PGPCC - Mentor session

Flow and Structure for MS3

Course 2 Week 2

“S3, EFS, EBS”

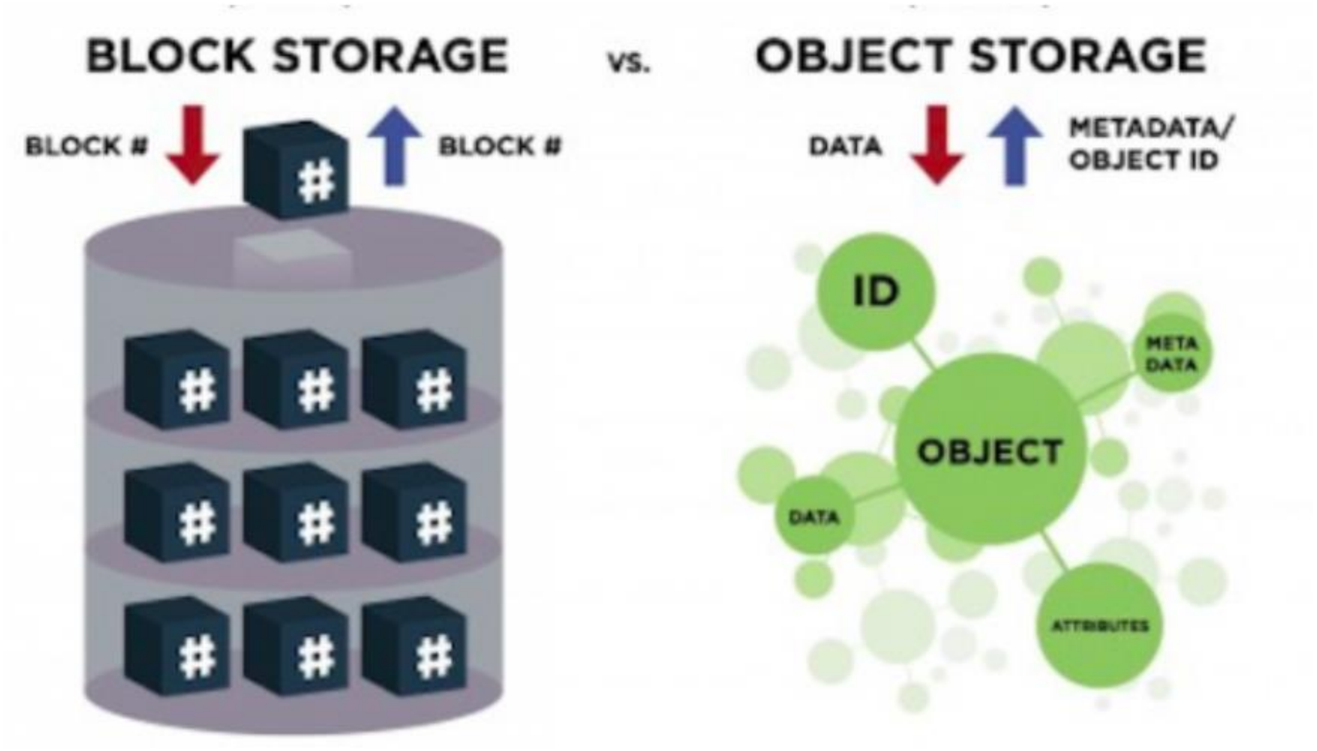
Agenda

- Total duration 120 mins
- Recap and Case study (50 mins)
- Cost Optimization Pillar (10 mins)
- Q&A from participants (60 mins)

AWS Week 3 – Storage

- Module 11 – EBS
- Module 12 – EFS
- Module 13 – S3 , Transition options, Mounting S3 on Local File system

Module 10 – Forms of Storage



- Block storage and Object storage

- **Block** – EBS, EFS, Instance
- storage

- **Object** – S3

- What is the difference between Block and Object - Check the diagram on left !

- Use cases.

Module 11 – EBS

Elastic Block Store – Check on elasticity in it! Expandableeee on the fly..as well.

