AWS

Amazon Web Services

Cloud Computing

- Cloud computing is a style of computing in which scalable and elastic IT-enabled capabilities are delivered as a service using internet technologies.

Introduction

- Amazon Web Services (AWS) is the world's most comprehensive and broadly adopted cloud platform, offering over 175 fully featured services from data centers globally. Millions of customers—including the fastest-growing start-ups, largest enterprises, and leading government agencies—are using AWS to lower costs, become more agile, and innovate faster.

The leading cloud platform.

- **1.Most functionality**
- AWS has significantly more services, and more features within those services, than any other cloud provider–from infrastructure technologies like compute, storage, and databases–to emerging technologies, such as machine learning and artificial intelligence, data lakes and analytics, and Internet of Things.
- AWS also has the deepest functionality within those services.

Contd..

- **2. Largest community of customers and partners**
- AWS has the largest and most dynamic community, with millions of active customers and tens of thousands of partners globally. Customers across virtually every industry and of every size, including start-ups, enterprises, and public sector organizations, are running every imaginable use case on AWS.

Contd..

- **3. Most secured**

- AWS is architected to be the most flexible and secure cloud computing environment available today.
- AWS core infrastructure is built to satisfy the security requirements for the military, global banks, and other high-sensitivity organizations. This is backed by a deep set of cloud security tools, with 230 security, compliance, and governance services and features. AWS supports 90 security standards and compliance certifications, and all 117 AWS services that store customer data offer the ability to encrypt that data.

Contd..

- **4. Fastest pace of innovation**

- With AWS, you can leverage the latest technologies to experiment and innovate more quickly. AWS is continually accelerating pace of innovation to invent entirely new technologies you can use to transform your business.
- For example, in 2014, AWS pioneered the serverless computing space with the launch of AWS Lambda, which lets developers run their code without provisioning or managing servers.
- AWS built Amazon SageMaker, a fully managed machine learning service that empowers everyday developers and scientists to use machine learning–without any previous experience.

Contd..

- **5. Most proven operational expertise**
- AWS has unmatched experience, maturity, reliability, security, and performance that you can depend upon for your most important applications. For over 13 years, AWS has been delivering cloud services to millions of customers around the world running a wide variety of use cases.
- AWS has the most operational experience, at greater scale, of any cloud provider.

AWS infrastructure.

24 Launched Regions

Each with multiple Availability Zones (AZ's)

3 Announced Regions

76 Availability Zones

1 Local Zone

For ultralow latency applications

2x More Regions

With multiple AZ's than the next largest cloud provider

245 Countries and Territories Served

97 Direct Connect Locations

216 Points of Presence

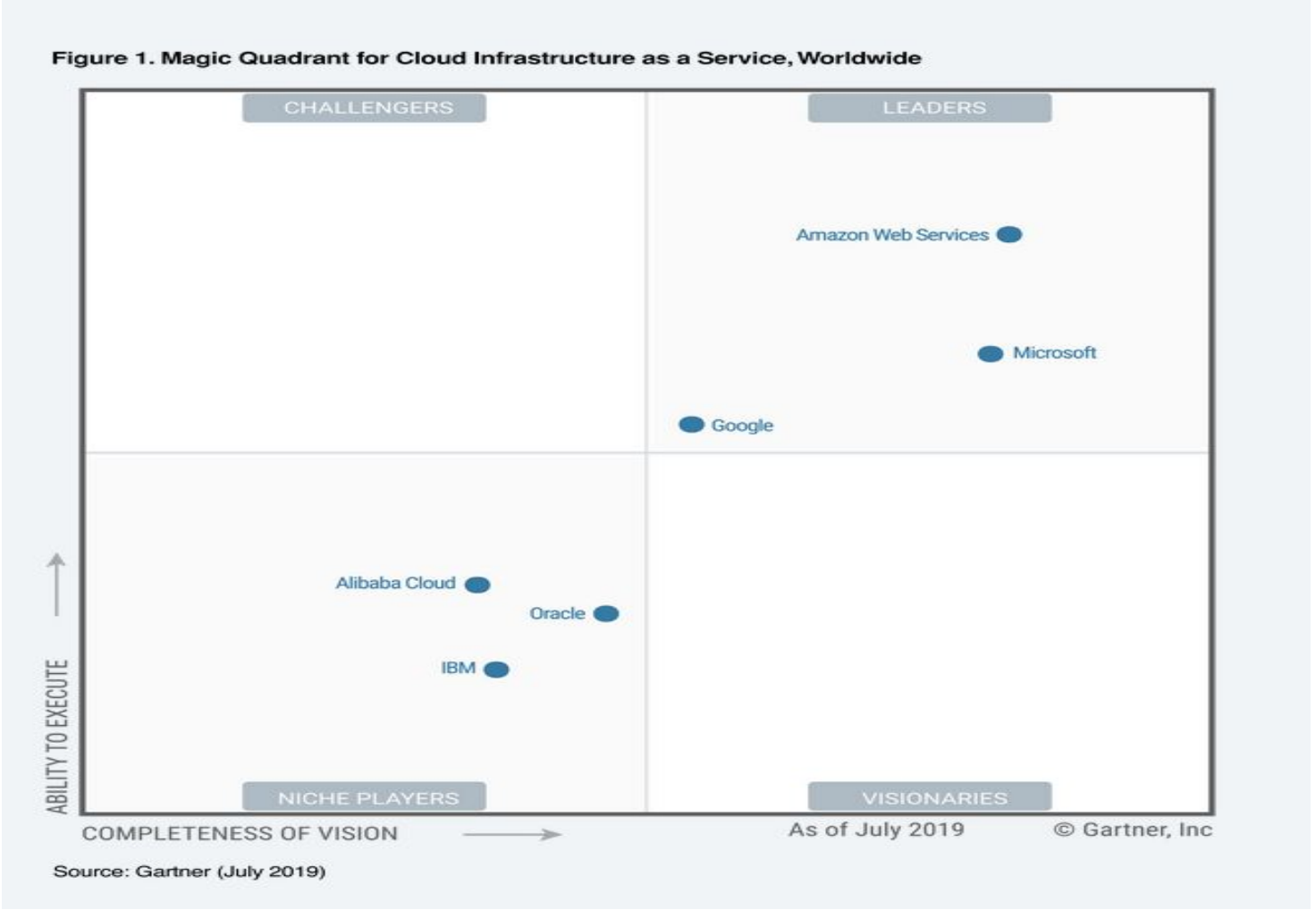
205 Edge Locations and 11 Regional Edge Caches

AWS Global Infrastructure Map

Contd..

- AWS now spans 76 Availability Zones within 24 geographic regions around the world, and has announced plans for nine more Availability Zones and three more AWS Regions in Indonesia, Japan, and Spain.

Magic Quadrant for Cloud Infrastructure as a Service, Worldwide (2019)



Trusted by…..


F1 Insights
POWERED BY aws
Formula One Group Uses Amazon SageMaker to Optimize Racing
Learn more »

Companies using AWS..

- Instagram
- Zoopla
- Smugmug
- Pinterest
- Netflix
- Dropbox
- Etsy
- Talkbox
- Playfish
- Ftopia

History

- 2002- AWS services launched
- 2006- Launched its cloud products
- 2012- Holds first customer event
- 2015- Reveals revenues achieved of $4.6 billion
- 2016- Surpassed $10 billon revenue target
- 2016- Release snowball and snowmobile
- 2018- Offers 120+ cloud services
- 2020- offers 175+ cloud services.

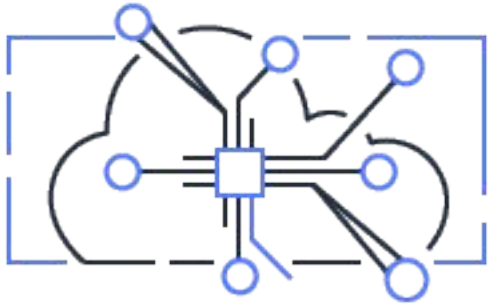
Contd..

- Hosting vs Cloud

What is Cloud Computing?

Cloud computing

- Cloud computing is providing developers and IT departments with the ability to focus on what matters most and avoid undifferentiated work like procurement, maintenance, and capacity planning.
- As cloud computing has grown in popularity, several different models and deployment strategies have emerged to help meet specific needs of different users. Each type of cloud service, and deployment method, provides you with different levels of control, flexibility, and management.



Contd..

- So basically cloud computing is on-demand usage of computing resources.
- In simple terms, using computer resources as service (on demand).

Cloud computing models

- There are three main models for cloud computing. Each model represents a different part of the cloud computing stack.

- Infrastructure as a service
- Platform as a service
- Software as a service

IaaS

- Infrastructure as a Service, sometimes abbreviated as IaaS, contains the basic building blocks for cloud IT and typically provide access to networking features, computers (virtual or on dedicated hardware), and data storage space.
- Infrastructure as a Service provides you with the highest level of flexibility and management control over your IT resources and is most similar to existing IT resources that many IT departments and developers are familiar with today.

PaaS

- Platforms as a service remove the need for organizations to manage the underlying infrastructure (usually hardware and operating systems) and allow you to focus on the deployment and management of your applications.
- This helps you be more efficient as you don’t need to worry about resource procurement, capacity planning, software maintenance, patching, or any of the other undifferentiated heavy lifting involved in running your application.

SaaS

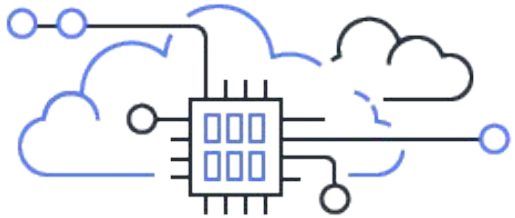
- Software as a Service provides you with a completed product that is run and managed by the service provider. In most cases, people referring to Software as a Service are referring to end-user applications.
- With a SaaS offering you do not have to think about how the service is maintained or how the underlying infrastructure is managed; you only need to think about how you will use that particular piece of software.
- A common example of a SaaS application is web-based email where you can send and receive email without having to manage feature additions to the email product or maintaining the servers and operating systems that the email program is running on.

Cloud computing Deployment Models

- Cloud / Public cloud
- Hybrid cloud
- On-premise / Private cloud

Cloud / public cloud deployment

- A cloud-based application is fully deployed in the cloud and all parts of the application run in the cloud. Applications in the cloud have either been created in the cloud or have been migrated from an existing infrastructure to take advantage of the benefits of cloud computing.
- Cloud-based applications can be built on low-level infrastructure pieces or can use higher level services that provide abstraction from the management, architecting, and scaling requirements of core infrastructure.



Hybrid Deployment

- A hybrid deployment is a way to connect infrastructure and applications between cloud-based resources and existing resources that are not located in the cloud.
- The most common method of hybrid deployment is between the cloud and existing on-premises infrastructure to extend, and grow, an organization's infrastructure into the cloud while connecting cloud resources to internal system.

On-premises

- Deploying resources on-premises, using virtualization and resource management tools, is sometimes called “private cloud”.
- On-premises deployment does not provide many of the benefits of cloud computing but is sometimes sought for its ability to provide dedicated resources.
- In most cases this deployment model is the same as legacy IT infrastructure while using application management and virtualization technologies to try and increase resource utilization.

AWS Platform.

Fundamentals.

- AWS Region
- Each Amazon EC2 Region is designed to be isolated from the other Amazon EC2 Regions. This achieves the greatest possible fault tolerance and stability.
- When you view your resources, you see only the resources that are tied to the Region that you specified. This is because Regions are isolated from each other, and we don't automatically replicate resources across Regions.

Availability Zones

- An Availability Zone (AZ) is one or more discrete data centers with redundant power, networking, and connectivity in an AWS Region.
- Availability Zones allow you to operate production applications and databases that are more highly available, fault tolerant, and scalable than would be possible from a single data center.
- All Availability Zones in an AWS Region are interconnected with high-bandwidth, low-latency networking, over fully redundant, dedicated metro fiber providing high-throughput, low-latency networking between Availability Zones.
- An Availability Zone is represented by a Region code followed by a letter identifier; for example, us-east-1a.

EC2

EC2

- Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides resizable compute capacity in the cloud. It is designed to make web-scale computing easier for developers.

EC2

- Amazon Elastic Compute Cloud (Amazon EC2) provides scalable computing capacity in the Amazon Web Services (AWS) cloud. Using Amazon EC2 eliminates your need to invest in hardware up front, so you can develop and deploy applications faster.
- You can use Amazon EC2 to launch as many or as few virtual servers as you need, configure security and networking, and manage storage. Amazon EC2 enables you to scale up or down to handle changes in requirements or spikes in popularity, reducing your need to forecast traffic.

Contd..

- EC2 is one of most popular of AWS offering

It mainly consists in the capability of :

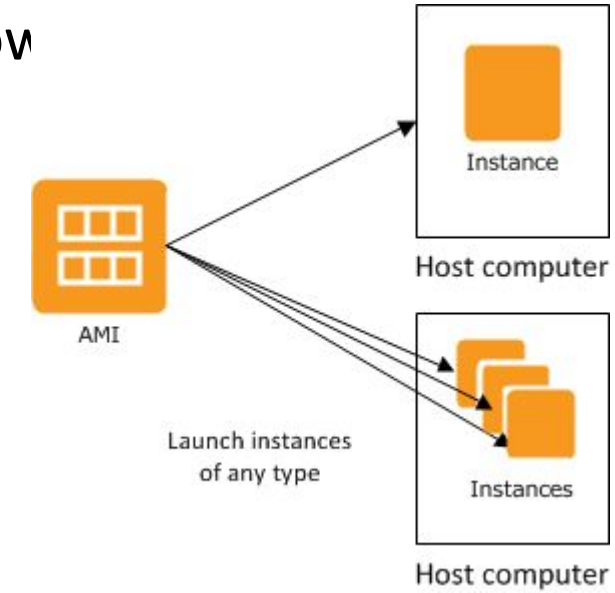
1. Renting virtual machines (EC2)
2. Storing data on virtual drives (EBS)
3. Distributing load across machines (ELB)
4. Scaling the services using an auto-scaling group (ASG)

PCI DSS Compliance

- Amazon Web Services (AWS) is certified as a PCI DSS 3.2 Level 1 Service Provider, the highest level of assessment available. The compliance assessment was conducted by Coalfire Systems Inc., an independent Qualified Security Assessor (QSA).
- Amazon EC2 supports the processing, storage, and transmission of credit card data by a merchant or service provider, and has been validated as being compliant with Payment Card Industry (PCI) Data Security Standard (DSS).
- For the list of AWS services that are PCI DSS compliant, see PCI tab on https://aws.amazon.com/compliance/services-in-scope/

Basics…

- 1. AMI :
- An *Amazon Machine Image (AMI)* is a template that contains a software configuration (for example, an operating system, an application server, and applications). From an AMI, you launch an *instance*, which is a copy of the AMI running as a virtual server in the cloud. You can launch multiple instances of an AMI, as shown in the follow



Which OS supported?

- Amazon EC2 currently supports a variety of operating systems including: Amazon Linux, Ubuntu, Windows Server, Red Hat Enterprise Linux, SUSE Linux Enterprise Server, openSUSE Leap, Fedora, Fedora CoreOS, Debian, CentOS, Gentoo Linux, Oracle Linux, and FreeBSD.
- And expanding…….

