

**Jon Bonso and  
Adrian Formaran**



# **TUTORIALS DOJO STUDY GUIDE AND CHEAT SHEETS**

AWS Certified  
Solutions Architect  
Associate Exam





## TABLE OF CONTENTS

<b>INTRODUCTION</b>	<b>4</b>
<b>AWS CERTIFIED SOLUTIONS ARCHITECT ASSOCIATE EXAM OVERVIEW</b>	<b>5</b>
<b>AWS CERTIFIED SOLUTIONS ARCHITECT ASSOCIATE EXAM - STUDY GUIDE AND TIPS</b>	<b>9</b>
Study Materials	9
New SAA-C02 AWS Services That You Should Prepare For	10
Core AWS Services to Focus On	11
Validate Your Knowledge	12
Some notes regarding your exam	16
<b>AWS CHEAT SHEETS</b>	<b>18</b>
AWS OVERVIEW	18
AWS Global infrastructure	18
AWS Pricing	21
AWS Well-Architected Framework - Five Pillars	23
AWS Well-Architected Framework - Design Principles	26
AWS Well-Architected Framework - Disaster Recovery	30
COMPUTE	31
Amazon EC2	33
AWS Elastic Beanstalk	44
AWS Lambda	48
Amazon Elastic Container Service (ECS)	53
AWS ParallelCluster	57
Elastic Fabric Adapter (EFA)	59
STORAGE	60
Amazon S3	60
Amazon EBS	75
Amazon EFS	82
Amazon S3 Glacier	87
AWS Storage Gateway	91
Amazon FSx	95
DATABASE	98
Amazon Aurora	99
Amazon DynamoDB	107
Amazon ElastiCache	123



---

Amazon Relational Database Service (RDS)	130
Amazon Redshift	142
<b>NETWORKING AND CONTENT DELIVERY</b>	<b>145</b>
Amazon API Gateway	145
Amazon CloudFront	148
AWS DirectConnect	156
AWS Elastic Load Balancing	160
Amazon Route 53	168
Amazon VPC	180
VPC Peering	192
AWS Global Accelerator	193
AWS Transit Gateway	195
<b>SECURITY AND IDENTITY</b>	<b>196</b>
AWS Identity and Access Management (IAM)	196
AWS Directory Service	203
AWS WAF	208
Amazon Macie	211
AWS Secrets Manager	214
Amazon GuardDuty	217
AWS Shield	222
Amazon Inspector	224
AWS Organizations	227
AWS Artifact	230
<b>MIGRATION</b>	<b>232</b>
AWS Snowball Edge	232
AWS Snowmobile	235
AWS DataSync	236
<b>MANAGEMENT</b>	<b>239</b>
AWS Auto Scaling	239
AWS CloudFormation	246
AWS CloudTrail	250
Amazon CloudWatch	252
AWS OpsWorks	256
AWS Config	260
AWS Systems Manager	264
<b>ANALYTICS</b>	<b>269</b>
Amazon Kinesis	269

---



Amazon Athena	279
Amazon Elasticsearch	281
Amazon Quicksight	286
Amazon EMR	289
<b>APPLICATION</b>	<b>295</b>
Amazon SQS	295
Amazon SNS	299
Amazon SWF	303
AWS Step Functions	307
<b>DESKTOP AND APP STREAMING</b>	<b>311</b>
Amazon WorkSpaces	311
<b>COMPARISON OF AWS SERVICES</b>	<b>313</b>
S3 vs EBS vs EFS	313
Amazon S3 vs Glacier	315
S3 Standard vs S3 Standard-IA vs S3OneZone-IA	316
RDS vs DynamoDB	317
CloudTrail vs CloudWatch	320
Security Group vs NACL	321
EBS-SSD vs HDD	323
S3 Pre-Signed URLs vs CloudFront Signed URLs vs Origin Access Identity	325
CloudWatch Agent vs SSM Agent vs Custom Daemon Scripts	326
Amazon SWF vs AWS Step Function vs Amazon SQS	327
Application Load Balancer vs Network Load Balancer vs Classic Load Balancer	329
S3 Transfer Acceleration vs Direct Connect vs VPN vs Snowball vs Snowmobile	331
Service Control Policies vs IAM Policies	334
EC2 Container Services ECS vs Lambda	335
SNI Custom SSL vs Dedicated IP Custom SSL	336
EC2 Instance Health Check vs ELB Health Check vs Auto Scaling and Custom Health Check	337
Backup and Restore vs Pilot Light vs Warm Standby vs Multi - site	338
Latency Routing vs Geoproximity Routing vs Geolocation Routing	340
Multi-AZ deployments vs. Multi-Region deployments vs. Read Replicas	342
<b>FINAL REMARKS AND TIPS</b>	<b>343</b>



## INTRODUCTION

As more and more businesses migrate their on-premises workloads to Amazon Web Services (AWS), the demand for highly skilled and certified AWS Professionals will continue to rise over the coming years ahead. Companies are now leveraging on the power of cloud computing to significantly lower their operating costs and dynamically scale their resources based on demand.

Gone are the days of over-provisioning your resources that turn out to be underutilized over time. With AWS, companies can now easily provision the number of resources that they actually need and pay only the computing resources they consume. AWS helps customers to significantly reduce upfront capital investment and replace it with lower variable costs. You can opt to pay your cloud resources using an on-demand pricing option with no long-term contracts or up-front commitments. You can easily discontinue your on-demand cloud resources if you don't need them to stop any recurring operational costs, thereby reducing your operating expenses.

This flexibility isn't available in a traditional on-premises environment where you have to maintain and pay for the resources even if you aren't using them. Moreover, companies can simply launch new AWS resources in seconds to scale and accommodate the surge of incoming requests to their enterprise applications. These are the financial and technical benefits, and the reason why thousands of companies are hiring skilled IT professionals to migrate their workload to the cloud. Conversely, this is also one of the reasons why there is a demand for certified AWS professionals.

The AWS Solutions Architect Associate certification has been consistently regarded as one of the highest-paying certifications in the IT Industry today. This eBook contains essential information about the AWS Certified Solutions Architect Associate exam, as well as the topics you have to review in order to pass it. You will learn the basics of the AWS Global Infrastructure and the relevant AWS services required to build a highly available and fault-tolerant cloud architecture.

**Note:** We took extra care to come up with these study guides and cheat sheets, however, this is meant to be just a supplementary resource when preparing for the exam. We highly recommend working on hands-on sessions and practice exams to further expand your knowledge and improve your test taking skills.



## AWS CERTIFIED SOLUTIONS ARCHITECT ASSOCIATE EXAM OVERVIEW

In 2013, Amazon Web Services (AWS) began the Global Certification Program with the primary purpose of validating the technical skills and knowledge for building secure and reliable cloud-based applications using the AWS platform. By successfully passing the AWS exam, individuals can prove their AWS expertise to their current and future employers. The AWS Certified Solutions Architect - Associate exam was the first AWS certification that was launched followed by the other two role-based certifications: Systems Operations (SysOps) Administrator and Developer Associate later that year.

AWS has continuously expanded the certification program since then, launching the Professional and Specialty-level certifications that cover various domains such as machine learning, data analytics, networking, and many others. As AWS services continue to evolve, a new and updated version of the AWS certification exams are released on a regular basis to reflect the service changes and to include new knowledge areas. After almost 5 years since its initial release, an updated version of the AWS Certified Solutions Architect - Associate certification was launched in February 2018 with an exam code of SAA-C01. And after two years, in March 2020, AWS released yet another version of the exam (SAA-C02).

### Exam Details

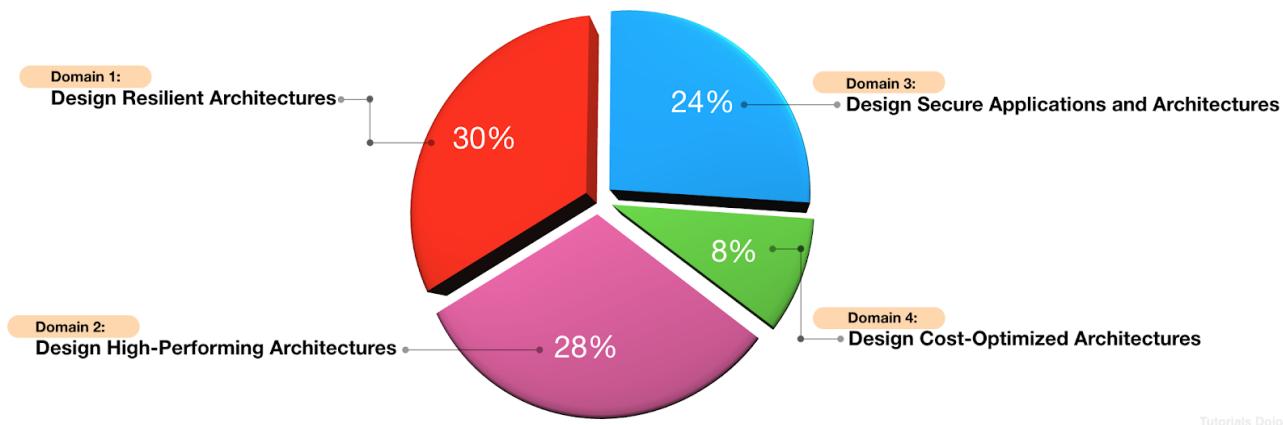
The AWS Certified Solutions Architect - Associate certification is intended for IT Professionals who perform a Solutions Architect or DevOps role and have substantial years of hands-on experience designing available, cost-efficient, fault-tolerant, and scalable distributed systems on the AWS platform. It is composed of scenario-based questions that can be either in multiple-choice or multiple response formats. The first question type has one correct answer and three incorrect responses, while the latter has two or more correct responses out of five or more options. You can take the exam from a local testing center or online from the comforts of your home.

<b>Exam Code:</b>	SAA-C02
<b>Release Date:</b>	March 2020
<b>Prerequisites:</b>	None
<b>No. of Questions:</b>	65
<b>Score Range:</b>	100/1000
<b>Cost:</b>	150 USD (Practice exam: 20 USD)
<b>Passing Score:</b>	720/1000
<b>Time Limit:</b>	2 hours 10 minutes (130 minutes)
<b>Format:</b>	Scenario-based. Multiple choice/multiple answers.
<b>Delivery Method:</b>	Testing center or online proctored exam

Don't be confused if you see in your Pearson Vue booking that the duration is 140 minutes since they included an additional 10 minutes for reading the Non-Disclosure Agreement (NDA) at the start of the exam and the survey at the end of it. If you booked in PSI, the exam duration time that you will see is 130 minutes.

## Exam Domains

The AWS Certified Solutions Architect - Associate (SAA-C02) exam has 4 different domains, each with corresponding weight and topic coverage. The exam domains are as follows: **Design Resilient Architectures** (30%), **Design High-Performing Architectures** (28%), **Design Secure Applications and Architectures** (24%), and **Design Cost-Optimized Architectures** (8%).



### Domain 1: Design Resilient Architectures (30%)

- 1.1 Design a multi-tier architecture solution
- 1.2 Design highly available and/or fault-tolerant architectures
- 1.3 Design decoupling mechanisms using AWS services
- 1.4 Choose appropriate resilient storage

### Domain 2: Design High-Performing Architectures (28%)

- 2.1 Identify elastic and scalable compute solutions for a workload
- 2.2 Select high-performing and scalable storage solutions for a workload
- 2.3 Select high-performing networking solutions for a workload
- 2.4 Choose high-performing database solutions for a workload

### Domain 3: Design Secure Applications and Architectures (24%)

- 3.1 Design secure access to AWS resources
- 3.2 Design secure application tiers
- 3.3 Select appropriate data security options



## Domain 4: Design Cost-Optimized Architectures (8%)

- 4.1 Identify cost-effective storage solutions
- 4.2 Identify cost-effective compute and database services
- 4.3 Design cost-optimized network architectures

## Exam Scoring System

You can get a score from 100 to 1,000 with a minimum passing score of **720** when you take the AWS Certified Solutions Architect - Associate exam. AWS is using a scaled scoring model to equate scores across multiple exam types that may have different difficulty levels. The complete score report will be sent to you by email after a few days. Right after you completed the actual exam, you'll immediately see a pass or fail notification on the testing screen. A "*Congratulations! You have successfully passed...*" message will be shown if you passed the exam.

Individuals who unfortunately do not pass the AWS exam must wait 14 days before they are allowed to retake the exam. Fortunately, there is no hard limit on exam attempts until you pass the exam. Take note that on each attempt, the full registration price of the AWS exam must be paid.

Within 5 business days of completing your exam, your AWS Certification Account will have a record of your complete exam results. The score report contains a table of your performance at each section/domain, which indicates whether you met the competency level required for these domains or not. AWS is using a compensatory scoring model, which means that you do not necessarily need to pass each and every individual section, only the overall examination. Each section has a specific score weighting that translates to the number of questions; hence, some sections have more questions than others. The Score Performance table highlights your strengths and weaknesses that you need to improve on.

Score Performance

Section	% of Scored Items	Needs Improvement	Meets Competencies
Section 1.0: Design Resilient Architectures	30%		
Section 2.0: Design High-Performing Architectures	28%		
Section 3.0: Design Secure Applications and Architectures	24%		
Section 4.0: Design Cost-Optimized Architectures	18%		

Tutorials Dojo



---

## Exam Benefits

If you successfully passed any AWS exam, you will be eligible for the following benefits:

- **Exam Discount** - You'll get a 50% discount voucher that you can apply for your recertification or any other exam you plan to pursue. To access your discount voucher code, go to the "Benefits" section of your AWS Certification Account, and apply the voucher when you register for your next exam.
- **Free Practice Exam** - To help you prepare for your next exam, AWS provides another voucher that you can use to take any official AWS practice exam for free. You can access your voucher code from the "Benefits" section of your AWS Certification Account.
- **AWS Certified Store** - All AWS certified professionals will be given access to exclusive AWS Certified merchandise. You can get your store access from the "Benefits" section of your AWS Certification Account.
- **Certification Digital Badges** - You can showcase your achievements to your colleagues and employers with digital badges on your email signatures, LinkedIn profile, or on your social media accounts. You can also show your Digital Badge to gain exclusive access to Certification Lounges at AWS re:Invent, regional Appreciation Receptions, and select AWS Summit events. To view your badges, simply go to the "Digital Badges" section of your AWS Certification Account.

You can visit the official AWS Certification FAQ page to view the frequently asked questions about getting AWS Certified and other information about the AWS Certification: <https://aws.amazon.com/certification/faqs/>.



## AWS CERTIFIED SOLUTIONS ARCHITECT ASSOCIATE EXAM - STUDY GUIDE AND TIPS

The [AWS Certified Solutions Architect Associate](#), or SAA for short, is one of the most sought after certifications in the Cloud industry. As a Solutions Architect, it is your responsibility to be familiar with the services that meet your customer requirements. Aside from that, you should also have the knowledge to create an efficient, secure, reliable, fault tolerant, and cost-effective infrastructure out of these services. Your AWS SA Associate exam will be based upon these topics.

Whitepapers, FAQs, and the AWS Documentation will be your primary study materials for this exam. Experience in building systems will also be helpful, since the exam consists of multiple scenario type questions. You can learn more details on your exam through the official [Exam Guide here](#). Do a quick read on it to be aware of how to prepare and what to expect on the exam itself.

### Study Materials

For the AWS Certified Solutions Architect Associate exam, we recommend going through the [FREE AWS Exam Readiness video course](#), [official AWS sample questions](#), AWS whitepapers, FAQs, [AWS cheat sheets](#), and [AWS practice exams](#).

The screenshot shows a slide from the AWS Certified Solutions Architect - Associate - Module 1 training course. The title is "Fault Tolerance". The text on the slide states: "The more loosely your system is coupled, the more easily it scales and the more fault-tolerant it can be." Below this text are two diagrams illustrating system coupling:

- Tightly coupled:** Shows three red "Web servers" boxes connected by a dense web of bidirectional arrows to three yellow "App servers" boxes below them.
- Loosely coupled:** Shows three orange "Web servers" boxes connected to a single central "Load balancer" box, which in turn connects to three yellow "App servers" boxes below it.

At the bottom of the slide, there is a navigation bar with icons for back, forward, and search, along with "PREV" and "NEXT" buttons.



---

We recommend that you read the following whitepapers for your review. They contain a lot of important concepts.

1. **Overview of Amazon Web Services**: This paper provides a good introduction on Cloud Computing, the AWS Global Infrastructure, and the available AWS Services. Reading this whitepaper before proceeding to the other whitepapers below will clear up many jargons found on the succeeding materials.
2. **AWS Well Architected Framework**: This paper is the most important one to read. It discusses the Five Pillars of a Well Architected Framework, with each pillar having a whitepaper of its own, and can all be found on this [webpage](#). Be sure to understand well architected framework not just conceptually, but also in actual practice and application.
3. **AWS Best Practices**: This paper teaches you the best practices to perform when running your applications in AWS. It points out the advantages of Cloud over traditional hosting infrastructures and how you can implement them to keep your applications up and running all the time. The SA Associate exam will include questions that will test your knowledge on the best practices through different example scenarios.
4. **Using Amazon Web Services for Disaster Recovery**: This paper explains the different types of disaster recovery plans that you can perform in AWS. It is your responsibility as a Solutions Architect to mitigate any potential downtime when disaster strikes. Depending on your RPO and RTO, a proper disaster recovery plan will be a deciding factor between business continuity and revenue loss.

### Additional Helpful Papers

1. **AWS Security Practices**: This paper supplements your study on the AWS services and features such as IAM, Security Groups, nACLs, etc. You should read this paper since security specific questions occasionally pop up in the exam.
2. **AWS Storage Services Overview**: This paper supplements your study on the different AWS Storage options such as S3, EBS, EFS, Glacier, etc. It contains a good detail of information and comparison for each storage service, which is crucial in knowing the best service to use for a situation.
3. **Building Fault-Tolerant Applications on AWS**: This paper discusses the many ways you can ensure your applications are fault-tolerant in AWS. It also contains multiple scenarios where the practices are applied and which AWS services were crucial for the scenario.

### New SAA-C02 AWS Services That You Should Prepare For

For the SAA-C02 exam version, you should also know the following services:

- AWS Global Accelerator
- Elastic Fabric Adapter (EFA)
- Elastic Network Adapter (ENA)
- AWS ParallelCluster
- Amazon FSx
- AWS DataSync



- AWS Directory Service
- High Performance Computing
- Aurora Serverless

... plus a few more services that we have recently added to our [AWS Certified Solutions Architect Associate Practice Exams](#).

