Class 31

**S3 Glacier:**

* S3 Glacier is used to store Archived data, which is important but not used many times, like bank pay slips, marriage video (in bank policy money transactions should be for 10 years but not used much).
* S3 Glacier is a cheaper storage, and also free up to 10GB data
* We can’t upload data directly in S3 Glacier, we use 3rd party tool to upload data, one of the tool is “FastGlacier” <https://fastglacier.com/download.aspx> download tool

i, IAM -> Users -> Add user ->Username: & Programmatic access -> Attach existing policies &

AdministratorAccess -> Access key ID & Secret access key (copy)

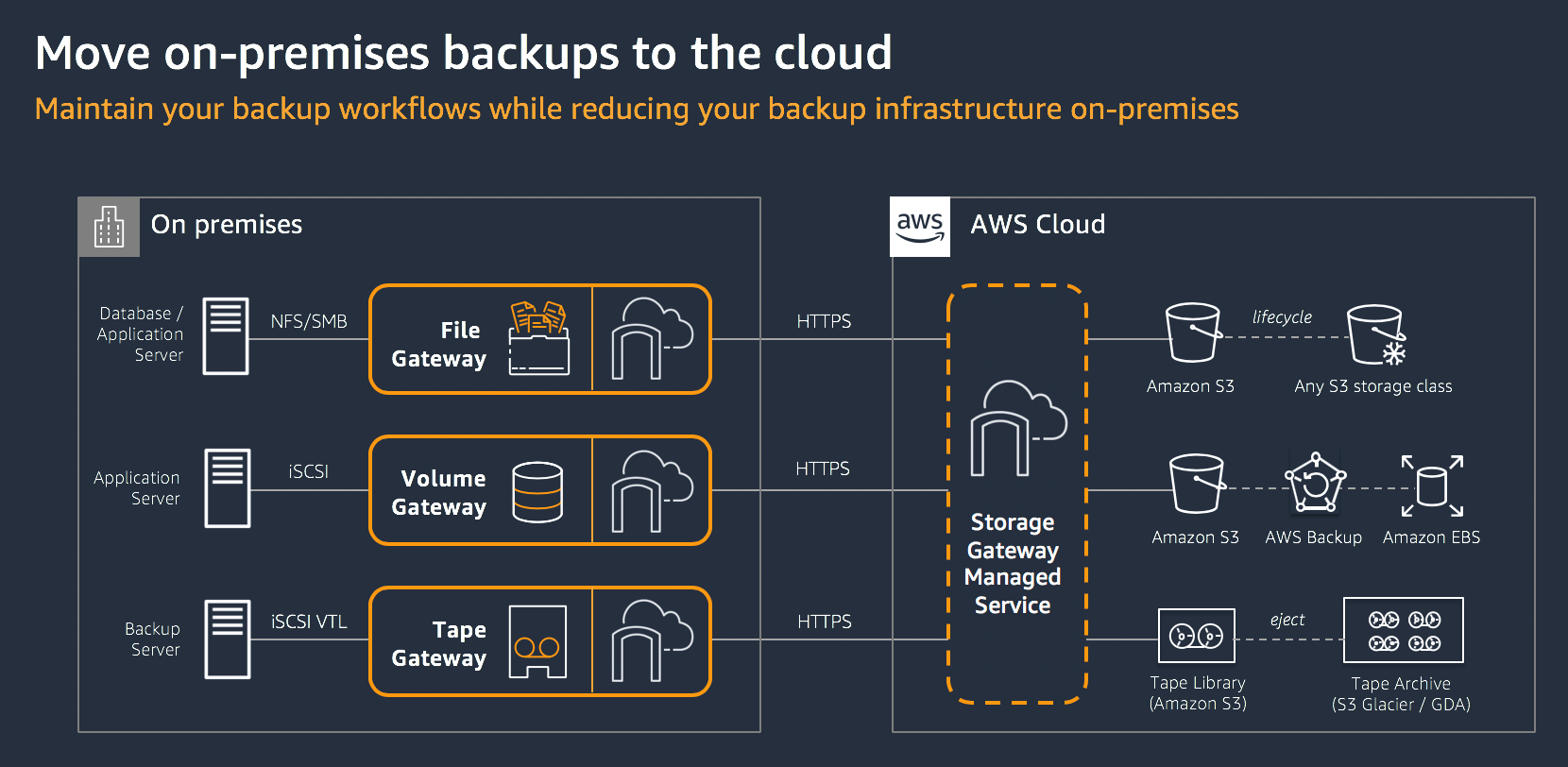
ii, install FastGlacier tool -> Account Name: & Secret key & Access Key (FastGlacier tool is used to upload