Basics..

- 2. Instances :
- An instance is a virtual server in the cloud. Its configuration at launch is a copy of the AMI that you specified when you launched the instance.
- You can launch different types of instances from a single AMI. An *instance type* essentially determines the hardware of the host computer used for your instance. Each instance type offers different compute and memory capabilities.
- Your AWS account has a limit on the number of instances that you can have running.

Security Groups….

- Security Groups are the fundamental of network security in AWS.
- A security group acts as a virtual firewall for your EC2 instances to control incoming and outgoing traffic.
- **Inbound rules** control the incoming traffic to your instance, and **outbound rules** control the outgoing traffic from your instance.
- When you launch an instance, you can specify one or more security groups. If you don't specify a security group, Amazon EC2 uses the default security group.
- If you have requirements that aren't fully met by security groups, you can maintain your own firewall on any of your instances in addition to using security groups.

Security Groups example…

- Web server rules:
- The following inbound rules allow HTTP and HTTPS access from any IP address. If IPv6 is enabled, you can add rules to control inbound HTTP and HTTPS traffic from IPv6 addresses.

Protocol type	Protocol number	Port	Source IP	Notes
TCP	6	80 (HTTP)	0.0.0.0/0	Allows inbound HTTP access from any IPv4 address
TCP	6	443 (HTTPS)	0.0.0.0/0	Allows inbound HTTPS access from any IPv4 address
TCP	6	80 (HTTP)	::/0	Allows inbound HTTP access from any IPv6 address
TCP	6	443 (HTTPS)	::/0	Allows inbound HTTPS access from any IPv6 address

Contd..

Protocol type	Protocol number	Port	Source IP	Notes
TCP	6	80 (HTTP)	0.0.0.0/0	Allows inbound HTTP access from any IPv4 address
TCP	6	443 (HTTPS)	0.0.0.0/0	Allows inbound HTTPS access from any IPv4 address
TCP	6	80 (HTTP)	::/0	Allows inbound HTTP access from any IPv6 address
TCP	6	443 (HTTPS)	::/0	Allows inbound HTTPS access from any IPv6 address

Contd..

- Rules to connect instances from your computer :

Protocol type	Protocol number	Port	Source IP
TCP	6	22 (SSH)	The public IPv4 address of your computer, or a range of IP addresses (in CIDR block notation) in your local network. If your VPC is enabled for IPv6 and your instance has an IPv6 address, you can enter an IPv6 address or range.
TCP	6	3389 (RDP)	The public IPv4 address of your computer, or a range of IP addresses (in CIDR block notation) in your local network. If your VPC is enabled for IPv6 and your instance has an IPv6 address, you can enter an IPv6 address or range.

Contd..

- They regulate:

1. Access to Ports
2. Authorised IP ranges – IPv4 and IPv6
3. Control of inbound network (from other to the instance)
4. Control of outbound network (from the instance to other)

Contd..

- Can be attached to multiple instances
- Locked down to a region
- Does live “outside” the EC2
- It’s good to maintain one separate security group for SSH access
- If your application is not accessible (time out), then it’s a security group issue
- If your application gives a “connection refused“ error, then it’s an application error or it’s not launched
- All inbound traffic is blocked by default
- All outbound traffic is authorised by default

Launching first Instance

- Demo.
- SSH using Linux… (or SSH client on windows)

User data in EC2

- When you launch an instance in Amazon EC2, you have the option of passing user data to the instance that can be used to perform common automated configuration tasks and even run scripts after the instance starts.
- You can pass two types of user data to Amazon EC2: shell scripts and cloud-init directives. You can also pass this data into the launch wizard as plain text, as a file (this is useful for launching instances using the command line tools), or as base64-encoded text (for API calls).

EC2 Instance types

- When you launch an instance, the instance type that you specify determines the hardware of the host computer used for your instance. Each instance type offers different compute, memory, and storage capabilities and are grouped in instance families based on these capabilities. Select an instance type based on the requirements of the application or software that you plan to run on your instance.
- Amazon EC2 provides each instance with a consistent and predictable amount of CPU capacity, regardless of its underlying hardware.
- https://aws.amazon.com/ec2/instance-types/

Instance Purchase types

- **On-Demand Instances** – Pay, by the second, for the instances that you launch.
- **Savings Plans** – Reduce your Amazon EC2 costs by making a commitment to a consistent amount of usage, in USD per hour, for a term of 1 or 3 years.
- **Reserved Instances** – Reduce your Amazon EC2 costs by making a commitment to a consistent instance configuration, including instance type and Region, for a term of 1 or 3 years.
- **Scheduled Instances** – Purchase instances that are always available on the specified recurring schedule, for a one-year term.

Instance Purchase types

- **Spot Instances** – Request unused EC2 instances, which can reduce your Amazon EC2 costs significantly.
- **Dedicated Hosts** – Pay for a physical host that is fully dedicated to running your instances, and bring your existing per-socket, per-core, or per-VM software licenses to reduce costs.
- **Dedicated Instances** – Pay, by the hour, for instances that run on single-tenant hardware.
- **Capacity Reservations** – Reserve capacity for your EC2 instances in a specific Availability Zone for any duration.

Placement Groups

- When you launch a new EC2 instance, the EC2 service attempts to place the instance in such a way that all of your instances are spread out across underlying hardware to minimize correlated failures. You can use placement groups to influence the placement of a group of interdependent instances to meet the needs of your workload.
- Following are placement strategies offered by AWS
- 1. Cluster
- 2. Spread
- 3. Partition.

Contd..

- **Cluster** – packs instances close together inside an Availability Zone. This strategy enables workloads to achieve the low-latency network performance necessary for tightly-coupled node-to-node communication that is typical of HPC applications.

- **Partition** – spreads your instances across logical partitions such that groups of instances in one partition do not share the underlying hardware with groups of instances in different partitions. This strategy is typically used by large distributed and replicated workloads, such as Hadoop, Cassandra, and Kafka.

- **Spread** – strictly places a small group of instances across distinct underlying hardware to reduce correlated failures.

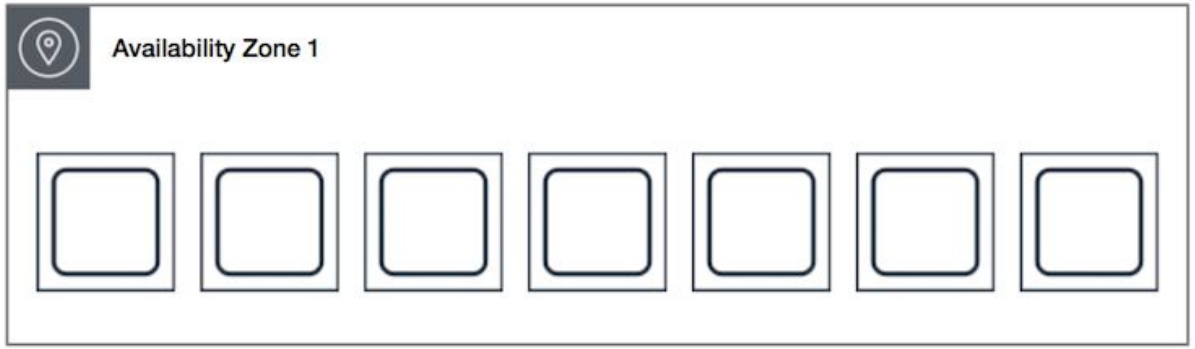
- There is no charge for creating a placement group.

Cluster placement group

- A cluster placement group is a logical grouping of instances within a single Availability Zone. Instances in the same cluster placement group enjoy a higher per-flow throughput limit of up to 10 Gbps for TCP/IP traffic and are placed in the same high-bisection bandwidth segment of the network.
- Cluster placement groups are recommended for applications that benefit from low network latency, high network throughput, or both. They are also recommended when the majority of the network traffic is between the instances in the group

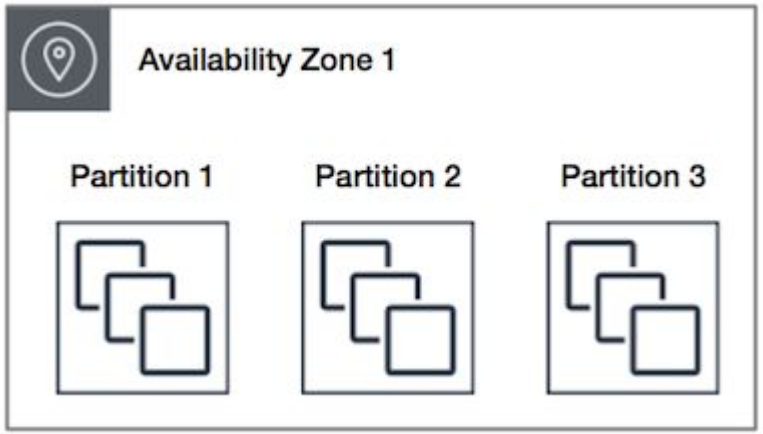
Spread placement groups

- A spread placement group is a group of instances that are each placed on distinct racks, with each rack having its own network and power source.
- The following image shows seven instances in a single Availability Zone that are placed into a spread placement group. The seven instances are placed on seven different racks.
- A spread placement group can span multiple Availability Zones in the same Region. You can have a maximum of seven running instances per Availability Zone per group.



Partition placement groups

- Partition placement groups help reduce the likelihood of correlated hardware failures for your application. When using partition placement groups, Amazon EC2 divides each group into logical segments called partitions.
- Amazon EC2 ensures that each partition within a placement group has its own set of racks. Each rack has its own network and power source. No two partitions within a placement group share the same racks, allowing you to isolate the impact of hardware failure within your application.



Elastic Load balancer

Introduction

- Elastic Load Balancing automatically distributes incoming application traffic across multiple targets, such as Amazon EC2 instances, containers, IP addresses, and Lambda functions.
- It can handle the varying load of your application traffic in a single Availability Zone or across multiple Availability Zones.
- Elastic Load Balancing offers three types of load balancers that all feature the high availability, automatic scaling, and robust security necessary to make your applications fault tolerant.

Application Load Balancer

- Application Load Balancer is best suited for load balancing of HTTP and HTTPS traffic and provides advanced request routing targeted at the delivery of modern application architectures, including microservices and containers.
- Operating at the individual request level (Layer 7), Application Load Balancer routes traffic to targets within Amazon Virtual Private Cloud (Amazon VPC) based on the content of the request.

Network Load balancer

- Network Load Balancer is best suited for load balancing of Transmission Control Protocol (TCP), User Datagram Protocol (UDP) and Transport Layer Security (TLS) traffic where extreme performance is required.
- Operating at the connection level (Layer 4), Network Load Balancer routes traffic to targets within Amazon Virtual Private Cloud (Amazon VPC) and is capable of handling millions of requests per second while maintaining ultra-low latencies. Network Load Balancer is also optimized to handle sudden and volatile traffic patterns

Classic load balancer

- Classic Load Balancer provides basic load balancing across multiple Amazon EC2 instances and operates at both the request level and connection level. Classic Load Balancer is intended for applications that were built within the EC2-Classic network.

Why Load Balancer

- Spread load across multiple instances
- Expose a single point of access (DNS) to your application
- Handle failures of instances
- Do regular health checks to your instances
- Provide SSL termination (HTTPS) for your websites
- Enforce stickiness with cookies
- High availability across zones
- Separate public traffic from private traffic

Auto Scaling Groups

Introduction

- An *Auto Scaling group* contains a collection of Amazon EC2 instances that are treated as a logical grouping for the purposes of automatic scaling and management.
- An Auto Scaling group also enables you to use Amazon EC2 Auto Scaling features such as health check replacements and scaling policies.
- Both maintaining the number of instances in an Auto Scaling group and automatic scaling are the core functionality of the Amazon EC2 Auto Scaling service.

Contd..

- The size of an Auto Scaling group depends on the number of instances that you set as the desired capacity. You can adjust its size to meet demand, either manually or by using automatic scaling.
- Two types of launch methods available
- 1. Launch Template.
- 2. Launch Configuration.

Contd.

- Launch template is similar to launch configuration which usually Auto Scaling group uses to launch EC2 instances. However, defining a launch template instead of a launch configuration allows you to have multiple versions of a template.
- launch configurations are used with Auto Scaling Groups. While launch templates are used when you launch an instance using the aws EC2 console, an AWS SDK, or a command line tool.

Why?

- Vertical Scaling: Increase instance size (= scale up / down)
  - • From: t2.nano - 0.5G of RAM, 1 vCPU
  - • To: u-12tb1.metal – 12.3 TB of RAM, 448 vCPUs (2018)
- Horizontal Scaling: Increase number of instances (= scale out / in)
  - • Auto Scaling Group
  - • Load Balancer
- • High Availability: Run instances for the same application across multi AZ
  - • Auto Scaling Group multi AZ
  - • Load Balancer multi AZ

EBS

Introduction

- Amazon Elastic Block Store (EBS) is an easy to use, high performance block storage service designed for use with Amazon Elastic Compute Cloud (EC2) for both throughput and transaction intensive workloads at any scale. A broad range of workloads, such as relational and non-relational databases, enterprise applications, containerized applications, big data analytics engines, file systems, and media workflows are widely deployed on Amazon EBS.

Contd..

- You can choose from four different volume types to balance optimal price and performance. You can achieve single digit-millisecond latency for high performance database workloads such as SAP HANA or gigabyte per second throughput for large, sequential workloads such as Hadoop. You can change volume types, tune performance, or increase volume size without disrupting your critical applications, so you have cost-effective storage when you need it.

Contd..