You can also check out the SAA-C02 official exam guide [here](#).

## Core AWS Services to Focus On

1. **EC2** - As the most fundamental compute service offered by AWS, you should know about EC2 inside out.
2. **Lambda** - Lambda is the common service used for serverless applications. Study how it is integrated with other AWS services to build a full stack serverless app.
3. **Elastic Load Balancer** - Load balancing is very important for a highly available system. Study about the different types of ELBs, and the features each of them supports.
4. **Auto Scaling** - Study what services in AWS can be auto scaled, what triggers scaling, and how auto scaling increases/decreases the number of instances.
5. **Elastic Block Store** - As the primary storage solution of EC2, study on the types of EBS volumes available. Also study how to secure, backup and restore EBS volumes.
6. **S3 / Glacier** - AWS offers many types of S3 storage depending on your needs. Study what these types are and what differs between them. Also review on the capabilities of S3 such as hosting a static website, securing access to objects using policies, lifecycle policies, etc. Learn as much about S3 as you can.
7. **Storage Gateway** - There are occasional questions about Storage Gateway in the exam. You should understand when and which type of Storage Gateway should be used compared to using services like S3 or EBS.
8. **EFS** - EFS is a service highly associated with EC2, much like EBS. Understand when to use EFS, compared to using S3, EBS or instance store. Exam questions involving EFS usually ask the trade off between cost and efficiency of the service compared to other storage services.
9. **RDS / Aurora** - Know how each RDS database differs from one another, and how they are different from Aurora. Determine what makes Aurora unique, and when it should be preferred from other databases (in terms of function, speed, cost, etc). Learn about parameter groups, option groups, and subnet groups.
10. **DynamoDB** - The exam includes lots of DynamoDB questions, so read as much about this service as you can. Consider how DynamoDB compares to RDS, Elasticache and Redshift. This service is also commonly used for serverless applications along with Lambda.
11. **Elasticache** - Familiarize yourself with Elasticache redis and its functions. Determine the areas/services where you can place a caching mechanism to improve data throughput, such as managing session state of an ELB, optimizing RDS instances, etc.



- 
12. **VPC / NACL and Security Groups** - Study every service that is used to create a VPC (subnets, route tables, internet gateways, nat gateways, VPN gateways, etc). Also, review on the differences of network access control lists and security groups, and during which situations they are applied.
  13. **Route 53** - Study the different types of records in Route 53. Study also the different routing policies. Know what hosted zones and domains are.
  14. **IAM** - Services such as IAM Users, Groups, Policies and Roles are the most important to learn. Study how IAM integrates with other services and how it secures your application through different policies. Also read on the best practices when using IAM.
  15. **CloudWatch** - Study how monitoring is done in AWS and what types of metrics are sent to CloudWatch. Also read upon Cloudwatch Logs, CloudWatch Alarms, and the custom metrics made available with CloudWatch Agent.
  16. **CloudTrail** - Familiarize yourself with how CloudTrail works, and what kinds of logs it stores as compared to CloudWatch Logs.
  17. **Kinesis** - Read about Kinesis sharding and Kinesis Data Streams. Have a high level understanding of how each type of Kinesis Stream works.
  18. **CloudFront** - Study how CloudFront helps speed up websites. Know what content sources CloudFront can serve from. Also check the kinds of certificates CloudFront accepts.
  19. **SQS** - Gather info on why SQS is helpful in decoupling systems. Study how messages in the queues are being managed (standard queues, FIFO queues, dead letter queues). Know the differences between SQS, SNS, SES, and Amazon MQ.
  20. **SNS** - Study the function of SNS and what services can be integrated with it. Also be familiar with the supported recipients of SNS notifications.
  21. **SWF / CloudFormation / OpsWorks** - Study how these services function. Differentiate the capabilities and use cases of each of them. Have a high level understanding of the kinds of scenarios they are usually used in.

The AWS Documentation and FAQs will be your primary source of information. Use them together with the Tutorials Dojo cheat sheets to learn the important details about these services. Lastly, try out these services yourself by signing up in AWS and performing some [lab exercises](#). Experiencing them on your own will help you in remembering what each service is capable of.

## Validate Your Knowledge

When you are feeling confident with your review, it is best to validate your knowledge through sample exams. You can take [this practice exam](#) from AWS for free as additional material, but do not expect your real exam to be on the same level of difficulty as this practice exam on the AWS website. [Tutorials Dojo](#) offers a very useful and well-reviewed set of practice tests for Solutions Architect Associate exam takers [here](#). Each test contains many unique questions which will surely help you verify if you have missed out on anything important that might appear on your exam.



If you have scored well on the [Tutorials Dojo AWS Certified Solutions Architect Associate practice tests](#) and you think you are ready, then go earn your certification with your head held high. If you think you are lacking in certain areas, better go review them again, and take note of any hints in the questions that will help you select the correct answers. If you are not that confident that you'll pass, then it would be best to reschedule your exam to another day, and take your time preparing for it. In the end, the efforts you have put in for this will surely reward you.

The screenshot shows a course page with the following sections:

- COURSE CONTENT**: A list of course modules:
  - Practice Exams - Timed Mode (1 Topic | 6 Quizzes)
  - Practice Exams - Review Mode (1 Topic | 6 Quizzes)
  - Practice Exams - Section-Based (1 Topic | 4 Quizzes)
  - Practice Exams - Topic-Based (1 Topic | 13 Quizzes)
  - Final Test (1 Topic | 1 Quiz)
- BONUS SECTION - FLASHCARDS**

To the right, there is a promotional box for the course:

- aws certified Solutions Architect Associate**
- Tutorials Dojo**
- Not Enrolled**
- Take this Course**
- \$14.99 USD**
- COURSE INCLUDES**
  - 6 Lessons
  - 5 Topics
  - 30 Quizzes

## Sample Practice Test Questions:

### Question 1

A Solutions Architect is designing an online medical system in AWS which will store sensitive Personally Identifiable Information (PII) of the users in an Amazon S3 bucket. Both the master keys and the unencrypted data should never be sent to AWS to comply with the strict compliance and regulatory requirements of the company.

Which S3 encryption technique should the Architect use?

1. Use S3 client-side encryption with a KMS-managed customer master key.
2. Use S3 client-side encryption with a client-side master key.
3. Use S3 server-side encryption with a KMS managed key.
4. Use S3 server-side encryption with customer provided key.



---

## Correct Answer: 2

**Client-side encryption** is the act of encrypting data before sending it to Amazon S3. To enable client-side encryption, you have the following options:

- Use an AWS KMS-managed customer master key.
- Use a client-side master key.

When using an AWS KMS-managed customer master key to enable client-side data encryption, you provide an AWS KMS customer master key ID (CMK ID) to AWS. On the other hand, when you use client-side master key for client-side data encryption, your client-side master keys and your unencrypted data are never sent to AWS. It's important that you safely manage your encryption keys because if you lose them, you can't decrypt your data.

This is how client-side encryption using client-side master key works:

**When uploading an object** - You provide a client-side master key to the Amazon S3 encryption client. The client uses the master key only to encrypt the data encryption key that it generates randomly. The process works like this:

1. The Amazon S3 encryption client generates a one-time-use symmetric key (also known as a data encryption key or data key) locally. It uses the data key to encrypt the data of a single Amazon S3 object. The client generates a separate data key for each object.
2. The client encrypts the data encryption key using the master key that you provide. The client uploads the encrypted data key and its material description as part of the object metadata. The client uses the material description to determine which client-side master key to use for decryption.
3. The client uploads the encrypted data to Amazon S3 and saves the encrypted data key as object metadata (x-amz-meta-x-amz-key) in Amazon S3.

**When downloading an object** - The client downloads the encrypted object from Amazon S3. Using the material description from the object's metadata, the client determines which master key to use to decrypt the data key. The client uses that master key to decrypt the data key and then uses the data key to decrypt the object.

Hence, the correct answer is Option 2.

Option 1 is incorrect because in client-side encryption with a KMS-managed customer master key, you provide an AWS KMS customer master key ID (CMK ID) to AWS. The scenario clearly indicates that both the master keys and the unencrypted data should never be sent to AWS.

Option 3 is incorrect because the scenario mentioned that the unencrypted data should never be sent to AWS, which means that you have to use client-side encryption in order to encrypt the data first before sending to



---

AWS. In this way, you can ensure that there is no unencrypted data being uploaded to AWS. In addition, the master key used by Server-Side Encryption with AWS KMS-Managed Keys (SSE-KMS) is uploaded and managed by AWS, which directly violates the requirement of not uploading the master key.

Option 4 is incorrect because just as mentioned in Option 3, you have to use client-side encryption in this scenario instead of server-side encryption. For the S3 server-side encryption with customer-provided key (SSE-C), you actually provide the encryption key as part of your request to upload the object to S3. Using this key, Amazon S3 manages both the encryption (as it writes to disks) and decryption (when you access your objects).

#### References:

<https://docs.aws.amazon.com/AmazonS3/latest/dev/UsingEncryption.html>

<https://docs.aws.amazon.com/AmazonS3/latest/dev/UsingClientSideEncryption.html>

## Question 2

You have launched a new enterprise application with a web server and a database. You are using a large EC2 Instance with one 500 GB EBS volume to host a relational database. Upon checking the performance, it shows that write throughput to the database needs to be improved.

Which of the following is the most suitable configuration to help you achieve this requirement? (Choose 2)

1. Set up a standard RAID 0 configuration with two EBS Volumes
2. Re-launch the instance with a Paravirtual (PV) AMI and enable Enhanced Networking
3. Use a standard RAID 1 configuration with two EBS Volumes
4. Set up the EC2 instance in a placement group
5. Increase the size of the EC2 Instance

#### Correct Answers: 1,5

The goal here is to increase the write performance of the database hosted in an EC2 instance. You can achieve this by either setting up a standard RAID 0 configuration or simply by increasing the size of the EC2 instance.

Some EC2 instance types can drive more I/O throughput than what you can provision for a single EBS volume. You can join multiple gp2, io1, st1, or sc1 volumes together in a RAID 0 configuration to use the available bandwidth for these instances.

With Amazon EBS, you can use any of the standard RAID configurations that you can use with a traditional bare metal server, as long as that particular RAID configuration is supported by the operating system for your instance. This is because all RAID is accomplished at the software level. For greater I/O performance than you



---

can achieve with a single volume, RAID 0 can stripe multiple volumes together; for on-instance redundancy, RAID 1 can mirror two volumes together.

Take note that HVM AMIs are required to take advantage of enhanced networking and GPU processing. In order to pass through instructions to specialized network and GPU devices, the OS needs to be able to have access to the native hardware platform which the HVM virtualization provides.

Option 2 is incorrect because although the Enhanced Networking feature can provide higher I/O performance and lower CPU utilization to your EC2 instance, you have to use an HVM AMI instead of PV AMI.

Option 3 is incorrect because the main use case for 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.

Option 4 is incorrect because the placement groups feature is primarily used for inter-instance communication.

#### References:

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/raid-config.html>

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EBSPerformance.html>

<https://aws.amazon.com/ec2/features/#enhanced-networking>

Check out our [\*\*AWS Certified Solutions Architect Associate Practice Exams\*\*](#) for more practice test questions in six different training modes, all of which have detailed explanations, references, and cheat sheets.

## Some notes regarding your exam

The Solutions Architect Associate exam loves to end questions that ask for highly available or cost-effective solutions. Be sure to understand the choices provided to you, and verify that they have correct details. Some choices are very misleading such that it seems it is the most appropriate answer to the question, but contains an incorrect detail about some service.

If you are unsure which options are correct in a multi-select question, try to eliminate some of the choices that you believe are false. This will help narrow down the feasible answers to that question. The same goes for multiple choice type questions. Be extra careful as well when selecting the number of answers you submit.

As mentioned in this review, you should be able to differentiate services that belong in one category with one another. Common comparisons include:

- EC2 vs ECS vs Lambda
- S3 vs EBS vs EFS



- CloudFormation vs OpsWorks vs Elastic Beanstalk
- SQS vs SNS vs SES vs MQ
- Security Group vs nACLs
- The different S3 storage types vs Glacier
- RDS vs DynamoDB vs ElastiCache
- RDS engines vs Aurora

The [Tutorials Dojo Comparison of AWS Services](#) contains excellent cheat sheets comparing these seemingly similar services which are crucial to solving the tricky scenario-based questions in the actual exam. By knowing each service's capabilities and use cases, you can consider these types of questions already half-solved.

Lastly, be on the lookout for “key terms” that will help you realize the answer much faster. Words such as millisecond latency, serverless, managed, highly available, most cost effective, fault tolerant, mobile, streaming, object storage, archival, polling, push notifications, etc are commonly seen in the exam. Time management is very important when taking AWS certification exams, so be sure to monitor the time you consume for each question.



## AWS CHEAT SHEETS

### AWS OVERVIEW

#### AWS Global infrastructure

- The AWS Global infrastructure is built around **Regions** and **Availability Zones (AZs)**.
- Regions** provide multiple, physically separated and isolated **Availability Zones** which are connected with low latency, high throughput, and highly redundant networking

The screenshot shows the AWS Management Console homepage. At the top right, a dropdown menu is open, showing the selected region "N. Virginia". A yellow circle highlights this selection. Below the menu, a large yellow circle contains the text "List of other AWS Regions". The "All services" sidebar on the left lists various AWS services under categories like Compute, Containers, and Management & Governance. The main content area displays service icons and names.

N. Virginia is the default AWS Region

List of other AWS Regions

Region	Code
US East (N. Virginia)	us-east-1
US East (Ohio)	us-east-2
US West (N. California)	us-west-1
US West (Oregon)	us-west-2
Africa (Cape Town)	af-south-1
Asia Pacific (Hong Kong)	ap-east-1
Asia Pacific (Mumbai)	ap-south-1
Asia Pacific (Seoul)	ap-northeast-2
Asia Pacific (Singapore)	ap-southeast-1
Asia Pacific (Sydney)	ap-southeast-2
Asia Pacific (Tokyo)	ap-northeast-1
Canada (Central)	ca-central-1
Europe (Frankfurt)	eu-central-1
Europe (Ireland)	eu-west-1
Europe (London)	eu-west-2
Europe (Milan)	eu-south-1
Europe (Paris)	eu-west-3
Europe (Stockholm)	eu-north-1
Middle East (Bahrain)	me-south-1
South America (São Paulo)	sa-east-1

- Availability Zones** offer highly availability, fault tolerance, and scalability.
  - Consist of one or more discrete data centers, each with redundant power, networking, and connectivity, housed in separate facilities.
  - An Availability Zone is represented by a **region code** followed by a **letter identifier**; for example, us-east-1a.



- An **AWS Local Zone** is a single datacenter designed to complement an existing AWS Region. It places AWS compute, storage, database, and other select services closer to large population, industry, and IT centers where no AWS Region exists today.
- **Edge locations** are locations where end users access services located at AWS.
- **Points of Presence** are locations which consist of Edge Locations and Regional Edge Cache servers.

View the Interactive AWS Global Infrastructure Map [here](#).

The screenshot shows the AWS Subnet creation interface. At the top, the region is set to "N. Virginia". The "Availability Zone" dropdown is open, showing "No preference" as the selected option. A red arrow points from the text "Region you are in" to the "N. Virginia" region selector. Another red arrow points from the "Availability Zone" dropdown to the list of availability zones below. The list is titled "Availability Zones under N. Virginia" and includes the following data:

Name	ID
No preference	
us-east-1a	use1-az2
us-east-1b	use1-az4
us-east-1c	use1-az6
us-east-1d	use1-az1
us-east-1e	use1-az3
us-east-1f	use1-az5



## Distribution Settings

The screenshot shows the 'Distribution Settings' section of the AWS CloudFront console. It includes fields for 'Price Class', 'AWS WAF Web ACL', and 'Alternate Domain Names (CNAMEs)'. A red box highlights the 'Edge Locations under CloudFront' dropdown menu, which contains three options: 'Use All Edge Locations (Best Performance)', 'Use Only U.S., Canada and Europe', and 'Use U.S., Canada, Europe, Asia, Middle East and Africa'. Below this, the 'SSL Certificate' section shows 'Default CloudFront Certificate (\*.cloudfront.net)' selected, with a note about HTTPS support. It also includes a 'Request or Import a Certificate with ACM' button and links to learn more about custom SSL/TLS certificates and ACM.

Price Class: Use All Edge Locations (Best Performance) ⓘ

AWS WAF Web ACL: Use Only U.S., Canada and Europe  
Use U.S., Canada, Europe, Asia, Middle East and Africa  
Use All Edge Locations (Best Performance)

Alternate Domain Names (CNAMEs): Edge Locations under CloudFront

SSL Certificate: Default CloudFront Certificate (\*.cloudfront.net)  
Choose this option if you want your users to use HTTPS or HTTP to access your content with the CloudFront domain name (such as <https://d111111abcdef8.cloudfront.net/logo.jpg>).  
Important: If you choose this option, CloudFront requires that browsers or devices support TLSv1 or later to access your content.

Custom SSL Certificate (example.com):  
Choose this option if you want your users to access your content by using an alternate domain name, such as <https://www.example.com/logo.jpg>. You can use a certificate stored in AWS Certificate Manager (ACM) in the US East (N. Virginia) Region, or you can use a certificate stored in IAM.

Request or Import a Certificate with ACM ⓘ

Learn more about using custom SSL/TLS certificates with CloudFront.  
Learn more about using ACM.

