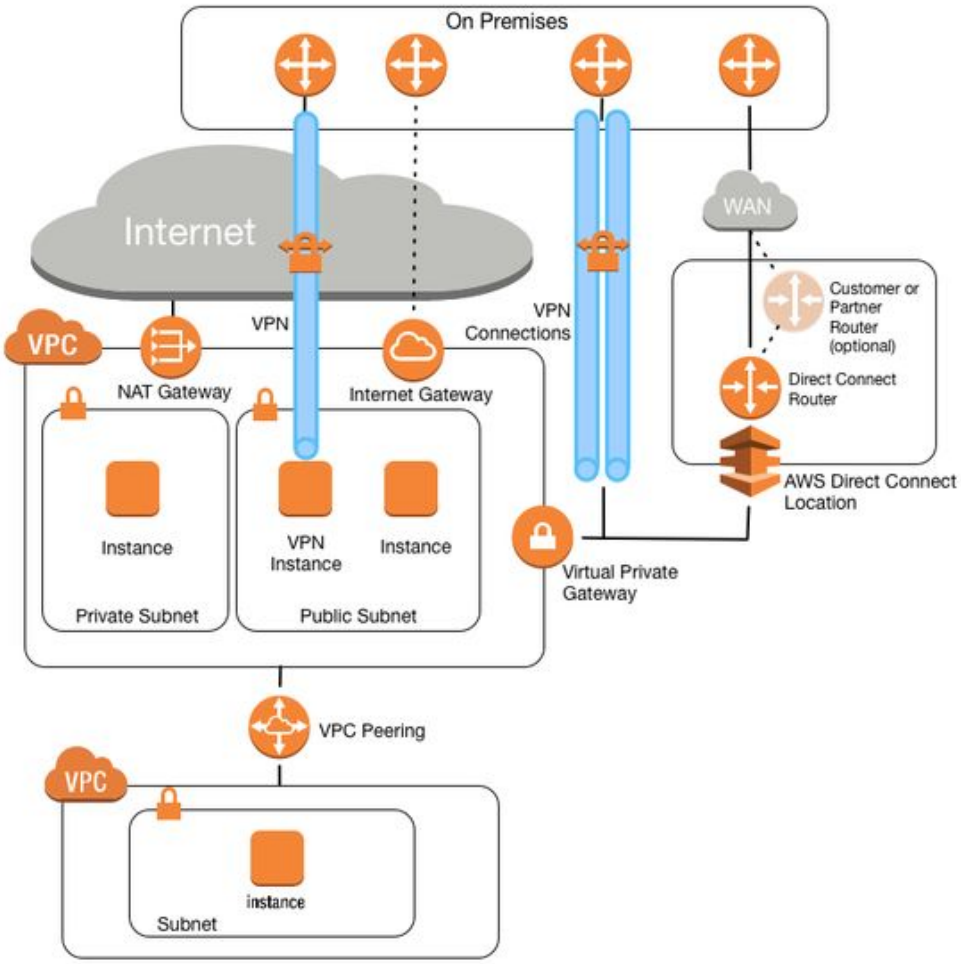
VPCs



Amazon VPC comprises a variety of objects that will be familiar to customers with existing networks:

- **A Virtual Private Cloud:** A logically isolated virtual network in the AWS cloud. You define a VPC's IP address space from ranges you select.
- **Subnet:** A segment of a VPC's IP address range where you can place groups of isolated resources.
- **Internet Gateway:** The Amazon VPC side of a connection to the public Internet.
- **NAT Gateway:** A highly available, managed Network Address Translation (NAT) service for your resources in a private subnet to access the Internet.
- **Virtual private gateway:** The Amazon VPC side of a VPN connection.
- **Peering Connection:** A peering connection enables you to route traffic via private IP addresses between two peered VPCs.
- **VPC Endpoints:** Enables private connectivity to services hosted in AWS, from within your VPC without using an Internet Gateway, VPN, Network Address Translation (NAT) devices, or firewall proxies.
- **Egress-only Internet Gateway:** A stateful gateway to provide egress only access for IPv6 traffic from the VPC to the Internet.

Amazon VPC provides three features that you can use to increase and monitor the security for your VPC. Security groups act as a firewall for associated Amazon EC2 instances, controlling both inbound and outbound traffic at the instance level. Network access control lists (NACLs) act as a firewall for associated subnets, controlling both inbound and outbound traffic at the subnet level. Flow logs capture information about the IP traffic going to and from network interfaces in your VPC

- When you create a **custom VPC**, a default Security Group, Access control List, and Route Table are created automatically. You must create your own subnets, Internet Gateway, and NAT Gateway (if you need one.)
- Default VPC :
    - access to Internet through Internet Gateway
    - public subnets with corresponding
    - route table.
    - public IPv4 addresses are assigned.

Difference between NAT Instance vs NAT Gateway?
NAT Instance is a single EC2 instance that is used by resources in a private subnet to access the internet. They are not on the way out. NAT Gateway are highly available EC2

What is a Route table? When should you use Route Table vs Security Groups?
https://www.reddit.com/r/aws/comments/cab816/difference_between_security_groups_route_tables/

Security Group	**NACL**	**Route Table**	**WAF (Web Application Firewall)**
Stateful - For every inbound rule, outbound rule is also supported. These automatically create temporary rules to allow return traffic from a TCP connection	Stateless - needs to define both inbound and outbound rules Since they are stateless, you MUST create rules to allow return traffic.	the route table specify how packets should flow,	A web application firewall (or WAF) filters, monitors, and blocks HTTP traffic to and from a web application. A WAF is differentiated from a regular firewall in that a WAF is able to filter the content of specific web applications while regular firewalls serve as a safety gate between servers. By inspecting HTTP traffic, it can prevent attacks stemming from web application security flaws, such as ● SQL Injection, ● Cross-site scripting (XSS), ● file inclusion, ● security misconfigurations. ● block traffic from IP address
EC2 Instances & Load Balancers	Gets applied at Subnet level		Attached to LB to do more complex types of filtering, like disallowing traffic from certain countries
ONLY Allow Rules	Allow & Deny Rules supported		
Runs all the rules	Runs the rules in order and if any rule matches, it doesn't execute the other rules		The following conditions: ● IP Match conditions ● String Match conditions ● Size Constraint conditions
Allow traffic based on IP Address / Port #			
Inside a subnet, only the security group CAN have an effect. This is how you would prevent 2 instances in the	Between subnets, the route table specify how packets should flow, and the NACL is what packets are allowed to flow. So, an entry in the route table has to be there to allow		

same subnet from using port 22 to SSH into each other, by having a security group that only allows traffic on port 22 from your bastion host, possibly in a different subnet.	traffic from that bastion host to the 2 instances. But, if you are using NACLs, you would have to allow traffic on port 22 into each subnet from the subnet that the bastion is in, and also allow traffic to go back on TCP ports 1024 to 65535 for ephemeral port		

Types of Load Balancers

- Application LB -- Needs at least 2 subnets ; can be internet facing or internal
- Network LB
- Classic LB

What are VPC Endpoints : https://docs.aws.amazon.com/vpc/latest/userguide/endpoint-services-overview.html

A VPC endpoint enables you to privately connect your VPC to supported AWS services and VPC endpoint services powered by AWS PrivateLink without requiring an internet gateway, NAT device, VPN connection, or AWS Direct Connect connection. Instances in your VPC do not require public IP addresses to communicate with resources in the service. Traffic between your VPC and the other service does not leave the Amazon network.
Endpoints are virtual devices. They are horizontally scaled, redundant, and highly available VPC components. They allow communication between instances in your VPC and services without imposing availability risks or bandwidth constraints on your network traffic.