- Amazon EBS provides the following volume types, which differ in performance characteristics and price, so that you can tailor your storage performance and cost to the needs of your applications.
- The volumes types fall into two categories:

1. SSD-backed volumes optimized for transactional workloads involving frequent read/write operations with small I/O size, where the dominant performance attribute is IOPS
2. HDD-backed volumes optimized for large streaming workloads where throughput (measured in MiB/s) is a better performance measure than IOPS

	Solid-state drives (SSD)		Hard disk drives (HDD)	
Volume type	General Purpose SSD (gp2)	Provisioned IOPS SSD (io1)	Throughput Optimized HDD (st1)	Cold HDD (sc1)
Description	General purpose SSD volume that balances price and performance for a wide variety of workloads	Highest-performance SSD volume for mission-critical low-latency or high-throughput workloads	Low-cost HDD volume designed for frequently accessed, throughput-intensive workloads	Lowest cost HDD volume designed for less frequently accessed workloads
Use cases	•Recommended for most workloads •System boot volumes •Virtual desktops •Low-latency interactive apps •Development and test environments	•Critical business applications that require sustained IOPS performance, or more than 16,000 IOPS or 250 MiB/s of throughput per volume •Large database workloads, such as: •MongoDB •Cassandra •Microsoft SQL Server •MySQL •PostgreSQL •Oracle	•Streaming workloads requiring consistent, fast throughput at a low price •Big data •Data warehouses •Log processing •Cannot be a boot volume	•Throughput-oriented storage for large volumes of data that is infrequently accessed •Scenarios where the lowest storage cost is important •Cannot be a boot volume
API name	gp2	io1	st1	sc1
Volume size	1 GiB - 16 TiB	4 GiB - 16 TiB	500 GiB - 16 TiB	500 GiB - 16 TiB
Max IOPS per volume	16,000 (16 KiB I/O) *	64,000 (16 KiB I/O) †	500 (1 MiB I/O)	250 (1 MiB I/O)
Max throughput per volume	250 MiB/s *	1,000 MiB/s †	500 MiB/s	250 MiB/s
Max IOPS per instance ††	80,000	80,000	80,000	80,000
Max throughput per instance ††	2,375 MB/s	2,375 MB/s	2,375 MB/s	2,375 MB/s
Dominant performance attribute	IOPS	IOPS	MiB/s	MiB/s

EBS Snapshots

- Incremental – only backup changed blocks
- EBS backups use IO and you shouldn't run them while your application is handling a lot of traffic
- Snapshots will be stored in S3 (but you won't directly see them)
- Not necessary to detach volume to do snapshot, but recommended
- Max 100,000 snapshots
- Can copy snapshots across AZ or Region
- Can make Image (AMI) from Snapshot
- EBS volumes restored by snapshots need to be pre-warmed (using fio or dd command to read the entire volume)

EBS Migration

- EBS Volumes are only locked to a specific AZ
- To migrate it to a different AZ (or region):

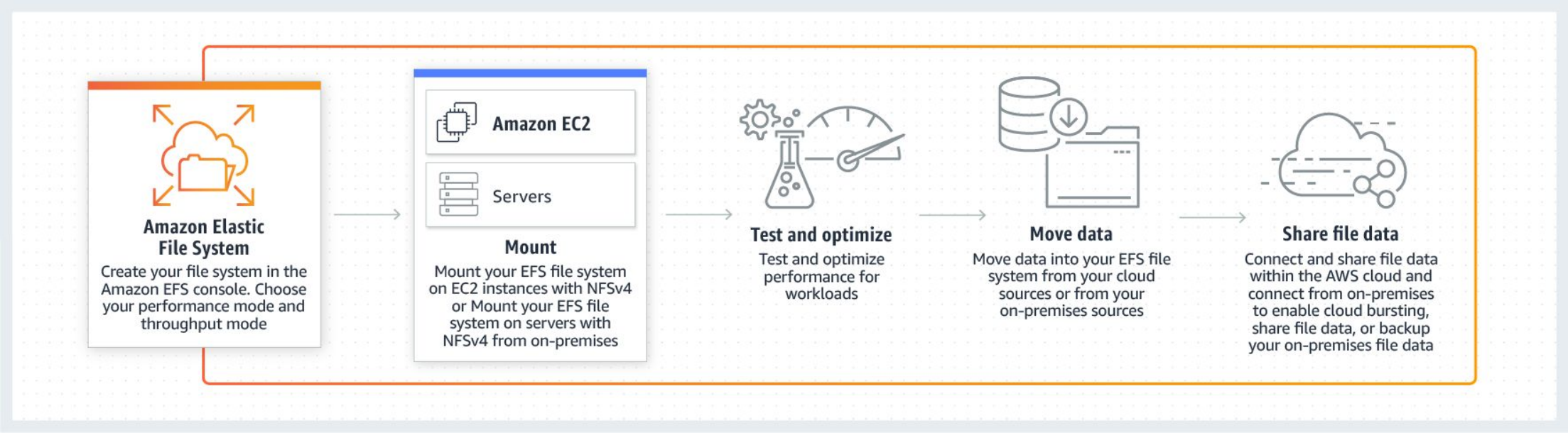
1. • Snapshot the volume
2. • (optional) Copy the volume to a different region
3. • Create a volume from the snapshot in the AZ of your choice

EFS

Introduction

- Amazon EFS provides scalable file storage for use with Amazon EC2. You can create an EFS file system and configure your instances to mount the file system. **You can use an EFS file system as a common data source for workloads and applications running on multiple instances in multiple AZs**

Contd..


Amazon Elastic File System
Create your file system in the Amazon EFS console. Choose your performance mode and throughput mode
Amazon EC2
Servers
Mount
Mount your EFS file system on EC2 instances with NFSv4 or Mount your EFS file system on servers with NFSv4 from on-premises
Test and optimize
Test and optimize performance for workloads
Move data
Move data into your EFS file system from your cloud sources or from your on-premises sources
Share file data
Connect and share file data within the AWS cloud and connect from on-premises to enable cloud bursting, share file data, or backup your on-premises file data

Contd..

- Managed NFS (network file system) that can be mounted on many EC2
- EFS works with EC2 instances in multi-AZ
- Highly available, scalable, expensive (3x gp2), pay per use/provisioned.

Relational Database Service (RDS)

Introduction

- Amazon Relational Database Service (Amazon RDS) makes it easy to set up, operate, and scale a relational database in the cloud. It provides cost-efficient and resizable capacity while automating time-consuming administration tasks such as hardware provisioning, database setup, patching and backups. It frees you to focus on your applications so you can give them the fast performance, high availability, security and compatibility they need.
- It’s a managed DB service for DB use SQL as a query language

Available Databases

- Amazon RDS is available on several database instance types - optimized for memory, performance or I/O - and provides you with six familiar database engines to choose from, including **Amazon Aurora, PostgreSQL, MySQL, MariaDB, Oracle Database, and SQL Server**.
- You can use the AWS Database Migration Service to easily migrate or replicate your existing databases to Amazon RDS.

Why to use RDS over EC2 ?

- Managed service:
- Continuous backups and restore to specific timestamp (Point in Time Restore)!
- Monitoring dashboards
- Read replicas for improved read performance
- Multi AZ setup for DR (Disaster Recovery)
- Maintenance windows for upgrades
- Scaling capability (vertical and horizontal)
- you can’t SSH into your instances (You need not actually).

Amazon Aurora

- Aurora is a proprietary technology from AWS (not open sourced)
- Postgres and MySQL are both supported as Aurora DB (that means your drivers will work as if Aurora was a Postgres or MySQL database)
- Aurora is “AWS cloud optimized” and claims 5x performance improvement over MySQL on RDS, over 3x the performance of Postgres on RDS
- Aurora storage automatically grows in increments of 10GB, up to 64 TB.
- Aurora can have 15 replicas while MySQL has 5, and the replication process is faster (sub 10 ms replica lag)
- Failover in Aurora is instantaneous. It’s HA native.
- Aurora costs more than RDS (20% more) – but is more efficient

AWS ElastiCache

Introduction

- ElastiCache is to get managed Redis or Memcached
- Caches are in-memory databases with really high performance, low latency
- Helps reduce load off of databases for read intensive workloads
- Helps make your application stateless
- Write Scaling using sharding
- Read Scaling using Read Replicas
- Multi AZ with Failover Capability
- AWS takes care of OS maintenance , optimizations, setup, configuration, monitoring, failure recovery and backups

Redis

Redis is an in-memory key-value store

- Super low latency (sub ms)
- Cache survive reboots by default (it's called persistence)
- Easy to host
- User sessions
- Leaderboard (for gaming)
- Distributed states
- Relieve pressure on databases (such as RDS)

Multi AZ with Automatic Failover for disaster recovery if you don't want to lose your cache data

Support for Read Replicas

Memcached

- Memcached is an in-memory object store
- **Cache doesn’t survive reboots**
- Use cases:
  - Quick retrieval of objects from memory
  - Cache often accessed objects
- Overall, Redis has largely grown in popularity and has better feature sets than Memcached.
- Redis is better for caching needs.

Elastic BeanStalk

Introduction

- AWS Elastic Beanstalk is an easy-to-use service for deploying and scaling web applications and services developed with Java, .NET, PHP, Node.js, Python, Ruby, Go, and Docker on familiar servers such as Apache, Nginx, Passenger, and IIS.

- You can simply upload your code and Elastic Beanstalk automatically handles the deployment, from capacity provisioning, load balancing, auto-scaling to application health monitoring. At the same time, you retain full control over the AWS resources powering your application and can access the underlying resources at any time.

- There is no additional charge for Elastic Beanstalk - you pay only for the AWS resources needed to store and run your applications.

Benefits

- Managed WebApp service.
- Fast and simple to begin.
- Developer productivity.
- Impossible to outgrow.
- Complete resource control.


aws

AWS Simple Storage Service

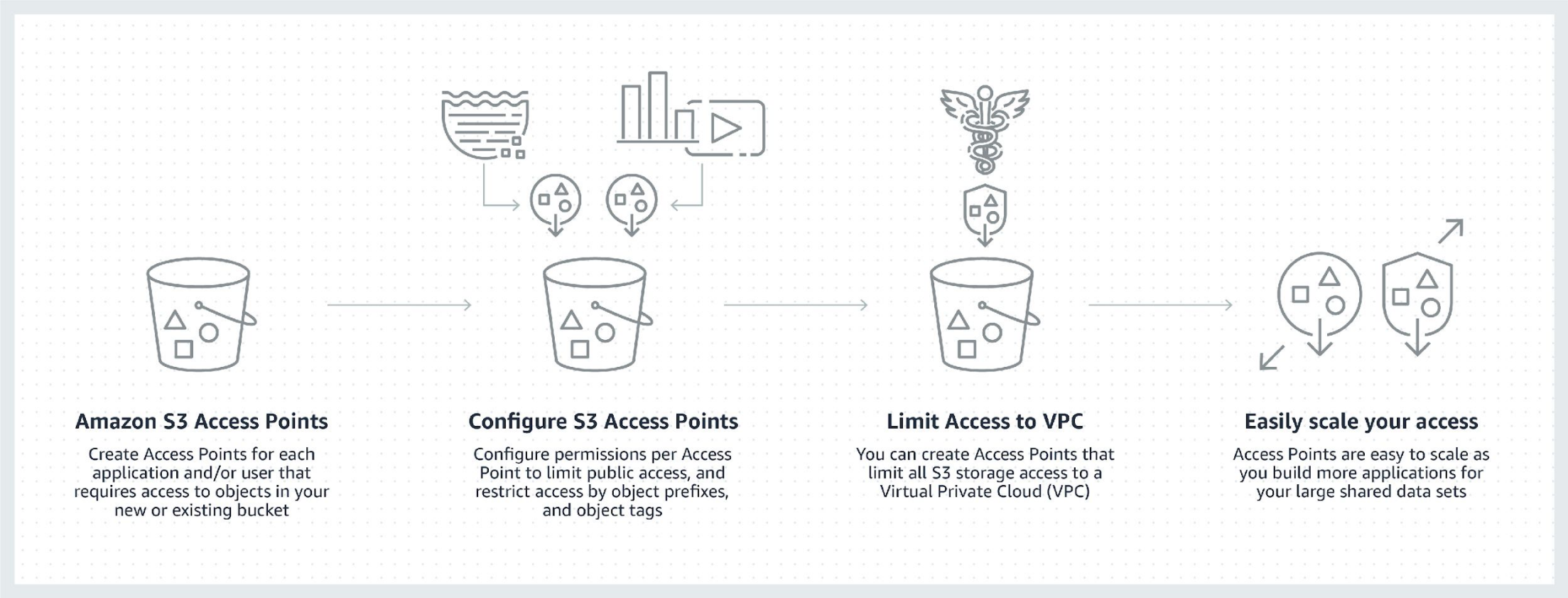
Introduction

- Amazon Simple Storage Service (Amazon S3) is an object storage service that offers industry-leading scalability, data availability, security, and performance. This means customers of all sizes and industries can use it to store and protect any amount of data for a range of use cases, such as websites, mobile applications, backup and restore, archive, enterprise applications, IoT devices, and big data analytics.
- Amazon S3 provides easy-to-use management features so you can organize your data and configure finely-tuned access controls to meet your specific business, organizational, and compliance requirements. Amazon S3 is designed for 99.999999999% (11 9's) of durability, and stores data for millions of applications for companies all around the world.

Introduction

- Amazon S3 is one of the main building blocks of AWS
- It’s advertised as ”infinitely scaling” storage
- It’s widely popular and deserves its own section
- Many websites use AWS S3 as a backbone
- Many AWS services uses AWS S3 as an integration as well

Introduction



Use cases

- **Backup and restore**
- **Disaster recovery (DR)**
- **Archive**
- **Data lakes and big data analytics**
- **Hybrid cloud storage**
- **Cloud-native applications**

Buckets

- Amazon S3 allows people to store objects (files) in “buckets” (directories)
- Buckets must have a globally unique name
- Buckets are defined at the region level
- Naming convention
  - No uppercase
  - No underscore
  - 3-63 characters long
  - Not an IP
  - Must start with lowercase letter or number

Objects

- Objects (files) have a Key. The key is the FULL path:
- • <my_bucket>/my_file.txt
- • There’s no concept of “directories” within buckets.(just a key with /)
- Object Values are the content of the body:
- • Max Size is 5TB
- Metadata (list of text key / value pairs – system or user metadata)
- • Tags (Unicode key / value pair – up to 10) – useful for security / lifecycle
- • Version ID (if versioning is enabled)

AWS S3 Versioning

- You can version your files in AWS S3
- It is enabled at the bucket level
- Same key overwrite will increment the “version”: 1, 2, 3….
- It is best practice to version your buckets
- Protect against unintended deletes (ability to restore a version)
- Easy roll back to previous version
- Any file that is not versioned prior to enabling versioning will have version “null”

S3 Encryption

S3 Encryption for Objects

- There are 4 methods of encrypting objects in S3

1. SSE-S3: encrypts S3 objects using keys handled & managed by AWS
2. SSE-KMS: leverage AWS Key Management Service to manage encryption keys
3. SSE-C: when you want to manage your own encryption keys
4. Client Side Encryption

Storage classes

- S3 Standard
- Amazon S3 Intelligent-Tiering
- Amazon S3 Standard-Infrequent Access
- Amazon S3 One Zone-Infrequent Access
- Amazon S3 Glacier
- Amazon S3 Glacier Deep Archive

S3 Standard

- Low latency and high throughput performance
- Designed for durability of 99.999999999% of objects across multiple Availability Zones
- Resilient against events that impact an entire Availability Zone
- Designed for 99.99% availability over a given year
- Backed with the Amazon S3 Service Level Agreement for availability
- Supports SSL for data in transit and encryption of data at rest
- S3 Lifecycle management for automatic migration of objects to other S3 Storage Classes

Contd..

- S3 Standard offers high durability, availability, and performance object storage for frequently accessed data. Because it delivers low latency and high throughput, S3 Standard is appropriate for a wide variety of use cases, including cloud applications, dynamic websites, content distribution, mobile and gaming applications, and big data analytics.
- S3 Storage Classes can be configured at the object level and a single bucket can contain objects stored across S3 Standard, S3 Intelligent-Tiering, S3 Standard-IA, and S3 One Zone-IA. You can also use S3 Lifecycle policies to automatically transition objects between storage classes without any application changes.

Amazon S3 Intelligent-Tiering

1. Same low latency and high throughput performance of S3 Standard
2. Small monthly monitoring and auto-tiering fee
3. Automatically moves objects between two access tiers based on changing access patterns
4. Designed for durability of 99.999999999% of objects across multiple Availability Zones
5. Resilient against events that impact an entire Availability Zone
6. Designed for 99.9% availability over a given year
7. Backed with the Amazon S3 Service Level Agreement for availability
8. Supports SSL for data in transit and encryption of data at rest
9. S3 Lifecycle management for automatic migration of objects to other S3 Storage Classes

Contd..

