**IAM:**

* AWS SDK:
  + Language-specific APIs
  + Enables you to access and manage AWS services programmatically
* IAM Credentials report:
  + A report that lists all your account’s users and the status of their various credentials
* IAM Access Advisor (user level):
  + Access Advisor shows the service permissions granted to a user and when those services were last accessed
  + You can use this info to revise your policies
* Shared Responsibility Model for IAM

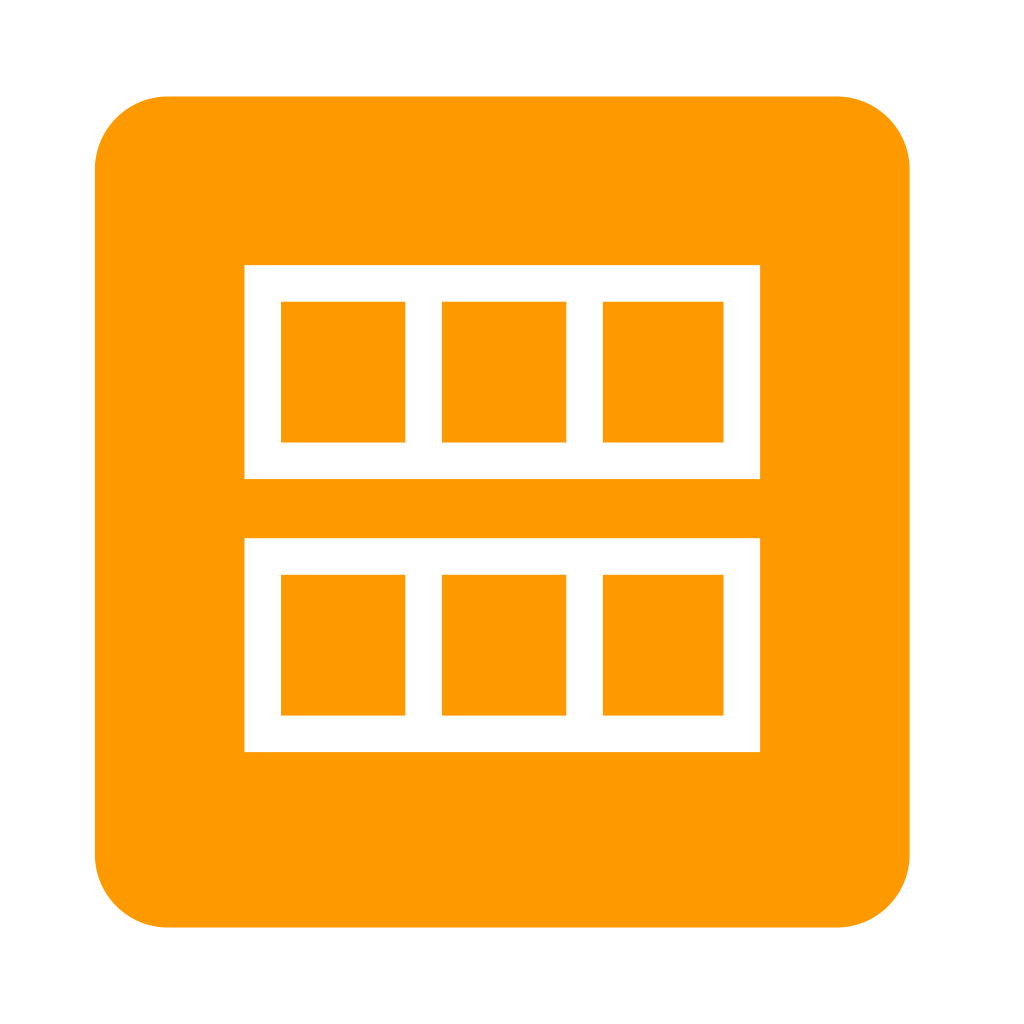
|  |  |
| --- | --- |
| **AWS** | **You** |
| Infrastructure (global network security) | Users, Groups, Roles, Policies management and monitoring |
| Configuration and vulnerability analysis | Enable MFA on all accounts |
| Compliance validation | Rotate all keys often |
|  | Use IAM tools to apply appropriate permissions |
|  | Analyze access patterns & review permissions |