EBS is replicated for HA and Durability within AZ

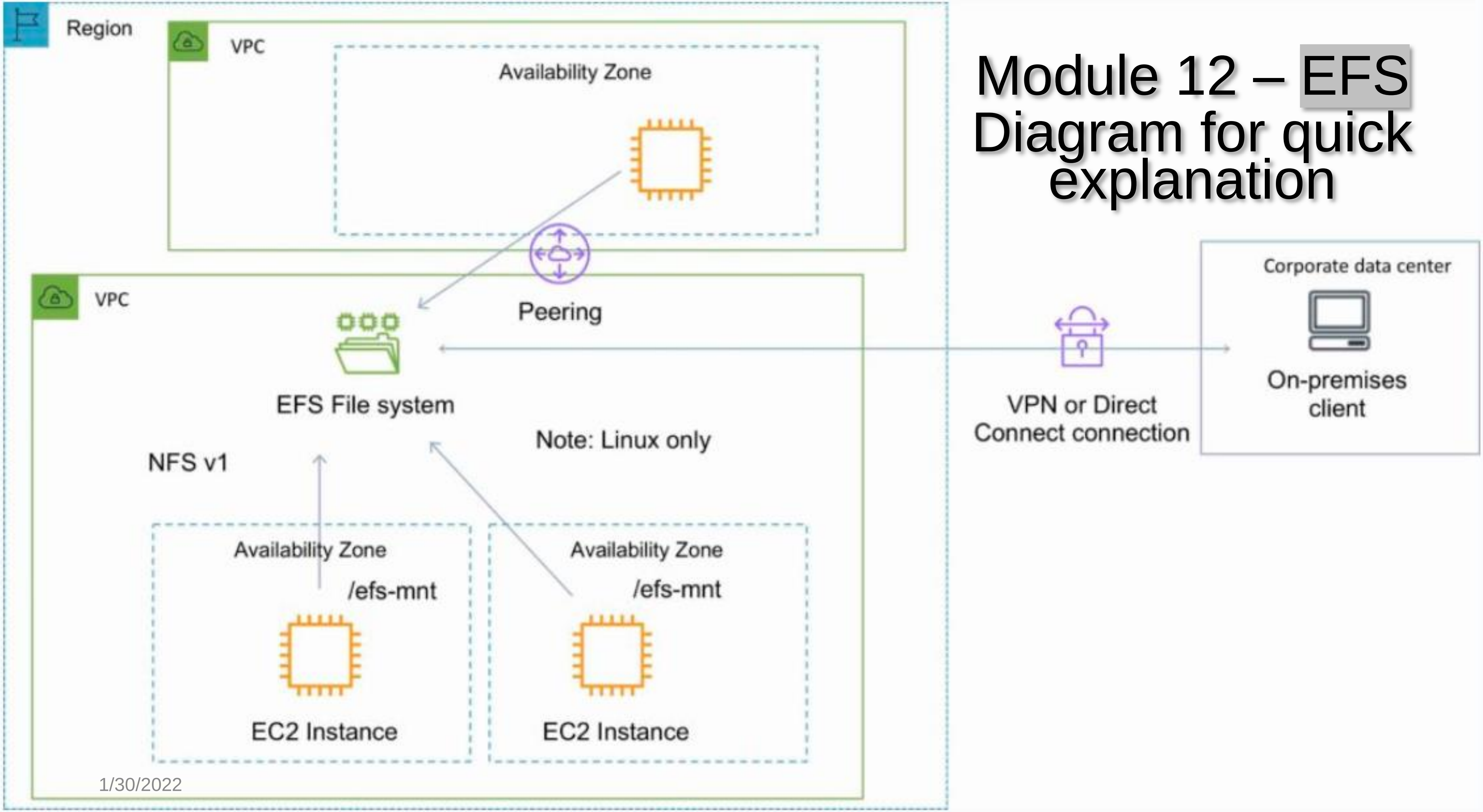
Different EBS volume – gp2, io1, sc1, st1, hdd standard

Following commands **lsblk** to see volumes, **mkfs** to create file system and **mount** to mount it.

EBS (AWS API’s and common use cases)

- SSD - Gp2 – Most workloads
- SSD - Io1 – Mission critical application – DB’s
- HDD - Sc1 – Big data and warehouse
- HDD - St1 – File servers
- HDD Standard – Workloads where data is Infrequently accessed

Module 12 – EFS Diagram for quick explanation



1/30/2022

Options during configure EFS

Select the VPC first and then **customize**

- **Automated backups** Charges apply
- **Enable Lifecycle management** – Aah so you can move to different storage class… Infrequently accesses (IA) – Range 7-90 days as on date
- **Performance mode** – General purpose (latency sensitive use cases), Max IO (higher throughput and ops/sec)
- **Throughput mode** – Bursting , provision – Basically you determine if throughput scales or not as size changes
- Enable **encryption at rest** checkbox
- Select **VPC and subnet** – will get clearer as you reach Week 3
- Now you can **optionally select policies**…root access prevention, read access, enforces in-transit encryption

✓ How you **access once it is configured** – Remember this is for Linux based systems only.. You **install efs utils** and use the commands applicable. Remember to configure SG!