- The S3 Intelligent-Tiering storage class is designed to optimize costs by automatically moving data to the most cost-effective access tier, without performance impact or operational overhead.
- It works by storing objects in two access tiers:
  - one tier that is optimized for frequent access and
  - another lower-cost tier that is optimized for infrequent access.

Contd..

- For a small monthly monitoring and automation fee per object, Amazon S3 monitors access patterns of the objects in S3 Intelligent-Tiering, and moves the ones that have not been accessed for 30 consecutive days to the infrequent access tier.
- If an object in the infrequent access tier is accessed, it is automatically moved back to the frequent access tier. There are no retrieval fees when using the S3 Intelligent-Tiering storage class, and no additional tiering fees when objects are moved between access tiers. It is the ideal storage class for long-lived data with access patterns that are unknown or unpredictable.

Amazon S3 Standard-Infrequent Access

- Same low latency and high throughput performance of S3 Standard
- Designed for durability of 99.999999999% of objects across multiple Availability Zones
- Resilient against events that impact an entire Availability Zone
- Data is resilient in the event of one entire Availability Zone destruction
- Designed for 99.9% availability over a given year
- Backed with the Amazon S3 Service Level Agreement for availability
- Supports SSL for data in transit and encryption of data at rest
- S3 Lifecycle management for automatic migration of objects to other S3 Storage Classes

Contd.

- S3 Standard-IA is for data that is accessed less frequently, but requires rapid access when needed. S3 Standard-IA offers the high durability, high throughput, and low latency of S3 Standard, with a **low per GB storage price and per GB retrieval fee**. This combination of low cost and high performance make S3 Standard-IA ideal for long-term storage, backups, and as a data store for disaster recovery files.

Amazon S3 One Zone-Infrequent Access

- Same low latency and high throughput performance of S3 Standard
- Designed for durability of 99.999999999% of objects in a single Availability Zone†
- Designed for 99.5% availability over a given year
- Backed with the Amazon S3 Service Level Agreement for availability
- Supports SSL for data in transit and encryption of data at rest
- S3 Lifecycle management for automatic migration of objects to other S3 Storage Classes

- † Because S3 One Zone-IA stores data in a single AWS Availability Zone, data stored in this storage class will be lost in the event of Availability Zone destruction.

Contd.

- S3 One Zone-IA is for data that is accessed less frequently, but requires rapid access when needed. Unlike other S3 Storage Classes which store data in a minimum of three Availability Zones (AZs), S3 One Zone-IA stores data in a single AZ and costs 20% less than S3 Standard-IA.
- S3 One Zone-IA is ideal for customers who want a lower-cost option for infrequently accessed data but do not require the availability and resilience of S3 Standard or S3 Standard-IA. It’s a good choice for storing secondary backup copies of on-premises data or easily re-creatable data. You can also use it as cost-effective storage for data that is replicated from another AWS Region using S3 Cross-Region Replication.

Archive

- Designed for durability of 99.999999999% of objects across multiple Availability Zones
- Data is resilient in the event of one entire Availability Zone destruction
- Supports SSL for data in transit and encryption of data at rest
- Low-cost design is ideal for long-term archive
- Configurable retrieval times, from minutes to hours
- S3 PUT API for direct uploads to S3 Glacier, and S3 Lifecycle management for automatic migration of objects

S3 Glacier

- S3 Glacier is a secure, durable, and low-cost storage class for data archiving. You can reliably store any amount of data at costs that are competitive with or cheaper than on-premises solutions.
- To keep costs low yet suitable for varying needs, S3 Glacier provides three retrieval options that range from a few minutes to hours. You can upload objects directly to S3 Glacier, or use S3 Lifecycle policies to transfer data between any of the S3 Storage Classes for active data (S3 Standard, S3 Intelligent-Tiering, S3 Standard-IA, and S3 One Zone-IA) and S3 Glacier

Deep Archive

- Designed for durability of 99.999999999% of objects across multiple Availability Zones
- Lowest cost storage class designed for long-term retention of data that will be retained for 7-10 years
- Ideal alternative to magnetic tape libraries
- Retrieval time within 12 hours
- S3 PUT API for direct uploads to S3 Glacier Deep Archive, and S3 Lifecycle management for automatic migration of objects

Glacier deep archive

- S3 Glacier Deep Archive is Amazon S3's lowest-cost storage class and supports long-term retention and digital preservation for data that may be accessed once or twice in a year. It is designed for customers — particularly those in highly-regulated industries, such as the Financial Services, Healthcare, and Public Sectors — that retain data sets for 7-10 years or longer to meet regulatory compliance requirements.
- S3 Glacier Deep Archive complements Amazon S3 Glacier, which is ideal for archives where data is regularly retrieved and some of the data may be needed in minutes. All objects stored in S3 Glacier Deep Archive are replicated and stored across at least three geographically-dispersed Availability Zones, protected by 99.999999999% of durability, and can be restored within 12 hours.

S3 Versioning

- Versioning enables you to keep multiple versions of an object in one bucket.
- It is enabled at the bucket level
- Same key overwrite will increment the “version”: 1, 2, 3….
- It is best practice to version your buckets
- Protect against unintended deletes (ability to restore a version)
- Easy roll back to previous version
- Any file that is not versioned prior to enabling versioning will have version “null”

S3 Encryption

- There are 4 methods of encrypting objects in S3
  - SSE-S3: encrypts S3 objects using keys handled & managed by AWS
  - SSE-KMS: leverage AWS Key Management Service to manage encryption keys
  - SSE-C: when you want to manage your own encryption keys
  - Client Side Encryption

SSE-S3

- SSE-S3: encryption using keys handled & managed by AWS S3
- Object is encrypted server side
- AES-256 encryption type
- Must set header: “x-amz-server-side-encryption": "AES256"

SSE-KMS

- SSE-KMS: encryption using keys handled & managed by KMS
- KMS Advantages: user control + audit trail
- Object is encrypted server side
- Must set header: “x-amz-server-side-encryption": ”aws:kms"

SSE-C

- SSE-C: server-side encryption using data keys fully managed by the customer outside of AWS
- Amazon S3 does not store the encryption key you provide
- HTTPS must be used
- Encryption key must provided in HTTP headers, for every HTTP request made

Client Side Encryption

- Client library such as the Amazon S3 Encryption Client
- Clients must encrypt data themselves before sending to S3
- Clients must decrypt data themselves when retrieving from S3
- Customer fully manages the keys and encryption cycle

S3 Security

- User based
  - IAM policies - which API calls should be allowed for a specific user from IAM console
- Resource Based
  - Bucket Policies - bucket wide rules from the S3 console
  - Object Access Control List (ACL) – finer grain
  - Bucket Access Control List (ACL) – less common

S3 Bucket Policies

- JSON based policies
  - Resources: buckets and objects
  - Actions: Set of API to Allow or Deny
  - Effect: Allow / Deny
  - Principal: The account or user to apply the policy to
- Use S3 bucket for policy to:
  - Grant public access to the bucket
  - Force objects to be encrypted at upload
  - Grant access to another account (Cross Account)

S3 Websites

- S3 can host static websites and have them accessible on the www
- The website URL will be:
- <bucket-name>.s3-website-<AWS-region>.amazonaws.com
- If you get a 403 (Forbidden) error, make sure the bucket policy allows public reads!

S3 CORS

- If you request data from another S3 bucket, you need to enable CORS
- Cross Origin Resource Sharing allows you to limit the number of websites that can request your files in S3 (and limit your costs)

AWS IAM

Introduction

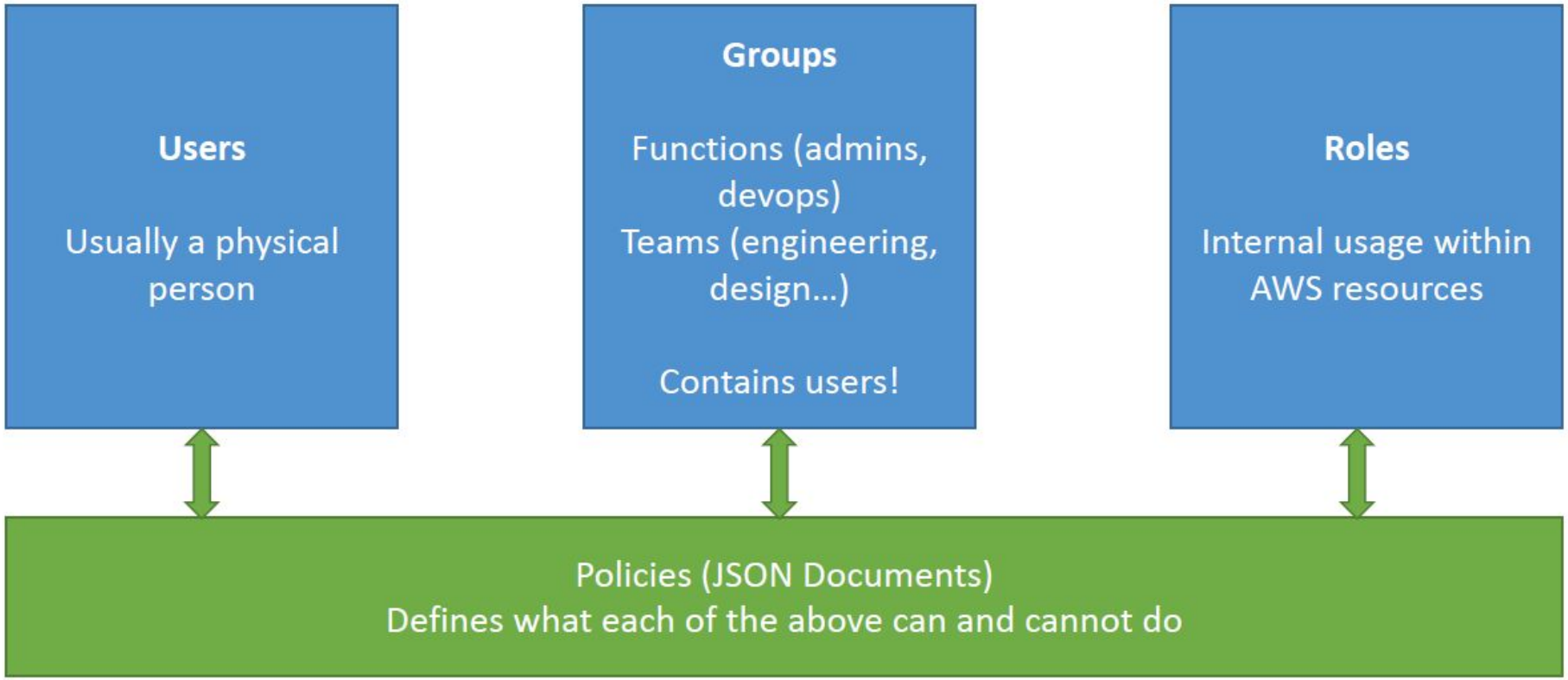
- IAM (Identity and Access Management)
- Your whole AWS security is there:
- Users
- Groups
- Roles
- Root account should never be used (and shared)
- Users must be created with proper permissions
- IAM is at the center of AWS
- Policies are written in JSON (JavaScript Object Notation)

Contd.

- AWS Identity and Access Management (IAM) enables you to manage access to AWS services and resources securely. Using IAM, you can create and manage AWS users and groups, and use permissions to allow and deny their access to AWS resources.

- IAM is a feature of your AWS account offered at no additional charge. You will be charged only for use of other AWS services by your users.

Use


Users
Usually a physical person
Groups
Functions (admins, devops)
Teams (engineering, design…)
Contains users!
Roles
Internal usage within AWS resources
Policies (JSON Documents)
Defines what each of the above can and cannot do

IAM

- AWS IAM allows you to:

1. Manage IAM users and their access
2. Manage IAM roles and their permissions
3. Manage federated users and their permissions

IAM Federation

- Big enterprises usually integrate their own repository of users with IAM
- This way, one can login into AWS using their company credentials
- Identity Federation uses the SAML standard (Active Directory)

AWS Route53

Introduction

- Route53 is a Managed DNS (Domain Name System)
- DNS is a collection of rules and records which helps clients understand how to reach a server through URLs.
- In AWS, the most common records are:
  - A: URL to IPv4
  - AAAA: URL to IPv6
  - CNAME: URL to URL
  - Alias: URL to **AWS resource**.(Public Resource/Bucket details….)

Contd.

- Amazon Route 53 is a highly available and scalable cloud Domain Name System (DNS) web service. It is designed to give developers and businesses an extremely reliable and cost effective way to route end users to Internet applications by translating names like www.example.com into the numeric IP addresses like 192.0.2.1 that computers use to connect to each other. Amazon Route 53 is fully compliant with IPv6 as well.
- Amazon Route 53 effectively connects user requests to infrastructure running in AWS – such as Amazon EC2 instances, Elastic Load Balancing load balancers, or Amazon S3 buckets – and can also be used to route users to infrastructure outside of AWS.

What we can do…

1. **Register domain names**
2. **Route internet traffic to the resources for your domain**
3. **Check the health of your resources**

4. You pay $0.50 per month per hosted zone

Health Check

- Have X health checks failed => unhealthy (default 3)
- After X health checks passed => health (default 3)
- Default Health Check Interval: 30s (can set to 10s – higher cost)
- About 15 health checkers will check the endpoint health
- => one request every 2 seconds on average
- Can have HTTP, TCP and HTTPS health checks (no SSL verification)
- Possibility of integrating the health check with CloudWatch

Routing policy

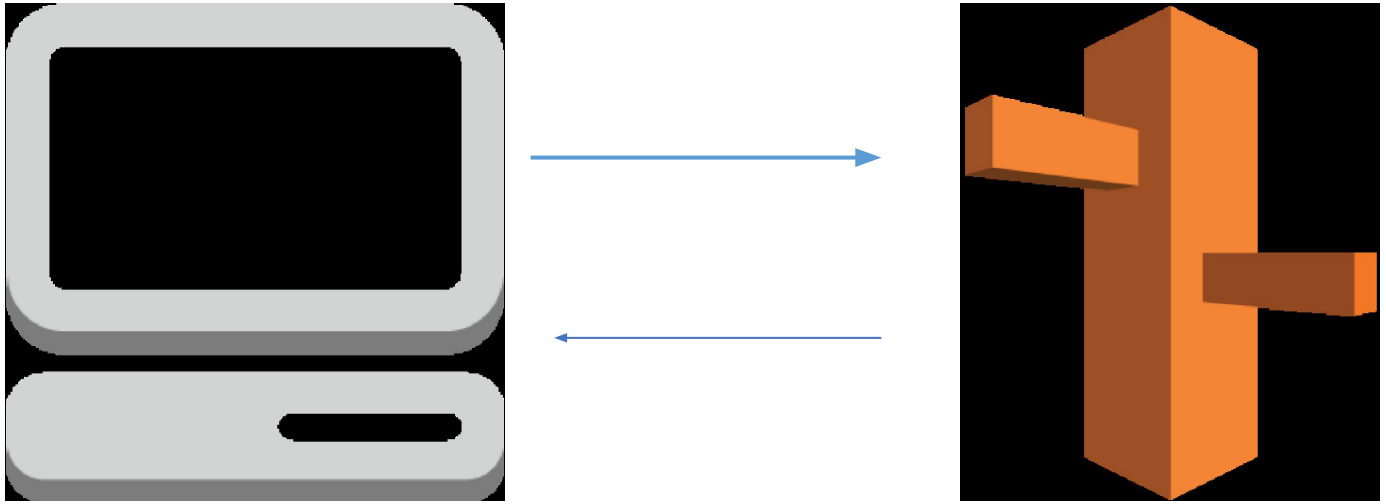
- **Simple routing policy** – Use to route internet traffic to a single resource that performs a given function for your domain, for example, a web server that serves content for the example.com website.
- **Failover routing policy** – Use when you want to configure active-passive failover.
- **Geolocation routing policy** – Use when you want to route internet traffic to your resources based on the location of your users.
- **Geoproximity routing policy** – Use when you want to route traffic based on the location of your resources and, optionally, shift traffic from resources in one location to resources in another.