**EC2:**

* EC2 = Elastic Compute Cloud = Infrastructure as a Service
  + Mainly consists in the capability of:
    - Renting virtual machine (EC2)
    - Storing data on virtual drives (EBS)
    - Distributing load across machines (ELB)
    - Scaling the service using an auto-scaling group (ASG)
  + Instance Types:
    - General purpose:
      * Great for a diversity of workloads such as web servers or code repositories
    - Compute Optimized:
      * Great for compute-intensive tasks that require high performance processors
    - Memory Optimized:
      * Fast performance for workloads that process large data sets in memory
    - Storage Optimized:
      * Greate for storage-intensive tasks that require high, sequential read and write access to large data sets on local storage
  + Security Groups:
    - Acting as a “firewall” on EC2 instances
  + Purchasing Options:
    - On-Demand Instances:
      * **Short workload**, un-interrupted workloads
      * Predictable pricing
      * **Pay by second**, no upfront payment
    - Reserved (1-3 yrs):
      * Reserved Instances: **long workloads**
      * Convertible reserved instances: long workloads with flexible instances
      * Reserved Instances’ Scope: Regional or Zonal
    - Savings plans (1-3 yrs): commitment to an **amount of usage**, **long workload**
    - Spot Instances:
      * **Short workload**
      * MOST cost-efficient (**cheap**)
      * Can **lose instances** (less reliable)
    - Dedicated Hosts:
      * Book an **entire physical server**
      * **Control instance placement**
      * Allows you to use your existing server-bound software licenses
      * **MOST expensive**
    - Dedicated Instances:
      * No other customers will share your hardware
      * NO control instance placement
      * Means you will have **your own instance on your own hardware**
    - Capacity Reservations:
      * **Reserve capacity in a specific AZ for any duration**
      * No time commitment
      * No billing discounts

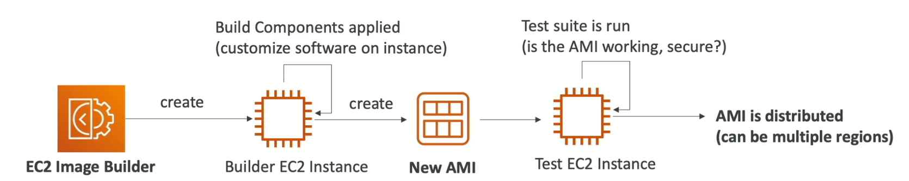


**EC2 Instance Storage:**

* EBS = Elastic Block Store
  + **Network drive** you attach to your instances while they run
  + Can only be **mounted to one instance at a time**
  + Bound to a **specific availability zone**
* EBS Snapshots:
  + Make a backup (snapshot) of EBS volume at a point in time
  + EBS Snapshot Archive:
    - Move snapshot to an “archive tier” that is 75% cheaper
    - 24-72 hrs to restore
* AMI = Amazone Machine Image
  + **Customization of EC2 instance**:
    - Add your own software, OS, monitoring,…
    - Faster boot / config time
  + Built for a **specific region**



* EC2 Image Builder
  + Automate creation, maintain, validate and test EC2 AMIs



* EC2 Instance Store
  + If you need **high-performance hardware disk**, use EC2 Instance Store
    - Better I/O
    - **Lose their storage if stopped**
    - Good for buffer / cache / scratch data / **temporary content**
    - Risk of data loss if hardware fails
* EFS = Elastic File System
  + Managed NFS (Network file system) that **can be mounted to 100s of EC2**