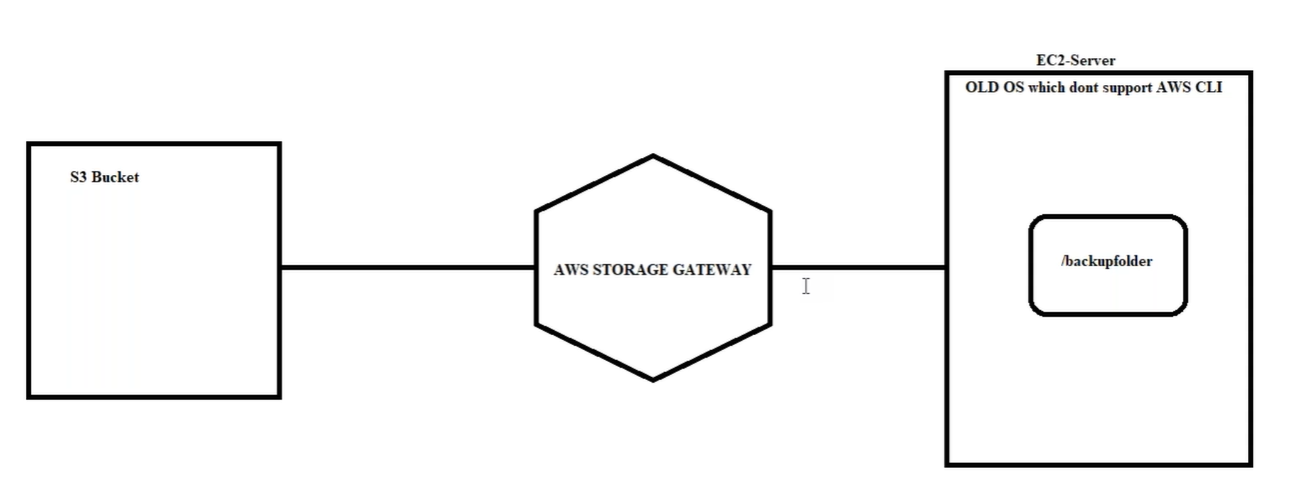
retrieve data, if we want to move with life cycle we no need)

Now we can upload the data

* In s3-> life cycle -> we keep data that should be 30 days in standard storage -> then move to glacier and be there for 5 years and then expire (like bank transactions data)

**Storage Gateway:** [**https://d2908q01vomqb2.cloudfront.net/e1822db470e60d090affd0956d743cb0e7cdf113/2019/11/23/Use-case-1-More-on-premises-backups-to-the-cloud.png**](https://d2908q01vomqb2.cloudfront.net/e1822db470e60d090affd0956d743cb0e7cdf113/2019/11/23/Use-case-1-More-on-premises-backups-to-the-cloud.png)





* Storage Gateway used to take data from on-premises to AWS DC

Note: in this example we take data Ec2 server to S3 Bucket using Storage Gateway, because S3 doesn’t have file system to mount to Ec2 server, like EBS volume and EFS. S3 is object storage like google drive, its not a file system storage.

Lab:

i, Ec2 -> Ubuntu Server 18.04 -> #!/bin/bash

apt update && apt install -y nfs-common -> AppServer

ii, S3 -> Create bucket -> SGWbucket

iii, Storage Gateway -> Create gateway -> File gateway -> Amazon EC2 (click on “Launch instance” that requires m4.xlarge & 10GB volume \* Name & SGW \* SGW server is deployed\* copy public ip) -> Public -> IP address: “past public ip here” -> Gateway time zone: Eastern time & Gateway name -> dheerajSGW ->

here 10Gb which we are added to server, display here and recommend for 150GB -> Disable logging

iii, Storage Gateway -> Create file share **(since we went with file gateway)** -> Amazon S3 bucket name: **sgwbucket1** & Access objects using: Network File System (NFS) -> One Zone-IA -> file share is created

here we get [mount path] of S3 - storage Gateway

i, Appserver -> copy Public DNS ->

puttu -> mkdir /myappdata

mount -t nfs -o nolock,hard 10.0.1.134:/sgwbucket1 /myappdata/ [copy mount path to directory]

cd /myappdata/

for i in {1..100}

do

echo $(date) > File$i

sleep 3

cat File$i

done

now since /myappdata/ in mounted to storage gateway the files created in folder can see in S3 bucket, here S3 bucket acts as GBS volume

Since S3 is object storage and no file system we can’t mount to server,

so, we add s3 to Storage gateway and mount Storage gateway to AppServer

**Workflow: [S3 – Storage Gateway (in file share add s3 bucket) (get mount path) – Appserver(mount)]**

**Q, what is Storage Gateway?**

**A: Storage Gateway is a mediator between On-premises and AWS cloud,**

**To take data or hardware drive backup and coping it to**

**S3 - if they are files,**

**Volume - if they are snapshots**

**Tape - if that is glacier data**

**AWS Backup:**

If we talk about Microsoft Azure, when it launched it directly came with Backup tool

AWS Backup is nothing but snapshots backup, of EFS, EC2, Storage Gateway

We also have lifecycle daily snapshot backup, but lifecycle is only for Ec2

i, AWS Backup -> **Backup vaults** -> Create Backup vault -> Backup vault name: awsb26backupvaults

ii, Ec2 -> instance -> ubuntu 18.0 -> Tags (Key: Name & Value - optional: AppServer / Key: **Env** & value: **prod**)

iv, putty -> cd / (change to root)

Mkdir /appdata

Wget tereformlink

for i in {1..10}

> do

> cp terraform\_0.13.5\_windows\_amd64.zip terraform$i.zip

> sleep 2

> done

iii, AWS Backup -> Backup plans -> Build a new plan -> Backup plan name: DailyAppBackup & Rule name: Rule 1 & Customize backup window & **Backup vault**: awsb26backupvaults

iii, open backup plan and click on "Assign resources"

AWS Backup -> Backup plans -> DailyAppBackup -> Assign resources -> Resource assignment name: AppServar & Key: **Env** & Value: **prod**

Now **create the backup**: Taking the Volume backup into snapshot

iii, AWS Backup -> Dashboard -> Create an on-demand backup -> select "Resource type: Ec2" -> select "Backup vault" (it will store “Jobs -> Backup jobs”)

Now **restore the backup** volume: restoring the Volume which is backup taken

iii, AWS Backup -> Protected resources-> open “Resource ID” -> select “Recovery point ID” and “Restore” (we can see “Jobs -> Restore jobs”)

* Now my AppServer instance is restored

**Workflow: [Backup vaults - Backup plans (add Backup vaults) (add tags: Env, prod) – Ec2 instance(create tags: Env, prod)-]**