There are two types of VPC endpoints: interface endpoints and gateway endpoints. Create the type of VPC endpoint required by the supported service.

Interface endpoints (powered by AWS PrivateLink)

An interface endpoint is an elastic network interface with a private IP address from the IP address range of your subnet that serves as an entry point for traffic destined to a supported service. Interface endpoints are powered by AWS PrivateLink, a technology that enables you to privately access services by using private IP addresses. AWS PrivateLink restricts all network traffic between your VPC and services to the Amazon network. You do not need an internet gateway, a NAT device, or a virtual private gateway.

Gateway endpoints

A gateway endpoint is a gateway that you specify as a target for a route in your route table for traffic destined to a supported AWS service. The following AWS services are supported:

- Amazon S3
- DynamoDB

VPC Flow Logs

- Captures information about the IP traffic going into & from network interfaces in the VPC
- Logs data is stored in Amazon CloudWatch Logs and it could be viewed & retrieved data from CloudWatch
- Can be created at VPC, Subnet and Network Interface Level
- It can be enabled for VPC peering; The VPC peering should be in the same accounts
- You cannot change its configurations
- Traffic to and from 169.254.169.254 for instance metadata is not monitored,
- Traffic to the reserved IP address for the default VPC router is not monitored
- DHCP traffic is not monitored;
- Traffic generated by instances when they contact Amazon DNS server is not monitored; but if you use your own DNS server, then all that traffic to DNS server is monitored

CIDR

A /28 subnet will only have 16 (2 ^ 4) addresses available. AWS reserves both the first four and last IP addresses in each subnet's CIDR block.

Subnets

- Currently you can create 200 subnets per VPC. If you would like to create more, please submit a case at the support center.
- The minimum size of a subnet is a /28 (or 16 IP addresses.) for IPv4. Subnets cannot be larger than the VPC in which they are created. For IPv6, the subnet size is fixed to be a /64. Only one IPv6 CIDR block can be allocated to a subnet.
-

VPC Peering Connections

- Peering connections can be created with VPCs in different regions. Inter-region VPC peering is available globally in all commercial regions (excluding China).
- AWS uses the existing infrastructure of a VPC to create a VPC peering connection; it is neither a gateway nor a VPN connection, and does not rely on a separate piece of physical hardware. There is no single point of failure for communication or a bandwidth bottleneck.
- Inter-Region VPC Peering operates on the same horizontally scaled, redundant, and highly available technology that powers VPC today. Inter-Region VPC Peering traffic goes over the AWS backbone that has in-built redundancy and dynamic bandwidth allocation. There is no single point of failure for communication. If an Inter-Region peering connection does go down, the traffic will not be routed over the internet.
- By default, a query for a public hostname of an instance in a peered VPC in a different region will resolve to a public IP address. Route 53 private DNS can be used to resolve to a private IP address with Inter-Region VPC Peering.
- Transitive peering relationships are not supported (meaning VPC A -> VPC B & VPC B -> VPC C, doesn't mean VPC A -> VPC C would occur)
- Traffic is not encrypted between instances in peered VPCs *within the region as the traffic* remains private and isolated – similar to how traffic between two instances in the same VPC is private and isolated. For Inter-region Traffic *is encrypted* using modern AEAD (Authenticated Encryption with Associated Data) algorithms. Key agreement and key management is handled by AWS.
- You can peer a VPC with a VPC belonging to another AWS account assuming the owner of the other VPC accepts your peering connection request.
- Peered VPCs must have non-overlapping IP ranges.
- No charge for creating VPC peering connections, however, data transfer across peering connections is charged.

Elastic Network Interface (ENI)

- More than 1 ENI can be attached, but the total number of network interfaces that can be attached to an EC2 instance depends on the instance type.
- ENI can only be attached to instances residing in the same Availability Zone.
- ENI can only be attached to instances in the same VPC as the interface.
- You can attach and detach secondary interfaces (eth1-ethn) on an EC2 instance, but you can't detach the eth0 interface.

Storage

EFS vs S3 vs EBS

- https://aws.amazon.com/efs/when-to-choose-efs/
- https://dzone.com/articles/confused-by-aws-storage-options-s3-ebs-amp-efs-explained
- https://help.acloud.guru/hc/en-us/articles/115002011194

- S3 Classes are;
    - Standard,
    - Standard-Infrequent Access,

    - One Zone-Infrequent Access,
    - Reduced Redundancy Storage
- For archive,
    - Glacier
    - Glacier Deep Archive.
- Reduced Redundancy Storage is the only S3 Class that *does not* offer 99.999999999% durability
- Snowball: peta-byte scale data transport solution that uses secure appliance to transfer large amount data into and out of aws. 50 TB & 80 TB storage. Copies the data to S3.
- Snowball Edge : 100TB ; It is compute and storage
- In general, multipart upload should be used for objects greater than 100 MB. It must be used for objects greater than 5 GB. You can set an object lifecycle to delete incomplete multipart uploads after a certain period of time. This will save storage costs on uploads that have been aborted for various reasons.

IAM Policies Vs S3 Bucket Policies vs S3 ACLs
https://aws.amazon.com/blogs/security/iam-policies-and-bucket-policies-and-acls-oh-my-controlling-access-to-s3-resources/

AWS Storage Gateway
https://docs.aws.amazon.com/storagegateway/latest/userguide/WhatIsStorageGateway.html

- AWS Storage Gateway connects an on-premises software appliance with *cloud-based storage* to provide seamless integration with data security features between your on-premises IT environment and the AWS storage infrastructure. You can use the service to store data in the AWS Cloud for scalable and cost-effective storage that helps maintain data security.
- AWS Storage Gateway offers both *volume-based and tape-based storage* solutions such as the following:
    - Gateway-Cached Volumes: store your data in S3 and caches frequently accessed data subsets locally
    - Gateway-Virtual Tape Library (Gateway-VTL): cloud-backed virtual tape storage backed up in Glacier/Deep Archive
    - Gateway-Stored Volumes: low-latency access to the entire dataset locally, then asynchronously back up point-in-time snapshots of this data to S3.

AWS CloudFront

- When you create a distribution, you receive the CloudFront domain name associated with that distribution. You use that domain name when creating the links to your objects. If you have another domain name that you'd rather use (for example, www.example.com), you can add a CNAME alias. When you create URLs to give end users access to objects in your CloudFront distribution, the URLs are either public URLs or signed URLs. Public URLs allow users to access the following objects:
    - Objects on which there are no restrictions
    - Objects in an Amazon S3 bucket that end users must access through CloudFront but that don't require a signed URL.
    - These objects can't be accessed using an Amazon S3 URL.
- You can store your content in an Amazon S3 bucket and use CloudFront to distribute the content. If you store your objects in an Amazon S3 bucket, you can either have your users get your objects directly from S3, or you can configure CloudFront to get your objects from S3 and distribute them to your users.
- Using CloudFront can be more cost effective if your users access your objects frequently because, at higher usage, the price for CloudFront data transfer is lower than the price for Amazon S3 data transfer. In addition, downloads are faster with CloudFront than with Amazon S3 alone because your objects are stored closer to your users.