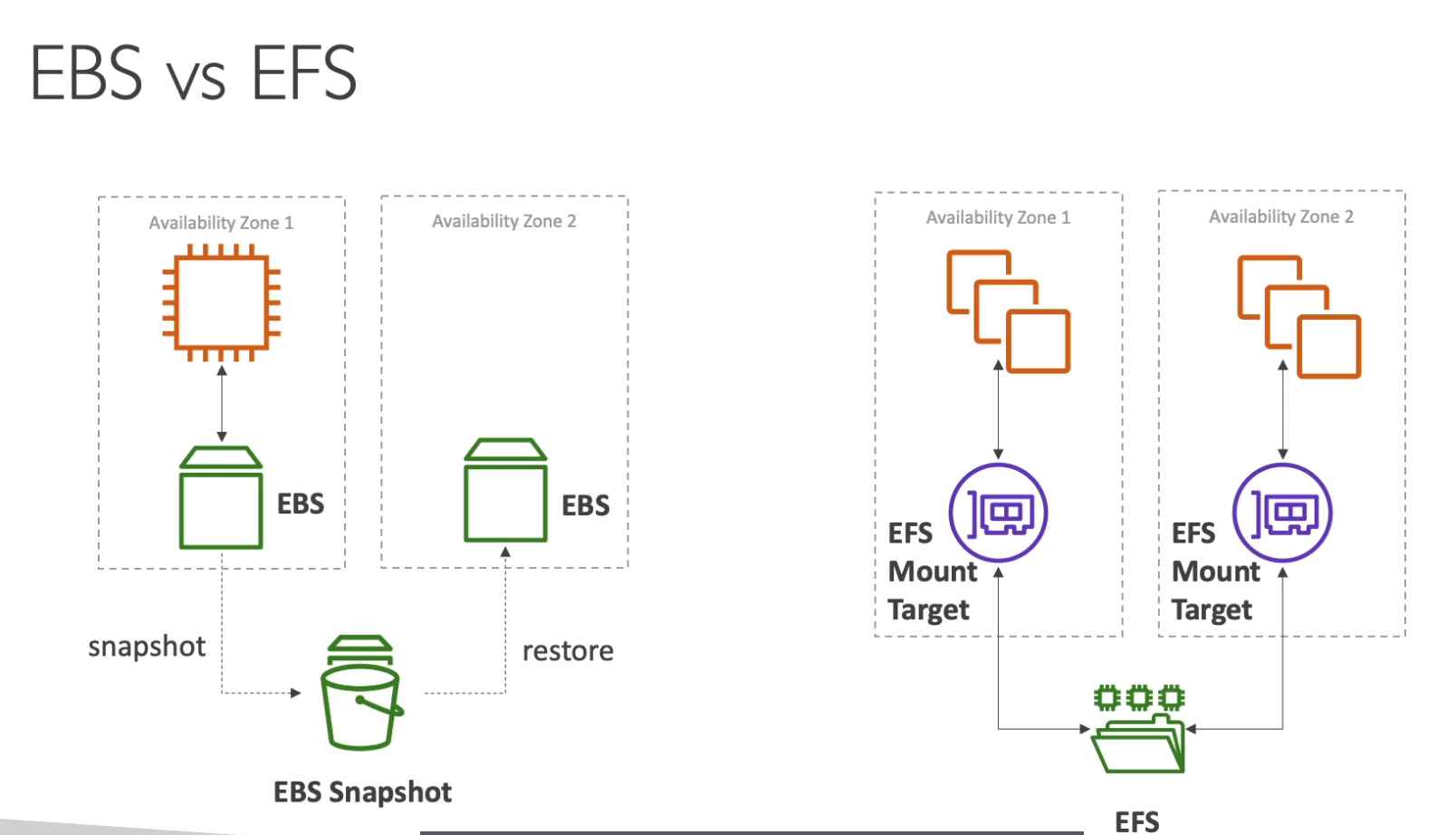


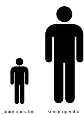
Figure . EBS is like an USB, while EFS is like Google Drive

* EFS Infrequent Access (EFS-IA): **storage class** that is **cost-optimized** for files **not accessed every day**
* Amazon FSx
  + Launch **3rd party high-performance file system** on AWS