Routing policy

- **Latency routing policy** – Use when you have resources in multiple locations and you want to route traffic to the resource that provides the best latency.
- **Multivalue answer routing policy** – Use when you want Route 53 to respond to DNS queries with up to eight healthy records selected at random.
- **Weighted routing policy** – Use to route traffic to multiple resources in proportions that you specify.

Simple routing

- With simple routing, you typically route traffic to a single resource,
- for example, to a web server for your website.



Failover Routing

- Failover routing lets you route traffic to a resource when the resource is healthy or to a different resource when the first resource is unhealthy.
- The primary and secondary records can route traffic to anything from an Amazon S3 bucket that is configured as a website to a complex tree of records.

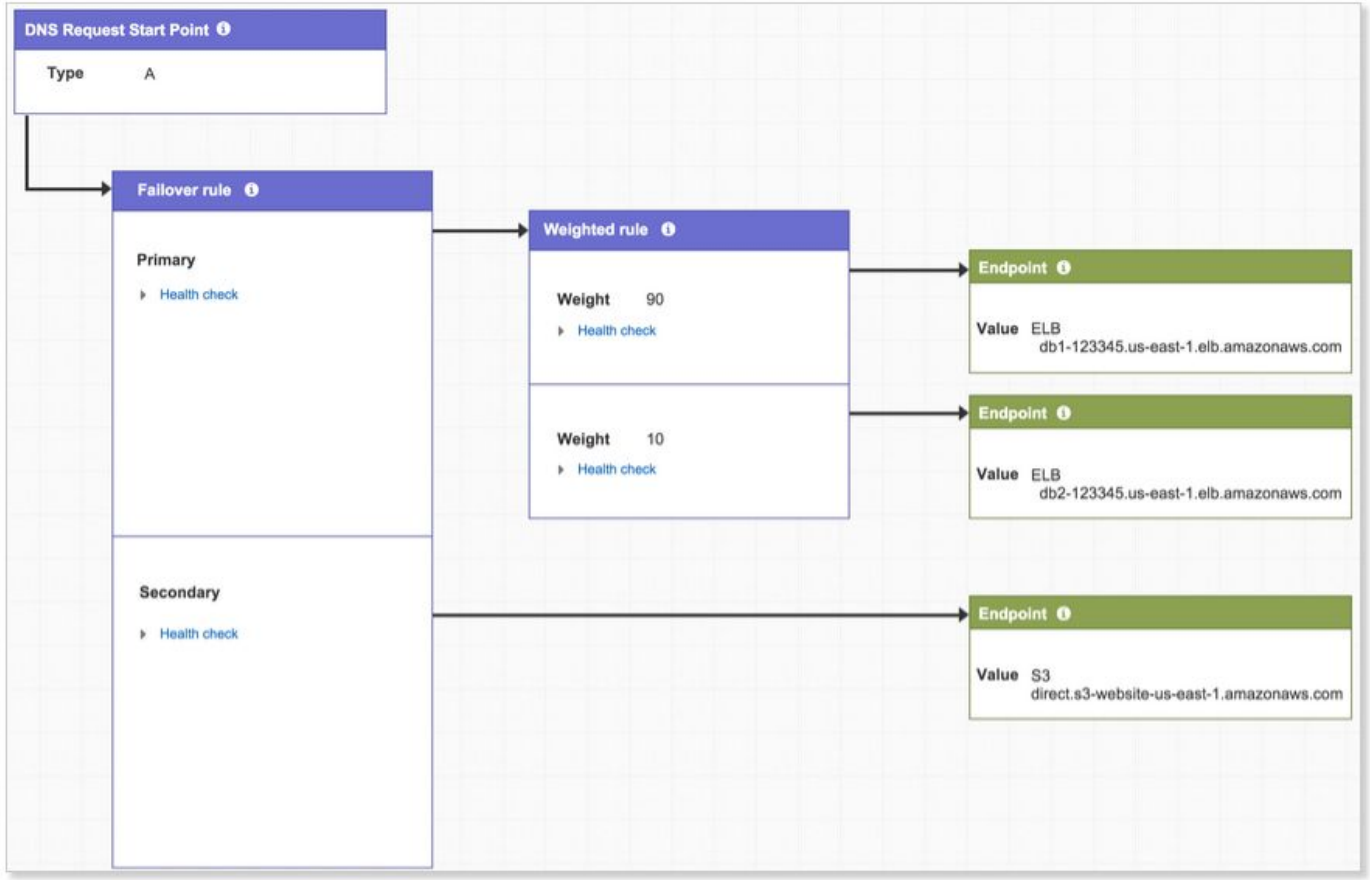
Geolocation routing

- Geolocation routing lets you choose the resources that serve your traffic based on the geographic location of your users, meaning the location that DNS queries originate from.
- For example, you might want all queries from Europe to be routed to an ELB load balancer in the Frankfurt region.
- This is routing based on user location
- Should create a “default” policy(in case there’s no match on location)

Weighted routing

- Control the % of the requests that go to specific endpoint
- Helpful to test 1% of traffic on new app version for example
- Helpful to split traffic between two regions
- Can be associated with Health Checks

Contd.


DNS Request Start Point
Type A
Failover rule
Primary
Health check
Secondary
Health check
Weighted rule
Weight 90
Health check
Weight 10
Health check
Endpoint
Value ELB
db1-123345.us-east-1.elb.amazonaws.com
Endpoint
Value ELB
db2-123345.us-east-1.elb.amazonaws.com
Endpoint
Value S3
direct.s3-website-us-east-1.amazonaws.com

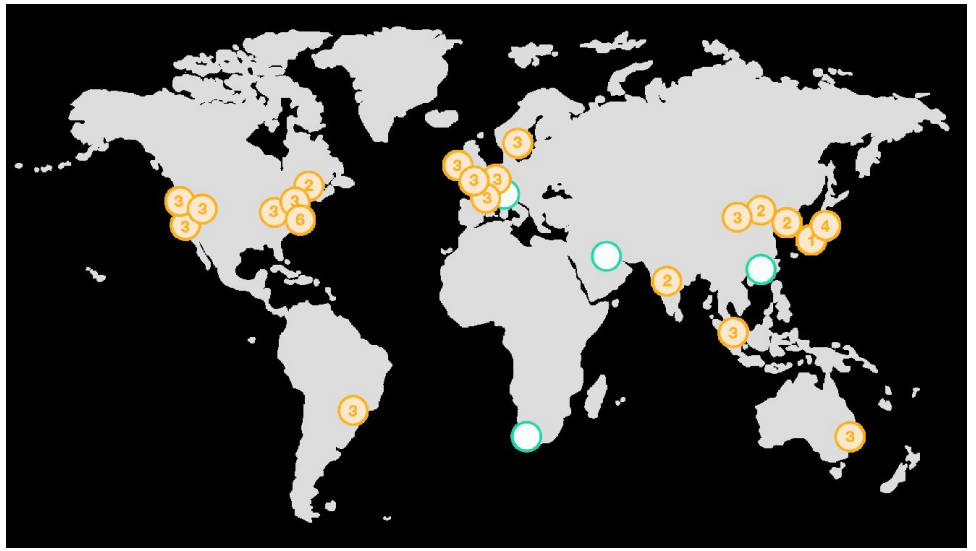
Multi Value Routing Policy

- Use when routing traffic to multiple resources
- Want to associate a Route 53 health checks with records
- Up to 8 healthy records are returned for each Multi Value query
- Multi Value is not a substitute for having an ELB

Name	Type	Value	TTL	Set ID	Health Check
www.example.com	A Record	192.0.2.2	60	Web1	A
www.example.com	A Record	198.51.100.2	60	Web2	B
www.example.com	A Record	203.0.113.2	60	Web3	C

Latency Routing policy

- Redirect to the server that has the least latency close to us
- Super helpful when latency of users is a priority
- Latency is evaluated in terms of user to designated AWS Region
- Germany may be directed to the US (if that's the lowest latency)



VPC

Introduction

- Amazon Virtual Private Cloud (Amazon VPC) lets you provision a logically isolated section of the AWS Cloud where you can launch AWS resources in a virtual network that you define. You have complete control over your virtual networking environment, including selection of your own IP address range, creation of subnets, and configuration of route tables and network gateways. You can use both IPv4 and IPv6 in your VPC for secure and easy access to resources and applications.

- You can easily customize the network configuration of your Amazon VPC. For example, you can create a public-facing subnet for your web servers that have access to the internet. You can also place your backend systems, such as databases or application servers, in a private-facing subnet with no internet access. You can use multiple layers of security, including security groups and network access control lists, to help control access to Amazon EC2 instances in each subnet.

CIDR

- They help to define an IP address range
  - We've seen WW.XX.YY.ZZ/32 == one IP
  - We've seen 0.0.0.0/0 == all IPs
  - But we can define for ex: 192.168.0.0/26: 192.168.0.0 – 192.168.0.63 (64 IP)

CIDR

- A CIDR has two components:
- The base IP (XX.XX.XX.XX)
- The Subnet Mask (/26)
- The base IP represents an IP contained in the range
- The subnet masks defines how many bits can change in the IP
- The subnet mask can take two forms. Examples:
  - 255.255.255.0 less common
  - /24 more common

Contd..

- The subnet masks basically allows part of the underlying IP to get additional next values from the base IP
- /32 allows for 1 IP = 2^0
- /31 allows for 2 IP = 2^1
- /30 allows for 4 IP = 2^2
- /29 allows for 8 IP = 2^3
- /28 allows for 16 IP = 2^4
- /27 allows for 32 IP = 2^5
- /26 allows for 64 IP = 2^6
- /25 allows for 128 IP = 2^7
- /24 allows for 256 IP = 2^8
- /16 allows for 65,536 IP = 2^16
- /0 allows for all IPs = 2^32

Examples

Understanding CIDRs

Little exercise

- 192.168.0.0/24 = … ?
- 192.168.0.0/16 = … ?
- 134.56.78.123/32 = … ?
- 0.0.0.0/0 ?
- When in doubt, use this website: **https://www.ipaddressguide.com/cidr**

Contd.

- All new accounts have a default VPC
- New instances are launched into default VPC if no subnet is specified
- Default VPC have internet connectivity and all instances have public IP
- We also get a public and a private DNS name

Contd.

- VPC = Virtual Private Cloud
- You can have multiple VPCs in a region (max 5 per region – soft limit)
- Max CIDR per VPC is 5. For each CIDR:
  - Min size is /28 = 16 IP Addresses
  - Max size is /16 = 65536 IP Addresses
- Because VPC is private, only the Private IP ranges are allowed:
  - 10.0.0.0 – 10.255.255.255 (10.0.0.0/8)
  - 172.16.0.0 – 172.31.255.255 (172.16.0.0/12)
  - 192.168.0.0 – 192.168.255.255 (192.168.0.0/16)
- Your VPC CIDR should not overlap with your other networks (ex: corporate)

Subnets

- AWS reserves 5 IPs address (first 4 and last 1 IP address) in each Subnet
- These 5 IPs are not available for use and cannot be assigned to an instance
- Ex, if CIDR block 10.0.0.0/24, reserved IP are:
  - 10.0.0.0: Network address
  - 10.0.0.1: Reserved by AWS for the VPC router
  - 10.0.0.2: Reserved by AWS for mapping to Amazon-provided DNS
  - 10.0.0.3: Reserved by AWS for future use
  - 10.0.0.255: Network broadcast address. AWS does not support broadcast in a VPC, therefore the address is reserved
- Exam Tip:
  - If you need 29 IP addresses for EC2 instances, you can't choose a Subnet of size /27 (32 IP)
  - You need at least 64 IP, Subnet size /26 (64-5 = 59 > 29, but 32-5 = 27 < 29)

Internet Gateways

- Internet gateways helps our VPC instances connect with the internet
- It scales horizontally and is HA and redundant
- Must be created separately from VPC
- One VPC can only be attached to one IGW and vice versa
- Internet Gateway is also a NAT for the instances that have a public IPv4
- Internet Gateways on their own do not allow internet access…
- Route tables must also be edited!

NAT Instances – Network Address Translation

- Allows instances in the private subnets to connect to the internet
- Must be launched in a public subnet
- Must disable EC2 flag: Source / Destination Check
- Must have Elastic IP attached to it
- Route table must be configured to route traffic from private subnets to NAT Instance

Overview

- CIDR: IP Range
- VPC: Virtual Private Cloud => we define a list of IPv4 & IPv6 CIDR
- Subnets: Tied to an AZ, we define a CIDR
- Internet Gateway: at the VPC level, provide IPv4 & IPv6 Internet Access
- Route Tables: must be edited to add routes from subnets to the IGW, VPC Peering Connections, VPC Endpoints, etc…
- NAT Instances: gives internet access to instances in private subnets. Old, must be setup in a public subnet, disable Source / Destination check flag
- NAT Gateway: managed by AWS, provides scalable internet access to private instances, IPv4 only
- Private DNS + Route 53: enable DNS Resolution + DNS hostnames (VPC)
- NACL: Stateless, subnet rules for inbound and outbound, don't forget ephemeral ports
- Security Groups: Stateful, operate at the EC2 instance level

NACL

- NACL are like a firewall which control traffic from and to subnet
- Default NACL allows everything outbound and everything inbound
- One NACL per Subnet, new Subnets are assigned the Default NACL
- Define NACL rules:
- Rules have a number (1-32766) and higher precedence with a lower number
- E.g. If you define #100 ALLOW <IP> and #200 DENY <IP> , IP will be allowed
- Last rule is an asterisk (*) and denies a request in case of no rule match
- AWS recommends adding rules by increment of 100
- Newly created NACL will deny everything
- NACL are a great way of blocking a specific IP at the subnet level

Contd..

Security Group	Network ACL
Operates at the instance level	Operates at the subnet level
Supports allow rules only	Supports allow rules and deny rules
Is stateful: Return traffic is automatically allowed, regardless of any rules	Is stateless: Return traffic must be explicitly allowed by rules
We evaluate all rules before deciding whether to allow traffic	We process rules in number order when deciding whether to allow traffic
Applies to an instance only if someone specifies the security group when launching the instance, or associates the security group with the instance later on	Automatically applies to all instances in the subnets it's associated with (therefore, you don't have to rely on users to specify the security group)


aws

CloudFront

Introduction

- Amazon CloudFront is a fast content delivery network (CDN) service that securely delivers data, videos, applications, and APIs to customers globally with low latency, high transfer speeds, all within a developer-friendly environment. CloudFront is integrated with AWS – both physical locations that are directly connected to the AWS global infrastructure, as well as other AWS services. CloudFront works seamlessly with services including AWS Shield for DDoS mitigation, Amazon S3, Elastic Load Balancing or Amazon EC2 as origins for your applications, and Lambda@Edge to run custom code closer to customers' users and to customize the user experience.
- Lastly, if you use AWS origins such as Amazon S3, Amazon EC2 or Elastic Load Balancing, you don't pay for any data transferred between these services and CloudFront.