Highlights to note for EFS

- Again reminding - For **Linux** based system only!
- **NFS** V4 Supports
- Pay as you go storage!
- It is meant for – Handle 1000's of connection …….
- Can scale to Petabyte
- Consistency – Read after write

Windows based

- **FSx** for Windows
- **FSx for Lustre** – Primarily for Machine learning apps...

Module 13 – S3 , Transition options, Mounting S3 on Local File system

- Object based store – **Not suitable for OS**
- Bucket and objects – Is S3 a truly **global service ?**

- **Storage class** – S3 Standard, S3 IA, S3 OneZone IA, S3 Intelligent Tier, S3 Glacier, S3 Glacier Deep Archive

  Understand the difference, pricing, retrieval time.

- **Versioning** – You enable at bucket level and primary purpose it is needed – can be great backups.
- **Lifecycle** – Moving to cost effective tier based on timelines…
- **Replication** – Same region and multi , Is that for DR or Performance ?
- Static website hosting , **CORS**
- **S3 Transfer acceleration**

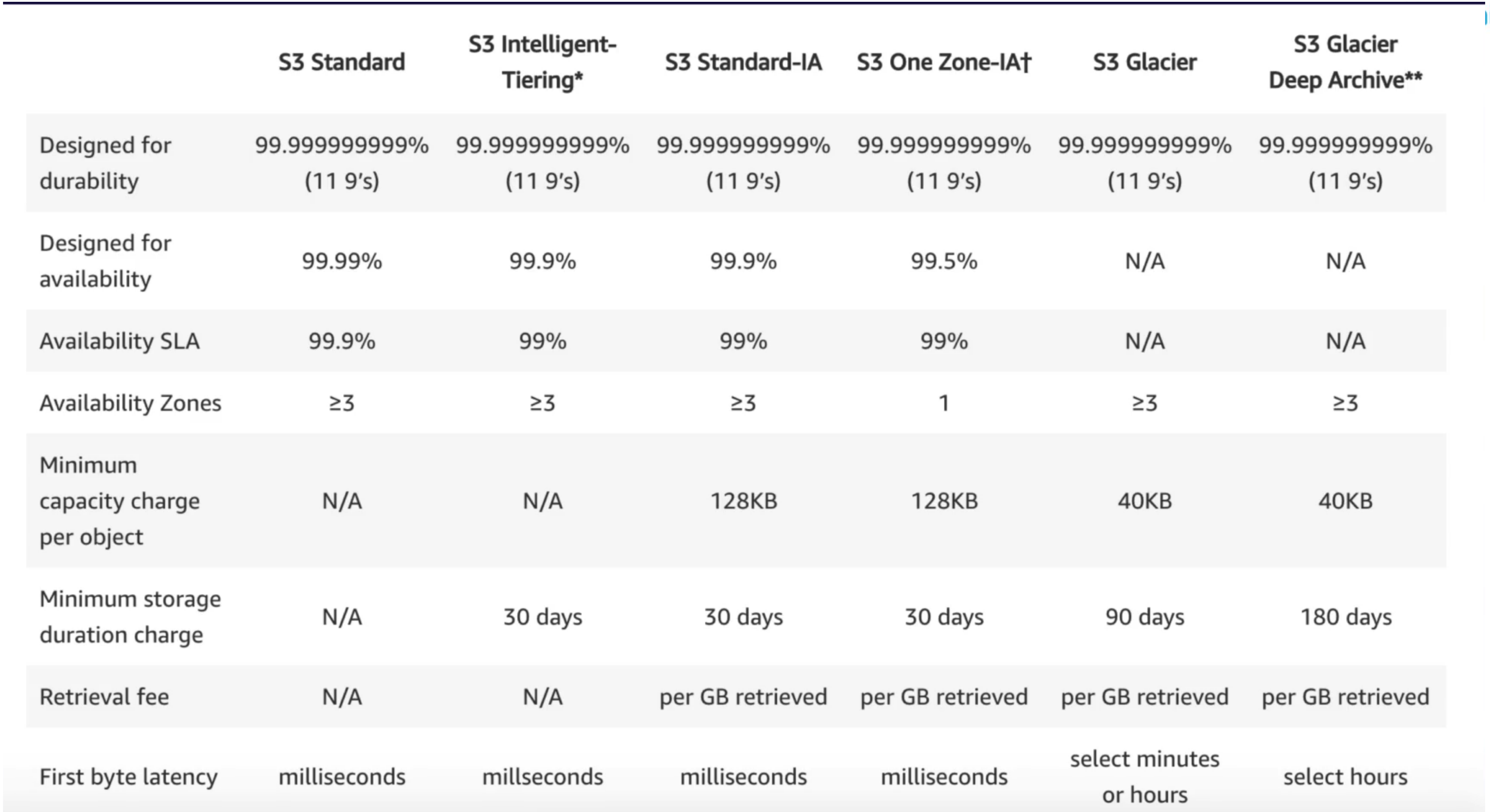
S3 pricing is based on – Storage, Requests, Data transfer, Transfer Acceleration if you use, Cross region replication