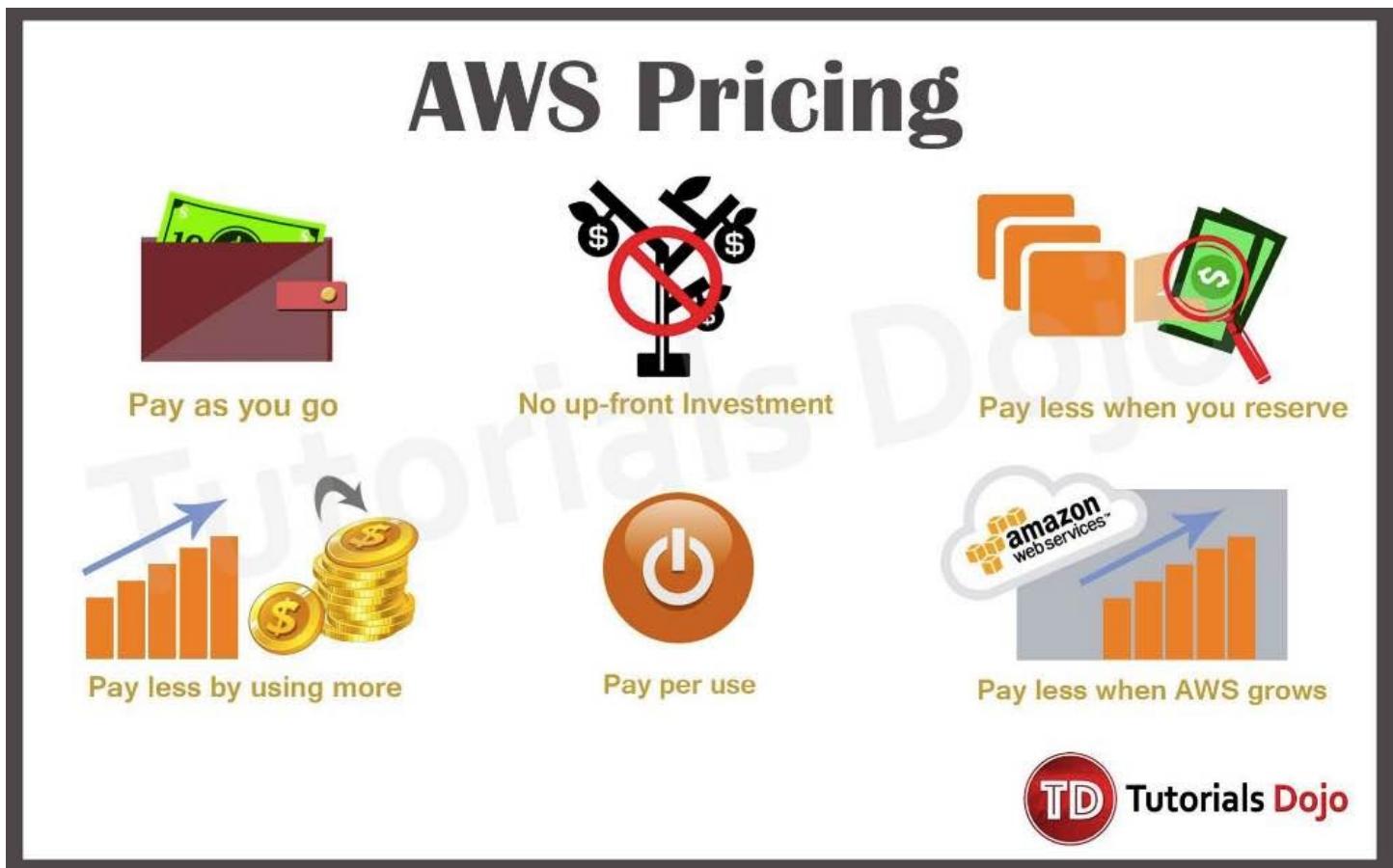
## Sources:

<https://aws.amazon.com/about-aws/global-infrastructure/>  
<https://docs.aws.amazon.com/aws-technical-content/latest/aws-overview/global-infrastructure.html>  
<https://www.infrastructure.aws/>



## AWS Pricing

- There are three fundamental drivers of cost with AWS:
  - Compute
  - Storage
  - Outbound data transfer.
- AWS offers pay-as-you-go for pricing.



- For certain services like **Amazon EC2** and **Amazon RDS**, you can invest in reserved capacity. With Reserved Instances, you can save up to 75% over equivalent on-demand capacity. When you buy Reserved Instances, the larger the upfront payment, the greater the discount.
  - With the **All Upfront** option, you pay for the entire Reserved Instance term with one upfront payment. This option provides you with the largest discount compared to On-Demand instance pricing.
  - With the **Partial Upfront** option, you make a low upfront payment and are then charged a discounted hourly rate for the instance for the duration of the Reserved Instance term.



- The **No Upfront** option does not require any upfront payment and provides a discounted hourly rate for the duration of the term.
- There are also volume-based discounts for services such as **Amazon S3**.
- For new accounts, AWS Free Tier is available.
  - Free Tier offers limited usage of AWS products at no charge for 12 months since the account was created. More details at <https://aws.amazon.com/free/>.
- You can estimate your monthly AWS bill using [\*\*AWS Pricing Calculator\*\*](#).
- The [\*\*AWS TCO calculator\*\*](#)
  - gives you the option to evaluate the savings from using AWS
  - compares costs and savings against on-premises and co-location environments
  - matches your current infrastructure to the most cost-effective AWS offering

**Sources:**

[https://d1.awsstatic.com/whitepapers/aws\\_pricing\\_overview.pdf](https://d1.awsstatic.com/whitepapers/aws_pricing_overview.pdf)

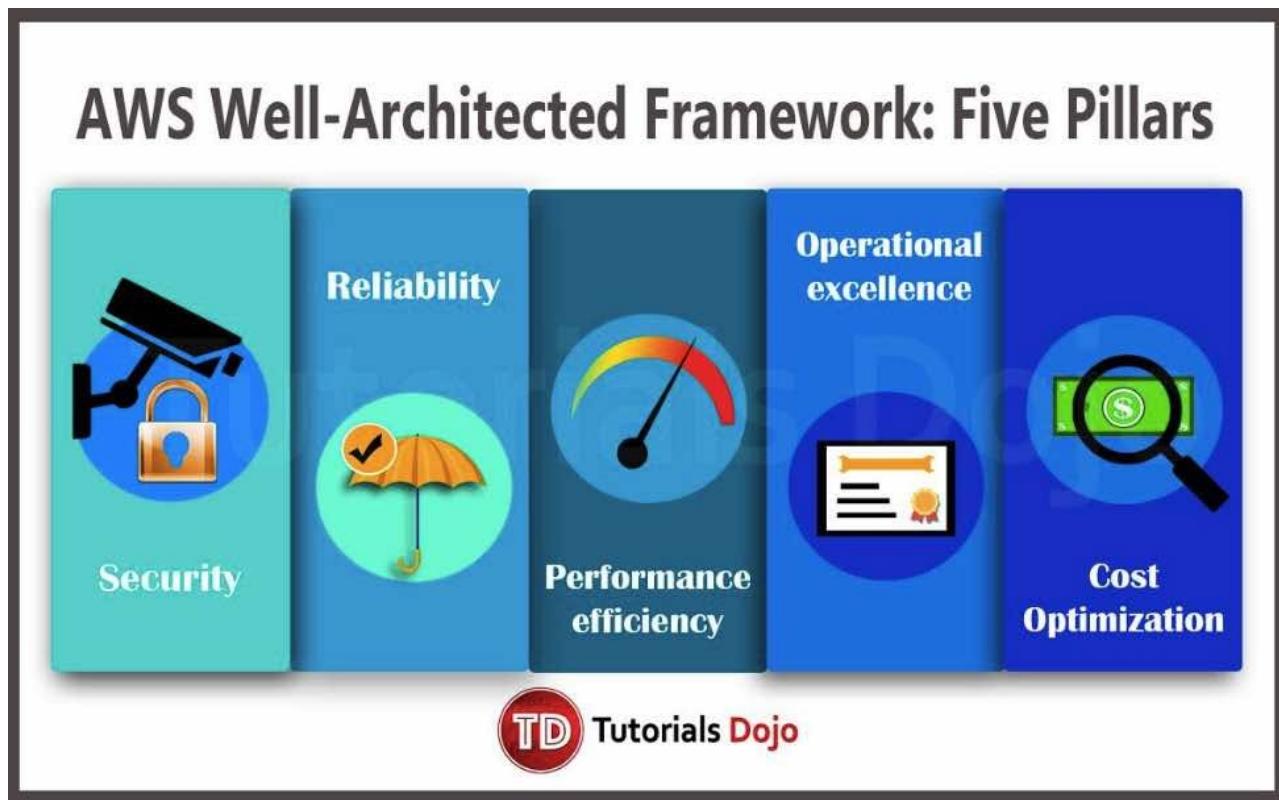
<https://aws.amazon.com/pricing/>

<https://aws.amazon.com/ec2/pricing/reserved-instances/pricing/>



## AWS Well-Architected Framework - Five Pillars

Having well-architected systems greatly increases the plausibility of business success which is why AWS created the AWS Well-Architected Framework. This framework is composed of five pillars that help you understand the pros and cons of decisions you make while building cloud architectures and systems on the AWS platform. You will learn the architectural best practices for designing and operating reliable, efficient, cost-effective and secure systems in the cloud by using the framework. It also provides a way to consistently measure your architectures against best practices and identify areas for improvement.



### 1. Operational Excellence

- The ability to run and monitor systems to deliver business value and to continually improve supporting processes and procedures.
- There are three best practice areas and tools for operational excellence in the cloud:
  - Prepare - AWS Config
  - Operate - Amazon CloudWatch
  - Evolve - Amazon Elasticsearch Service
- Key AWS service:
  - AWS CloudFormation for creating templates. (See AWS Management Tools Cheat Sheet)



## 2. Security

- The ability to protect information, systems, and assets while delivering business value through risk assessments and mitigation strategies.
- There are five best practice areas and tools for security in the cloud:
  - Identity and Access Management - IAM, Multi-Factor Authentication, AWS Organizations
  - Detective Controls - AWS CloudTrail, AWS Config, Amazon GuardDuty
  - Infrastructure Protection - Amazon VPC, Amazon CloudFront with AWS Shield, AWS WAF
  - Data Protection - ELB, Amazon Elastic Block Store (Amazon EBS), Amazon S3, and Amazon Relational Database Service (Amazon RDS) encryption, Amazon Macie, AWS Key Management Service (AWS KMS)
  - Incident Response - IAM, Amazon CloudWatch Events
- Key AWS service:
  - AWS Identity and Access Management (IAM)

## 3. Reliability

- The ability of a system to recover from infrastructure or service disruptions, dynamically acquire computing resources to meet demand, and mitigate disruptions such as misconfigurations or transient network issues.
- There are three best practice areas and tools for reliability in the cloud:
  - Foundations - IAM, Amazon VPC, AWS Trusted Advisor, AWS Shield
  - Change Management - AWS CloudTrail, AWS Config, Auto Scaling, Amazon CloudWatch
  - Failure Management - AWS CloudFormation, Amazon S3, AWS KMS, Amazon Glacier
- Key AWS service:
  - Amazon CloudWatch

## 4. Performance Efficiency

- The ability to use computing resources efficiently to meet system requirements, and to maintain that efficiency as demand changes and technologies evolve.
- There are four best practice areas for performance efficiency in the cloud:
  - Selection - Auto Scaling for Compute, Amazon EBS and S3 for Storage, Amazon RDS and DynamoDB for Database, Route53, VPC, and AWS Direct Connect for Network
  - Review - AWS Blog and What's New section of the website
  - Monitoring - Amazon CloudWatch
  - Tradeoffs - Amazon ElastiCache, Amazon CloudFront, AWS Snowball, Amazon RDS read replicas.
- Key AWS service:
  - Amazon CloudWatch

## 5. Cost Optimization



- The ability to avoid or eliminate unneeded cost or suboptimal resources.
- There are four best practice areas and tools for cost optimization in the cloud:
  - Cost-Effective Resources - Cost Explorer, Amazon CloudWatch and Trusted Advisor, Amazon Aurora for RDS, AWS Direct Connect with Amazon CloudFront
  - Matching supply and demand - Auto Scaling
  - Expenditure Awareness - AWS Cost Explorer, AWS Budgets
  - Optimizing Over Time - AWS News Blog and the What's New section on the AWS website, AWS Trusted Advisor
- Key AWS service:
  - Cost Explorer

**Source:**

[https://d1.awsstatic.com/whitepapers/architecture/AWS\\_Well-Architected\\_Framework.pdf](https://d1.awsstatic.com/whitepapers/architecture/AWS_Well-Architected_Framework.pdf)



## AWS Well-Architected Framework - Design Principles

### 1. Scalability

- **Scaling Horizontally** - an increase in the number of resources
- **Scaling Vertically** - an increase in the specifications of an individual resource

### 2. Disposable Resources Instead of Fixed Servers

- **Instantiating Compute Resources** - automate setting up of new resources along with their configuration and code
- **Infrastructure as Code** - AWS assets are programmable. You can apply techniques, practices, and tools from software development to make your whole infrastructure reusable, maintainable, extensible, and testable.

### 3. Automation

- **Serverless Management and Deployment** - being serverless shifts your focus to automation of your code deployment. AWS handles the management tasks for you.
- **Infrastructure Management and Deployment** - AWS automatically handles details, such as resource provisioning, load balancing, auto scaling, and monitoring, so you can focus on resource deployment.
- **Alarms and Events** - AWS services will continuously monitor your resources and initiate events when certain metrics or conditions are met.

### 4. Loose Coupling

- **Well-Defined Interfaces** - reduce interdependencies in a system by allowing various components to interact with each other only through specific, technology agnostic interfaces, such as RESTful APIs.
- **Service Discovery** - applications that are deployed as a set of smaller services should be able to be consumed without prior knowledge of their network topology details. Apart from hiding complexity, this also allows infrastructure details to change at any time.
- **Asynchronous Integration** - interacting components that do not need an immediate response and where an acknowledgement that a request has been registered will suffice, should integrate through an intermediate durable storage layer.
- **Distributed Systems Best Practices** - build applications that handle component failure in a graceful manner.

### 5. Services, Not Servers

- **Managed Services** - provide building blocks that developers can consume to power their applications, such as databases, machine learning, analytics, queuing, search, email, notifications, and more.



- **Serverless Architectures** - allow you to build both event-driven and synchronous services without managing server infrastructure, which can reduce the operational complexity of running applications.

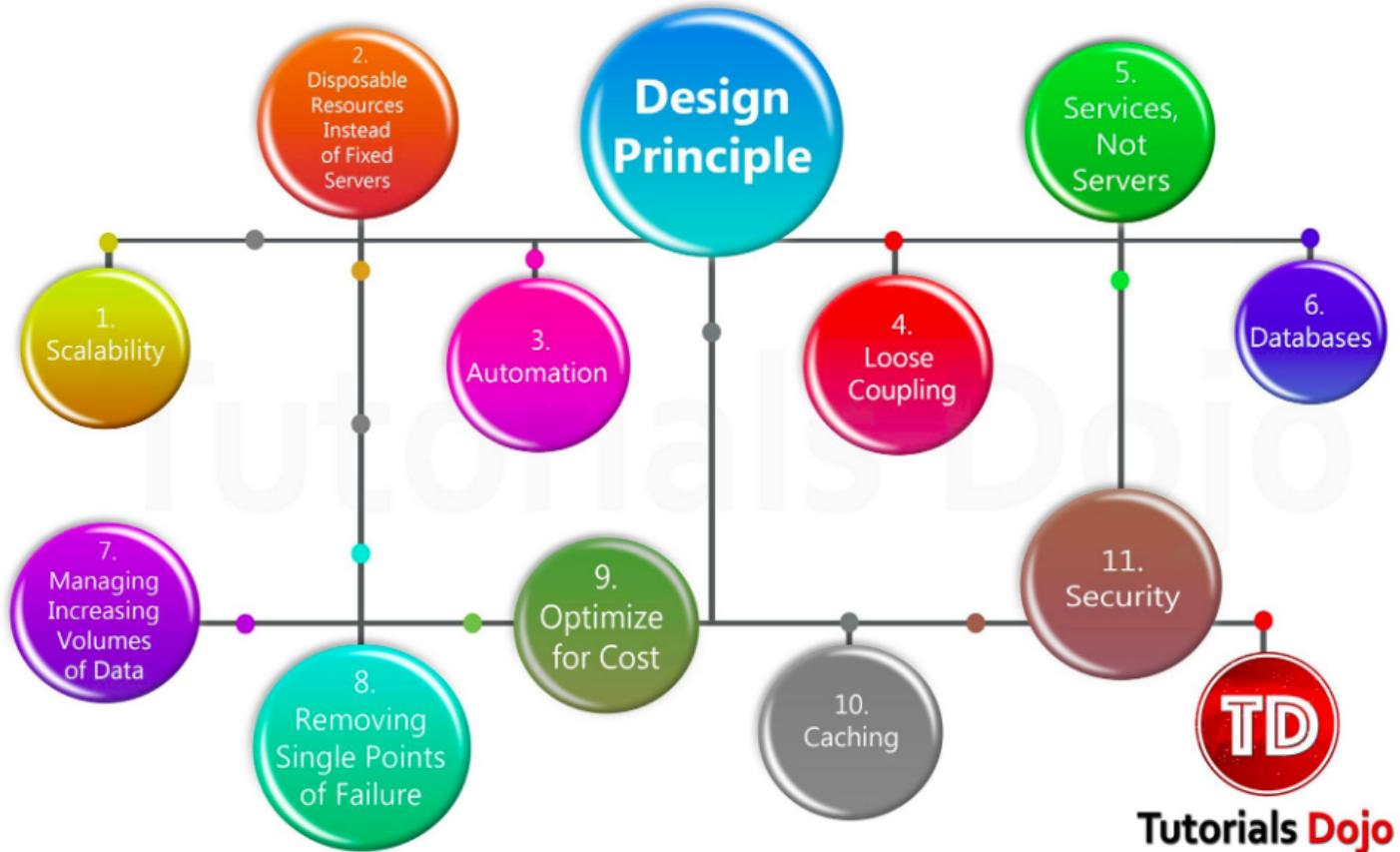
## 6. Databases

- Choose the Right Database Technology for Each Workload
- **Relational Databases** provide a powerful query language, flexible indexing capabilities, strong integrity controls, and the ability to combine data from multiple tables in a fast and efficient manner.
- **NoSQL Databases** trade some of the query and transaction capabilities of relational databases for a more flexible data model that seamlessly scales horizontally. It uses a variety of data models, including graphs, key-value pairs, and JSON documents, and are widely recognized for ease of development, scalable performance, high availability, and resilience.
- **Data Warehouses** are a specialized type of relational database, which is optimized for analysis and reporting of large amounts of data.
- **Graph Databases** uses graph structures for queries.
  - Search Functionalities
    - Search is often confused with query. A query is a formal database query, which is addressed in formal terms to a specific data set. Search enables datasets to be queried that are not precisely structured.
    - A search service can be used to index and search both structured and free text format and can support functionality that is not available in other databases, such as customizable result ranking, faceting for filtering, synonyms, and stemming.

## 7. Managing Increasing Volumes of Data

- **Data Lake** - an architectural approach that allows you to store massive amounts of data in a central location so that it's readily available to be categorized, processed, analyzed, and consumed by diverse groups within your organization.