Contd..

- • Content Delivery Network (CDN)
- • Improves read performance, content is cached at the edge
- • 136 Point of Presence globally (edge locations)
- • Popular with S3 but works with EC2, Load Balancing
- • Can help protect against network attacks
- • Can provide SSL encryption (HTTPS) at the edge using ACM
- • CloudFront can use SSL encryption (HTTPS) to talk to your applications
- • Support RTMP Protocol (videos / media)

How it works

- Amazon CloudFront is a web service that speeds up distribution of your static and dynamic web content, such as .html, .css, .js, and image files, to your users. CloudFront delivers your content through a worldwide network of data centers called edge locations. When a user requests content that you're serving with CloudFront, the user is routed to the edge location that provides the lowest latency (time delay), so that content is delivered with the best possible performance.
    - If the content is already in the edge location with the lowest latency, CloudFront delivers it immediately.
    - If the content is not in that edge location, CloudFront retrieves it from an origin that you've defined—such as an Amazon S3 bucket, a MediaPackage channel, or an HTTP server (for example, a web server) that you have identified as the source for the definitive version of your content.

Contd..

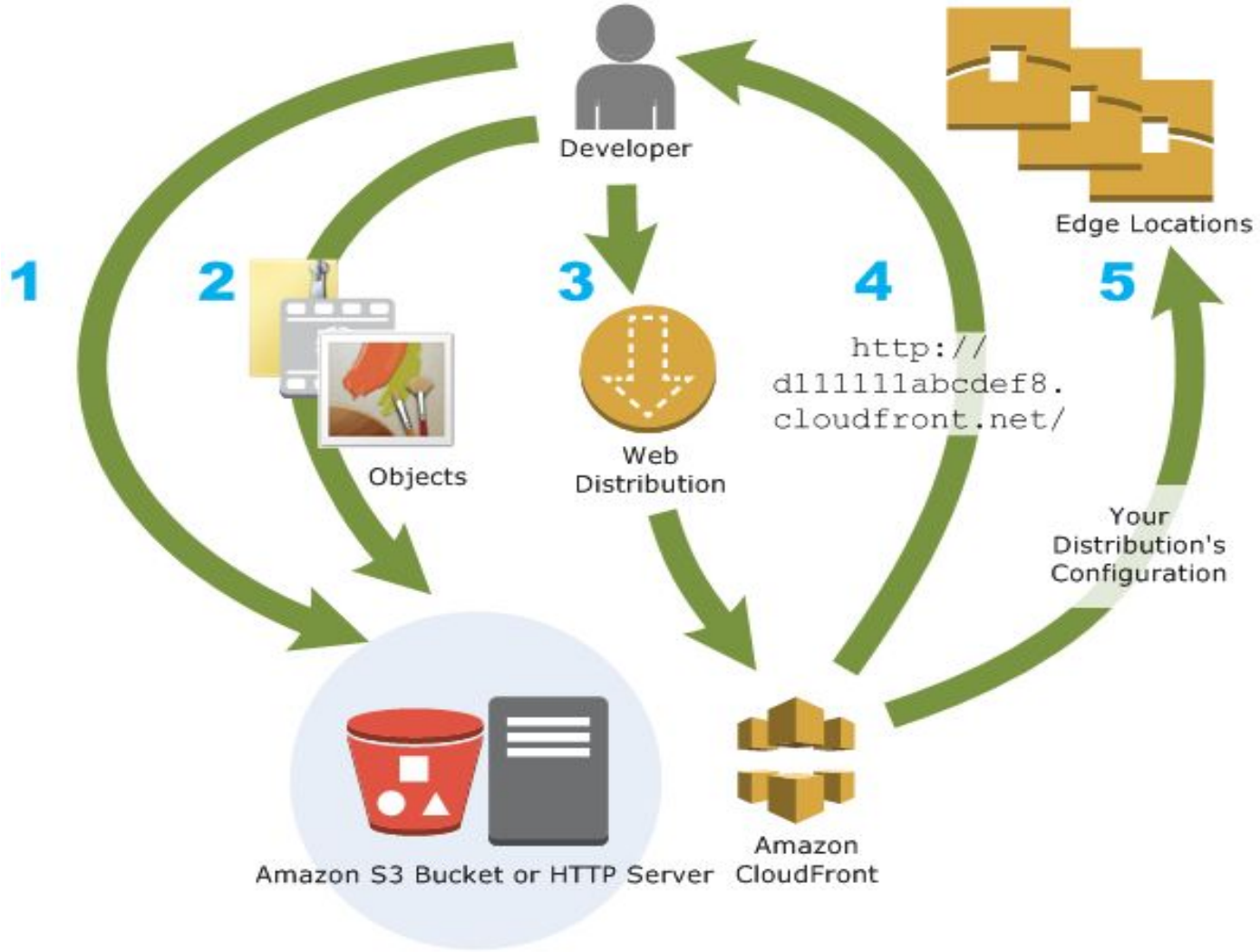
- As an example, suppose that you're serving an image from a traditional web server, not from CloudFront. For example, you might serve an image, sunsetphoto.png, using the URL http://example.com/sunsetphoto.png.

- Your users can easily navigate to this URL and see the image. But they probably don't know that their request was routed from one network to another—through the complex collection of interconnected networks that comprise the internet—until the image was found.

Contd..

- CloudFront speeds up the distribution of your content by routing each user request through the AWS backbone network to the edge location that can best serve your content. Typically, this is a CloudFront edge server that provides the fastest delivery to the viewer. Using the AWS network dramatically reduces the number of networks that your users' requests must pass through, which improves performance. Users get lower latency—the time it takes to load the first byte of the file—and higher data transfer rates.
- You also get increased reliability and availability because copies of your files (also known as *objects)* are now held (or cached) in multiple edge locations around the world.

Contd..


Developer
Edge Locations
1
2
3
4
5
Objects
Web Distribution
http://
d111111abcdef8.
cloudfront.net/
Your Distribution's Configuration
Amazon S3 Bucket or HTTP Server
Amazon CloudFront

Snowball

Introduction

- **Migrate Petabyte-Scale Data to the Cloud.**
- Snowball is a petabyte-scale data transport solution that uses secure appliances to transfer large amounts of data into and out of the AWS cloud. Using Snowball addresses common challenges with large-scale data transfers including high network costs, long transfer times, and security concerns.

Contd.

- Physical data transport solution that helps moving TBs or PBs of data in or out of AWS
- Alternative to moving data over the network (and paying network fees)
- Secure, tamper resistant, uses KMS 256 bit encryption
- Tracking using SNS and text messages. E-ink shipping label
- Pay per data transfer job
- Use cases: large data cloud migrations, DC decommission, disaster recovery
- If it takes more than a week to transfer over the network, use Snowball devices!

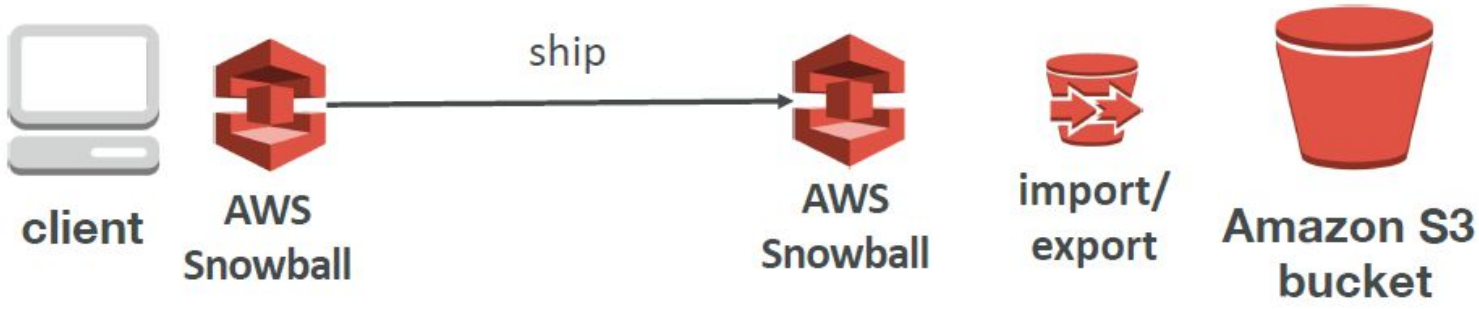
Snowball Process

1. Request snowball devices from the AWS console for delivery
2. Install the snowball client on your servers
3. Connect the snowball to your servers and copy files using the client
4. Ship back the device when you're done (goes to the right AWS facility)
5. Data will be loaded into an S3 bucket
6. Snowball is completely wiped
7. Tracking is done using SNS, text messages and the AWS console

Diagram

- Direct upload to S3:

- With snowball



Snowball Edge

- Snowball Edges add computational capability to the device
- 100 TB capacity with either:
  - Storage optimized – 24 vCPU
  - Compute optimized – 52 vCPU & optional GPU
- Supports a custom EC2 AMI so you can perform processing on the go
- Supports custom Lambda functions
- Very useful to pre-process the data while moving
- Use case: data migration, image collation, IoT capture, machine learning

AWS Snowmobile

- Transfer exabytes of data (1 EB = 1,000 PB = 1,000,000 TBs)
- Each Snowmobile has 100 PB of capacity (use multiple in parallel)
- Better than Snowball if you transfer more than 10 PB.

Hybrid Cloud for Storage

- AWS is pushing for ”hybrid cloud”
- Part of your infrastructure is on the cloud
- Part of your infrastructure is on-premise
- This can be due to
- Long cloud migrations
- Security requirements
- Compliance requirements
- IT strategy

AWS Storage Gateway

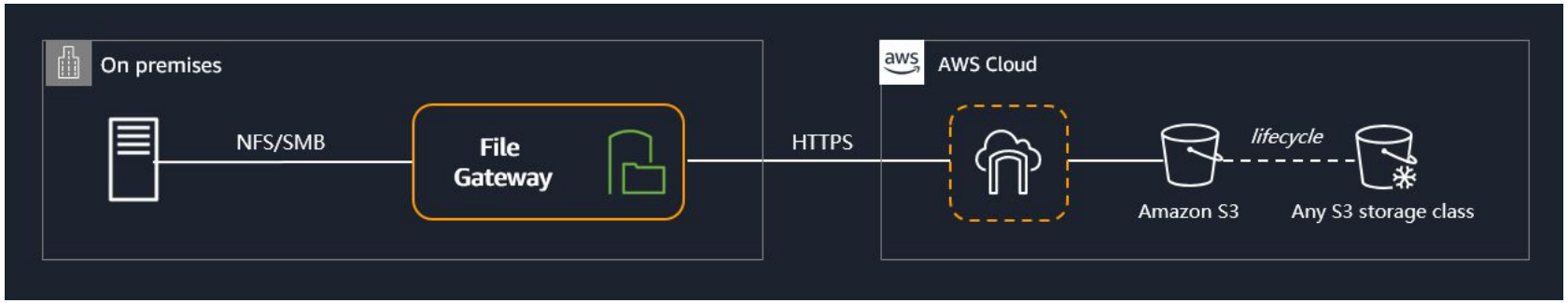
Introduction

- AWS Storage Gateway is a **hybrid cloud storage service** that gives you on-premises access to virtually unlimited cloud storage. Customers use Storage Gateway to simplify storage management and reduce costs for key hybrid cloud storage use cases. These include moving backups to the cloud, using on-premises file shares backed by cloud storage, and providing low latency access to data in AWS for on-premises applications.
- To support these use cases, Storage Gateway offers three different types of gateways –
  1. File Gateway
  2. Tape Gateway
  3. Volume Gateway

Contd..

- Bridge between on-premise data and cloud data in S3
- Use cases:
  1. disaster recovery,
  2. backup
  3. restore
  4. tiered storage

File Gateway



File Gateway

- File Gateway provides a seamless way to connect to the cloud in order to store application data files and backup images as durable objects in Amazon S3 cloud storage. File Gateway offers SMB or NFS-based access to data in Amazon S3 with local caching. It can be used for on-premises applications, and for Amazon EC2-based applications that need file protocol access to S3 object storage.

File Gateway

- Configured S3 buckets are accessible using the NFS and SMB protocol
- Supports S3 standard, S3 IA, S3 One Zone IA
- Bucket access using IAM roles for each File Gateway
- Most recently used data is cached in the file gateway
- Can be mounted on many servers

Volume Gateway

- The Volume Gateway provides full volumes on-premises while also storing full copies of your volumes in the AWS cloud. Volume Gateway also provides Amazon EBS Snapshots of your data for backup, disaster recovery, and migration.
- It's easy to get started with the Volume Gateway: Deploy it as a virtual machine or hardware appliance, give it local disk resources, connect it to your applications, and start using your hybrid cloud storage for block data.

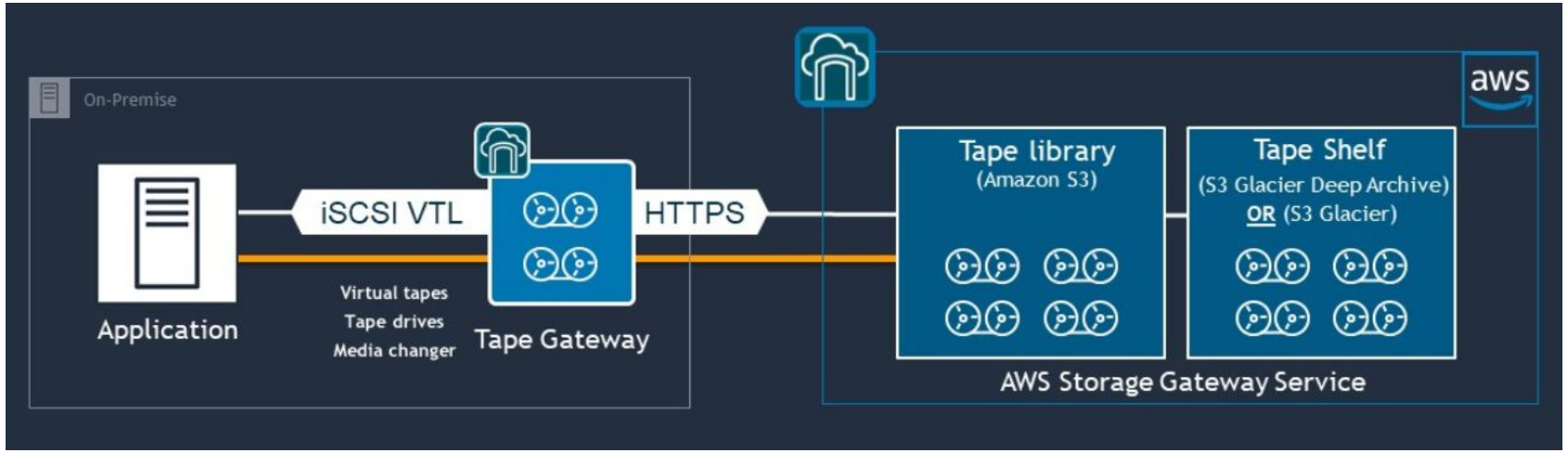
Volume Gateway

- Block storage using iSCSI protocol backed by S3
- Backed by EBS snapshots which can help restore on-premise volumes!
- Cached volumes: low latency access to most recent data
- Stored volumes: entire dataset is on premise, scheduled backups to S3

Tape Gateway

- Some companies have backup processes using physical tapes (!)
- With Tape Gateway, companies use the same processes but in the cloud
- Virtual Tape Library (VTL) backed by Amazon S3 and Glacier
- Back up data using existing tape-based processes (and iSCSI interface)
- Works with leading backup software vendors

- For as long as IT groups have run important applications, they've needed backups of their key application data. Unfortunately, many organizations are still hindered by physical backup technologies that are decades old, and waste valuable operational time, capital, and flexibility. AWS Storage Gateway offers IT organizations a seamless way to transfer backup jobs from tape or Virtual Tape Library systems to the cloud – while keeping trusted backup tools and processes in place.