S3 features continued….

- **Security** in S3
- **MFA Delete**
- **S3 Object lock**
- **S3 Glacier Vault lock**
- **Prefix and performance**
- **S3 Select** option – What does it help in ?
- Finally can it be **mounted** as local drive ?

Yes can be mounted as drive…(FUSE plugin). Commands check DIY section…

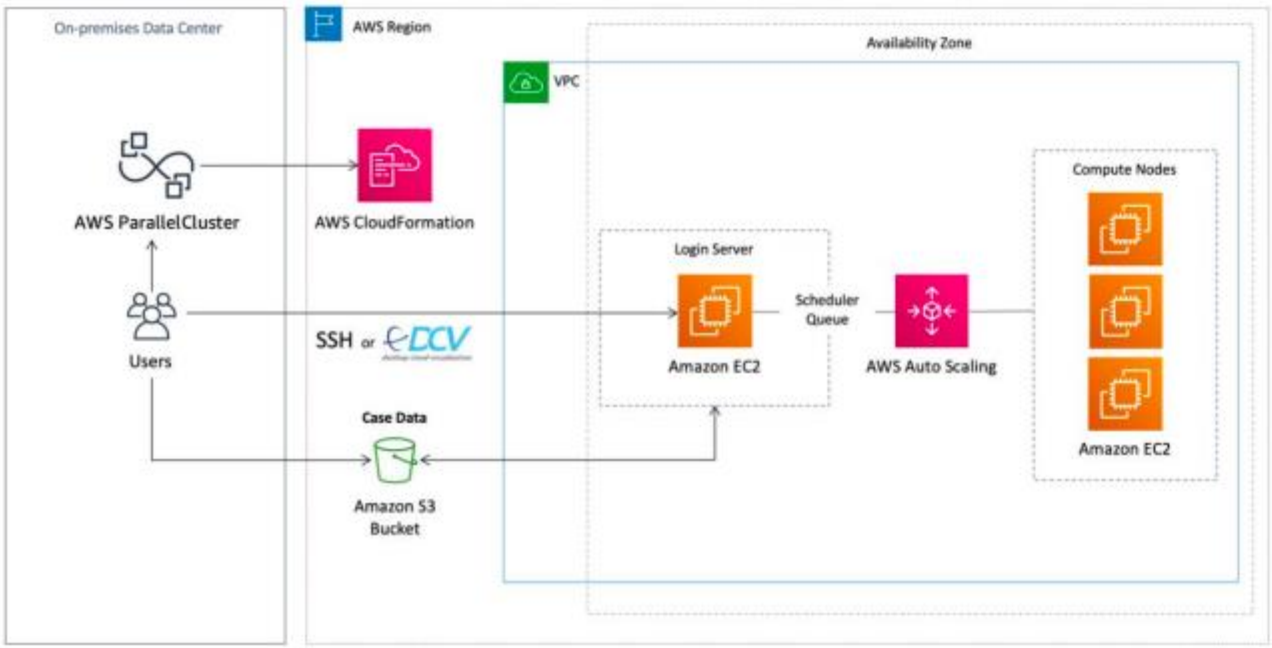
sudo apt install s3fs
mkdir <directory-name>
s3fs <bucket name> <directory-name> -o iam_role='ec2-multirole'