- When you configure an origin group, you can pair two AWS origins, which would be Amazon S3 buckets or Amazon EC2 instances, or you can pair two custom origins, such as on-premise HTTP servers. (RTMP distributions are NOT compatible). If you are using Amazon S3 buckets to host your video content, you could implement an EC2 instance or S3 bucket as a secondary origin server.

SQS

- SQS is pull based
- Max message size : 256kb; Kept - 1min to 14 days; retention period : 4days
- More than max size, it writes to S3.
- Visibility Timeout - amount of time the message will be invisible after it gets picked up; Max timeout : 12 hrs
- Short polling vs Long polling: Short returns immediately ; Long polling won't return a response until a message arrives or long poll timeout. ; Long poll saves costs
- Message Oriented API

SWF

- Simple Workflow Service : Webservice that makes it easy to coordinate work across distributed application components. It enables applications for a wide range of use cases including media media processing, web application back ends, business process workflows. analytic pipelines and it's basically used to be designed as a coordination of tasks..

- Task represent invocation of various processing steps in an application which can be performed by executable code, web service calls, human actions, scripts
- SWF can last for 1 year
- Task Oriented API

Kinesis

Kinesis is a platform on AWS to send your streaming data to. Kinesis makes it easy to load and analyze streaming data and also provides the ability for you to build your own custom application for your business needs.
Types:
Kinesis Streams: The Producers can publish the data to Kinesis and it allows you to persistently store your streaming data for 24 hrs - 7 days, while your Consumers can process the data and store the processed data in DynamoDB, S3, Elastic Map Reduce (EMR), Redshift; Kinesis Streams consist of **Shards**
Kinesis Firehose: Inside Firehose, there are no Shards and no persistent. As soon as the data is received, it needs to be processed / analyze the data. Lamda is optional; Output the result to S3 or Redshift, Elasticsearch Cluster.
Kinesis Analytics: Works with Kinesis Streams or Kinesis Firehose and it should be used when you want to analyze the data (inside Kinesis) on the fly and store the data to redshift, S3, Elasticsearch.

Route53

Amazon Route 53 provides highly available and scalable Domain Name System (DNS), domain name registration, and health-checking web services. 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 example.com into the numeric IP addresses, such as 192.0.2.1, that computers use to connect to each other. You can combine your DNS with health-checking services to route traffic to healthy endpoints or to independently monitor and/or alarm on endpoints. You can also purchase and manage domain names such as

example.com and automatically configure DNS settings for your domains. 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.

Route 53 has the following routing policies -

- Simple,
- Weighted,
- Latency,
- Failover,
- Multivalue answer:
  - Route 53 now supports multivalue answers in response to DNS queries. While not a substitute for a load balancer, the ability to return multiple health-checkable IP addresses in response to DNS queries is a way to use DNS to improve availability and load balancing. If you want to route traffic randomly to multiple resources, such as web servers, you can create one multivalue answer record for each resource and, optionally, associate an Amazon Route 53 health check with each record. Amazon Route 53 supports up to eight healthy records in response to each DNS query.
- Geoproximity (based on Latitude / Longitude)
- Geolocation (based on countries)

Alias Record:

- Amazon Route 53 offers a special type of record called an 'Alias' record that lets you map your zone apex (example.com) DNS name to the DNS name for your ELB/S3, CloudFront, Elastic Beanstalk (PAAS), VPC Endpoint, API Gateway
- Alias Records can also point to AWS Resources that are hosted in other accounts by manually entering the ARN

CNAME Record

Private DNS: is a Route 53 feature that lets you have authoritative DNS within your VPCs without exposing your DNS records (including the name of the resource and its IP address(es) to the Internet.

Can I configure DNS Failover based on internal health metrics, such as CPU load, network, or memory?
Yes. Amazon Route 53's metric based health checks let you perform DNS failover based on any metric that is available within Amazon CloudWatch, including AWS-provided metrics and custom metrics from your own application. When you create a metric based health check within Amazon Route 53, the health check becomes unhealthy whenever its associated Amazon CloudWatch metric enters an alarm state.

Auto Scaling

- Launch templates (recommended over launch configurations) can include parameters for launching multiple instance types with multiple purchase options in the same template; launch template allows versioning too
- A launch template is similar to a launch configuration, in that it specifies instance configuration information. Included are the ID of the Amazon Machine Image (AMI), the instance type, a key pair, security groups, and the other parameters that you use to launch EC2 instances.

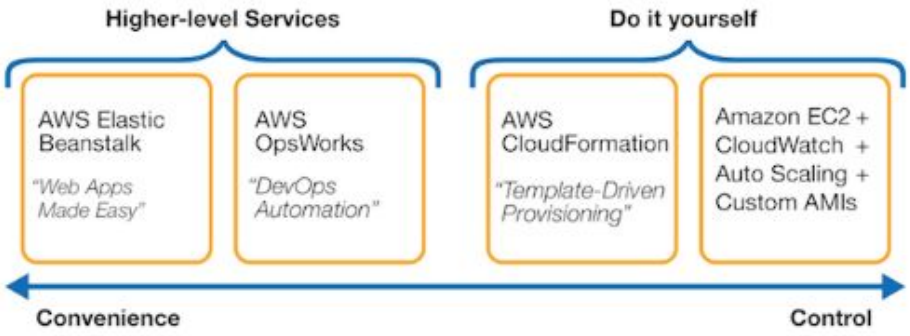
AutoScaling scales-in according to a hierarchy of decisions. Please see the link for further details.
http://docs.aws.amazon.com/AutoScaling/latest/DeveloperGuide/AutoScalingBehavior.InstanceTermination.html

Elastic Beanstalk vs CloudFormation

https://stackoverflow.com/questions/14422151/what-is-the-difference-between-elastic-beanstalk-and-cloudformation-for-a-net-p

- They're actually pretty different. Elastic Beanstalk is intended to make developers' lives easier. CloudFormation is intended to make systems engineers' lives easier.
- Elastic Beanstalk is a PaaS-like layer ontop of AWS's IaaS services which abstracts away the underlying EC2 instances, Elastic Load Balancers, auto scaling groups, etc. This makes it a lot easier for developers, who don't want to be dealing with all the systems stuff, to get their application quickly deployed on AWS. It's very similar to other PaaS products such as Heroku, EngineYard, Google App Engine, etc. With Elastic Beanstalk, you don't need to understand how any of the underlying magic works.
- CloudFormation, on the other hand, doesn't automatically do anything. It's simply a way to define all the resources needed for deployment in a huge JSON file. So a CloudFormation template might actually create two ElasticBeanstalk environments (production and staging), a couple of ElasticCache clusters, a DyanmoDB table, and then the proper DNS in Route53. I then upload this template to AWS, walk away, and 45 minutes later everything is ready and waiting. Since it's just a plain-text JSON file, I can stick it in my source control which provides a great way to version my application deployments. It also ensures that I have a repeatable, "known good" configuration that I can quickly deploy in a different region.

48



AWS CloudFormation: "Template-Driven Provisioning"

> AWS CloudFormation gives developers and systems administrators an easy way to create and manage a collection of related AWS resources, provisioning and updating them in an orderly and predictable fashion.