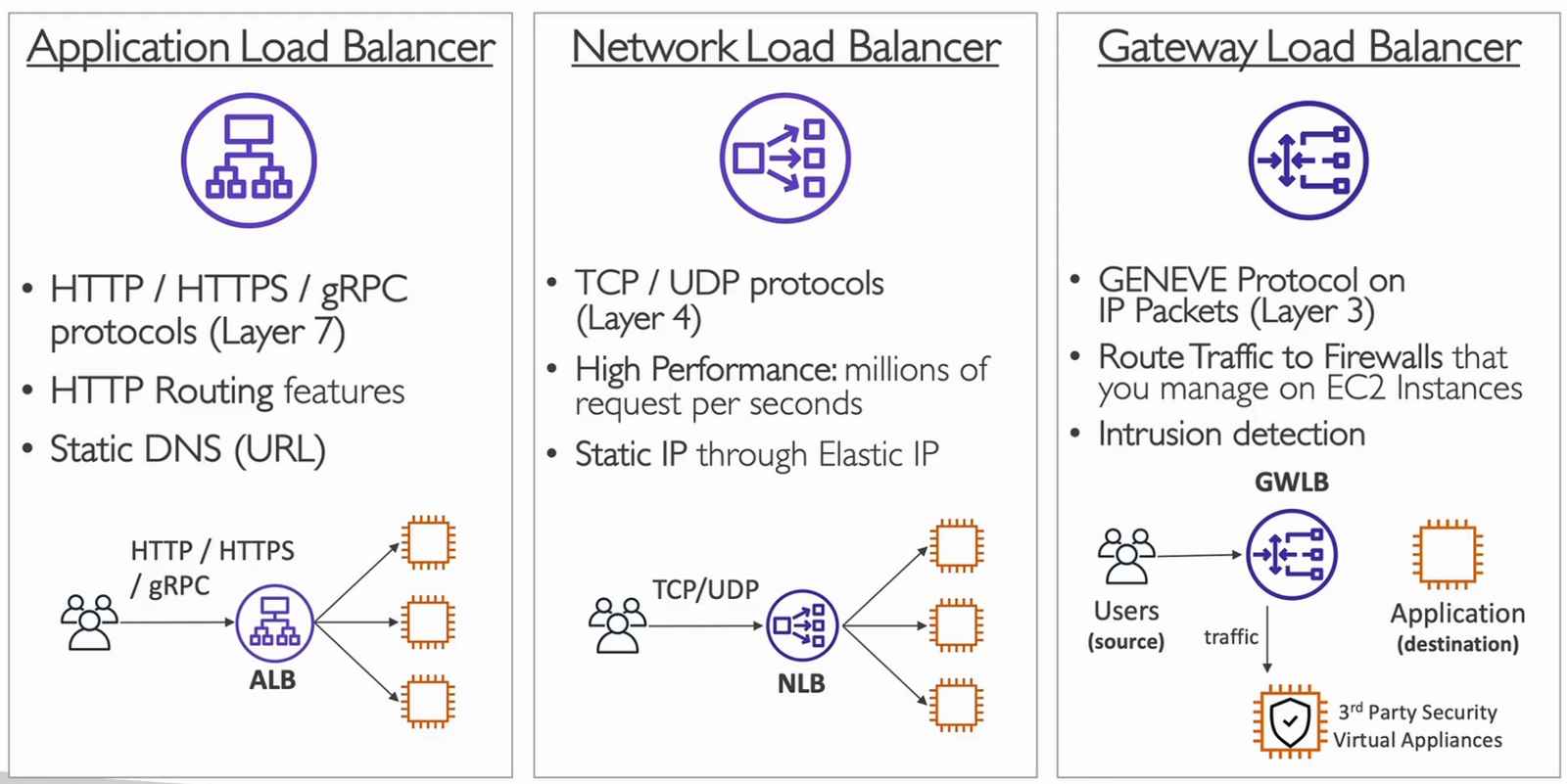
* + FSx for Windows:
    - Network File System (NFS) for Windows server
  + FSx for Lustre:
    - **High performance computing** linux file system
* Shared Responsibility Model for EC2:

|  |  |
| --- | --- |
| **AWS** | **You** |
| Infrastructure | Setting up backup / snapshot procedure |
| Replicate for data for EBS volumes & EFS drives | Setting up data encryption |
| Replacing faulty hardware | Responsibility of any data on the drives |
| Ensuring their employees cannot access your data | Understanding the risk of using EC2 instance store |

**ELB & ASG:**

* Vertical Scalability: increasing **the size** of the instance
* Horizontal Scalability: increasing **the number of instances / systems** for your app
* Scalability: ability to **accommodate a larger load** by making the **hardware stronger (scale up)**, or by **adding notes (scale out)**
* Elasticity: once a system is scalable, elasticity means that there will be **some “auto-scaling” so that the system can scale based on the load**. This is “cloud-friendly”: pay-per-use, match demand, optimize costs
* Agility: new IT resources are only a click away, which means that you **reduce the time** **to** **make those resources available** to your developers from weeks to just minutes

