect on-prem infrastructure to AWS.**
   * **Deployed as a VM or EC2 instance.**
   * **Requires a cache disk to optimize storage performance.**
3. **AWS Services Used**
   * **S3 (Object Storage)**
   * **EBS (Block Storage)**
   * **Glacier (Archival Storage)**
   * **FSx (Windows File Storage, not used in this setup)**

**Setup Process**

**Step 1: Create File Gateway**

1. **Go to AWS Storage Gateway Console.**
2. **Create a File Gateway.**

**Step 2: Launch an EC2 Instance**

1. **Choose m5.xlarge instance type.**
2. **Configure:** 
   * **Root Volume: 80GB.**
   * **Additional Volume: 150GB.**

**Step 3: Configure Security**

1. **Enable ALL TRAFFIC for the default Storage Gateway.**
2. **(For production, use proper security protocols instead of allowing all traffic.)**

**Step 4: Connect Storage Gateway**

1. **Provide the Public IP of the Storage Gateway.**

**Step 5: Allocate Cache Storage**

1. **Wait until the gateway becomes active.**
2. **Allocate 150GB volume as Cache.**

**Step 6: Create an S3 Private Bucket**

1. **Navigate to S3 Console.**
2. **Create a private bucket for storing data.**

**Step 7: Create a File Share**

1. **Use Storage Gateway Console.**
2. **Select the S3 bucket created earlier.**
3. **Choose NFS protocol.**
4. **Keep all other settings as default.**

**Testing the File Gateway**

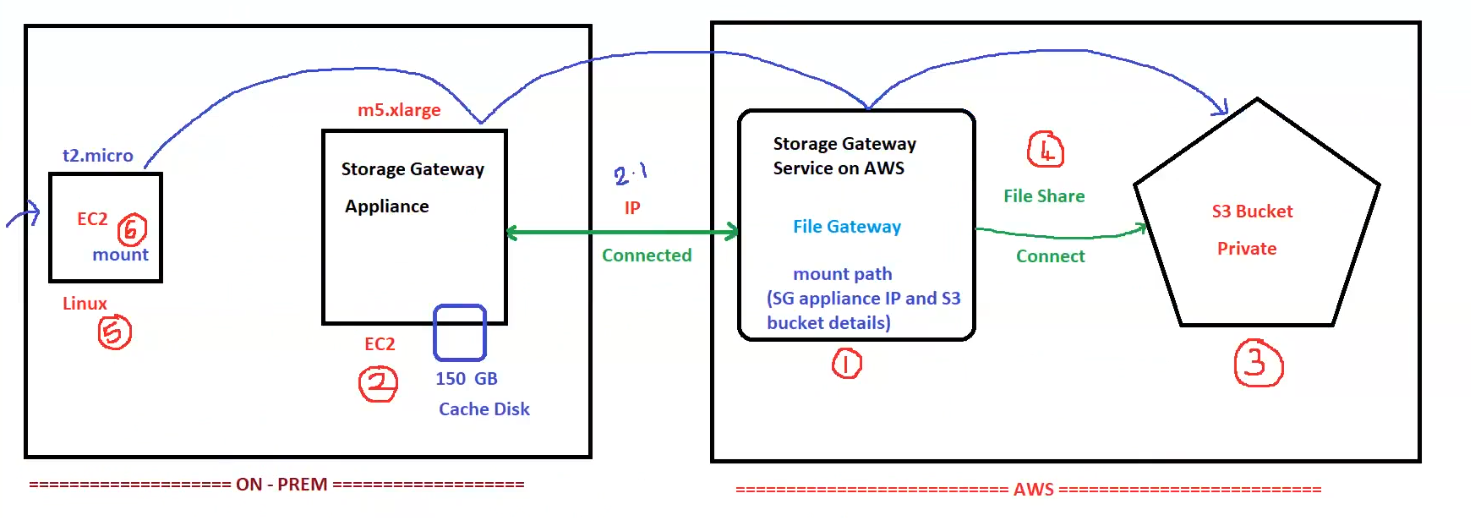
**Step 8: Launch a Test EC2 Instance**

1. **Launch a t2.micro Linux instance.**
2. **Install NFS utilities:**
3. **sudo -s**
4. **yum install -y nfs-utils**
5. **Create a directory to mount the file share:**
6. **mkdir rajsga**
7. **Mount the NFS file share:**
8. **sudo mount -t nfs -o nolock,hard 172.31.29.188:/530pm-s3-bkt rajsga/**
9. **Verify by creating a test file:**
10. **cd rajsga**
11. **touch file1.txt**

**Conclusion**

* **The Storage Gateway Appliance acts as a bridge between on-premises infrastructure and AWS.**
* **File Gateway stores files in S3, while Volume Gateway and Tape Gateway use block storage.**
* **Security must be configured properly to avoid unnecessary exposure.**
* **Testing ensures proper mounting and access to cloud storage.**

## File Gateway Architecture

****

**AWS Storage Gateway - File Gateway Architecture**

The diagram illustrates how an **on-premises environment** connects to AWS using **AWS Storage Gateway (File Gateway)** to store data in **Amazon S3**.

**Step-by-Step Process**

**On-Premises Setup (Left Side)**

1. **Deploy Storage Gateway Service (Step 2)**
   * Launch an **m5.xlarge EC2 instance**.
   * Attach a **150GB Cache Disk**.
   * Acts as the **Storage Gateway Appliance**.
2. **Connect Storage Gateway to AWS (Step 2.1)**
   * Register the Storage Gateway in AWS.
   * Provide the **Public IP** to AWS Storage Gateway Service.
3. **Launch a Linux Instance for Testing (Step 5)**
   * Deploy a **t2.micro EC2 instance**.
   * Used to test and mount the file share.
4. **Mount File Share (Step 6)**
   * The Linux EC2 instance will mount the **File Gateway share**.

**AWS Cloud Setup (Right Side)**

1. **Create a File Gateway in AWS (Step 1)**
   * Configure a **File Gateway** in the AWS Storage Gateway Service.
   * AWS will provide a **mount path** (SG appliance IP + S3 bucket details).
2. **Create an S3 Bucket (Step 3)**
   * A **private Amazon S3 bucket** is created to store the data.
3. **Create a File Share (Step 4)**
   * Connect the File Gateway to the S3 bucket.
   * This allows on-premises instances to store and retrieve files in S3.

**Flow of Data**

1. **On-premises EC2 (Storage Gateway Appliance) sends data to AWS Storage Gateway.**
2. **Storage Gateway connects to S3, storing files in the configured bucket.**
3. **A Linux EC2 instance (on-premises) mounts the File Gateway share for read/write access.**
4. **Any data stored in the mounted directory is automatically stored in S3.**

**Key AWS Services Used**

* **Amazon EC2**: Runs the Storage Gateway and test instance.
* **AWS Storage Gateway (File Gateway)**: Bridges on-premises storage with Amazon S3.
* **Amazon S3**: Stores files securely in the cloud.

**Conclusion**

This setup allows an **on-premises data center to store files in AWS S3** through **Storage Gateway (File Gateway)**. The on-prem **Linux instance can access the cloud storage as if it were a local drive** while benefiting from AWS scalability and durability.