CloudFormation (CFn) is a lightweight, low-level abstraction over existing AWS APIs. Using a static JSON/YAML template document, you declare a set of Resources (such as an EC2 instance or an S3 bucket) that correspond to CRUD operations on the AWS APIs.

When you create a CloudFormation stack, CloudFormation calls the corresponding APIs to create the associated Resources, and when you delete a stack, CloudFormation calls the corresponding APIs to delete them. Most (but not all) AWS APIs are supported.

AWS Elastic Beanstalk: "Web Apps Made Easy"

> 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.

Elastic Beanstalk (EB) is a higher-level, managed 'platform as a service' (PaaS) for hosting web applications, similar in scope to Heroku. Rather than deal with low-level AWS resources directly, EB provides a fully-managed platform where you create an application environment using a web interface, select which platform your application uses, create and upload a source bundle, and EB handles the rest.

Using EB, you get all sorts of built-in features for monitoring your application environment and deploying new versions of your application.

Under the hood, EB uses CloudFormation to create and manage the application's various AWS resources. You can customize and extend the default EB environment by adding CloudFormation Resources to an EB configuration file deployed with your application.

Conclusion

If your application is a standard web-tier application using one of Elastic Beanstalk's supported platforms, and you want easy-to-manage, highly-scalable hosting for your application, **use Elastic Beanstalk**.

If you:

- Want to manage all of your application's AWS resources directly;
- Want to manage or heavily customize your instance-provisioning or deployment process;
- Need to use an application platform not supported by Elastic Beanstalk; or
- Just don't want/need any of the higher-level Elastic Beanstalk features

then **use CloudFormation directly** and avoid the added configuration layer of Elastic Beanstalk.

share edit follow edited Jul 31 '18 at 17:35 answered Jan 4 '17 at 17:45

EC2

- EC2 is a compute instance. Other example of compute instances are Lambdas, AWS Fargate (serverless compute for containers that works with ECS or EKS)
- If you wish to run more than 20 On-Demand instances, complete the Amazon EC2 instance request form.
- EC2 instances have the ability to store your root device data on Amazon EBS or the local instance store. Using EBS, data on the root device will persist independently from the lifetime of the instance. This enables you to stop and restart the instance at a subsequent time. Local instance stores (are backed by S3) only persist during the life of the instance. This is an inexpensive way to launch instances where data is not stored to the root device.

If you are using an Amazon EBS volume as a root partition, you will need to set the Delete On Terminate flag to "N" if you want your Amazon EBS volume to persist outside the life of the instance.

- While you are able to attach multiple EBS volumes to a single instance, attaching multiple EC2 instances to one volume is not supported at this time.
- Snapshots can be done in real time while the volume is attached and in use. For consistency, it is better to stop the instance and then take the snapshot of the EBS volume.
- SLA guarantees a Monthly Uptime Percentage of at least 99.99% for Amazon EC2 and Amazon EBS within a Region.
- Instance Types: Accelerated Computing instances | Compute Optimized instances | General Purpose instances | High Memory instances | Memory Optimized instances | Previous Generation instances | Storage Optimized instances
-
- CloudWatch Metrics are received and aggregated at 1 minute intervals. You can retrieve metrics data for any Amazon EC2 instance up to 2 weeks from the time you started to monitor it.
- There is no Data Transfer charge between two Amazon Web Services within the same region (i.e. between Amazon EC2 US West and another AWS service in the US West). Data transferred between AWS services in different regions will be charged as Internet Data Transfer on both sides of the transfer.
- Regional Data Transfer rates apply if at least one of the following is true, but is only charged once for a given instance even if both are true:
    - The other instance is in a different Availability Zone, regardless of which type of address is used.
    - Public or Elastic IP addresses are used, regardless of which Availability Zone the other instance is in.
- One Availability Zone name (for example, us-east-1a) in two AWS customer accounts may relate to different physical Availability Zones.
- All new EC2 instance types will use the Nitro Hypervisor, but in the near term, some new instance types will use Xen depending on the requirements of the platform.
- A cluster placement group is a logical entity that enables creating a cluster of instances by launching instances as part of a group. The cluster of instances then provides low latency connectivity between instances in the group. Cluster placement groups are created through the Amazon EC2 API or AWS Management Console.
- High Memory Cluster Instances provide customers with large amounts of memory and CPU capabilities per instance in addition to high network capabilities. These instance types are ideal for memory intensive workloads including in-memory analytics systems, graph analysis and many science and engineering applications

    High Memory Cluster Instances use the same cluster placement group functionality as Cluster Compute Instances for grouping instances into clusters – allowing applications to get the low-latency, high bandwidth network performance required for tightly-coupled node-to-node communication typical of many HPC and other network intensive applications.
- Use Savings Plans or Regional RIs to reduce your bill while committing to a one- or three-year term. Savings Plans offer significant savings over On Demand, just like EC2 RIs, but automatically reduce customers' bills on compute usage across any AWS region, even as usage changes. Use Capacity Reservations if you need the additional confidence in your ability to launch instances. Capacity Reservations can be created for any duration and can be managed independently of your Savings Plans or RIs. If you have Savings Plans or Regional RIs, they will automatically apply to matching Capacity Reservations. This gives you the flexibility to selectively add Capacity Reservations to a portion of your instance footprint and still reduce your bill for that usage.

- Reserved Instances (RI) : Standard RIs offer a significant discount on EC2 instance usage when you commit to a particular instance family. Convertible RIs offer you the option to change your instance configuration during the term, and still receive a discount on your EC2 usage.
- Scheduled Reserved Instances (Scheduled Instances) enable you to purchase capacity reservations that recur on a daily, weekly, or monthly basis, with a specified start time and duration, for a one-year term.
- Spot blocks (type of Spot instances) are designed not to be interrupted and will run continuously for the duration you select, independent of Spot market price. In rare situations, Spot blocks may be interrupted due to AWS capacity needs. In these cases, we will provide a two-minute warning before we terminate your instance (termination notice), and you will not be charged for the affected instance(s).
- Instance Lifecycle:
    - Rebooting an instance is equivalent to rebooting an operating system. The instance remains on the same host computer and maintains its public DNS name, private IP address, and any data on its instance store volumes. It typically takes a few minutes for the reboot to complete, but the time it takes to reboot depends on the instance configuration.
    - Stop / Start:
        - When you stop and start your instance, you lose any data on the instance store volumes on the previous host computer.
        - Your instance retains its private IPv4 address, which means that an Elastic IP address associated with the private IPv4 address or network interface is still associated with your instance. If your instance has an IPv6 address, it retains its IPv6 address.
    - Hibernate:When you hibernate an instance, we signal the operating system to perform hibernation (suspend-to-disk), which saves the contents from the instance memory (RAM) to your Amazon EBS root volume. We persist the instance's Amazon EBS root volume and any attached Amazon EBS data volumes. When you start your instance, the Amazon EBS root volume is restored to its previous state and the RAM contents are reloaded. Previously attached data volumes are reattached and the instance retains its instance ID.
- Amazon VPC traffic mirroring makes it easy for customers to replicate network traffic to and from an Amazon EC2 instance and forward it to out-of-band security and monitoring appliances for use-cases such as content inspection, threat monitoring, and troubleshooting. These appliances can be deployed on an individual EC2 instance or a fleet of instances behind a Network Load Balancer (NLB) with User Datagram Protocol (UDP) listener.
    - The traffic mirroring feature copies network traffic from Elastic Network Interface (ENI) of EC2 instances in your Amazon VPC. The mirrored traffic can be sent to another EC2 instance or to an NLB with a UDP listener. Traffic mirroring encapsulates all copied traffic with VXLAN headers. The mirror source and destination (monitoring appliances) can be in the same VPC or in a different VPC, connected via VPC peering or AWS Transit Gateway.