## AWS Well-Architected Framework



### 8. Removing Single Points of Failure

- **Introducing Redundancy**
  - **Standby redundancy** - when a resource fails, functionality is recovered on a secondary resource with the failover process. The failover typically requires some time before it completes, and during this period the resource remains unavailable. This is often used for stateful components such as relational databases.
  - Active redundancy - requests are distributed to multiple redundant compute resources. When one of them fails, the rest can simply absorb a larger share of the workload.
- **Detect Failure** - use health checks and collect logs
- **Durable Data Storage**
  - **Synchronous replication** - only acknowledges a transaction after it has been durably stored in both the primary storage and its replicas. It is ideal for protecting the integrity of data from the event of a failure of the primary node.



- **Asynchronous replication** - decouples the primary node from its replicas at the expense of introducing replication lag. This means that changes on the primary node are not immediately reflected on its replicas.
- **Quorum-based replication** - combines synchronous and asynchronous replication by defining a minimum number of nodes that must participate in a successful write operation.
- **Automated Multi-Data Center Resilience** - utilize AWS Regions and Availability Zones (Multi-AZ Principle). (See Disaster Recovery section)
- **Fault Isolation and Traditional Horizontal Scaling** - Shuffle Sharding

## 9. Optimize for Cost

- **Right Sizing** - AWS offers a broad range of resource types and configurations for many use cases.
- **Elasticity** - save money with AWS by taking advantage of the platform's elasticity.
- **Take Advantage of the Variety of Purchasing Options** - Reserved Instances vs Spot Instances (See AWS Pricing)

## 10. Caching

- **Application Data Caching** - store and retrieve information from fast, managed, in-memory caches.
- **Edge Caching** - serve content by infrastructure that is closer to viewers, which lowers latency and gives high, sustained data transfer rates necessary to deliver large popular objects to end users at scale.

## 11. Security

- **Use AWS Features for Defense in Depth** - secure multiple levels of your infrastructure from network down to application and database.
- **Share Security Responsibility with AWS** - AWS handles security OF the Cloud while customers handle security IN the Cloud.
- **Reduce Privileged Access** - implement Principle of Least Privilege controls.
- **Security as Code** - firewall rules, network access controls, internal/external subnets, and operating system hardening can all be captured in a template that defines a *Golden Environment*.
- **Real-Time Auditing** - implement continuous monitoring and automation of controls on AWS to minimize exposure to security risks.

### Source:

[https://d0.awsstatic.com/whitepapers/AWS\\_Cloud\\_Best\\_Practices.pdf](https://d0.awsstatic.com/whitepapers/AWS_Cloud_Best_Practices.pdf)



## AWS Well-Architected Framework - Disaster Recovery

- **RTO** is the time it takes after a disruption to restore a business process to its service level.
- **RPO** is the acceptable amount of data loss measured in time before the disaster occurs.
- Disaster Recovery With AWS
  - **Backup and Restore** - storing backup data on S3 and recover data quickly and reliably.
  - **Pilot Light** for Quick Recovery into AWS - quicker recovery time than backup and restore because core pieces of the system are already running and are continually kept up to date.
  - **Warm Standby** Solution - a scaled-down version of a fully functional environment is always running in the cloud
  - **Multi-Site** Solution - run your infrastructure on another site, in an active-active configuration.
  - AWS Production to an AWS DR Solution **Using Multiple AWS Regions** - take advantage of AWS' multiple availability zones
- Services
  - **S3** as a destination for backup data that might be needed quickly to perform a restore.
  - **Import/Export** for transferring very large data sets by shipping storage devices directly to AWS.
  - **Glacier** for longer-term data storage where retrieval times of several hours are adequate.
  - **Storage Gateway** copies snapshots of your on-premises data volumes to S3 for backup. You can create local volumes or EBS volumes from these snapshots.
  - Preconfigured servers bundled as **Amazon Machine Images (AMIs)**.
  - **Elastic Load Balancing (ELB)** for distributing traffic to multiple instances.
  - **Route 53** for routing production traffic to different sites that deliver the same application or service.
  - **Elastic IP address** for static IP addresses.
  - **Virtual Private Cloud (VPC)** for provisioning a private, isolated section of the AWS cloud.
  - **Direct Connect** for a dedicated network connection from your premises to AWS.
  - **Relational Database Service (RDS)** for scale of a relational database in the cloud.
  - **DynamoDB** for a fully managed NoSQL database service to store and retrieve any amount of data and serve any level of request traffic.
  - **Redshift** for a petabyte-scale data warehouse service that analyzes all your data using existing business intelligence tools.
  - **CloudFormation** for creating a collection of related AWS resources and provision them in an orderly and predictable fashion.
  - **Elastic Beanstalk** is a service for deploying and scaling web applications and services developed.
  - **OpsWorks** is an application management service for deploying and operating applications of all types and sizes.

### Source:

<https://www.slideshare.net/AmazonWebServices/disaster-recovery-options-with-aws>

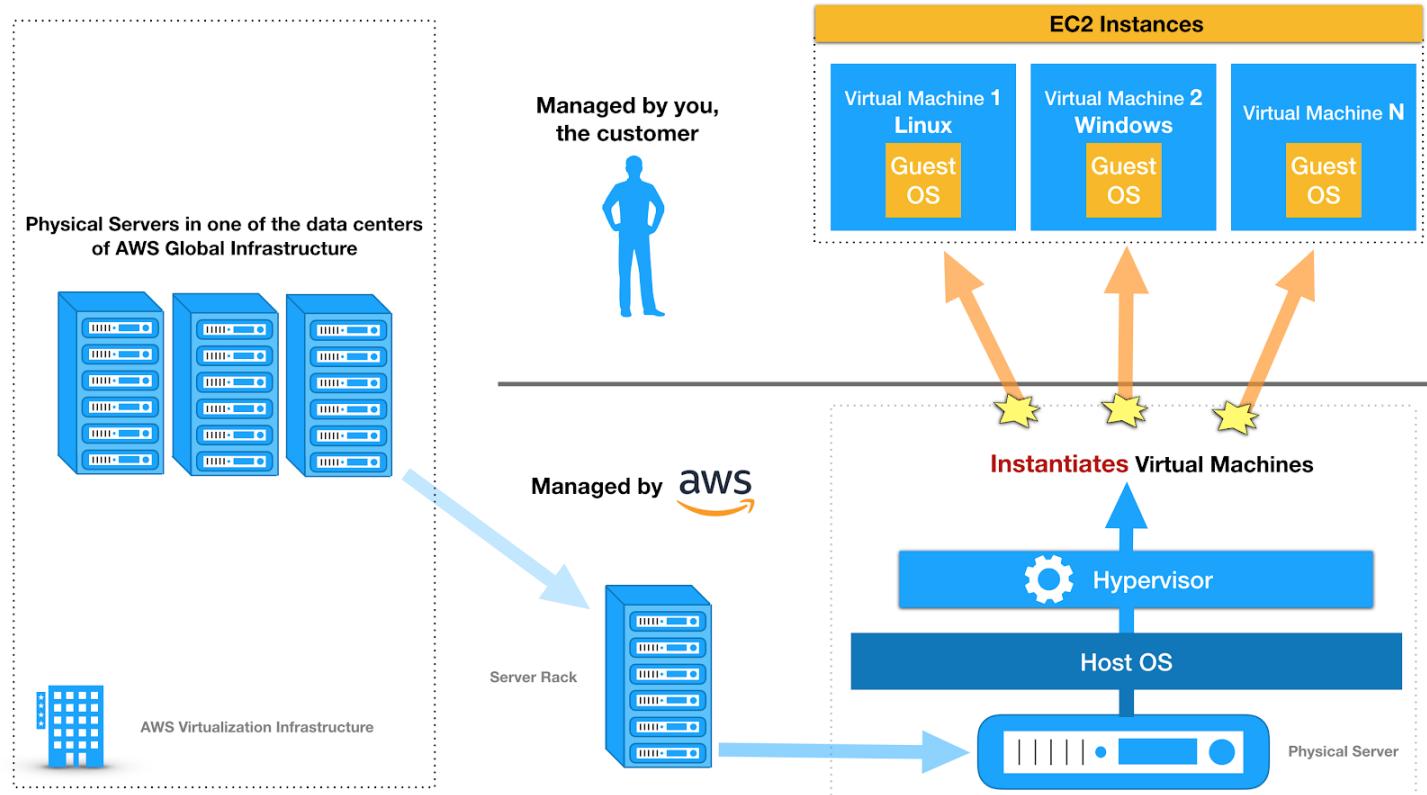


## COMPUTE

AWS provides a variety of cost-effective and flexible computing services to meet the needs of your organization such as Amazon Elastic Compute Cloud (EC2), Amazon Elastic Container Service (ECS), Amazon Elastic Container Service for Kubernetes (EKS), Amazon Lightsail, AWS Batch, and AWS Lambda to name a few. For some services like Amazon EC2, you have extensive control of the underlying resources while for others, AWS has full control.

With these computing services in AWS, you can dynamically provision a number of resources and pay only the computing resources you actually consume. This significantly reduces the upfront capital investment required and replaces it with lower variable costs. Instead of the traditional long-term contracts or up-front commitments, you can opt to pay your compute resources in AWS using an On-Demand or Spot pricing option to easily discontinue your cloud resources if you don't need them, effectively reducing your operating expenses. Amazon EC2 is a commonly used AWS service which you can integrate with various features and services like Amazon Machine Image, Instance Store, Elastic Block Store, Elastic Network Interface, Elastic IP, Auto Scaling, Elastic Load Balancer, Placements Groups, Enhanced Networking, Security Groups and so much more.

Have you ever heard people say “Amazon Linux EC2 **Instance**” instead of “Amazon Linux EC2 **Server**” when they launch a compute resource in AWS? It is because AWS is programmatically creating a new virtual machine (VM) **instance**, rather than providing you with an actual physical **server**, when you launch an EC2 Instance. AWS has a powerful virtualization infrastructure that is composed of physical servers that they manage. Each physical server has a host operating system that runs a virtual machine monitor (VMM), also known as a hypervisor, which instantiates multiple VM “instances” that you can use. These instances use guest operating systems that you can manage.



AWS manages, operates, and controls the components from the host operating system and virtualization layer down to the physical security of the facilities in which the service operates. Conversely, the customer is responsible for the management of the guest operating system such as installing patches and doing the necessary security configuration.

You can also use these compute services in AWS to run your High Performance Computing (HPC) applications. Basically, HPC requires a higher storage I/O and large amounts of memory to perform a complex task. Moving your HPC workloads to AWS eliminates the unnecessary wait times and long job queues that are associated with limited on-premises HPC resources. Since there are no upfront capital expenditures or lengthy procurement cycles, you can get significant cost savings whenever you process time-flexible, stateless workloads.



## Amazon EC2

- A Linux-based/Windows-based virtual server that you can provision.
- You are limited to running On-Demand Instances per your vCPU-based On-Demand Instance limit, purchasing 20 Reserved Instances, and requesting Spot Instances per your dynamic Spot limit per region.

### Features

- The **AWS Nitro System** is the underlying platform of the next generation of EC2 instances. Traditionally, hypervisors protect the physical hardware and bios, virtualize the CPU, storage, networking, and provide a rich set of management capabilities. With the Nitro System, these functions are offloaded to dedicated hardware and software, thereby reducing costs of your instances in the process. Hence, the Nitro Hypervisor delivers performance that is indistinguishable from bare metal.
- Server environments called **instances**.
- Package OS and additional installations in a reusable template called **Amazon Machine Images**.
- Various configurations of CPU, memory, storage, and networking capacity for your instances, known as **instance types**
  - t-type and m-type for general purpose
  - c-type for compute optimized
  - r-type, x-type and z-type for memory optimized
  - d-type, h-type and i-type for storage optimized
  - f-type, g-type and p-type for accelerated computing
- Secure login information for your instances using **key pairs**
- Storage volumes for temporary data that are deleted when you STOP or TERMINATE your instance, known as **instance store volumes**. Take note that you can stop an EBS-backed instance but not an Instance Store-backed instance. You can only either start or terminate an Instance Store-backed instance.
- Persistent storage volumes for your data using **Elastic Block Store volumes** (see aws storage services).
- Multiple physical locations for deploying your resources, such as instances and EBS volumes, known as **regions** and **Availability Zones** (see AWS overview).
- A firewall that enables you to specify the protocols, ports, and source IP ranges that can reach your instances using **security groups** (see aws networking and content delivery).
- Static IPv4 addresses for dynamic cloud computing, known as **Elastic IP addresses** (see aws networking and content delivery).
- Metadata, known as **tags**, that you can create and assign to your EC2 resources
- Virtual networks you can create that are logically isolated from the rest of the AWS cloud, and that you can optionally connect to your own network, known as **virtual private clouds** or **VPCs** (see aws networking and content delivery).
- Add a script that will be run on instance boot called **user-data**.



- **Host Recovery for Amazon EC2** automatically restarts your instances on a new host in the event of an unexpected hardware failure on a Dedicated Host.
- **EC2 Hibernation** is available for On-Demand and Reserved Instances running on freshly launched M3, M4, M5, C3, C4, C5, R3, R4, and R5 instances running Amazon Linux and Ubuntu 18.04 LTS. You can enable hibernation for your EBS-backed instances at launch. You can then hibernate and resume your instances through the AWS Management Console, or though the AWS SDK and CLI using the existing stop-instances and start-instances commands. Hibernation requires an EC2 instance be an encrypted EBS-backed instance.

## Instance States

- **Start** - run your instance normally. You are continuously billed while your instance is running.
- **Stop** - is just a normal instance shutdown. You may restart it again anytime. All EBS volumes remain attached, but data in instance store volumes are deleted. You won't be charged for usage while instance is stopped. You can attach or detach EBS volumes. You can also create an AMI from the instance, and change the kernel, RAM disk, and instance type while in this state.
- **Terminate** - instance performs a normal shutdown and gets deleted. You won't be able to restart an instance once you terminate it. The root device volume is deleted by default, but any attached EBS volumes are preserved by default. Data in instance store volumes are deleted.
- To prevent accidental termination, *disable instance termination*.

## Root Device Volumes

- The root device volume contains the image used to boot the instance.
- Instance Store-backed Instances
  - Any data on the instance store volumes is deleted when the instance is terminated (instance store-backed instances do not support the Stop action) or if it fails (such as if an underlying drive has issues).
  - You should also back up critical data from your instance store volumes to persistent storage on a regular basis.
- Amazon EBS-backed Instances
  - An Amazon EBS-backed instance can be stopped and later restarted without affecting data stored in the attached volumes.
  - When in a stopped state, you can modify the properties of the instance, change its size, or update the kernel it is using, or you can attach your root volume to a different running instance for debugging or any other purpose.
  - By default, the root device volume for an AMI backed by Amazon EBS is deleted when the instance terminates.

## AMI

- Includes the following:



- A template for the root volume for the instance (OS, application server, and applications)
- Launch permissions that control which AWS accounts can use the AMI to launch instances
- A block device mapping that specifies the volumes to attach to the instance when it's launched
- Backed by Amazon EBS - root device for an instance launched from the AMI is an Amazon EBS volume. AMIs backed by Amazon EBS snapshots can use EBS encryption.
- Backed by Instance Store - root device for an instance launched from the AMI is an instance store volume created from a template stored in S3.

Characteristic	Amazon EBS-Backed AMI	Amazon Instance Store-Backed AMI
Boot time for an instance	Usually less than 1 minute	Usually less than 5 minutes
Size limit for a root device	16 TiB	10 GiB
Root device volume	Amazon EBS volume	Instance store volume
Data persistence	By default, the root volume is deleted when the instance terminates.* Data on any other Amazon EBS volumes persists after instance termination by default.	Data on any instance store volumes persists only during the life of the instance.
Modifications	The instance type, kernel, RAM disk, and user data can be changed while the instance is stopped.	Instance attributes are fixed for the life of an instance.
Charges	You're charged for instance usage, Amazon EBS volume usage, and storing your AMI as an Amazon EBS snapshot.	You're charged for instance usage and storing your AMI in Amazon S3.
AMI creation/bundling	Uses a single command/call	Requires installation and use of AMI tools
Stopped state	Can be placed in stopped state where instance is not running, but the root volume is persisted in Amazon EBS	Cannot be in stopped state; instances are running or terminated

- You can copy AMIs to different regions.

## Pricing

- On-Demand - pay for the instances that you use by the second, with no long-term commitments or upfront payments.
- Reserved - make a low, one-time, up-front payment for an instance, reserve it for a one- or three-year term, and pay a significantly lower hourly rate for these instances. It has two offering classes: Standard and Convertible. The Standard class provides the most significant discount but you can only modify



some of its attributes during the term. It can also be sold in the Reserved Instance Marketplace. The Convertible class provides a lower discount than Standard Reserved Instances, but can be exchanged for another Convertible Reserved Instance with different instance attributes. However, this one cannot be sold in the Reserved Instance Marketplace.

- Spot - request unused EC2 instances, which can lower your costs significantly. Spot Instances are available at up to a 90% discount compared to On-Demand prices.

Spot Instances		On-Demand Instances
Launch time	Can only be launched immediately if the Spot Request is active and capacity is available.	Can only be launched immediately if you make a manual launch request and capacity is available.
Available capacity	If capacity is not available, the Spot Request continues to automatically make the launch request until capacity becomes available.	If capacity is not available when you make a launch request, you get an insufficient capacity error (ICE).
Hourly price	The hourly price for Spot Instances varies based on demand.	The hourly price for On-Demand Instances is static.
Instance interruption	You can't stop and start an Amazon EBS-backed Spot Instance; only the Amazon EC2 Spot service can do this. The Amazon EC2 Spot service can interrupt an individual Spot Instance if capacity is no longer available, the Spot price exceeds your maximum price, or demand for Spot Instances increases.	You determine when an On-Demand Instance is interrupted (stopped or terminated).

- 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.
- There is a data transfer charge when copying AMI from one region to another
- EBS pricing is different from instance pricing. (see AWS storage services)
- AWS imposes a small hourly charge if an Elastic IP address is not associated with a running instance, or if it is associated with a stopped instance or an unattached network interface.
- You are charged for any additional Elastic IP addresses associated with an instance.
- If data is transferred between these two instances, it is charged at "Data Transfer Out from EC2 to Another AWS Region" for the first instance and at "Data Transfer In from Another AWS Region" for the second instance.