	S3 Standard	S3 Intelligent-Tiering*	S3 Standard-IA	S3 One Zone-IA†	S3 Glacier	S3 Glacier Deep Archive**
Designed for durability	99.999999999% (11 9's)	99.999999999% (11 9's)	99.999999999% (11 9's)	99.999999999% (11 9's)	99.999999999% (11 9's)	99.999999999% (11 9's)
Designed for availability	99.99%	99.9%	99.9%	99.5%	N/A	N/A
Availability SLA	99.9%	99%	99%	99%	N/A	N/A
Availability Zones	≥3	≥3	≥3	1	≥3	≥3
Minimum capacity charge per object	N/A	N/A	128KB	128KB	40KB	40KB
Minimum storage duration charge	N/A	30 days	30 days	30 days	90 days	180 days
Retrieval fee	N/A	N/A	per GB retrieved	per GB retrieved	per GB retrieved	per GB retrieved
First byte latency	milliseconds	millseconds	milliseconds	milliseconds	select minutes or hours	select hours

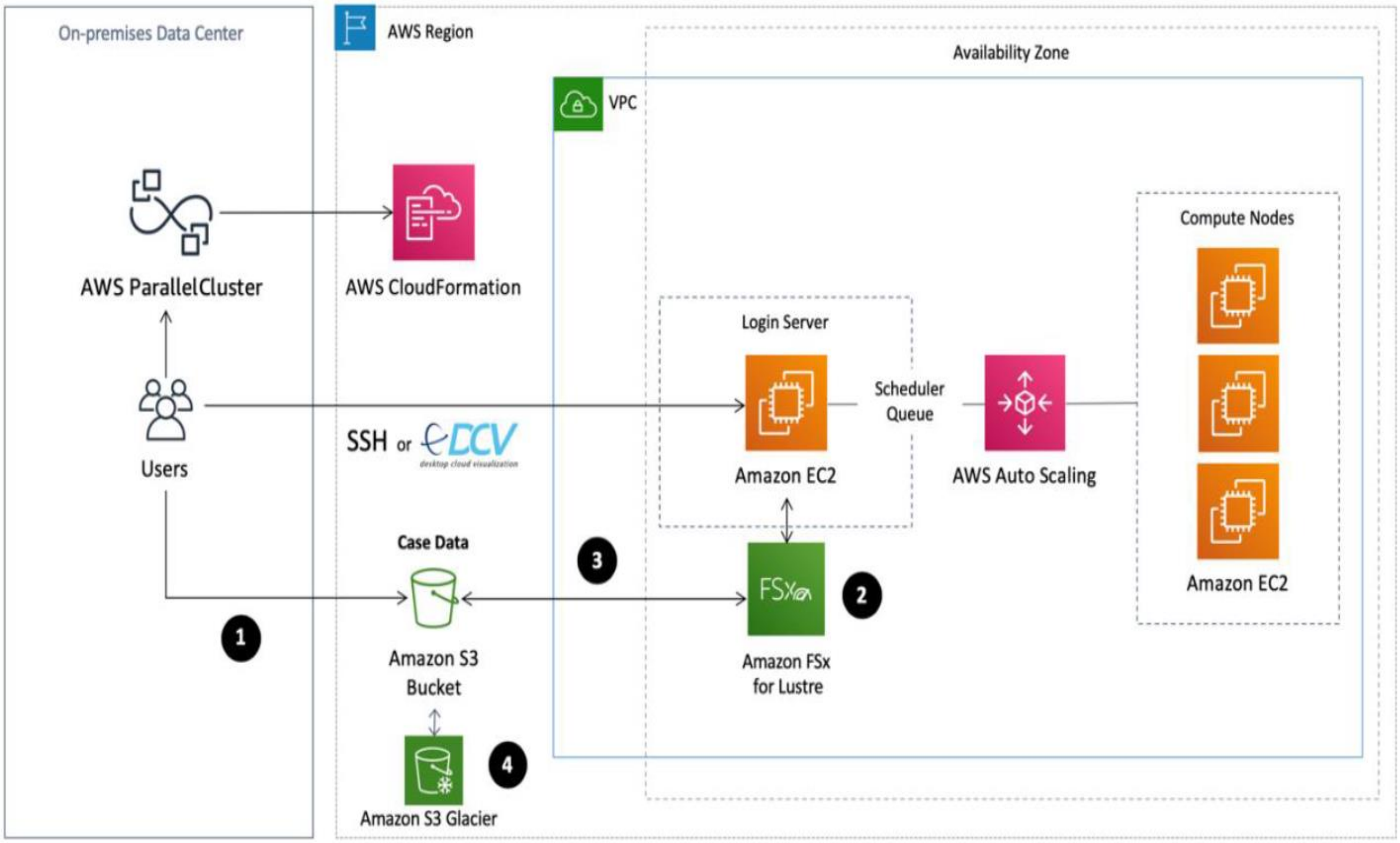
Case: Compute & Storage

- Fluid dynamics is the study of the motion of fluids, usually in the presence of an object. Typical fluid flows of interest to engineers and scientist include: flow in pipes, through engines, and around objects, such as buildings, automobiles, and airplanes
- **Computational fluid dynamics (CFD)** is the study of these flows through computer simulation and modeling
- CFD workloads are typically MPI-based (message passing interface), **tightly coupled workloads** relying on a **large number of cores across many nodes**. **Many** of the AWS instance types, such as the **compute family instance types**, are designed to include support for this type of workload.