* ELB = Elastic Load Balancer
  + **Servers** that **forward internet traffic** to multiple **servers (EC2 instances) downstream**
  + 4 kinds of load balancers:
    - **Application** Load Balancer (**HTTP/HTTPS** only) – Layer 7
    - **Network** Load Balancer (ultra-high performance, allows **TCP**) – Layer 4
    - **Gateway** Load Balancer (**3rd party appliances**) – Layer 3
    - Classic Load Balancer – Layer 4 & 7





* ASG = Auto Scaling Group
  + **Scale out** (**add** EC2 Instances) to **match increase** load
  + **Scale in** (**remove** EC2 Instances) to **match decrease** load
  + Ensure minimum / maximum number of machines running
  + Automatically **register new instances** to a load balancer
  + **Replace unhealthy** instances
  + (Optional) Scaling Strategies:
    - Manual Scaling: update size of ASG manually
    - Dynamic Scaling: respond to changing demand
      * Simple / Step Scaling:
        + Ex: CPU > 70%, add 2 units
      * Target Tracking Scaling:
        + Ex: Average ASG CPU stay around 40%
      * Scheduled Scaling:
        + Ex: increase the min, cap to 10 at 5pm on Fridays
      * Predictive Scaling:
        + Use ML to predict future traffic ahead of time

**Amazon S3:**

* Allows people to **store objects** (files) in “**buckets**” (**directories**)
* **Globally unique name** (across all regions all accounts)
* Max object file is 5TB
* S3 Security:
  + **User-Based**:
    - **IAM-Policies**: which API calls should be allowed for a specific user from IAM
  + **Resource-Based:**
    - **Bucket Polices:** bucket wide rules from the S3 console – allow cross account
    - **Object Access Control List (OACL):** finer grain (can be disabled)
    - **Bucket Access Control List (BACL):** less common (can be disabled)
    - **Encryption:** encrypt objects in Amazon S3 using encryption keys
* Static Website Hosting:
  + Host static websites and have them accessible on the Internet
* Versioning:
  + Enabled at **Bucket Level**
  + **Multiple versions** of files, **prevent accidental deletes**
* Replication:
  + **Must enable versioning** in source and destination buckets
  + **Cross-region Replication (CRR)**
    - Compliance
    - Lower Latency access
    - Replication across accounts
  + **Same-region Replication (SRR)**
    - Log aggregation
    - Live replication between production and test accounts
* Storage Classes:
  + Standard
    - **99.99% availability**
    - **Frequently Accessed** Data
    - Use case: big data analytics, mobile & gaming app, content distribution,…
  + Infrequent Access (IA)
    - **Less frequently accessed**, but requires **rapid access** when needed
    - Lower cost than Standard
    - **Standard-Infrequent Access** (S3 Standard-IA)
      * 99.99% availability
      * Use case: disaster recovery, backups
    - **One Zone-Infrequent Access**
      * 99.99% durability in **single AZ**, **data lost when AZ is destroyed**
      * 99.5% availability
      * Use case: storing 2nd backup copies of on-premise data, or data you can recreate
  + Glacier Storage
    - **Low-cost** object storage for **archiving/backup**
    - Price: price for storage + **object retrieval cost**
    - **Glacier Instant Retrieval**
      * Millisecond retrieval, great **for data accessed once a quarter**
      * **Minimum storage** duration of **90 days**
    - **Glacier Flexible Retrieval**
      * Expedited (1-5 mins), standard (3-5 hrs), bulk (5-12 hrs) – free
      * **Minimum storage** duration of **90 days**
    - Glacier Deep Archive
      * **Long-term storage**
      * Standard (12 hrs), bulk (48 hrs)
      * **Minimum storage** duration of **180 days**
  + Intelligent Tiering
    - Small monthly monitoring & auto-tiering fee
    - **Moves objects automatically** between access tiers **based on usage**
    - **No retrieval charges**
* AWS Snow
  + High-secure, **physical devices**
    - **Collect and process** data at the **edge**
    - **Migrate data into and out of AWS**
  + Price: **device usage** and **data transfer out of AWS**
  + OpsHub: **desktop app** to manage **Snow Family devices**
* AWS Storage Gateway
  + **Hybrid storage service** to **extend on-promises storage to S3**
  + Use case: disaster recovery, backup, restore & tiered storage