- EC2 instances within a VPC communicate with Amazon EC2 instances not within a VPC
    - Yes. If an Internet gateway has been configured, Amazon VPC traffic bound for Amazon EC2 instances not within a VPC traverses the Internet gateway and then enters the public AWS network to reach the EC2 instance. If an Internet gateway has not been configured, or if the instance is in a subnet configured to route through the virtual private gateway, the traffic traverses the VPN connection, egresses from your datacenter, and then re-enters the public AWS network.
- EC2 instances within a VPC in one region communicate with Amazon EC2 instances within a VPC in another region?
    - Yes. Instances in one region can communicate with each other using Inter-Region VPC Peering, public IP addresses, NAT gateway, NAT instances, VPN Connections or Direct Connect connections.
- Primary private IP addresses are retained for the instance's or interface's lifetime. Secondary private IP addresses can be assigned, unassigned, or moved between interfaces or instances at any time.
- **Auto Recovery:** Examples of problems that cause system status checks to fail include:
    - Loss of network connectivity
    - Loss of system power
    - Software issues on the physical host
    - Hardware issues on the physical host that impact network reachability

- Assuming a Linux operating system without separate hourly charges is in use, partial instance-hours are billed to the next hour for Dedicated instances only & Reserved, Spot and On-Demand instances are now billed on a per-second basis, with a one-minute minimum charge.
- Billing commences when Amazon EC2 initiates the boot sequence of an AMI instance. Billing ends when the instance terminates, which could occur through a web services command, by running "shutdown -h", or through instance failure.

EC2 Auto Scaling Groups

- If you have an EC2 Auto Scaling group (ASG) with running instances and you choose to delete the ASG, the instances will be terminated and the ASG will be deleted.
- Amazon SNS coordinates and manages the delivery or sending of notifications to subscribing clients or endpoints. You can configure EC2 Auto Scaling to send an SNS notification whenever your EC2 Auto Scaling group scales.
- When you create an EC2 Auto Scaling group, you must specify a launch configuration. You can specify your launch configuration with multiple EC2 Auto Scaling groups. However, you can only specify one launch configuration for an EC2 Auto Scaling group at a time, and you can't modify a launch configuration after you've created it. Therefore, if you want to change the launch configuration for your EC2 Auto Scaling group, you must create a launch configuration and then update your EC2 Auto Scaling group with the new launch configuration. When you change the launch configuration for your EC2 Auto Scaling group, any new instances are launched using the new configuration parameters, but existing instances are not affected.
- EC2 Auto Scaling groups are regional constructs. They can span Availability Zones, but not AWS regions.
- If you don't use Elastic Load Balancing (ELB) how would users be directed to the other servers in a group if there was a failure? You can integrate with Route53 (which Amazon EC2 Auto Scaling does not currently support out of the box, but many customers use). You can also use your own reverse proxy, or for internal microservices, can use service discovery solutions.
- You can provision and automatically scale EC2 capacity across different EC2 instance types, Availability Zones, and On-Demand, RIs and Spot purchase options in a single Auto Scaling Group.
- Amazon EC2 Auto Scaling fleet management for EC2 instances carries no additional fees. The dynamic scaling capabilities of Amazon EC2 Auto Scaling are enabled by Amazon CloudWatch and also carry no additional fees. Amazon EC2 and Amazon CloudWatch service fees apply and are billed separately.

CloudWatch

- Amazon CloudWatch is a monitoring service for AWS cloud resources and the applications you run on AWS. You can use Amazon CloudWatch to collect and track metrics, collect and monitor log files, and set alarms. Amazon CloudWatch can monitor AWS resources such as Amazon EC2 instances, Amazon DynamoDB tables, and Amazon RDS DB instances, as well as custom metrics generated by your applications and services, and any log files your applications generate. You can use Amazon CloudWatch to gain system-wide visibility into resource utilization, application performance, and operational health. You can use these insights to react and keep your application running smoothly.

EBS

https://aws.amazon.com/ebs/features/

- Amazon EBS allows you to create storage volumes and attach them to Amazon EC2 instances. Once attached, you can create a file system on top of these volumes, run a database, or use them in any other way you would use block storage. Amazon EBS volumes are placed in a specific Availability Zone where they are automatically replicated to protect you from the

failure of a single component. All EBS volume types offer durable snapshot capabilities and are designed for 99.999% availability.

- Amazon EBS provides four current generation volume types and are divided into two major categories: SSD-backed storage for transactional workloads and HDD-backed storage for throughput intensive workloads
- Throughput Optimized HDD (st1) volume types: ST1 volumes are backed by hard disk drives (HDDs) and are ideal for frequently accessed, throughput intensive workloads with large datasets and large I/O sizes, such as MapReduce, Kafka, log processing, data warehouse, and ETL workloads
- Cold HDD (sc1) :SC1 volumes are backed by hard disk drives (HDDs) and provide the lowest cost per GB of all EBS volume types. It is ideal for less frequently accessed workloads with large, cold datasets.
- **RAID 1** is to provide mirroring, redundancy, and fault-tolerance. **RAID 0** is a more suitable option for providing faster read and write operations, compared with RAID 1.

Elastic IP Address

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/elastic-ip-addresses-eip.html

- Elastic IP vs Public IP:
  - Public IP addresses are dynamic - i.e. if you stop/start your instance you get reassigned a new public IP.
  - Elastic IPs get allocated to your account, and stay the same - it's up to you to attach them to an instance or not. You could say they are static public IP addresses.
  - By default, all accounts are limited to 5 Elastic IP addresses per region. Elastic IP addresses, AWS impose a small hourly charge for each address when it is not associated to a running instance, as public IP addresses are sparse.
- The private IP address remains associated with the network interface when the instance is stopped and restarted, and is released when the instance is terminated. The public address is associated exclusively with the instance until it is stopped, terminated or replaced with an Elastic IP address.
- To use an Elastic IP address, you first allocate one to your account, and then associate it with your instance or a network interface.
- When you associate an Elastic IP address with an instance, it is also associated with the instance's primary network interface. When you associate an Elastic IP address with a network interface that is attached to an instance, it is also associated with the instance.
- When you associate an Elastic IP address with an instance or its primary network interface, the instance's public IPv4 address (if it had one) is released back into Amazon's pool of public IPv4 addresses. You cannot reuse a public IPv4 address, and you cannot convert a public IPv4 address to an Elastic IP address. For more information, see Public IPv4 addresses and external DNS hostnames.
- You can disassociate an Elastic IP address from a resource, and reassociate it with a different resource. Any open connections to an instance continue to work for a time even after you disassociate its Elastic IP address and reassociate it with another instance. We recommend that you reopen these connections using the reassociated Elastic IP address.
- A disassociated Elastic IP address remains allocated to your account until you explicitly release it.
- The EIP addresses will only be reachable from the Internet (not over the VPN connection). Each EIP address must be associated with a unique private IP address on the instance. EIP addresses should only be used on instances in subnets configured to route their traffic directly to the Internet gateway. EIPs cannot be used on instances in subnets configured to use a NAT gateway or a NAT instance to access the Internet. This is applicable only for IPv4. Amazon VPCs do not support EIPs for IPv6 at this time.