## Security



- Use IAM to control access to your instances (see AWS Security and Identity Service).
  - IAM policies
  - IAM roles
- Restrict access by only allowing trusted hosts or networks to access ports on your instance.
- A **security group** acts as a virtual firewall that controls the traffic for one or more instances.
  - Create different security groups to deal with instances that have different security requirements.
  - You can add rules to each security group that allow traffic to or from its associated instances.
  - You can modify the rules for a security group at any time.
  - New rules are automatically applied to all instances that are associated with the security group.
  - Evaluates all the rules from all the security groups that are associated with an instance to decide whether to allow traffic or not.
  - By default, security groups allow **all outbound traffic**.
  - Security group rules are **always permissive**; you can't create rules that deny access.
  - Security groups are **stateful**
- If you don't specify a security group when you launch an instance, the instance is automatically associated with the **default security group** for the VPC, which has the following rules:
  - Allows all inbound traffic from other instances associated with the default security group

## Details

Security group name	Security group ID	Description
default	sg-03ad7f0ed3e0053ee	default VPC security group

Owner	Inbound rules count	Outbound rules count
[REDACTED]	1 Permission entry	1 Permission entry

Inbound rules	Outbound rules	Tags	
<b>Inbound rules</b>			
<b>Type</b>			
Type	Protocol	Port range	Source
All traffic	All	All	sg-03ad7f0ed3e0053ee (default)



- Allows all outbound traffic from the instance.

### Details

Security group name default	Security group ID sg-03ad7f0ed3e0053ee	Description default VPC security group
Owner [REDACTED]	Inbound rules count 1 Permission entry	Outbound rules count 1 Permission entry

Inbound rules    **Outbound rules**    Tags

### Outbound rules

Type	Protocol	Port range	Destination
All traffic	All	All	0.0.0.0/0

- Disable password-based logins for instances launched from your AMI, since passwords can be cracked or found.
- You can replicate the network traffic from an EC2 instance within your Amazon VPC and forward that traffic to security and monitoring appliances for content inspection, threat monitoring, and troubleshooting.

## Networking

- An **Elastic IP address** is a static IPv4 address designed for dynamic cloud computing. With it, you can mask the failure of an instance or software by rapidly remapping the address to another instance in your account.
- If you have not enabled auto-assign public IP address for your instance, you need to associate an Elastic IP address with your instance to enable communication with the internet.
- An Elastic IP address is for use in a specific region only.
- By default, all AWS accounts are limited to five (5) Elastic IP addresses per region, because public (IPv4) internet addresses are a scarce public resource.
- By default EC2 instances come with a private IP.



- An elastic **network interface** is a logical networking component in a VPC that represents a virtual network card, which directs traffic to your instance
- Every instance in a VPC has a default network interface, called the **primary network interface** (eth0). You cannot detach a primary network interface from an instance.
- You can create and attach additional network interfaces. The maximum number of network interfaces that you can use varies by instance type.
- You can attach a network interface to an instance in a different subnet as long as its within the same AZ
- Default interfaces are terminated with instance termination.
- Scale with **EC2 Scaling Groups** and distribute traffic among instances using **Elastic Load Balancer**.
- You can configure EC2 instances as **bastion hosts** (aka jump boxes) in order to access your VPC instances for management, using SSH or RDP protocols

## Monitoring

- EC2 items to monitor
  - CPU utilization, Network utilization, Disk performance, Disk Reads/Writes using EC2 metrics
  - Memory utilization, disk swap utilization, disk space utilization, page file utilization, log collection using a monitoring agent/CloudWatch Logs
- Automated monitoring tools include:
  - System Status Checks - monitor the AWS systems required to use your instance to ensure they are working properly. These checks detect problems with your instance that require AWS involvement to repair.
  - Instance Status Checks - monitor the software and network configuration of your individual instance. These checks detect problems that require your involvement to repair.
  - Amazon CloudWatch Alarms - watch a single metric over a time period you specify, and perform one or more actions based on the value of the metric relative to a given threshold over a number of time periods.
  - Amazon CloudWatch Events - automate your AWS services and respond automatically to system events.
  - Amazon CloudWatch Logs - monitor, store, and access your log files from Amazon EC2 instances, AWS CloudTrail, or other sources.
- Monitor your EC2 instances with CloudWatch. By default, EC2 sends metric data to CloudWatch in 5-minute periods.

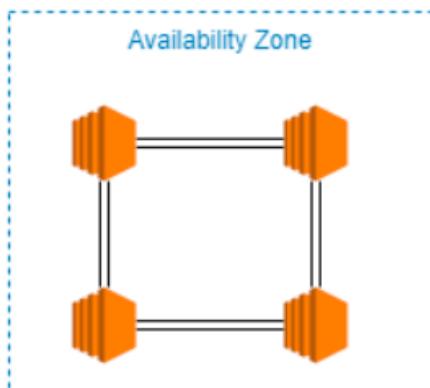
## Instance Metadata and User Data

- **Instance metadata** is data about your instance that you can use to configure or manage the running instance.
- Instance metadata and user data are not protected by cryptographic methods.

- View all categories of instance metadata from within a running instance at <http://169.254.169.254/latest/meta-data/>
- You can pass two types of user data to EC2: shell scripts and cloud-init directives.
- User data is limited to 16 KB.
- If you stop an instance, modify its user data, and start the instance, the updated user data is not executed when you start the instance.
- Retrieve user data from within a running instance at <http://169.254.169.254/latest/user-data>

## Placement Groups

- You can launch or start instances in a **placement group**, which determines how instances are placed on underlying hardware.
  - Cluster - clusters instances into a low-latency group in a single Availability Zone. Recommended for applications that benefit from low network latency, high network throughput, or both, and if the majority of the network traffic is between the instances in the group.
  - Spread - spreads instances across underlying hardware. Recommended for applications that have a small number of critical instances that should be kept separate from each other. Note: A spread placement group can span multiple Availability Zones, and you can have a maximum of seven running instances per Availability Zone per group.
- Partition placement groups is an Amazon EC2 placement strategy that helps reduce the likelihood of correlated failures for large distributed and replicated workloads such as HDFS, HBase and Cassandra running on EC2.



Cluster Placement



Spread Placement

- Partition placement groups spread EC2 instances across logical partitions and ensure that instances in different partitions do not share the same underlying hardware. In addition, partition placement groups



---

offer visibility into the partitions and allow topology aware applications to use this information to make intelligent data replication decisions, increasing data availability and durability.

## Rules

- The name you specify for a placement group must be unique within your AWS account for the region.
- You can't merge placement groups.
- An instance can be launched in one placement group at a time; it cannot span multiple placement groups.
- Instances with a tenancy of host cannot be launched in placement groups.

## Storage

- **EBS** (see AWS Storage Services)
  - Provides durable, block-level storage volumes that you can attach to a running instance.
  - Use as a primary storage device for data that requires frequent and granular updates.
  - To keep a backup copy of your data, create a snapshot of an EBS volume, which is stored in S3. You can create an EBS volume from a snapshot, and attach it to another instance.
- **Instance Store**
  - Provides temporary block-level storage for instances.
  - The data on an instance store volume persists only during the life of the associated instance; if you stop or terminate an instance, any data on instance store volumes is lost.
- **Elastic File System (EFS)** (see AWS Storage Services)
  - 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.
- **S3** (see AWS Storage Services)
  - Provides access to reliable and inexpensive data storage infrastructure.
  - Storage for EBS snapshots and instance store-backed AMIs.
- **Resources and Tagging**
  - EC2 resources include images, instances, volumes, and snapshots. When you create a resource, AWS assigns the resource a *unique resource ID*.
  - Some resources can be used in all regions (global), and some resources are specific to the region or Availability Zone in which they reside.



Resource	Type	Description
AWS account	Global	You can use the same AWS account in all regions.
Key pairs	Global or Regional	The key pairs that you create using EC2 are tied to the region where you created them. You can create your own RSA key pair and upload it to the region in which you want to use it; therefore, you can make your key pair globally available by uploading it to each region.
Amazon EC2 resource identifiers	Regional	Each resource identifier, such as an AMI ID, instance ID, EBS volume ID, or EBS snapshot ID, is tied to its region and can be used only in the region where you created the resource.
User-supplied resource names	Regional	Each resource name, such as a security group name or key pair name, is tied to its region and can be used only in the region where you created the resource. Although you can create resources with the same name in multiple regions, they aren't related to each other.
AMIs	Regional	An AMI is tied to the region where its files are located within S3. You can copy an AMI from one region to another.
Elastic IP addresses	Regional	An Elastic IP address is tied to a region and can be associated only with an instance in the same region.
Security groups	Regional	A security group is tied to a region and can be assigned only to instances in the same region. You can't enable an instance to communicate with an instance outside its region using security group rules.
EBS snapshots	Regional	An EBS snapshot is tied to its region and can only be used to create volumes in the same region. You can copy a snapshot from one region to another.
EBS volumes	Availability Zone	An EBS volume is tied to its Availability Zone and can be attached only to instances in the same Availability Zone.
Instances	Availability Zone	An instance is tied to the Availability Zones in which you launched it. However, its instance ID is tied to the region.



- You can optionally assign your own metadata to each resource with **tags**, which consists of a key and an optional value that you both define.

## Networking

- **Enhanced Networking** - It provides higher bandwidth, higher packet per second (PPS) performance, and consistent lower inter-instance latencies, which is being used in Placement Groups. It uses single root I/O virtualization (SR-IOV) to provide high-performance networking capabilities. SR-IOV is a method of device virtualization that provides higher I/O performance and lower CPU utilization when compared to traditional virtualized network interfaces.
- **Elastic Fabric Adapter (EFA)** - This is a network device that you can attach to your EC2 instance to significantly accelerate machine learning applications and High Performance Computing (HPC). It empowers your computing resources to achieve the application performance of an on-premises HPC cluster, with the elasticity and scalability provided by AWS. Compared with a TCP transport that is traditionally used in cloud-based HPC systems, EFA provides lower and more consistent latency and higher throughput as it enhances the performance of inter-instance communication.

## Sources:

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/>

<https://aws.amazon.com/ec2/features/>

<https://aws.amazon.com/ec2/pricing/>

<https://aws.amazon.com/ec2/faqs/>



## AWS Elastic Beanstalk

- Allows you to quickly deploy and manage applications in the AWS Cloud without worrying about the infrastructure that runs those applications.
- Elastic Beanstalk automatically handles the details of capacity provisioning, load balancing, scaling, and application health monitoring for your applications.
- It is a Platform-as-a-Service
- Elastic Beanstalk supports the following languages:
  - Go
  - Java
  - .NET
  - Node.js
  - PHP
  - Python
  - Ruby
- Elastic Beanstalk supports the following web containers:
  - Tomcat
  - Passenger
  - Puma
- Elastic Beanstalk supports Docker containers.
- Your application's domain name is in the format: `subdomain.region.elasticbeanstalk.com`

## Environment Pages

- The **Configuration** page shows the resources provisioned for this environment. This page also lets you configure some of the provisioned resources.
- The **Health** page shows the status and detailed health information about the EC2 instances running your application.
- The **Monitoring** page shows the statistics for the environment, such as average latency and CPU utilization. You also use this page to create alarms for the metrics that you are monitoring.
- The **Events** page shows any informational or error messages from services that this environment is using.
- The **Tags** page shows tags – key-value pairs that are applied to resources in the environment. You use this page to manage your environment's tags.



All Applications > My First Elastic Beanstalk Application > MyFirstElasticBeanstalkApplication-env (Environment ID: e-xd5gu4u3r7, URL: testapp1077642.us-west-2.elasticbeanstalk.com) Actions ▾

Dashboard Overview Refresh

Configuration

Logs

Health

Monitoring

Alarms

Managed updates

Events

Tags

Recent Events

Health Ok Causes

Running Version Sample Application Upload and Deploy

Platform Python 3.6 running on 64bit Amazon Linux/2.9.7 Change

Show All

Time	Type	Details
2020-04-05 12:44:13 UTC+0800	INFO	Environment health has transitioned from Pending to Ok. Initialization completed 16 seconds ago and took 2 minutes.
2020-04-05 12:43:28 UTC+0800	INFO	Successfully launched environment: MyFirstElasticBeanstalkApplication-env
2020-04-05 12:43:28 UTC+0800	INFO	Application available at testapp1077642.us-west-2.elasticbeanstalk.com.
2020-04-05 12:43:13 UTC+0800	INFO	Added instance [i-0d9d496e37d436518] to your environment.
2020-04-05 12:42:42 UTC+0800	INFO	Waiting for EC2 instances to launch. This may take a few minutes.

## Elastic Beanstalk Concepts

- **Application** - a logical collection of Elastic Beanstalk components, including environments, versions, and environment configurations. It is conceptually similar to a folder.
- **Application Version** - refers to a specific, labeled iteration of deployable code for a web application. An application version points to an Amazon S3 object that contains the deployable code. Applications can have many versions and each application version is unique.
- **Environment** - a version that is deployed on to AWS resources. Each environment runs only a single application version at a time, however you can run the same version or different versions in many environments at the same time.
- **Environment Tier** - determines whether Elastic Beanstalk provisions resources to support an application that handles HTTP requests or an application that pulls tasks from a queue. An application that serves HTTP requests runs in a **web server environment**. An environment that pulls tasks from an Amazon SQS queue runs in a **worker environment**.
- **Environment Configuration** - identifies a collection of parameters and settings that define how an environment and its associated resources behave.
- Configuration Template - a starting point for creating unique environment configurations.
- There is a limit to the number of application versions you can have. You can avoid hitting the limit by applying an *application version lifecycle policy* to your applications to tell Elastic Beanstalk to delete application versions that are old, or to delete application versions when the total number of versions for an application exceeds a specified number.

## Environment Types

- Load-balancing, Autoscaling Environment - automatically starts additional instances to accommodate increasing load on your application.



- Single-Instance Environment - contains one Amazon EC2 instance with an Elastic IP address.

## Environment Configurations

- Your environment contains:
  - Your **EC2 virtual machines** configured to run web apps on the platform that you choose.
  - An **Auto Scaling group** that ensures that there is always one instance running in a single-instance environment, and allows configuration of the group with a range of instances to run in a load-balanced environment.
  - When you enable load balancing, Elastic Beanstalk creates an **Elastic Load Balancing load balancer** to distributes traffic among your environment's instances.
  - Elastic Beanstalk provides integration with **Amazon RDS** to help you add a database instance to your Elastic Beanstalk environment : **MySQL, PostgreSQL, Oracle, or SQL Server**. When you add a database instance to your environment, Elastic Beanstalk provides connection information to your application by setting environment properties for the database hostname, port, user name, password, and database name.
  - You can use **environment properties** to pass secrets, endpoints, debug settings, and other information to your application. Environment properties help you run your application in multiple environments for different purposes, such as development, testing, staging, and production.
  - You can configure your environment to use **Amazon SNS** to notify you of important events that affect your application.
  - Your environment is available to users at a **subdomain of elasticbeanstalk.com**. When you create an environment, you can choose a unique subdomain that represents your application.

## Monitoring

- Elastic Beanstalk Monitoring console displays your environment's status and application health at a glance.
- Elastic Beanstalk reports the health of a web server environment depending on how the application running in it responds to the health check.
- **Enhanced health reporting** is a feature that you can enable on your environment to allow AWS Elastic Beanstalk to gather additional information about resources in your environment. Elastic Beanstalk analyzes the information gathered to provide a better picture of overall environment health and aid in the identification of issues that can cause your application to become unavailable.
- You can create alarms for metrics to help you monitor changes to your environment so that you can easily identify and mitigate problems before they occur.
- EC2 instances in your Elastic Beanstalk environment generate logs that you can view to troubleshoot issues with your application or configuration files.

## Security



- When you create an environment, Elastic Beanstalk prompts you to provide two AWS IAM roles: a **service role** and an **instance profile**.
  - Service Roles - assumed by Elastic Beanstalk to use other AWS services on your behalf.
  - Instance Profiles - applied to the instances in your environment and allows them to retrieve application versions from S3, upload logs to S3, and perform other tasks that vary depending on the environment type and platform.
- User Policies - allow users to create and manage Elastic Beanstalk applications and environments.

## Pricing

- There is no additional charge for Elastic Beanstalk. You pay only for the underlying AWS resources that your application consumes.

## Sources:

<https://docs.aws.amazon.com/elasticbeanstalk/latest/dg>

<https://aws.amazon.com/elasticbeanstalk/details/>

<https://aws.amazon.com/elasticbeanstalk/pricing/>

<https://aws.amazon.com/elasticbeanstalk/faqs/>



## AWS Lambda

- A serverless compute service.
- Lambda executes your code only when needed and scales automatically.
- Lambda functions are stateless - no affinity to the underlying infrastructure.
- You choose the amount of memory you want to allocate to your functions and AWS Lambda allocates proportional CPU power, network bandwidth, and disk I/O.
- AWS Lambda is SOC, HIPAA, PCI, ISO compliant.
- Supports the following languages:
  - Node.js
  - Java
  - C#
  - Go
  - Python
  - Ruby

## Components of a Lambda Application

- **Function** – a script or program that runs in Lambda. Lambda passes invocation events to your function. The function processes an event and returns a response.
- **Runtimes** – Lambda runtimes allow functions in different languages to run in the same base execution environment. The runtime sits in-between the Lambda service and your function code, relaying invocation events, context information, and responses between the two.
- **Layers** – Lambda layers are a distribution mechanism for libraries, custom runtimes, and other function dependencies. Layers let you manage your in-development function code independently from the unchanging code and resources that it uses.
- **Event source** – an AWS service or a custom service that triggers your function and executes its logic.
- **Downstream resources** – an AWS service that your Lambda function calls once it is triggered.
- **Log streams** – While Lambda automatically monitors your function invocations and reports metrics to CloudWatch, you can annotate your function code with custom logging statements that allow you to analyze the execution flow and performance of your Lambda function.
- AWS Serverless Application Model

## Lambda Functions

- You upload your application code in the form of one or more *Lambda functions*. Lambda stores code in Amazon S3 and encrypts it at rest.
- To create a Lambda function, you first package your code and dependencies in a deployment package. Then, you upload the deployment package to create your Lambda function.
- After your Lambda function is in production, Lambda automatically monitors functions on your behalf, reporting metrics through Amazon CloudWatch.



- Configure **basic function settings** including the description, memory usage, execution timeout, and role that the function will use to execute your code.

Lambda > Functions > test > Edit basic settings

## Edit basic settings

### Basic settings

Description - *optional*

Memory (MB) [Info](#)  
Your function is allocated CPU proportional to the memory configured.