- A common quick-start approach for CFD users getting started with AWS is to combine a **custom Amazon Machine Image (AMI) with the AWS CLI and a bash script**. After launching an Amazon EC2 instance, software can be added to the instance and an AMI is created to be used as the starting point for all compute nodes.
- It is typical to set up the SSH files and the bashrc file before creating the custom AMI or “golden image”. Although many CFD solvers do not require a **shared file location**, one can easily be created with an exported NFS volume

DataLifecycle in CFD Case



Data lifecycle has 4 stages
1) Transferring input data to AWS (can be from on-premises)
2) Running your simulation with input (hot) data
3) Storing your output (warm) data
4) Archiving inactive (cold) data

Manual and scripted approaches of data transfer can use the AWS CLI for transferring data to S3, which helps optimize the data transfer to S3 by executing parallel uploads.

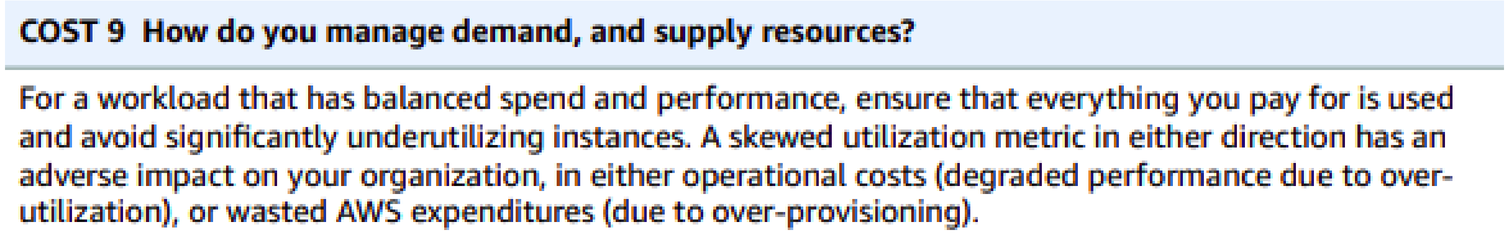
Storage use in CFD

Type	Description	Use
EBS	Block Storage	Block storage or export as a Network File System (NFS) share
FSx for Lustre	Managed Lustre	Fast parallel high-performance file system optimized for HPC workloads
S3	Object Storage	Store case files, input, and output data
Glacier	Archival Storage	Long-term storage of archival data
EFS	Managed NFS	Network File System (NFS) to share files across multiple instances. Occasionally used for home directories. Not generally recommended for CFD cases.

Well architected framework - “Cost Optimization” Pillar

- **Automation**: **Auto Scaling**, which ensures elasticity at minimal effort, and increases **staff productivity** by eliminating manual capacity planning work
- **Decommission resources**: Use automation to reduce or remove the associated costs of the decommissioning process. Designing your workload to perform automated decommissioning will reduce the overall workload costs during its lifetime. You can use **AWS Auto Scaling to perform the decommissioning process**
- **Cost**: Spot is also integrated into multiple AWS services, such as EC2 Auto Scaling groups (ASGs),
- **Dynamic Supply**: Leverage the elasticity of the cloud to supply resources to meet changing demand**. Take advantage of APIs or service features to programmatically vary** the amount of cloud resources in your architecture dynamically
- **Scheduling**: Auto Scaling can implement manual, scheduled or demand based scaling, you can also use metrics and alarms from Amazon CloudWatch to trigger scaling events for your workload.

- “Well architected framework” document “Manage demand and supply resources” section
  - https://docs.aws.amazon.com/wellarchitected/latest/framework/wellarchitected-framework.pdf



COST 9 How do you manage demand, and supply resources?

For a workload that has balanced spend and performance, ensure that everything you pay for is used and avoid significantly underutilizing instances. A skewed utilization metric in either direction has an adverse impact on your organization, in either operational costs (degraded performance due to over-utilization), or wasted AWS expenditures (due to over-provisioning).

Q&A

Thank you

Happy Learning!