ELB

- Elastic Load Balancing offers two types of load balancers that both feature high availability, automatic scaling, and robust security. These include the Classic Load Balancer that routes traffic based on either application or network level information, and the Application Load Balancer that routes traffic based on advanced application level information that includes the content of the request.
- The Classic Load Balancer is ideal for simple load balancing of traffic across multiple EC2 instances, while the Application Load Balancer is ideal for applications needing advanced routing capabilities, microservices, and container-based architectures. The Classic Load Balancer supports load balancing of applications using HTTP, HTTPS (Secure HTTP), SSL (Secure TC21P) and TCP protocols.
- Elastic Load Balancing guarantees a monthly availability of at least 99.99% for your load balancers (Classic, Application or Network).
- You can integrate your Application Load Balancer with AWS WAF, a web application firewall that helps protect web applications from attacks by allowing you to configure rules based on IP addresses, HTTP headers, and custom URI strings. Using these rules, AWS WAF can block, allow, or monitor (count) web requests for your web application
- Network Load Balancers support both TCP, UDP, and TCP+UDP (Layer 4) listeners, as well as TLS listeners. It is architected to handle millions of requests/sec, sudden volatile traffic patterns and provides extremely low latencies. In addition Network Load Balancer also supports TLS termination, preserves the source IP of the clients, and provides stable IP support and Zonal isolation. It also supports long-running connections that are very useful for WebSocket type applications.

ELB Vs Route53
(https://stackoverflow.com/questions/57321793/elastic-load-balancer-elb-and-route-53-in-aws)

Both Route53 and ELB are used to distribute the network traffic. These AWS services appear similar but there are minor differences between them.

ELB distributes traffic among Multiple Availability Zone but not to multiple Regions. Route53 can distribute traffic among multiple Regions. In short, ELBs are intended to load balance across EC2 instances in a single region whereas DNS load-balancing (Route53) is intended to help balance traffic across regions.

Both Route53 and ELB perform health check and route traffic to only healthy resources. Route53 weighted routing has health checks and removes unhealthy targets from its list. However, DNS is cached so unhealthy targets will still be in the visitors cache for some time. On the other hand, ELB is not cached and will remove unhealthy targets from the target group immediately.

Use both Route53 and ELB: Route53 provides integration with ELB. You can use both Route53 and ELB in your AWS infrastructure. If you have AWS resources in multiple regions, you can use Route53 to balance the load among those regions. Inside the region, you can use ELB to load balance among the instances running in various Availability Zones.

IAM

- **IAM Users**
- **IAM User Groups**
- **IAM Role:** An IAM role is an IAM entity that defines a set of permissions for making AWS service requests. IAM roles are not associated with a specific user or group. Instead, trusted entities assume roles, such as IAM users, applications, or AWS services such as EC2.

    - A service-linked role is a type of role that links to an AWS service (also known as a linked service) such that only the linked service can assume the role. Using these roles, you can delegate permissions to AWS services to create and manage AWS resources on your behalf.
    - You can only associate one IAM role with an EC2 instance at this time. This limit of one role per instance cannot be increased.
    - Any application running on the instance that is using the role that was deleted will be denied access immediately.
    -
- **IAM Policies (Control Access):** A policy is an object in AWS that, when associated with an identity or resource, defines their permissions. When you create a permissions policy to restrict access to a resource, you can choose an *identity-based policy or a resource-based policy..*
    - *Identity-based policies* are attached to an IAM user, group, or role. These policies let you specify what that identity can do (its permissions). For example, you can attach the policy to the IAM user named John, stating that he is allowed to perform the Amazon EC2 RunInstances action. The policy could further state that John is allowed to get items from an Amazon DynamoDB table named MyCompany. You can also allow John to manage his own IAM security credentials. Identity-based policies can be managed or inline.
    - *Resource-based policies* are attached to a resource. For example, you can attach resource-based policies to Amazon S3 buckets, Amazon SQS queues, and AWS Key Management Service encryption keys. For a list of services that support resource-based policies, see AWS Services That Work with IAM.
    - With resource-based policies, you can specify who has access to the resource and what actions they can perform on it.

- An IAM user has permanent long-term credentials and is used to directly interact with AWS services. An IAM group is primarily a management convenience to manage the same set of permissions for a set of IAM users. An IAM role is an AWS Identity and Access Management (IAM) entity with permissions to make AWS service requests. IAM roles cannot make direct requests to AWS services; they are meant to be assumed by authorized entities, such as IAM users, applications, or AWS services such as EC2. Use IAM roles to delegate access within or between AWS accounts.
- AWS Identity and Access Management (IAM) supports identity federation for delegated access to the AWS Management Console or AWS APIs. With identity federation, external identities are granted secure access to resources in your AWS account without having to create IAM users. These external identities can come from your corporate identity provider (such as Microsoft Active Directory or from the AWS Directory Service) or from a web identity provider (such as Amazon Cognito, Login with Amazon, Facebook, Google, or any OpenID Connect-compatible provider).
    - Federated users (external identities) are users you manage outside of AWS in your corporate directory, but to whom you grant access to your AWS account using temporary security credentials. They differ from IAM users, which are created and maintained in your AWS account.
- Three ways IAM authenticates a principal: *User Name/Password, Access Key, Access Key/Session Token*. IAM allows you to create a password policy enforcing password complexity and expiration. An Access Key is a combination of an access key ID (20 characters) and an access secret key (40 characters). When a process operates under an assumed role, the temporary security token provides an access key for authentication. In addition to the access key, the token also includes a session token.

Web Identity Federation / Amazon Cognito

Web Identity federation lets you give your users access to AWS resources after they have successfully authenticated with a web-based identity provider like Amazon, FB, Google. Following successful authentication, the user received an authentication code from the Web ID provider, which they can trade for temporary AWS security credentials.

Cognito User Pools: All about actual users; It is user based, handles things like user registration, authentication and account recovery;

Cognito Identity Pools: Authorize access to your AWS resources

S3

- For S3 Standard, S3 Standard-IA, and S3 Glacier storage classes, your objects are automatically stored across multiple devices spanning a minimum of three Availability Zones, each separated by miles across an AWS Region. Objects stored in the S3 One Zone-IA storage class are stored redundantly within a single Availability Zone in the AWS Region.
- No Data Transfer charge for data transferred within an Amazon S3 Region via a COPY request. Data transferred via a COPY request between AWS Regions is charged at rates specified in the pricing section of the Amazon S3 detail page. There is no Data Transfer charge for data transferred between Amazon EC2 and Amazon S3 within the same region. However, data transferred between Amazon EC2 and Amazon S3 across all other regions is charged at rates specified

Storage Class	Availability	Durability
S3 Standard	99.99%	99.999999999%
S3 Standard-IA	99.9%	99.999999999%
S3 One Zone-IA	99.5%	99.999999999%
S3 Glacier & GlacierDeep Archive	99.99%	99.999999999%

- The charge for successful Requester Pays requests is straightforward: the requester pays for the data transfer and the request; the bucket owner pays for the data storage. However, the bucket owner is charged for the anonymous requests.