128 MB

Timeout [Info](#)  
0 min 3 sec

Execution role  
Choose a role that defines the permissions of your function. To create a custom role, go to the [IAM console](#).

Use an existing role  
 Create a new role from AWS policy templates

Existing role  
Choose an existing role that you've created to be used with this Lambda function. The role must have permission to upload logs to Amazon CloudWatch Logs.

service-role/test-role-10ewwcvh

[View the test-role-10ewwcvh role on the IAM console.](#)

[Cancel](#) [Save](#)

- Environment variables** are always encrypted at rest, and can be encrypted in transit as well.
- Versions and aliases** are secondary resources that you can create to manage function deployment and invocation.
- A **layer** is a ZIP archive that contains libraries, a custom runtime, or other dependencies. Use layers to manage your function's dependencies independently and keep your deployment package small.

## Invoking Functions



- Lambda supports **synchronous** and **asynchronous invocation** of a Lambda function. You can control the invocation type only when you invoke a Lambda function (referred to as *on-demand invocation*).
- An **event source** is the entity that publishes events, and a Lambda function is the custom code that processes the events.
- *Event source mapping* maps an event source to a Lambda function. It enables automatic invocation of your Lambda function when events occur.

### Configuring a Lambda Function to Access Resources in a VPC

In AWS Lambda, you can set up your function to establish a connection to your virtual private cloud (VPC). With this connection, your function can access the private resources of your VPC during execution like EC2, RDS and many others.

By default, AWS executes your Lambda function code securely within a VPC. Alternatively, you can enable your Lambda function to access resources inside your private VPC by providing additional VPC-specific configuration information such as VPC subnet IDs and security group IDs. It uses this information to set up elastic network interfaces which enable your Lambda function to connect securely to other resources within your VPC.



Custom VPC

VPC  
Choose a VPC for your function to access.

vpc-039af9582d172c330 (10.0.0.0/16) C

Subnets  
Select the VPC subnets for Lambda to use to set up your VPC configuration.

subnet-0f7f5201919833eee (10.0.0.0/24) us-west-2a X  
Name: Public subnet [REDACTED]  
BusinessUnit: Marketplace Env: prod

subnet-0c7e78ae88ca44068 (10.0.1.0/24) us-west-2a X  
Name: DBSubnet BusinessUnit: Marketplace Env: prod [REDACTED]

Security groups  
Choose the VPC security groups for Lambda to use to set up your VPC configuration. The table below shows the inbound and outbound rules for the security groups that you choose.

sg-03ad7f0ed3e0053ee (default) X  
default VPC security group

Inbound rules      Outbound rules

Security group ID	Protocol	Ports	Source
sg-03ad7f0ed3e0053ee	All	All	sg-03ad7f0ed3e0053ee

## Lambda@Edge

- Lets you run Lambda functions to customize content that CloudFront delivers, executing the functions in AWS locations closer to the viewer. The functions run in response to CloudFront events, without provisioning or managing servers.
- You can use Lambda functions to change CloudFront requests and responses at the following points:



- 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)
- You can automate your serverless application's release process using AWS CodePipeline and AWS CodeDeploy.
- Lambda will automatically track the behavior of your Lambda function invocations and provide feedback that you can monitor. In addition, it provides metrics that allows you to analyze the full function invocation spectrum, including event source integration and whether downstream resources perform as expected.

## Pricing

- You are charged based on the total number of requests for your functions and the duration, the time it takes for your code to execute.

## Sources:

<https://docs.aws.amazon.com/lambda/latest/dg>  
<https://aws.amazon.com/lambda/features/>  
<https://aws.amazon.com/lambda/pricing/>  
<https://aws.amazon.com/lambda/faqs/>



## Amazon Elastic Container Service (ECS)

- A container management service to run, stop, and manage Docker containers on a cluster.
- ECS can be used to create a consistent deployment and build experience, manage, and scale batch and **Extract-Transform-Load (ETL)** workloads, and build sophisticated application architectures on a microservices model.
- Amazon ECS is a regional service.

## Features

- You can create ECS clusters within a new or existing VPC.
- After a cluster is up and running, you can define task definitions and services that specify which Docker container images to run across your clusters.
- AWS Compute SLA guarantees a Monthly Uptime Percentage of at least 99.99% for Amazon ECS.

## Components

- Containers and Images
  - Your application components must be architected to run in **containers** — containing everything that your software application needs to run: code, runtime, system tools, system libraries, etc.
  - Containers are created from a read-only template called an **image**.
  - Images are typically built from a **Dockerfile**, a plain text file that specifies all of the components that are included in the container. These images are then stored in a **registry** from which they can be downloaded and run on your cluster.
  - When you launch a container instance, you have the option of passing *user data* to the instance. The data can be used to perform common automated configuration tasks and even run scripts when the instance boots.
  - Docker Volumes can be a local instance store volume, EBS volume or EFS volume. Connect your Docker containers to these volumes using Docker drivers and plugins.
- Task Components
  - **Task definitions** specify various parameters for your application. It is a text file, in JSON format, that describes one or more containers, up to a maximum of ten, that form your application.
  - Task definitions are split into separate parts:
    - Task family - the name of the task, and each family can have multiple revisions.
    - IAM task role - specifies the permissions that containers in the task should have.
    - Network mode - determines how the networking is configured for your containers.
    - Container definitions - specify which image to use, how much CPU and memory the container are allocated, and many more options.
    - Volumes - allow you to share data between containers and even persist the data on the container instance when the containers are no longer running.



- Task placement constraints - lets you customize how your tasks are placed within the infrastructure.
- Launch types - determines which infrastructure your tasks use.
- Tasks and Scheduling
  - A **task** is the instantiation of a task definition within a cluster. After you have created a task definition for your application, you can specify the number of tasks that will run on your cluster.
    - Each task that uses the Fargate launch type has its own isolation boundary and does not share the underlying kernel, CPU resources, memory resources, or elastic network interface with another task.
  - The **task scheduler** is responsible for placing tasks within your cluster. There are several different scheduling options available.
    - REPLICA – places and maintains the desired number of tasks across your cluster. By default, the service scheduler spreads tasks across Availability Zones. You can use task placement strategies and constraints to customize task placement decisions.
    - DAEMON – deploys exactly one task on each active container instance that meets all of the task placement constraints that you specify in your cluster. When using this strategy, there is no need to specify a desired number of tasks, a task placement strategy, or use Service Auto Scaling policies.
  - You can upload a new version of your application task definition, and the ECS scheduler automatically starts new containers using the updated image and stop containers running the previous version.
- Clusters
  - When you run tasks using ECS, you place them in a **cluster**, which is a logical grouping of resources.
  - Clusters are Region-specific.
  - Clusters can contain tasks using both the Fargate and EC2 launch types.
  - When using the Fargate launch type with tasks within your cluster, ECS manages your cluster resources.
  - When using the EC2 launch type, then your clusters are a group of container instances you manage. These clusters can contain multiple different container instance types, but each container instance may only be part of one cluster at a time.
  - Before you can delete a cluster, you must delete the services and deregister the container instances inside that cluster.
- Services
  - ECS allows you to run and maintain a specified number of instances of a task definition simultaneously in a cluster.
  - In addition to maintaining the desired count of tasks in your service, you can optionally run your service behind a load balancer.
  - There are two deployment strategies in ECS:
    - **Rolling Update**



- This involves the service scheduler replacing the current running version of the container with the latest version.
- The number of tasks ECS adds or removes from the service during a rolling update is controlled by the deployment configuration, which consists of the minimum and maximum number of tasks allowed during a service deployment.

### Blue/Green Deployment with AWS CodeDeploy

- This deployment type allows you to verify a new deployment of a service before sending production traffic to it.
- The service must be configured to use either an Application Load Balancer or Network Load Balancer.

- Container Agent
  - The **container agent** runs on each infrastructure resource within an ECS cluster.
  - It sends information about the resource's current running tasks and resource utilization to ECS, and starts and stops tasks whenever it receives a request from ECS.
  - Container agent is only supported on Amazon EC2 instances.
- You can attach multiple target groups to your Amazon ECS services that are running on either Amazon EC2 or AWS Fargate. This allows you to maintain a single ECS service that can serve traffic from both internal and external load balancers and support multiple path based routing rules and applications that need to expose more than one port.

## AWS Fargate

- You can use Fargate with ECS to run containers without having to manage servers or clusters of EC2 instances.
- You no longer have to provision, configure, or scale clusters of virtual machines to run containers.
- Fargate only supports container images hosted on Elastic Container Registry (ECR) or Docker Hub.

### Task Definitions for Fargate Launch Type

- Fargate task definitions require that the network mode is set to awsvpc. The awsvpc network mode provides each task with its own elastic network interface.
- Fargate task definitions require that you specify CPU and memory at the task level.
- Fargate task definitions only support the awslogs log driver for the log configuration. This configures your Fargate tasks to send log information to Amazon CloudWatch Logs.
- Task storage is **ephemeral**. After a Fargate task stops, the storage is deleted.
- Put multiple containers in the same task definition if:
  - Containers share a common lifecycle.
  - Containers are required to be run on the same underlying host.
  - You want your containers to share resources.
  - Your containers share data volumes.



- Otherwise, define your containers in separate tasks definitions so that you can scale, provision, and deprovision them separately.

### Task Definitions for EC2 Launch Type

- Create task definitions that group the containers that are used for a common purpose, and separate the different components into multiple task definitions.
- After you have your task definitions, you can create services from them to maintain the availability of your desired tasks.
- For EC2 tasks, the following are the types of data volumes that can be used:
  - Docker volumes
  - Bind mounts
- Private repositories are only supported by the EC2 Launch Type.

### Monitoring

- You can configure your container instances to send log information to CloudWatch Logs. This enables you to view different logs from your container instances in one convenient location.
- With CloudWatch Alarms, watch a single metric over a time period that you specify, and perform one or more actions based on the value of the metric relative to a given threshold over a number of time periods.
- Share log files between accounts, monitor CloudTrail log files in real time by sending them to CloudWatch Logs.

### Tagging

- ECS resources, including task definitions, clusters, tasks, services, and container instances, are assigned an Amazon Resource Name (ARN) and a unique resource identifier (ID). These resources can be tagged with values that you define, to help you organize and identify them.

### Pricing

- With Fargate, you pay for the amount of vCPU and memory resources that your containerized application requests. vCPU and memory resources are calculated from the time your container images are pulled until the Amazon ECS Task terminates.
- There is no additional charge for EC2 launch type. You pay for AWS resources (e.g. EC2 instances or EBS volumes) you create to store and run your application.

### Sources:

<https://docs.aws.amazon.com/AmazonECS/latest/developerguide>Welcome.html>  
<https://aws.amazon.com/ecs/features/>  
<https://aws.amazon.com/ecs/pricing/>  
<https://aws.amazon.com/ecs/faqs/>



## AWS ParallelCluster

- An AWS-supported open source cluster management tool for deploying and managing High Performance Computing (HPC) clusters on AWS. ParallelCluster uses a simple text file to model and provision all the resources needed for your HPC applications in an automated and secure manner.
- AWS ParallelCluster provisions a master instance for build and control, a cluster of compute instances, a shared filesystem, and a batch scheduler. You can also extend and customize your use cases using custom pre-install and post-install bootstrap actions.
- You have four supported schedulers to use along with ParallelCluster:
  - SGE (Son of Grid Engine)
  - Torque
  - Slurm
  - AWS Batch
- AWS ParallelCluster supports
  - On-Demand,
  - Reserved,
  - and Spot Instances
- Networking
  - AWS ParallelCluster uses Amazon Virtual Private Cloud (VPC) for networking. The VPC must have DNS Resolution = yes, DNS Hostnames = yes and DHCP options with the correct domain-name for the Region.
  - AWS ParallelCluster supports the following high-level configurations:
    - One subnet for both master and compute instances.
    - Two subnets, with the master in one public subnet, and compute instances in a private subnet. The subnets can be new or existing.
  - AWS ParallelCluster can also be deployed to use an HTTP proxy for all AWS requests.
- Storage
  - By default, AWS ParallelCluster automatically configures an external volume of 15 GB of Elastic Block Storage (EBS) attached to the cluster's master node and exported to the cluster's compute nodes via Network File System (NFS).
  - AWS ParallelCluster is also compatible with Amazon Elastic File System (EFS), RAID, and Amazon FSx for Lustre file systems.
  - You can configure AWS ParallelCluster with Amazon S3 object storage as the source of job inputs or as a destination for job output.
- Cluster Configuration
  - By default, AWS ParallelCluster uses the file `~/.parallelcluster/config` for all configuration parameters. A custom configuration file may be specified via the `-c` or `--config` command line option or the `AWS_PCLUSTER_CONFIG_FILE` environment variable.
  - The following sections are required:
    - **[global]** section and **[aws]** section.



- 
- At least one [cluster] section and one [vpc] section.
  - Cluster Processes
    - When a cluster is running, a process called a **jobwatcher** monitors the configured scheduler ( SGE , Slurm , or Torque ) and each minute, it evaluates the queue in order to decide when to scale up.
    - The **sqswatcher** process monitors for Amazon SQS messages that are sent by Auto Scaling, to notify you of state changes within the cluster.
    - The **nodewatcher** process runs on each node in the compute fleet and terminates instances that have been idle for a set amount of time.
  - Pricing
    - AWS ParallelCluster is available at no additional charge. You pay only for the AWS resources needed to run your applications.
  - Limitations
    - AWS ParallelCluster does not support building Windows clusters.
    - AWS ParallelCluster does not currently support mixed instance types for a cluster. However, you can pick one instance type for the master node and another instance type for the compute nodes.

#### Sources:

<https://aws.amazon.com/hpc/parallelcluster/>

<https://docs.aws.amazon.com/parallelcluster/latest/ug/what-is-aws-parallelcluster.html>

<https://aws.amazon.com/hpc/faqs/>

<https://aws.amazon.com/blogsopensource/aws-parallelcluster/>



## Elastic Fabric Adapter (EFA)

- An **Elastic Fabric Adapter (EFA)** is a **network device** that you can attach to your Amazon **EC2 instance** to accelerate High Performance Computing (HPC) and machine learning applications.
- An EFA is an Elastic Network Adapter (ENA) with an additional OS-bypass functionality.
- EFA integrates with
  - Libfabric 1.9.0 and it supports Open MPI 4.0.2 and Intel MPI 2019 Update 6 for HPC applications, and
  - Nvidia Collective Communications Library (NCCL) for machine learning applications
- With an EFA, HPC applications use Intel Message Passing Interface (MPI) or Nvidia Collective Communications Library (NCCL) to interface with the **Libfabric API**. The Libfabric API bypasses the operating system kernel and communicates directly with the EFA device to place packets on the network.
- Supported AMIs
  - Amazon Linux
  - Amazon Linux 2
  - RHEL 7.6 and RHEL 7.7
  - CentOS 7
  - Ubuntu 16.04 and Ubuntu 18.04
- Examples of HPC Applications
  - computational fluid dynamics (CFD)
  - crash simulations
  - weather simulations
- Limitations
  - You can attach only **one** EFA per instance.
  - EFA OS-bypass traffic is **limited to a single subnet**. EFA traffic cannot be sent from one subnet to another. Normal IP traffic from the EFA can be sent from one subnet to another.
  - EFA OS-bypass traffic is **not routable**. Normal IP traffic from the EFA remains routable.
  - The EFA must be a member of a security group that allows all inbound and outbound traffic to and from the security group itself.
- Pricing
  - EFA is available as an optional EC2 networking feature that you can enable on any supported EC2 instance at no additional cost.

### Sources:

<https://aws.amazon.com/hpc/efa/>

<https://aws.amazon.com/hpc/faqs/>

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/efa.html>



## STORAGE

### Amazon S3

- S3 stores data as objects within **buckets**.
- An **object** consists of a file and optionally any metadata that describes that file.
- A **key** is the unique identifier for an object within a bucket.
- Storage capacity is virtually unlimited.

### Buckets

- For each bucket, you can:
  - Control access to it (create, delete, and list objects in the bucket)
  - View access logs for it and its objects
  - Choose the geographical region where to store the bucket and its contents.
- **Bucket name** must be a unique DNS-compliant name.
  - The name must be unique across all existing bucket names in Amazon S3.
  - After you create the bucket you cannot change the name.
  - The bucket name is visible in the URL that points to the objects that you're going to put in your bucket.
- By default, you can create up to 100 buckets in each of your AWS accounts.
- You can't change its Region after creation.
- You can host static websites by configuring your bucket for website hosting.

### Data Consistency Model

- read-after-write consistency for PUTS of new objects in your S3 bucket in all regions
- eventual consistency for read-after-write HEAD or GET requests
- eventual consistency for overwrite PUTS and Deletes in all regions

### Storage Classes

- Storage Classes for Frequently Accessed Objects
  - S3 **STANDARD** for **general-purpose** storage of frequently accessed data.
  - S3 **REDUCED\_REDUNDANCY (RRS)** for noncritical, reproducible data that can be stored with less redundancy than the STANDARD storage class. Not recommended because STANDARD storage class is more cost effective.
- Storage Classes for Infrequently Accessed Objects
  - S3 **STANDARD\_IA** for long-lived, but **less frequently accessed** data. It stores the object data redundantly across multiple geographically separated AZs.



- S3 **ONEZONE\_IA** stores the object data in only one AZ. Less expensive than STANDARD\_IA, but data is not resilient to the physical loss of the AZ.
- These two storage classes are suitable for objects larger than 128 KB that you plan to store for **at least 30 days**. If an object is less than 128 KB, Amazon S3 charges you for 128 KB. If you delete an object before the 30-day minimum, you are charged for 30 days.
- Amazon S3 Intelligent Tiering
  - S3 Intelligent-Tiering is a storage class designed for customers who want to optimize storage costs automatically when data access patterns change, without performance impact or operational overhead.
  - S3 Intelligent-Tiering is the first cloud object storage class that delivers automatic cost savings by moving data between two access tiers – frequent access and infrequent access – when access patterns change, and is ideal for data with unknown or changing access patterns.
  - S3 Intelligent-Tiering monitors access patterns and moves objects that have not been accessed for 30 consecutive days to the infrequent access tier. If an object in the infrequent access tier is accessed later, it is automatically moved back to the frequent access tier.
  - There are no retrieval fees in S3 Intelligent-Tiering.
- GLACIER
  - For long-term **archive**
  - Archived objects are not available for real-time access. You must first restore the objects before you can access them.
  - You cannot specify GLACIER as the storage class at the time that you create an object.
  - Glacier objects are visible through S3 only.
  - Retrieval Options
    - **Expedited** - allows you to quickly access your data when occasional urgent requests for a subset of archives are required. For all but the largest archived objects, data accessed are typically made available within 1–5 minutes. There are two types of Expedited retrievals: On-Demand requests are similar to EC2 On-Demand instances and are available most of the time. Provisioned requests are guaranteed to be available when you need them.
    - **Standard** - allows you to access any of your archived objects within several hours. Standard retrievals typically complete within 3–5 hours. This is the default option for retrieval requests that do not specify the retrieval option.
    - **Bulk** - Glacier's lowest-cost retrieval option, enabling you to retrieve large amounts, even petabytes, of data inexpensively in a day. Bulk retrievals typically complete within 5–12 hours.
  - For S3 Standard, S3 Standard-IA, and Glacier storage classes, your objects are automatically stored across multiple devices spanning a **minimum of three** Availability Zones.