On-Premise
Application
iSCSI VTL
Virtual tapes
Tape drives
Media changer
Tape Gateway
HTTPS
Tape library
(Amazon S3)
Tape Shelf
(S3 Glacier Deep Archive)
OR (S3 Glacier)
aws
AWS Storage Gateway Service

Summary

- On premise data to the cloud => Storage Gateway
- File access / NFS => File Gateway (backed by S3)
- Volumes / Block Storage / iSCSI => Volume gateway (backed by S3 with EBS snapshots)
- VTL Tape solution / Backup with iSCSI = > Tape Gateway

(backed by S3 and Glacier)

AWS Athena

Introduction

- Amazon Athena is an interactive query service that makes it easy to analyze data in Amazon S3 using standard SQL. Athena is serverless, so there is no infrastructure to manage, and you pay only for the queries that you run.

- Athena is easy to use. Simply point to your data in Amazon S3, define the schema, and start querying using standard SQL. Most results are delivered within seconds. With Athena, there’s no need for complex ETL jobs to prepare your data for analysis. This makes it easy for anyone with SQL skills to quickly analyze large-scale datasets.

Contd..

- Serverless service to perform analytics directly against S3 files
- Uses SQL language to query the files
- Has a JDBC / ODBC driver
- Charged per query and amount of data scanned
- Supports CSV, JSON, ORC, Avro, and Parquet (built on Presto)
- Use cases: Business intelligence / analytics / reporting, analyze & query

VPC Flow Logs, ELB Logs, CloudTrail trails, etc...

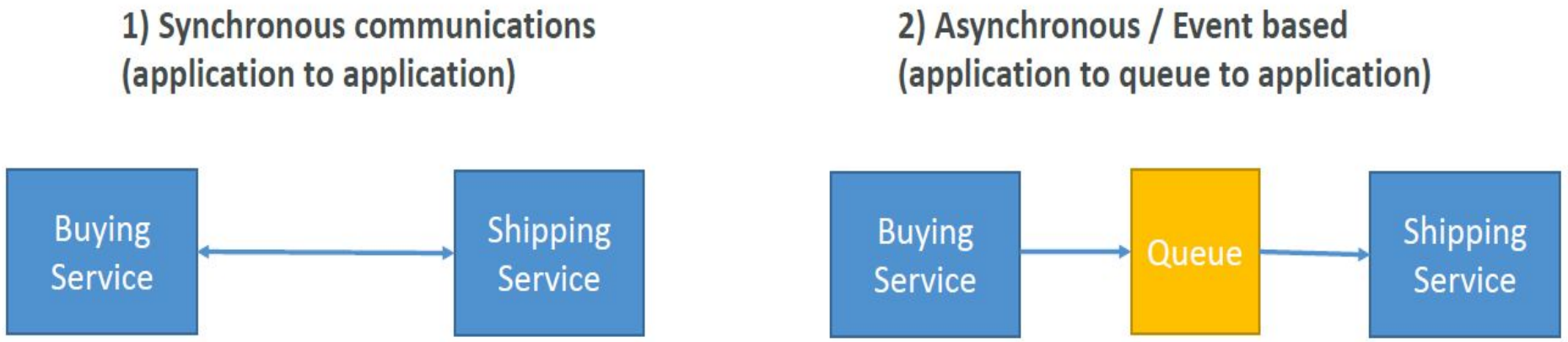
AWS SQS

Introduction

- Amazon Simple Queue Service (SQS) is a fully managed message queuing service that enables you to decouple and scale microservices, distributed systems, and serverless applications. SQS eliminates the complexity and overhead associated with managing and operating message oriented middleware, and empowers developers to focus on differentiating work. Using SQS, you can send, store, and receive messages between software components at any volume, without losing messages or requiring other services to be available.
- SQS offers two types of message queues.
  - Standard queues offer maximum throughput, best-effort ordering, and at-least-once delivery.
  - SQS FIFO queues are designed to guarantee that messages are processed exactly once, in the exact order that they are sent.

Concept

- multiple applications need to communicate with one another many times.
- There are two patterns of application communication



Contd..

- Synchronous between applications can be problematic if there are sudden spikes of traffic
- What if you need to suddenly encode 1000 videos but usually it’s 10?
- In that case, it’s better to decouple your applications,
  1. SQS: queue model
  2. SNS: pub/sub model
  3. Kinesis: real-time streaming model

Standard Queue

- Oldest offering (over 10 years old)
- Fully managed
- Scales from 1 message per second to 10,000s per second
- Default retention of messages: 4 days, maximum of 14 days
- No limit to how many messages can be in the queue
- Low latency (<10 ms on publish and receive)
- Horizontal scaling in terms of number of consumers
- Can have duplicate messages (at least once delivery, occasionally)
- Can have out of order messages (best effort ordering)
- Limitation of 256KB per message sent

Delay Queue

- Delay a message (consumers don’t see it immediately) up to 15 minutes
- Default is 0 seconds (message is available right away)
- Can set a default at queue level
- Can override the default using the DelaySeconds parameter

Visibility Timeout

- When a consumer polls a message from a queue, the message is “invisible” to other consumers for a defined period… the Visibility Timeout:
- Set between 0 seconds and 12 hours (default 30 seconds)
- If too high (15 minutes) and consumer fails to process the message,

you must wait a long time before processing the message again

- If too low (30 seconds) and consumer needs time to process the message (2 minutes), another consumer will receive the message and the message will be processed more than once
- ChangeMessageVisibility API to change the visibility while processing a message
- DeleteMessage API to tell SQS the message was successfully processed

Dead Letter Queue(DLQ)

- If a consumer fails to process a message within the Visibility Timeout…the message goes back to the queue!
- We can set a threshold of how many times a message can go back to the queue – it's called a “redrive policy”
- After the threshold is exceeded, the message goes into a dead letter queue(DLQ)
- We have to create a DLQ first and then designate it dead letter queue
- Make sure to process the messages in the DLQ before they expire!

Long Polling

- When a consumer requests message from the queue, it can optionally “wait” for messages to arrive if there are none in the queue
- This is called Long Polling
- LongPolling decreases the number of API calls made to SQS while increasing the efficiency and latency of your application.
- The wait time can be between 1 sec to 20 sec (20 sec preferable)
- Long Polling is preferable to Short Polling
- Long polling can be enabled at the queue level or at the API level using WaitTimeSeconds

FIFO Queue

- Newer offering (First In - First out) – not available in all regions!
- Name of the queue must end in .fifo
- Lower throughput (up to 3,000 per second with batching, 300/s without)
- Messages are processed in order by the consumer
- Messages are sent exactly once
- No per message delay (only per queue delay)
- Ability to do content based de-duplication
- 5-minute interval de-duplication using “Duplication ID”
- Message Groups:
  - Possibility to group messages for FIFO ordering using “Message GroupID”
  - Only one worker can be assigned per message group so that messages are processed in order
  - Message group is just an extra tag on the message!

AWS SNS

Introduction

- Amazon Simple Notification Service (SNS) is a fully managed messaging service for both system-to-system and app-to-person (A2P) communication. It enables you to communicate between systems through publish/subscribe (pub/sub) patterns that enable messaging between decoupled microservice applications or to communicate directly to users via SMS, mobile push and email.
- The system-to-system pub/sub functionality provides topics for high-throughput, push-based, many-to-many messaging. Using Amazon SNS topics, your publisher systems can fanout messages to a large number of subscriber systems or customer endpoints including Amazon SQS queues, AWS Lambda functions and HTTP/S, for parallel processing.

- The “event producer” only sends message to one SNS topic
- As many “event receivers” (subscriptions) as we want to listen to the SNS topic notifications
- Each subscriber to the topic will get all the messages (note: new feature to filter messages)
- Up to 10,000,000 subscriptions per topic
- 100,000 topics limit
- Subscribers can be:
  - SQS
  - HTTP / HTTPS (with delivery retries – how many times)
  - Lambda
  - Emails
  - SMS messages
  - Mobile Notifications

AWS Kinesis

- Kinesis is a managed alternative to Apache Kafka
- Great for application logs, metrics, IoT, clickstreams
- Great for “real-time” big data
- Great for streaming processing frameworks (Spark, NiFi, etc…)
- Data is automatically replicated to 3 AZ
- Kinesis Streams: low latency streaming ingest at scale
- Kinesis Analytics: perform real-time analytics on streams using SQL
- Kinesis Firehose: load streams into S3, Redshift, ElasticSearch…

AWS Lambda

Introduction

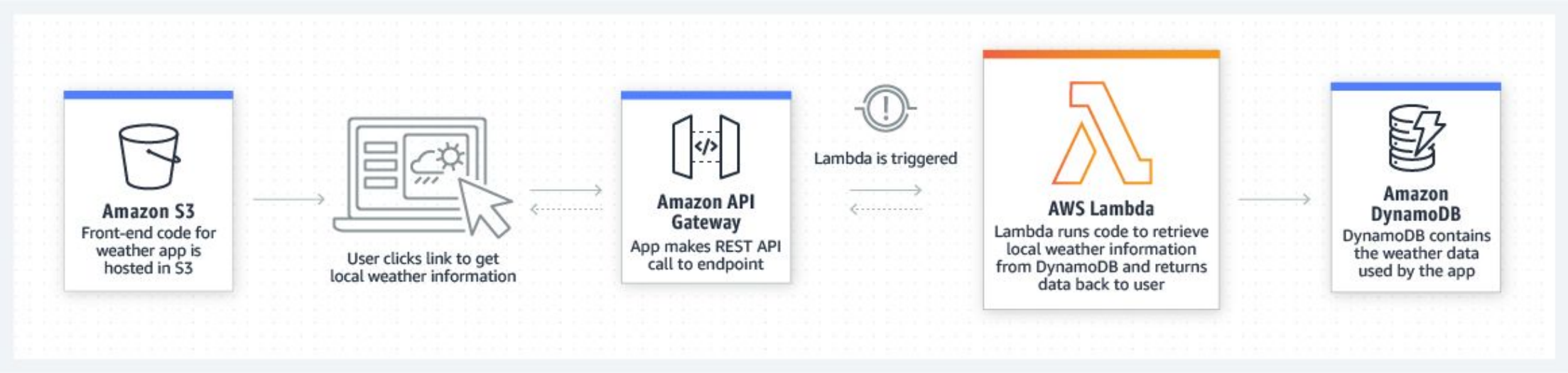
- AWS Lambda lets you run code without provisioning or managing servers. You pay only for the compute time you consume.

- With Lambda, you can run code for virtually any type of application or backend service - all with zero administration. Just upload your code and Lambda takes care of everything required to run and scale your code with high availability. You can set up your code to automatically trigger from other AWS services or call it directly from any web or mobile app.

Contd.


Upload your code to AWS Lambda or write code in Lambda's code editor
Set up your code to trigger from other AWS services, HTTP endpoints, or in-app activity
AWS Lambda
Lambda runs your code only when triggered, using only the compute resources needed
Just pay for the compute time you use

Sample Use case



Contd..


User posts status update
Amazon API Gateway
App makes REST API call to endpoint
Lambda is triggered
AWS Lambda
Lambda runs code to look up friends list and initiate status update notification
Amazon SNS
Status update notification is pushed to user's friends
User's friends receive status update notification

AWS Sample GitHub repo

- Sample codes hosted by AWS on GitHub repo.

- https://github.com/aws-samples

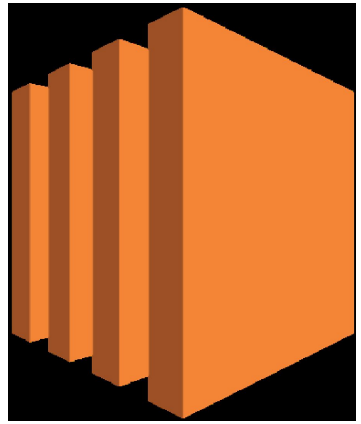
Lambda Functions

- The code you run on AWS Lambda is called a “Lambda function.” After you create your Lambda function it is always ready to run as soon as it is triggered, similar to a formula in a spreadsheet. Each function includes your code as well as some associated configuration information, including the function name and resource requirements. Lambda functions are “stateless,” with no affinity to the underlying infrastructure, so that Lambda can rapidly launch as many copies of the function as needed to scale to the rate of incoming events.

- After you upload your code to AWS Lambda, you can associate your function with specific AWS resources (e.g. a particular Amazon S3 bucket, Amazon DynamoDB table, Amazon Kinesis stream, or Amazon SNS notification). Then, when the resource changes, Lambda will execute your function and manage the compute resources as needed in order to keep up with incoming requests.

EC2 vs Lambda

- Virtual Servers in the Cloud
- Limited by RAM and CPU
- Continuously running
- Scaling means intervention to add / remove servers



Lambda

- Virtual functions – no servers to manage!
- Limited by time - short executions
- Run on-demand
- Scaling is automated!

Contd.

- Easy Pricing:
  - Pay per request and compute time
  - Free tier of 1,000,000 AWS Lambda requests and 400,000 GBs of compute time
- Integrated with the whole AWS Stack
- Integrated with many programming languages
- Easy monitoring through AWS CloudWatch
- Easy to get more resources per functions (up to 3GB of RAM!)
- Increasing RAM will also improve CPU and network!

Programming Languages

- Node.js (JavaScript)
- Python
- Java (Java 8/11 compatible)
- C# (.NET Core)
- Golang
- C# / PowerShell

Configuration

- Timeout: default 3 seconds, max of 300s (Note: new limit 15 minutes)
- Environment variables
- Allocated memory (128M to 3G)
- Ability to deploy within a VPC + assign security groups
- IAM execution role must be attached to the Lambda function

Limits

- Execution:
  - Memory allocation: 128 MB – 3008 MB (64 MB increments)
  - Maximum execution time: 300 seconds (5 minutes), now 15 minutes but 5 for exam
  - Disk capacity in the “function container” (in /tmp): 512 MB
  - Concurrency limits: 1000
- Deployment:
  - Lambda function deployment size (compressed .zip): 50 MB
  - Size of uncompressed deployment (code + dependencies): 250 MB
  - Can use the /tmp directory to load other files at startup
  - Size of environment variables: 4 KB

Lambda@Edge

- You have deployed a CDN using CloudFront
- What if you wanted to run a global AWS Lambda alongside?