Security

- S3 is secure by default. Upon creation, only the resource owners have access to Amazon S3 resources they create. Amazon S3 supports user authentication to control access to data. You can use access control mechanisms such as bucket policies and Access Control Lists (ACLs) to selectively grant permissions to users and groups of users.
- Four mechanisms for controlling access to Amazon S3 resources:
    - **IAM policies:**IAM enables organizations with multiple employees to create and manage multiple users under a single AWS account. With IAM policies, customers can grant IAM users fine-grained control to their Amazon S3 bucket or objects while also retaining full control over everything the users do.
    - **Bucket policies:** With bucket policies, customers can define rules which apply broadly across all requests to their Amazon S3 resources, such as
        - granting write privileges to a subset of Amazon S3 resources.
        - restrict access based on an aspect of the request, such as HTTP referrer and IP address.

    - When a bucket policy is applied the permissions assigned apply to all objects within that Bucket. This policy introduces a new attribute called 'principal,' these principals can be IAM users, federated users, another AWS account, or even other AWS services and it defines which 'principals' should be allowed or denied access to various S3 resources.
    - Principals are not used within IAM policies, as the 'principal' element is defined by who is associated with that policy via the user, group or role association. As Bucket policies are assigned to Buckets, we need to have this additional parameter of 'Principals' within the policy.
    - **Access Control Lists (ACLs):** With ACLs, customers can grant specific permissions (i.e. READ, WRITE, FULL_CONTROL) to specific users for an individual bucket or object.
    - **Query String Authentication:** With Query String Authentication, customers can create a URL to an Amazon S3 object which is only valid for a limited time.

- By default, all Amazon S3 resources—buckets, objects, and related subresources (for example, lifecycle configuration and website configuration)—are private: only the resource owner, an AWS account that created it, can access the resource. The resource owner can optionally grant access permissions to others by writing an access policy.
- Amazon S3 offers access policy options broadly categorized as resource-based policies and user policies. Access policies you attach to your resources (buckets and objects) are referred to as resource-based policies. For example, bucket policies and access control lists (ACLs) are resource-based policies. You can also attach access policies to users in your account. These are called user policies. You may choose to use resource-based policies, user policies, or some combination of these to manage permissions to your Amazon S3 resources. The introductory topics provide general guidelines for managing permissions.
- You can limit access to your bucket from a specific Amazon VPC Endpoint or a set of endpoints using Amazon S3 bucket policies. S3 bucket policies now support a condition, aws:sourceVpce, that you can use to restrict access.

- **Amazon S3 Access Points** simplifies managing data access at scale for applications using shared data sets on S3. With S3 Access Points, you can now easily create hundreds of access points per bucket, representing a new way of provisioning access to shared data sets. Access Points provide a customized path into a bucket, with a unique hostname and access policy that enforces the specific permissions and network controls for any request made through the access point.
- An access point is a separate Amazon resource created for a bucket with an
    - Amazon Resource Name (ARN)
    - hostname (in the format of *https://[access_point_name]-[account ID].s3-accesspoint.[region].amazonaws.com*)
    - an access control policy,
    - network origin control.
-
- **Access Analyzer for S3** is a feature that monitors your access policies, ensuring that the policies provide only the intended access to your S3 resources. Access Analyzer for S3 evaluates your bucket access policies and enables you to discover and swiftly remediate buckets with potentially unintended access.
-

S3 Urls

S3 Access Styles	Format
Virtual Hosted Style URL	`https://bucket-name.s3.Region.amazonaws.com/key name` `https://bucket-name.s3-Region.amazonaws.com/key name`
Path-Style Access URL	`https://s3.Region.amazonaws.com/bucket-name/key name`
Static Website URL	`http://bucket-name.s3-website.Region.amazonaws.com/object-name` `http://bucket-name.s3-website-Region.amazonaws.com/object-name`
Legacy Global Endpoint URL	`https://bucket-name.s3.amazonaws.com`
S3 Access Point	`https://AccessPointName-AccountId.s3-accesspoint.region.amazonaws.com`
Through AWS Services	`s3://bucket-name/key-name`

S3 Object Tagging

- S3 object tags are key-value pairs applied to S3 objects which can be created, updated or deleted at any time during the lifetime of the object. With these, you'll have the ability to create Identity and Access Management (IAM) policies, setup S3

Lifecycle policies, and customize storage metrics. These object-level tags can then manage transitions between storage classes and expire objects in the background.
- Object tags can be replicated across AWS Regions using Cross-Region Replication. For customers with Cross-Region Replication already enabled, new permissions are required in order for tags to replicate.
- Object tags are priced based on the quantity of tags and a request cost for adding tags.

S3 Object Lock
- A new Amazon S3 feature that blocks object version deletion during a customer-defined retention period so that you can enforce retention policies as an added layer of data protection or for regulatory compliance. You can migrate workloads from existing write-once-read-many (WORM) systems into Amazon S3, and configure S3 Object Lock at the object- and bucket-levels to prevent object version deletions prior to pre-defined Retain Until Dates or Legal Hold Dates. S3 Object Lock protection is maintained regardless of which storage class the object resides in and throughout S3 Lifecycle transitions between storage classes.
- Object locks can prevent objects from modification or deletion; objection versioning does not - it only maintains versions of an object until those versions are deleted.
- Users can enable Object locks only on new S3 buckets.
- **Retention Modes:** - *governance* mode and *compliance* mode. Compliance mode prevents objects from being deleted or updated by users, including the root user. Governance mode allows objects to be modified, but prevents objects from being deleted.
- **Retention Period:** protects an object version for a fixed amount of time; After the retention period expires, the object version can be overwritten or deleted unless you also placed a legal hold on the object version.
- **Legal Holds:** Object Lock also enables you to place a legal hold on an object version (cannot be set at bucket level). Like a retention period, a legal hold prevents an object version from being overwritten or deleted. However, a legal hold doesn't have an associated retention period and remains in effect until removed. Legal holds can be freely placed and removed by any user who has the s3:PutObjectLegalHold permission.
- Legal holds are independent from retention periods. As long as the bucket that contains the object has Object Lock enabled, you can place and remove legal holds regardless of whether the specified object version has a retention period set. Placing a legal hold on an object version doesn't affect the retention mode or retention period for that object version. For example, suppose that you place a legal hold on an object version while the object version is also protected by a retention period. If the retention period expires, the object doesn't lose its WORM protection. Rather, the legal hold continues to protect the object until an authorized user explicitly removes it. Similarly, if you remove a legal hold while an object version has a retention period in effect, the object version remains protected until the retention period expires.

Notifications
- S3 Event Notifications can be set up on objects stored in S3. An event could be set up to notify when a delete is performed. Logging is turned off by default but can be enabled. When enabled, they can track requests to your S3 bucket.
- S3 Server Access Logging tracks detailed information not currently available with CloudWatch metrics or CloudTrail logs. To track requests for access to your bucket, you can enable access logging. Each access log record provides details about a single access request, such as the requester, bucket name, request time, request action, response status, and error code, if any. Access log information can be useful in security and access audits. It can also help you learn about your customer base and understand your Amazon S3 bill.
    - For more on server access logs (https://docs.aws.amazon.com/AmazonS3/latest/dev/ServerLogs.html)
    - For more on event notifications (https://docs.aws.amazon.com/AmazonS3/latest/dev/NotificationHowTo.html)
- Amazon S3 Standard - Infrequent Access (Standard - IA) is an Amazon S3 storage class for data that is accessed less frequently, but requires rapid access when needed. Standard - IA offers the high durability, throughput, and low latency of Amazon S3 Standard, with a low per GB storage price and per GB retrieval fee. This combination of low cost and high performance make Standard - IA ideal for long-term storage, backups, and as a data store for disaster recovery. The Standard - IA storage class is set at the object level and can exist in the same bucket as Standard, allowing you to use lifecycle policies to automatically transition objects between storage classes without any application changes.