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

\* S3 Intelligent-Tiering charges a small tiering fee and has a minimum eligible object size of 128KB for auto-tiering. Smaller objects may be stored but will always be charged at the Frequent Access tier rates.

\*\* 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.

Tutorials Dojo

- Amazon S3 Glacier Deep Archive

- A new Amazon S3 storage class providing secure and durable object storage for long-term retention of data that is accessed rarely in a year.
- S3 Glacier Deep Archive offers the lowest cost storage in the cloud, at prices lower than storing and maintaining data in on-premises magnetic tape libraries or archiving data offsite.
- All objects stored in the S3 Glacier Deep Archive storage class are replicated and stored across at least three geographically-dispersed Availability Zones, protected by 99.999999999% durability, and can be restored within 12 hours or less.
- S3 Glacier Deep Archive also offers a bulk retrieval option, where you can retrieve petabytes of data within 48 hours.

## S3 API

- REST - use standard HTTP requests to create, fetch, and delete buckets and objects. You can use S3 virtual hosting to address a bucket in a REST API call by using the HTTP Host header.
- SOAP

## Bucket Configurations



Subresource	Description
<i>location</i>	Specify the AWS Region where you want S3 to create the bucket.
<i>policy and ACL (access control list)</i>	All your resources are private by default. Use bucket policy and ACL options to grant and manage bucket-level permissions.
<i>cors (cross-origin resource sharing)</i>	You can configure your bucket to allow cross-origin requests. CORS defines a way for client web applications that are loaded in one domain to interact with resources in a different domain.
<i>website</i>	You can configure your bucket for static website hosting.
<i>logging</i>	Logging enables you to track requests for access to your bucket. 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.
<i>event notification</i>	You can enable your bucket to send you notifications of specified bucket events.
<i>versioning</i>	AWS recommends VERSIONING AS A BEST PRACTICE to recover objects from being deleted or overwritten by mistake.
<i>lifecycle</i>	You can define lifecycle rules for objects in your bucket that have a well-defined lifecycle.
<i>cross-region replication</i>	Cross-region replication is the automatic, asynchronous copying of objects across buckets in different AWS Regions.
<i>tagging</i>	S3 provides the <i>tagging</i> subresource to store and manage tags on a bucket. AWS generates a cost allocation report with usage and costs aggregated by your tags.
<i>requestPayment</i>	By default, the AWS account that creates the bucket (the bucket owner) pays for downloads from the bucket. The bucket owner can specify that the person requesting the download will be charged for the download.



<i>transfer acceleration</i>	Transfer Acceleration enables fast, easy, and secure transfers of files over long distances between your client and an S3 bucket. It takes advantage of Amazon CloudFront's globally distributed <b>edge locations</b> .
------------------------------	--

## Objects

- Are private by default. Grant permissions to other users.
- Each S3 object has **data**, a **key**, and **metadata**.
- You cannot modify object metadata after object is uploaded.
- Two kinds of metadata
  - System metadata

Name	Description	Can User Modify the Value?
Date	Current date and time.	No
Content-Length	Object size in bytes.	No
Last-Modified	Object creation date or the last modified date, whichever is the latest.	No
Content-MD5	The base64-encoded 128-bit MD5 digest of the object.	No
x-amz-server-side-encryption	Indicates whether server-side encryption is enabled for the object, and whether that encryption is from the AWS Key Management Service (SSE-KMS) or from AWS managed encryption (SSE-S3).	Yes
x-amz-version-id	Object version. When you enable versioning on a bucket, Amazon S3 assigns a version number to objects added to the bucket.	No
x-amz-delete-marker	In a bucket that has versioning enabled, this Boolean marker indicates whether the object is a delete marker.	No
x-amz-storage-class	Storage class used for storing the object.	Yes



x-amz-website-redirect-location	Redirects requests for the associated object to another object in the same bucket or an external URL.	Yes
x-amz-server-side-encryption-aws-kms-key-id	If x-amz-server-side-encryption is present and has the value of aws:kms, this indicates the ID of the AWS Key Management Service (AWS KMS) master encryption key that was used for the object.	Yes
x-amz-server-side-encryption-customer-algorithm	Indicates whether server-side encryption with customer-provided encryption keys (SSE-C) is enabled.	Yes

- User-defined metadata - key-value pair that you provide.
- You can upload and copy objects of up to **5 GB** in size in a single operation. For objects greater than 5 GB up to **5 TB**, you must use the **multipart upload API**.
- Tagging
  - You can associate up to 10 tags with an object. Tags associated with an object must have unique tag keys.
  - A tag key can be up to 128 Unicode characters in length and tag values can be up to 256 Unicode characters in length.
  - Key and values are case sensitive.
- Object Delete
  - Deleting Objects from a Version-Enabled Bucket
    - Specify a non-versioned delete request - specify only the object's key, and not the version ID.
    - Specify a versioned delete request - specify both the key and also a version ID.
  - Deleting Objects from an MFA-Enabled Bucket
    - If you provide an invalid MFA token, the request always fails.
    - If you are not deleting a versioned object, and you don't provide an MFA token, the delete succeeds.
- S3 Select
  - S3 Select is an Amazon S3 capability designed to pull out only the data you need from an object, which can dramatically improve the performance and reduce the cost of applications that need to access data in S3.
  - Amazon S3 Select works on objects stored in CSV and JSON format, Apache Parquet format, JSON Arrays, and BZIP2 compression for CSV and JSON objects.
  - CloudWatch Metrics for S3 Select lets you monitor S3 Select usage for your applications. These metrics are available at 1-minute intervals and lets you quickly identify and act on operational issues.



- Lifecycle Management
  - A *lifecycle configuration* is a set of rules that define actions that are applied to a group of objects.

- Transition actions—Define when objects transition to another storage class. For S3-IA and S3-One-Zone, the objects must be stored at least 30 days in the current storage class before you can transition them to another class.

Lifecycle rule X

(1) Name and scope      (2) **Transitions**      (3) Expiration      (4) Review

### Storage class transition

There are per-request fees when using lifecycle to transition data to any S3 or S3 Glacier storage class. [Learn more](#) or see [Amazon S3 pricing](#)

Current version     Previous versions

For current versions of objects [+ Add transition](#)

**!** A minimum of 30 days is required before transitioning to the Standard-IA storage class

Enter an integer value greater than or equal to 30.

Object creation	Days after creation
Transition to Standard-IA after	<input type="text" value="0"/> X

For previous versions of objects [+ Add transition](#)

Object becomes a previous version	Days after objects become
-----------------------------------	---------------------------

[Previous](#) [Next](#)



## Lifecycle rule

① Name and scope    ② Transitions    ③ Expiration    ④ Review

Current version     Previous versions

For current versions of objects [+ Add transition](#)

Object creation	Days after creation
Transition to Glacier after	<input type="text" value="0"/> X

**A** Transitioning small objects to Glacier or Glacier Deep Archive will increase costs.

Lifecycle transitions include a per-object transition cost. For example, if you were to transition all objects currently in this bucket to Glacier or Glacier Deep Archive, the transition cost would be approximately \$0.01 (USD). You can reduce this cost by limiting the number of objects to transition (by prefix, tag, or version) or by aggregating objects before transitioning them. [Learn more](#) or see [Amazon S3 pricing](#)

I acknowledge that this lifecycle rule will increase the one-time lifecycle request cost if it transitions small objects.

For previous versions of objects [+ Add transition](#)

[Previous](#) [Next](#)

- Expiration actions—Define when objects expire. S3 deletes expired objects on your behalf.



Lifecycle rule X

1 Name and scope    2 Transitions    3 Expiration    4 Review

### Configure expiration

Current version     Previous versions

Expire current version of object ?  
After  days from object creation

Permanently delete previous versions ?  
After  days from becoming a previous version

Clean up expired object delete markers and incomplete multipart uploads

Clean up expired object delete markers ?

You cannot enable clean up expired object delete markers if you enable Expiration.

Clean up incomplete multipart uploads ?

[Previous](#) [Next](#)

## Pricing

- S3 charges you only for what you actually use, with no hidden fees and no overage charges
- No charge for creating a bucket, but only for storing objects in the bucket and for transferring objects in and out of the bucket.

Charge	Comments



Storage	You pay for storing objects in your S3 buckets. The rate you're charged depends on your objects' size, how long you stored the objects during the month, and the storage class.
Requests	You pay for requests, for example, GET requests, made against your S3 buckets and objects. This includes lifecycle requests. The rates for requests depend on what kind of request you're making.
Retrievals	You pay for retrieving objects that are stored in STANDARD_IA, ONEZONE_IA, and GLACIER storage.
Early Deletes	If you delete an object stored in STANDARD_IA, ONEZONE_IA, or GLACIER storage before the minimum storage commitment has passed, you pay an early deletion fee for that object.
Storage Management	You pay for the storage management features that are enabled on your account's buckets.
Bandwidth	<p>You pay for all bandwidth into and out of S3, except for the following:</p> <ul style="list-style-type: none"><li>● Data transferred in from the internet</li><li>● Data transferred out to an Amazon EC2 instance, when the instance is in the same AWS Region as the S3 bucket</li><li>● Data transferred out to Amazon CloudFront</li></ul> <p>You also pay a fee for any data transferred using Amazon S3 Transfer Acceleration.</p>

## Networking

- Hosted-style access
  - Amazon S3 routes any virtual hosted-style requests to the US East (N. Virginia) region by default if you use the endpoint s3.amazonaws.com, instead of the region-specific endpoint.
  - Format:
    - <http://bucket.s3.amazonaws.com>
    - <http://bucket.s3-aws-region.amazonaws.com>
- Path-style access
  - In a path-style URL, the endpoint you use must match the Region in which the bucket resides.
  - Format:
    - US East (N. Virginia) Region endpoint, <http://s3.amazonaws.com/bucket>
    - Region-specific endpoint, <http://s3-aws-region.amazonaws.com/bucket>
- Customize S3 URLs with CNAMEs - the bucket name must be the same as the CNAME.



- **Amazon S3 Transfer Acceleration** enables fast, easy, and secure transfers of files over long distances between your client and an S3 bucket. It takes advantage of Amazon CloudFront's globally distributed **edge locations**.
- Transfer Acceleration cannot be disabled, and can only be suspended.
- Transfer Acceleration URL is: *bucket.s3-accelerate.amazonaws.com*

## Security

- Policies contain the following:
  - **Resources** – buckets and objects
  - **Actions** – set of operations
  - **Effect** – can be either allow or deny. Need to explicitly grant allow to a resource.
  - **Principal** – the account, service or user who is allowed access to the actions and resources in the statement.
- Resource Based Policies
  - Bucket Policies
    - Provides **centralized access control** to buckets and objects based on a variety of conditions, including S3 operations, requesters, resources, and aspects of the request (e.g., IP address).
    - Can either **add or deny permissions** across all (or a subset) of objects within a bucket.
    - IAM users need additional permissions from root account to perform bucket operations.
    - Bucket policies are limited to 20 KB in size.
  - Access Control Lists
    - A list of grants identifying grantee and permission granted.
    - ACLs use an S3-specific XML schema.
    - You can grant permissions only to other AWS accounts, not to users in your account.
    - You cannot grant conditional permissions, nor explicitly deny permissions.
    - Object ACLs are limited to 100 granted permissions per ACL.
    - The only recommended use case for the bucket ACL is to grant **write** permissions to the **S3 Log Delivery group**.
- User Policies
  - AWS IAM (see AWS Security and Identity Services)
    - IAM User Access Keys
    - Temporary Security Credentials
- Versioning
  - Use versioning to keep multiple versions of an object in one bucket.
  - Versioning protects you from the consequences of unintended overwrites and deletions.
  - You can also use versioning to archive objects so you have access to previous versions.
  - Since versioning is disabled by default, need to EXPLICITLY enable.
  - When you PUT an object in a versioning-enabled bucket, the non-current version is not overwritten.



- When you DELETE an object, all versions remain in the bucket and Amazon S3 inserts a delete marker.
- Performing a simple GET Object request when the current version is a delete marker returns a 404 Not Found error. You can, however, GET a non-current version of an object by specifying its version ID.
- You can permanently delete an object by specifying the version you want to delete. Only the owner of an Amazon S3 bucket can permanently delete a version.
- Encryption
  - Server-side Encryption using
    - **Amazon S3-Managed Keys (SSE-S3)**
    - **AWS KMS-Managed Keys (SSE-KMS)**
    - **Customer-Provided Keys (SSE-C)**
  - Client-side Encryption using
    - AWS KMS-managed customer master key
    - client-side master key
- MFA Delete
  - MFA delete grants additional authentication for either of the following operations:
    - Change the versioning state of your bucket
    - Permanently delete an object version
  - MFA Delete requires two forms of authentication together:
    - Your security credentials
    - The concatenation of a valid serial number, a space, and the six-digit code displayed on an approved authentication device
- Cross-Account Access
  - You can provide another AWS account access to an object that is stored in an Amazon Simple Storage Service (Amazon S3) bucket. These are the methods on how to grant cross-account access to objects that are stored in your own Amazon S3 bucket:
    - Resource-based policies and AWS Identity and Access Management (IAM) policies for programmatic-only access to S3 bucket objects
    - Resource-based Access Control List (ACL) and IAM policies for programmatic-only access to S3 bucket objects
    - Cross-account IAM roles for programmatic and console access to S3 bucket objects
- Requester Pays Buckets
  - Bucket owners pay for all of the Amazon S3 storage and data transfer costs associated with their bucket. To save on costs, you can enable the Requester Pays feature so the requester will pay the cost of the request and the data download from the bucket instead of the bucket owner. Take note that the bucket owner always pays the cost of storing data.
- Monitoring
  - Automated monitoring tools to watch S3:



- Amazon CloudWatch Alarms – Watch a single metric over a time period that you specify, and perform one or more actions based on the value of the metric relative to a given threshold over a number of time periods.
- AWS CloudTrail Log Monitoring – Share log files between accounts, monitor CloudTrail log files in real time by sending them to CloudWatch Logs, write log processing applications in Java, and validate that your log files have not changed after delivery by CloudTrail.
- Monitoring with CloudWatch
  - Daily Storage Metrics for Buckets - You can monitor bucket storage using CloudWatch, which collects and processes storage data from S3 into readable, daily metrics.
  - Request metrics - You can choose to monitor S3 requests to quickly identify and act on operational issues. The metrics are available at 1 minute intervals after some latency to process.
- You can have a maximum of 1000 metrics configurations per bucket.
- Supported event activities that occur in S3 are recorded in a CloudTrail event along with other AWS service events in **Event history**.
- Website Hosting
  - Enable website hosting in your bucket **Properties**.
  - Your static website is available via the region-specific website endpoint.
  - You must make the objects that you want to serve publicly readable by writing a bucket policy that grants everyone s3:GetObject permission.

Key Difference	REST API Endpoint	Website Endpoint
Access control	Supports both public and private content.	Supports only publicly readable content.
Error message handling	Returns an XML-formatted error response.	Returns an HTML document.
Redirection support	Not applicable	Supports both object-level and bucket-level redirects.
Requests supported	Supports all bucket and object operations	Supports only GET and HEAD requests on objects.
Responses to GET and HEAD requests at the root of a bucket	Returns a list of the object keys in the bucket.	Returns the index document that is specified in the website configuration.
Secure Sockets Layer (SSL) support	Supports SSL connections.	Does not support SSL connections.



- S3 Events Notification
  - To enable notifications, add a *notification configuration* identifying the events to be published, and the destinations where to send the event notifications.
  - Can publish following events:
    - A new object created event
    - An object removal event
    - A Reduced Redundancy Storage (RRS) object lost event
  - Supports the following destinations for your events:
    - Amazon Simple Notification Service (Amazon SNS) topic
    - Amazon Simple Queue Service (Amazon SQS) queue
    - AWS Lambda
- Cross Region Replication
  - Enables automatic, asynchronous copying of objects across buckets in different AWS Regions.
  - When to use:
    - Comply with compliance requirements
    - Minimize latency
    - Increase operational efficiency
    - Maintain object copies under different ownership
  - Requirements of CRR:
    - Both source and destination buckets must have versioning enabled.
    - The source and destination buckets must be in different AWS Regions.
    - S3 must have permissions to replicate objects from the source bucket to the destination bucket on your behalf.
    - If the owner of the source bucket doesn't own the object in the bucket, the object owner must grant the bucket owner READ and READ\_ACP permissions with the object ACL.
  - Only the following are replicated:
    - Objects created after you add a replication configuration.
    - Both unencrypted objects and objects encrypted using Amazon S3 managed keys (SSE-S3) or AWS KMS managed keys (SSE-KMS), although you must explicitly enable the option to replicate objects encrypted using KMS keys. The replicated copy of the object is encrypted using the same type of server-side encryption that was used for the source object.
    - Object metadata.
    - Only objects in the source bucket for which the bucket owner has permissions to read objects and access control lists.
    - Object ACL updates, unless you direct S3 to change the replica ownership when source and destination buckets aren't owned by the same accounts.
    - Object tags.
  - What isn't replicated



- Objects that existed before you added the replication configuration to the bucket.
- Objects created with server-side encryption using customer-provided (SSE-C) encryption keys.
- Objects created with server-side encryption using AWS KMS-managed encryption (SSE-KMS) keys.
- Objects in the source bucket that the bucket owner doesn't have permissions for.
- Updates to bucket-level subresources.
- Actions performed by lifecycle configuration.
- Objects in the source bucket that are replicas created by another cross-region replication. You can replicate objects from a source bucket to **only one destination bucket**.
- CRR delete operations
  - If you make a DELETE request without specifying an object version ID, S3 adds a delete marker.
  - If you specify an object version ID to delete in a DELETE request, S3 deletes that object version in the source bucket, but it doesn't replicate the deletion in the destination bucket. This protects data from malicious deletions.