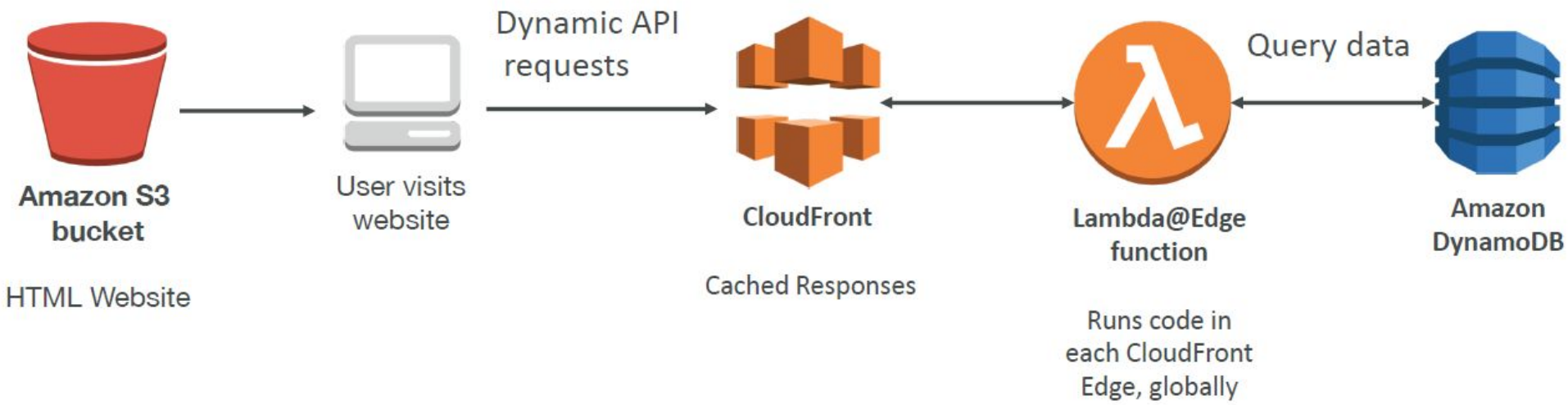
Lambda@Edge:deploy Lambda functions alongside your CloudFront CDN

  - Build more responsive applications
  - You don’t manage servers, Lambda is deployed globally
  - Customize the CDN content
  - Pay only for what you use

Contd..

- You can use Lambda to change CloudFront requests and responses:
- After CloudFront receives a request from a viewer (viewer request)
- Before CloudFront forwards the request to the origin (origin request)
- After CloudFront receives the response from the origin (origin response)
- Before CloudFront forwards the response to the viewer (viewer response)

Contd..


Amazon S3 bucket
HTML Website
User visits website
Dynamic API requests
CloudFront
Cached Responses
Lambda@Edge function
Runs code in each CloudFront Edge, globally
Query data
Amazon DynamoDB

Use cases

- Website Security and Privacy
- Dynamic Web Application at the Edge
- Search Engine Optimization (SEO)
- Intelligently Route Across Origins and Data Centers
- Bot Mitigation at the Edge
- Real-time Image Transformation
- A/B Testing
- User Authentication and Authorization
- User Prioritization
- User Tracking and Analytics

Amazon Cognito

- Simple and Secure User Sign-Up, Sign-In, and Access Control
- Amazon Cognito lets you add user sign-up, sign-in, and access control to your web and mobile apps quickly and easily. Amazon Cognito scales to millions of users and supports sign-in with social identity providers, such as Facebook, Google, and Amazon, and enterprise identity providers via SAML 2.0.

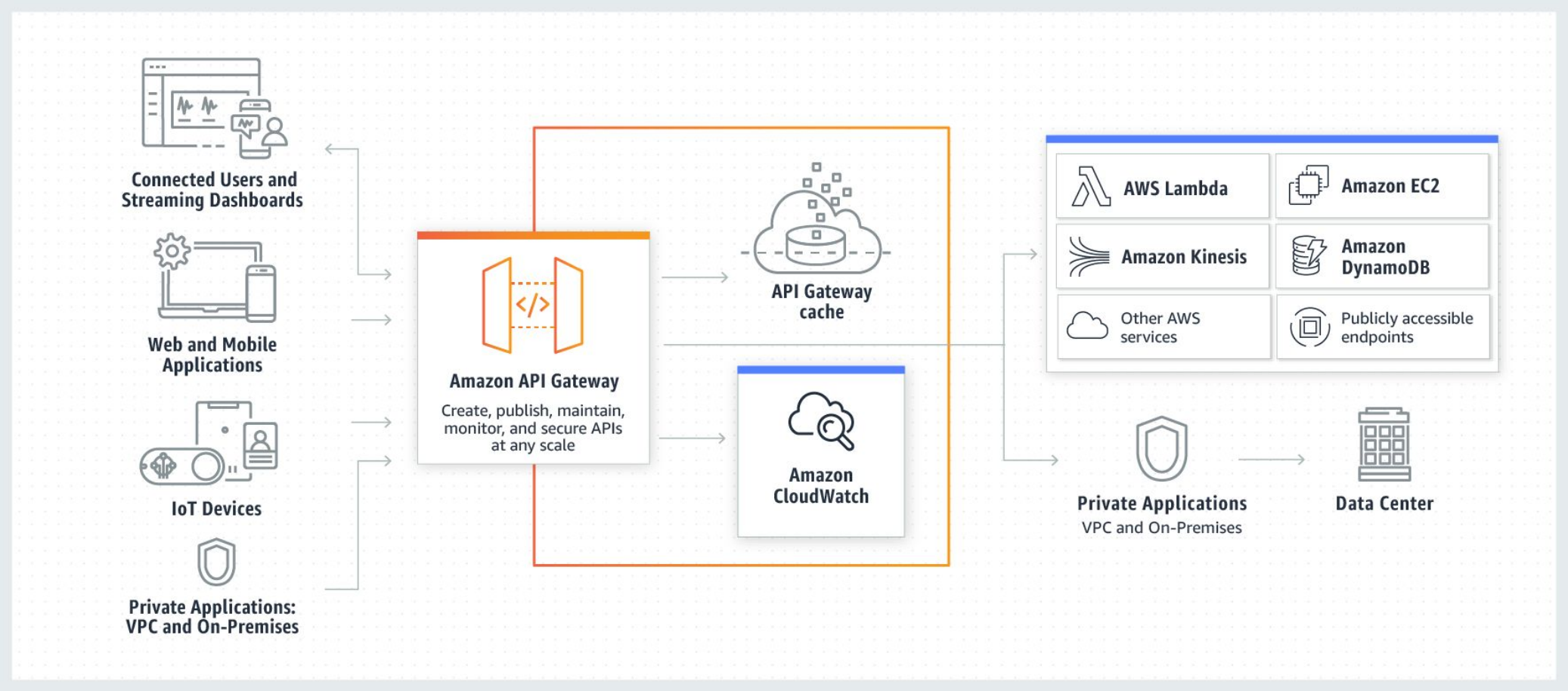
- We want to give our users an identity so that they can interact with our application.
- Cognito User Pools:
  - Sign in functionality for app users
  - Integrate with API Gateway
- Cognito Identity Pools (Federated Identity):
  - Provide AWS credentials to users so they can access AWS resources directly
  - Integrate with Cognito User Pools as an identity provider
- Cognito Sync:
  - Synchronize data from device to Cognito.
  - deprecated and replaced by AppSync

AWS API Gateway

Introduction

- Amazon API Gateway is a fully managed service that makes it easy for developers to create, publish, maintain, monitor, and secure APIs at any scale. APIs act as the "front door" for applications to access data, business logic, or functionality from your backend services. Using API Gateway, you can create RESTful APIs and WebSocket APIs that enable real-time two-way communication applications.
-
- API Gateway handles all the tasks involved in accepting and processing up to hundreds of thousands of concurrent API calls, including traffic management, CORS support, authorization and access control, throttling, monitoring, and API version management.

Contd..


Connected Users and Streaming Dashboards
Web and Mobile Applications
IoT Devices
Private Applications: VPC and On-Premises
Amazon API Gateway
Create, publish, maintain, monitor, and secure APIs at any scale
API Gateway cache
Amazon CloudWatch
AWS Lambda
Amazon EC2
Amazon Kinesis
Amazon DynamoDB
Other AWS services
Publicly accessible endpoints
Private Applications
VPC and On-Premises
Data Center

Contd..

- AWS Lambda + API Gateway: No infrastructure to manage
- Handle API versioning (v1, v2…)
- Handle different environments (dev, test, prod…)
- Handle security (Authentication and Authorization)
- Create API keys, handle request throttling
- Swagger / Open API import to quickly define APIs
- Transform and validate requests and responses
- Generate SDK and API specifications
- Cache API responses

Integrations

- Outside of VPC:
  - AWS Lambda (most popular / powerful)
  - Endpoints on EC2
  - Load Balancers
  - Any AWS Service
  - External and publicly accessible HTTP endpoints
- Inside of VPC:
  - AWS Lambda in your VPC
  - EC2 endpoints in your VPC

Security

- IAM:
  - Great for users / roles already within your AWS account
  - Handle authentication + authorization
  - Leverages Sig v4
- Custom Authorizer:
  - Great for 3rd party tokens
  - Very flexible in terms of what IAM policy is returned
  - Handle Authentication + Authorization
  - Pay per Lambda invocation
- Cognito User Pool:
  - You manage your own user pool (can be backed by Facebook, Google login etc…)
  - No need to write any custom code
  - Must implement authorization in the backend

DynamoDB

Introduction

- Amazon DynamoDB is a key-value and document database that delivers single-digit millisecond performance at any scale. It's a fully managed, multiregion, multimaster, durable database with built-in security, backup and restore, and in-memory caching for internet-scale applications. DynamoDB can handle more than 10 trillion requests per day and can support peaks of more than 20 million requests per second.
- Many of the world's fastest growing businesses such as Lyft, Airbnb, and Redfin as well as enterprises such as Samsung, Toyota, and Capital One depend on the scale and performance of DynamoDB to support their mission-critical workloads.

Contd..

- Fully Managed, Highly available with replication across 3 AZ
- **NoSQL database** - not a relational database
- Scales to massive workloads, distributed database
- Millions of requests per seconds, trillions of row, 100s of TB of storage
- Fast and consistent in performance (low latency on retrieval)
- Integrated with IAM for security, authorization and administration
- Enables event driven programming with DynamoDB Streams
- Low cost and auto scaling capabilities

Basics

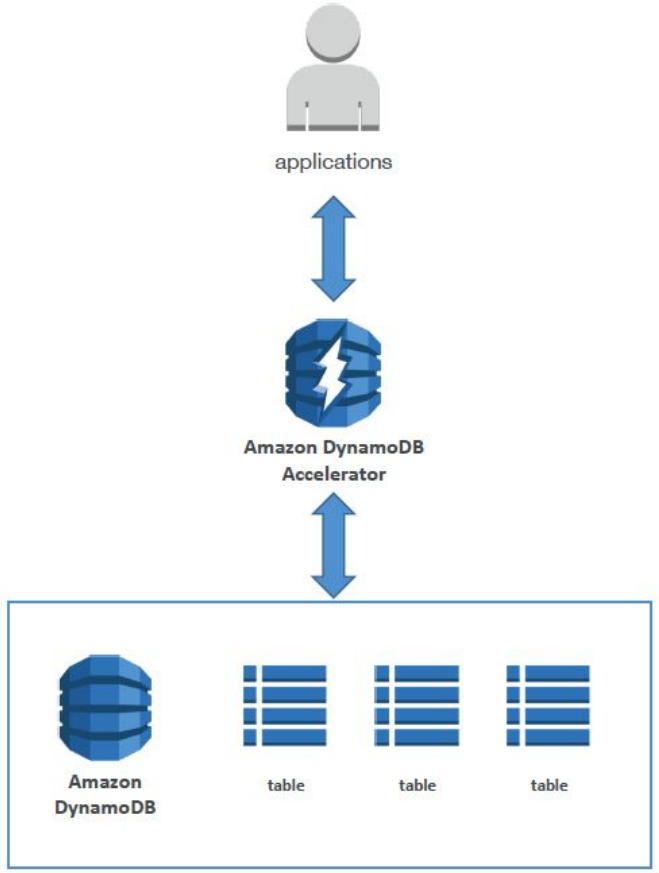
- DynamoDB is made of tables
- Each table has a primary key (must be decided at creation time)
- Each table can have an infinite number of items (= rows)
- Each item has attributes (can be added over time – can be null)
- Maximum size of a item is 400KB
- Data types supported are:
- Scalar Types: String, Number, Binary, Boolean, Null
- Document Types: List, Map
- Set Types: String Set, Number Set, Binary Set

Contd..

- Table must have provisioned read and write capacity units
- Read Capacity Units (RCU): throughput for reads ($0.00013 per RCU)
  - 1 RCU = 1 strongly consistent read of 4 KB per second
  - 1 RCU = 2 eventually consistent read of 4 KB per second
- Write Capacity Units (WCU): throughput for writes ($0.00065 per WCU)
  - 1 WCU = 1 write of 1 KB per second
- Option to setup auto-scaling of throughput to meet demand
- Throughput can be exceeded temporarily using “burst credit”
- If burst credit are empty, you’ll get a “ProvisionedThroughputException”.
- It’s then advised to do an exponential back-off retry

DAX

- • DAX = DynamoDB Accelerator
- • Seamless cache for DynamoDB, no application rewrite
- • Writes go through DAX to DynamoDB
- • Micro second latency for cached reads & queries
- • Solves the Hot Key problem (too many reads)
- • 5 minutes TTL for cache by default
- • Up to 10 nodes in the cluster
- • Multi AZ (3 nodes minimum recommended for
- production)
- • Secure (Encryption at rest with KMS, VPC, IAM,
- CloudTrail…)



Whitepapers

Important

- Well Architected Framework Whitepaper
- Well Architected Tool
- AWS Trusted Advisor
- Reference architectures resources (for real-world)
- Disaster Recovery on AWS Whitepaper

Contd

Well Architected Framework

5 Pillars

- 1) Operational Excellence
- 2) Security
- 3) Reliability
- 4) Performance Efficiency
- 5) Cost Optimization

For exam

- You can read about some AWS White Papers here:
  - Architecting for the Cloud: AWS Best Practices
  - AWS Well-Architected Framework
  - AWS Disaster Recovery (https://aws.amazon.com/disaster-recovery/)

FAQ

- Read each service’s FAQ
- • FAQ = Frequently asked questions
- • Example: https://aws.amazon.com/vpc/faqs/

Exams

- You'll have to register online at https://www.aws.training/
- Fee for the exam is 150 USD
- Provide two identity documents (ID, Credit Card, details are in emails sent to you)
- No notes are allowed, no pen is allowed, no speaking
- 65 questions will be asked in 130 minutes
- At the end you can optionally review all the questions / answers
- You will know right away if you passed / failed the exams
- You will not know which answers were right / wrong
- You will know the overall score a few days later (email notification)
- To pass you need a score of a least 720 out of 1000
- If you fail, you can retake the exam again 14 days later


aws