CloudWatch Metrics

- CloudWatch Request Metrics will be available in CloudWatch within 15 minutes after they are enabled. CloudWatch Storage Metrics are enabled by default for all buckets, and reported once per day.
- CloudWatch storage metrics are provided free. Cloudwatch request metrics are priced as custom metrics

S3 Transfer Acceleration

- Amazon S3 Transfer Acceleration enables fast, easy, and secure transfers of files over long distances between your client and your Amazon S3 bucket. S3 Transfer Acceleration leverages Amazon CloudFront's globally distributed AWS Edge Locations. As data arrives at an AWS Edge Location, data is routed to your Amazon S3 bucket over an optimized network path.
- If you are uploading to a centralized bucket from geographically dispersed locations, or if you regularly transfer GBs or TBs of data across continents, you may save hours or days of data transfer time with S3 Transfer Acceleration.
- If you have objects that are smaller than 1GB or if the data set is less than 1GB in size, you should consider using Amazon CloudFront's PUT/POST commands for optimal performance.
- **S3 Select:** to retrieve a smaller, targeted data set from an object using simple SQL statements
- **Athena:** is an interactive query service that makes it easy to analyze data in Amazon S3 using standard SQL queries.
    - It allows you to query data located in S3 using the standard SQL
    - Serverless
    - Commonly used to analyse log data stored in S3.
- **Redshift Spectrum:** enables you to run queries against exabytes of unstructured data in Amazon S3 with no loading or ETL required.

Macie

- Amazon Macie is an AI-powered security service that helps you prevent data loss by automatically discovering, classifying, and protecting sensitive data stored in Amazon S3
- Macie uses AI to analyze data in S3 and helps identify PII
- Can also be used to analyse CloudTrail logs for suspicious API activity
- Includes Dashboards, Reports and Alerting
- Great for PCI-DSS compliance and preventing ID theft

AWS Global Accelerator

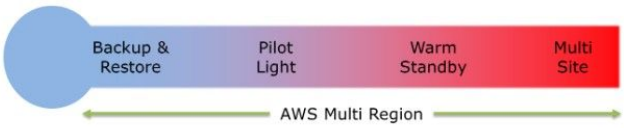
- It is a service in which you create accelerators to improve availability and performance of your application for local and global users. User request goes to AWS Edge network and then the request traverses to AWS Backbone network to get forwarded to the VPC / EC2 / ELB / Endpoints
- It assigns 2 static IPs or alternatively you can bring your own.
- You can control traffic using traffic dials or weights
- Global Accelerator can be created on the following endpoints: Network LB, App LB, EC2 Instances, Elastic IP
- AWS Global Accelerator and Amazon CloudFront are separate services that use the AWS global network and its edge locations around the world. CloudFront improves performance for both cacheable content (such as images and videos) and dynamic content (such as API acceleration and dynamic site delivery). Global Accelerator improves performance for a wide range of applications over TCP or UDP by proxying packets at the edge to applications running in one or more AWS Regions. Global Accelerator is a good fit for non-HTTP use cases, such as gaming (UDP), IoT (MQTT), or Voice over IP, as well as for HTTP use cases that specifically require static IP addresses or deterministic, fast regional failover. Both services integrate with AWS Shield for DDoS protection.

- GA and ELB services solve the challenge of routing user requests to healthy application endpoints. AWS Global Accelerator relies on ELB to provide the traditional load balancing features such as support for internal and non-AWS endpoints, pre-warming, and Layer 7 routing. However, while ELB provides load balancing within one Region, AWS Global Accelerator provides traffic management across multiple Regions.
-

Billing

- AWS consolidated billing enables the organization to consolidate payments for multiple Amazon Web Services (AWS) accounts within a single organization by making a single paying account. Consolidated billing enables the organization to see a combined view of the AWS charges incurred by each account as well as obtain a detailed cost report for each of the individual AWS accounts associated with the paying account. The payee account will not have any other access than billing data of linked accounts. http://docs.aws.amazon.com/awsaccountbilling/latest/about/consolidatedbilling.html
- To add a particular account (linked) to the master (payee) account, the payee account has to request the linked account to join consolidated billing. Once the linked account accepts the request henceforth all charges incurred by the linked account will be paid by the payee account.
- With reference to forecasting with Cost Explorer in your AWS billing account, a forecast is a prediction of how much you will use AWS over the next three months, based on your past usage. Forecasting provides an estimate of what your AWS bill will be, which then enables you to use alarms and budgets for amounts that you are predicted to use, and because forecasts are predictions, the forecasted billing amounts are estimated, and might differ from your actual charges for each statement period.
- Billing is not region specific; The CloundWatch alarm metric isn't specific to a region ; because billing is a worldwide metric, not specific to any region.
- Total Cost of Ownership (TCO) Calculator: The Total Cost of Ownership (TCO) Calculator is a free service offered by AWS to allow you to compare the cost of on-premises servers and AWS cloud services. It requires users to answer a short series of questions and then provides a detailed report on the potential AWS cloud environment that would mirror their on-premise systems.

Disaster Recovery



- **Backup and restore:** This simple and low cost DR approach ; backs up your data and applications from anywhere to the AWS cloud for use during recovery from a disaster.
- **Pilot light:** you simply replicate part of your IT structure for a limited set of most critical core services so that the AWS cloud environment seamlessly takes over in the event of a disaster. A small part of your infrastructure is always running simultaneously syncing mutable data (as databases or documents), while other parts of your infrastructure are switched off and used only during testing. When the time comes for recovery, you can rapidly provision a full-scale production environment around the critical core.
- **Warm standby:** in which a scaled-down version of a fully functional environment is always running in the cloud. A warm standby solution extends the pilot light elements and preparation. It further

decreases the recovery time because some services are always running. By identifying your business-critical systems, you can fully duplicate these systems on AWS and have them always on.

- **Multi-site:** A multi-site solution runs on AWS as well as on your existing on-site infrastructure in an active- active configuration. The data replication method that you employ will be determined by the recovery point that you choose, either Recovery Time Objective (the maximum allowable downtime before degraded operations are restored) or Recovery Point Objective (the maximum allowable time window whereby you will accept the loss of transactions during the DR process).

APPENDIX

AWS Shared Security Model
https://aws.amazon.com/compliance/shared-responsibility-model/

White Papers - AWS training Learning Path
https://aws.amazon.com/certification/certification-prep

FAQs

1. **Amazon VPC**
2. **Amazon EC2**
3. **Amazon S3**
4. **Amazon Route 53**
5. **Amazon RDS**
6. **Amazon SQS**
7. **Elastic Load Balancer**
8. **CloudFront**
9. **Lambdas**
10. **IAM**