S3 Batch Operations is a new feature that makes it simple to manage billions of objects stored in Amazon S3. Customers can make changes to object properties and metadata, and perform other storage management tasks – such as copying objects between buckets, replacing tag sets, modifying access controls, and restoring archived objects from Amazon S3 Glacier – for any number of S3 objects in minutes.

**Sources:**

<https://docs.aws.amazon.com/AmazonS3/latest/dev/Welcome.html>

<https://aws.amazon.com/s3/faqs/>



## Amazon EBS

- **Block level storage** volumes for use with EC2 instances.
- Well-suited for use as the primary storage for file systems, databases, or for any applications that require fine granular updates and access to raw, unformatted, block-level storage.
- Well-suited to both database-style applications (random reads and writes), and to throughput-intensive applications (long, continuous reads and writes).
- New EBS volumes receive their maximum performance the moment that they are available and do not require initialization (formerly known as pre-warming). However, storage blocks on volumes that were restored from snapshots must be initialized (pulled down from Amazon S3 and written to the volume) before you can access the block.
- Termination protection is turned off by default and must be manually enabled (keeps the volume/data when the instance is terminated)
- You can have up to 5,000 EBS volumes by default
- You can have up to 10,000 snapshots by default

## Features

- Different types of storage options: **General Purpose SSD (gp2)**, **Provisioned IOPS SSD (io1)**, **Throughput Optimized HDD (st1)**, and **Cold HDD (sc1)** volumes up to **16 TiB** in size.
- You can mount multiple volumes on the same instance, and you can mount a volume to multiple instances at a time using Amazon EBS Multi-Attach.
- You can create a file system on top of these volumes, or use them in any other way you would use a block device (like a hard drive).
- You can use encrypted EBS volumes to meet data-at-rest encryption requirements for regulated/audited data and applications.
- You can create point-in-time **snapshots** of EBS volumes, which are persisted to Amazon S3. Similar to AMIs. Snapshots can be copied across AWS regions.
- Volumes are created in a specific AZ, and can then be attached to any instances in that same AZ. To make a volume available outside of the AZ, you can create a snapshot and restore that snapshot to a new volume anywhere in that region.
- You can copy snapshots to other regions and then restore them to new volumes there, making it easier to leverage multiple AWS regions for geographical expansion, data center migration, and disaster recovery.
- Performance metrics, such as bandwidth, throughput, latency, and average queue length, provided by Amazon CloudWatch, allow you to monitor the performance of your volumes to make sure that you are providing enough performance for your applications without paying for resources you don't need.
- You can detach an EBS volume from an instance explicitly or by terminating the instance. However, if the instance is running, you must first unmount the volume from the instance.
- If an EBS volume is the root device of an instance, you must stop the instance before you can detach the volume.



- You can use AWS Backup, an automated and centralized backup service, to protect EBS volumes and your other AWS resources. AWS Backup is integrated with Amazon DynamoDB, Amazon EBS, Amazon RDS, Amazon EFS, and AWS Storage Gateway to give you a fully managed AWS backup solution.
- With AWS Backup, you can configure backups for EBS volumes, automate backup scheduling, set retention policies, and monitor backup and restore activity.

## Types of EBS Volumes

- General Purpose SSD (gp2)
  - Base performance of 3 IOPS/GiB, with the ability to burst to 3,000 IOPS for extended periods of time.
  - Support up to 16,000 IOPS and 250 MB/s of throughput.
  - The **burst duration** of a volume is dependent on the size of the volume, the burst IOPS required, and the credit balance when the burst begins. Burst IO duration is computed using the following formula:

$$\text{Burst duration} = (\text{Credit balance}) / [(\text{Burst IOPS}) - 3 \times (\text{Volume size in GiB})]$$

- If your gp2 volume uses all of its I/O credit balance, the maximum IOPS performance of the volume remains at the baseline IOPS performance level and the volume's maximum throughput is reduced to the baseline IOPS multiplied by the maximum I/O size.
- Throughput for a gp2 volume can be calculated using the following formula, up to the throughput limit of 160 MiB/s:

$$\text{Throughput in MiB/s} = (\text{Volume size in GiB}) \times (\text{IOPS per GiB}) \times (\text{I/O size in KiB})$$

- Provisioned IOPS SSD (io1)
  - Designed for I/O-intensive workloads, particularly database workloads, which are sensitive to storage performance and consistency.
  - Allows you to specify a consistent IOPS rate when you create the volume
- Throughput Optimized HDD (st1)
  - Low-cost magnetic storage that focuses on throughput rather than IOPS.
  - Throughput of up to 500 MiB/s.
  - Subject to throughput and throughput-credit caps, the available throughput of an st1 volume is expressed by the following formula:

$$(\text{Volume size})(\text{Credit accumulation rate per TiB}) = \text{Throughput}$$

- Cold HDD (sc1)
  - Low-cost magnetic storage that focuses on throughput rather than IOPS.
  - Throughput of up to 250 MiB/s.



	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)
Use Cases	<ul style="list-style-type: none"><li>-Recommended for most workloads</li><li>-System boot volumes</li><li>-Virtual desktops</li><li>-Low-latency interactive apps</li><li>-Development and test environments</li></ul>	<ul style="list-style-type: none"><li>-Critical business applications that require sustained IOPS performance, or more than 16,000 IOPS or 250 MiB/s of throughput per volume. When attached to Nitro system EC2 instances, peak performance can go up to 64,000 IOPS and 1,000 MB/s of throughput per volume.</li><li>-Large database workloads.</li></ul>	<ul style="list-style-type: none"><li>-Streaming workloads requiring consistent, fast throughput at a low price</li><li>-Big data</li><li>-Data warehouse</li><li>-Log processing</li><li><b>Cannot be a boot volume</b></li></ul>	<ul style="list-style-type: none"><li>-Throughput-oriented storage for large volumes of data that is infrequently accessed</li><li>-Scenarios where the lowest storage cost is important</li><li><b>Cannot be a boot volume</b></li></ul>
Dominant Performance Attribute	IOPS	IOPS	MiB/s	MiB/s



FEATURES	SSD Solid State Drive	HDD Hard Disk Drive
<b>Best for workloads with:</b>	<b><i>small, random</i></b> I/O operations	<b><i>large, sequential</i></b> I/O operations
<b>Can be used as a bootable volume?</b>	Yes	No
<b>Suitable Use Cases</b>	<ul style="list-style-type: none"><li>- Best for <b>transactional workloads</b></li><li>- Critical business applications that require sustained IOPS performance</li><li>- Large database workloads such as MongoDB, Oracle, Microsoft SQL Server and many others...</li></ul>	<ul style="list-style-type: none"><li>- Best for <b><i>large streaming workloads</i></b> requiring consistent, fast throughput at a low price</li><li>- Big data, Data warehouses, Log processing</li><li>- Throughput-oriented storage for large volumes of data that is <b><i>infrequently</i></b> accessed</li></ul>
<b>Cost</b>	moderate / high 	low 
<b>Dominant Performance Attribute</b>	IOPS	Throughput (MiB/s)



## Encryption

- Data stored at rest on an encrypted volume, disk I/O, and snapshots created from it are all encrypted.
- Also provides encryption for data in-transit from EC2 to EBS since encryption occurs on the servers that hosts EC2 instances.
- The following types of data are encrypted:
  - Data at rest inside the volume
  - All data moving between the volume and the instance
  - All snapshots created from the volume
  - All volumes created from those snapshots



- Uses AWS Key Management Service (AWS KMS) master keys when creating encrypted volumes and any snapshots created from your encrypted volumes.
- Volumes restored from encrypted snapshots are automatically encrypted.
- EBS encryption is only available on certain instance types.
- There is no direct way to encrypt an existing unencrypted volume, or to remove encryption from an encrypted volume. However, you can migrate data between encrypted and unencrypted volumes.
- You can now enable Amazon Elastic Block Store (EBS) Encryption by Default, ensuring that all new EBS volumes created in your account are encrypted.

## Monitoring

- Cloudwatch Monitoring two types: Basic and Detailed monitoring
- Volume status checks provide you the information that you need to determine whether your EBS volumes are impaired, and help you control how a potentially inconsistent volume is handled. List of statuses include:
  - Ok - normal volume
  - Warning - degraded volume
  - Impaired - stalled volume
  - Insufficient-data - insufficient data
- Volume events include a start time that indicates the time at which an event occurred, and a duration that indicates how long I/O for the volume was disabled. The end time is added to the event when I/O for the volume is enabled.
- Volume events are:
  - Awaiting Action: Enable IO
  - IO Enabled
  - IO Auto-Enabled
  - Normal
  - Degraded
  - Severely Degraded
  - Stalled

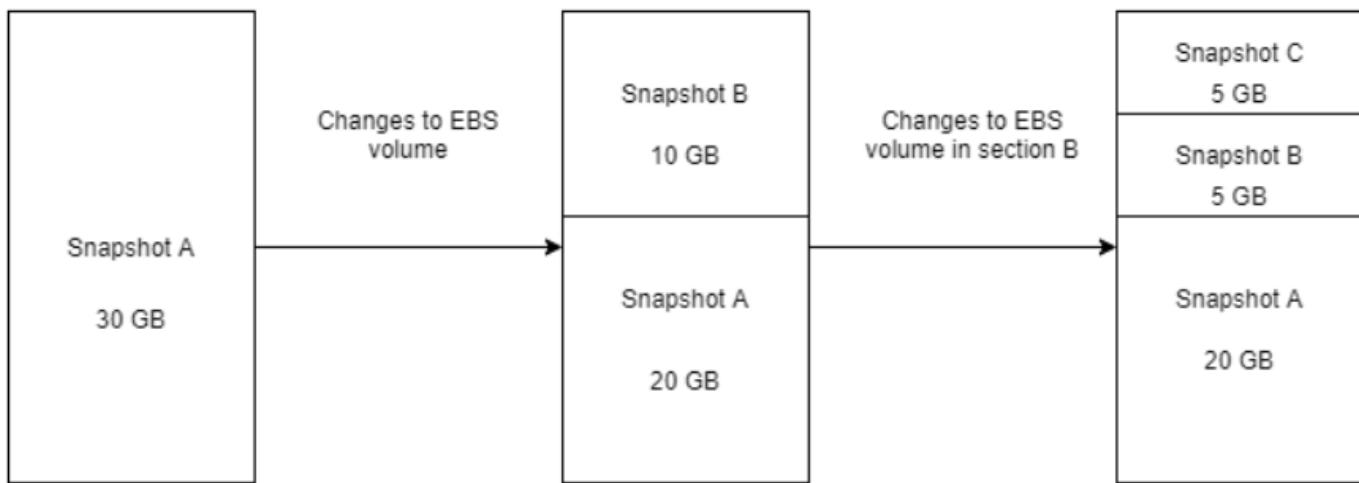
## Modifying the Size, IOPS, or Type of an EBS Volume on Linux

- If your current-generation EBS volume is attached to a current-generation EC2 instance type, you can increase its size, change its volume type, or (for an io1 volume) adjust its IOPS performance, all without detaching it.
- EBS currently supports a maximum volume size of 16 TiB.
- Two partitioning schemes commonly used on Linux and Windows systems: master boot record (MBR) and GUID partition table (GPT).
- An EBS volume being modified goes through a sequence of states. The volume enters first the **Modifying** state, then the **Optimizing** state, and finally the **Complete** state.

- You can expand a partition to a new size. Expand by using `parted` or `gdisk`.
- Use a file system-specific command to resize the file system to the larger size of the new volume. These commands work even if the volume to extend is the root volume. For ext2, ext3, and ext4 file systems, this command is `resize2fs`. For XFS file systems, this command is `xfs_growfs`.
- Decreasing the size of an EBS volume is not supported.

## EBS Snapshots

- Back up the data on your EBS volumes to S3 by taking point-in-time snapshots.
- Snapshots are **incremental** backups, which means that only the blocks on the device that have changed after your most recent snapshot are saved. This minimizes the time required to create the snapshot and saves on storage costs by not duplicating data.



- When you delete a snapshot, only the data unique to that snapshot is removed.
- You can share a snapshot across AWS accounts by modifying its access permissions.
- You can make copies of your own snapshots as well as snapshots that have been shared with you.
- A snapshot is constrained to the Region where it was created.
- EBS snapshots broadly support EBS encryption.
- You can't delete a snapshot of the root device of an EBS volume used by a registered AMI. You must first deregister the AMI before you can delete the snapshot.
- Each account can have up to **5 concurrent snapshot copy requests** to a single destination Region.
- User-defined tags are not copied from the source snapshot to the new snapshot.
- Snapshots are constrained to the Region in which they were created. To share a snapshot with another Region, copy the snapshot to that Region.
- Snapshots that you intend to share must instead be encrypted with a custom CMK.



## Amazon EBS-Optimized Instances

- Provides the best performance for your EBS volumes by minimizing contention between EBS I/O and other traffic from your instance.
- EBS-optimized instances deliver dedicated bandwidth between 425 Mbps and 14,000 Mbps to EBS.
- For instance types that are EBS-optimized by default, there is no need to enable EBS optimization and no effect if you disable EBS optimization.

## Pricing

- You are charged by the amount you provision in GB per month until you release the storage.
- Provisioned storage for *gp2* volumes, provisioned storage and provisioned IOPS for *io1* volumes, provisioned storage for *st1* and *sc1* volumes will be billed in per-second increments, with a 60 second minimum.
- With Provisioned IOPS SSD (*io1*) volumes, you are also charged by the amount you provision in IOPS per month.
- After you detach a volume, you are still charged for volume storage as long as the storage amount exceeds the limit of the AWS Free Tier. You must delete a volume to avoid incurring further charges.
- Snapshot storage is based on the amount of space your data consumes in Amazon S3.
- Copying a snapshot to a new Region does incur new storage costs.
- When you enable EBS optimization for an instance that is not EBS-optimized by default, you pay an additional low, hourly fee for the dedicated capacity.

## Improving Performance

- Use EBS-Optimized Instances
- Understand How Performance is Calculated
- Understand Your Workload
- Be Aware of the Performance Penalty When Initializing Volumes from Snapshots
- Factors That Can Degrade HDD Performance
- Increase Read-Ahead for High-Throughput, Read-Heavy Workloads on *st1* and *sc1*
- Use a Modern Linux Kernel
- Use RAID 0 (Redundant Array of Independent Disks) to Maximize Utilization of Instance Resources
- Track Performance Using Amazon CloudWatch

## Sources:

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/AmazonEBS.html>

<https://aws.amazon.com/ebs/faqs/>



## Amazon EFS

A fully-managed **file storage service** that makes it easy to set up and scale file storage in the Amazon Cloud.

### Features

- The service manages all the file storage infrastructure for you, avoiding the complexity of deploying, patching, and maintaining complex file system configurations.
- EFS supports the Network File System version 4 protocol.
- Multiple Amazon EC2 instances can access an EFS file system at the same time, providing a common data source for workloads and applications running on more than one instance or server.
- EFS file systems store data and metadata across multiple Availability Zones in an AWS Region.
- EFS file systems can grow to petabyte scale, drive high levels of throughput, and allow massively parallel access from EC2 instances to your data.
- EFS provides file system access semantics, such as strong data consistency and file locking.
- EFS enables you to control access to your file systems through Portable Operating System Interface (POSIX) permissions.
- Moving your EFS file data can be managed simply with AWS DataSync - a managed data transfer service that makes it faster and simpler to move data between on-premises storage and Amazon EFS.
- You can schedule automatic incremental backups of your EFS file system using the EFS-to-EFS Backup solution.
- Amazon EFS Infrequent Access (EFS IA) is a new storage class for Amazon EFS that is cost-optimized for files that are accessed less frequently. Customers can use EFS IA by creating a new file system and enabling Lifecycle Management. With Lifecycle Management enabled, EFS automatically will move files that have not been accessed for 30 days from the Standard storage class to the Infrequent Access storage class.

### Performance Modes

- General purpose performance mode (default)
  - Ideal for latency-sensitive use cases.
- Max I/O mode
  - Can scale to higher levels of aggregate throughput and operations per second with a tradeoff of slightly higher latencies for file operations.

### Throughput Modes

- Bursting Throughput mode (default)
  - Throughput scales as your file system grows.
- Provisioned Throughput mode



- You specify the throughput of your file system independent of the amount of data stored.

## Mount Targets

- To access your EFS file system in a VPC, you create one or more **mount targets** in the VPC. A mount target provides an IP address for an NFSv4 endpoint.
- You can create one mount target in each Availability Zone in a region.
- You mount your file system using its DNS name, which will resolve to the IP address of the EFS mount target. Format of DNS is *File-system-id.efs.aws-region.amazonaws.com*
- When using Amazon EFS with an on-premises server, your on-premises server must have a Linux based operating system.

## Components of a File System

- ID
- creation token
- creation time
- file system size in bytes
- number of mount targets created for the file system
- file system state
- mount target

## Data Consistency in EFS

- EFS provides the **open-after-close consistency semantics** that applications expect from NFS.
- Write operations will be durably stored across Availability Zones.
- Applications that perform synchronous data access and perform non-appending writes will have **read-after-write consistency** for data access.

## Managing File Systems

- You can create encrypted file systems. EFS supports encryption in transit and encryption at rest.
- Managing file system network accessibility refers to managing the mount targets:
  - Creating and deleting mount targets in a VPC
  - Updating the mount target configuration
- You can create new tags, update values of existing tags, or delete tags associated with a file system.
- The following list explains the metered data size for different types of file system objects.
  - **Regular files** – the metered data size of a regular file is the logical size of the file rounded to the next 4-KiB increment, except that it may be less for sparse files.
    - A sparse file is a file to which data is not written to all positions of the file before its logical size is reached. For a sparse file, if the actual storage used is less than the logical



---

size rounded to the next 4-KiB increment, Amazon EFS reports actual storage used as the metered data size.

- **Directories** – the metered data size of a directory is the actual storage used for the directory entries and the data structure that holds them, rounded to the next 4 KiB increment. The metered data size doesn't include the actual storage used by the file data.
- **Symbolic links and special files** – the metered data size for these objects is always 4 KiB.
- **File system deletion is a destructive action that you can't undo.** You lose the file system and any data you have in it, and you can't restore the data. You should always unmount a file system before you delete it.
- You can use AWS DataSync to automatically, efficiently, and securely copy files between two Amazon EFS resources, including file systems in different AWS Regions and ones owned by different AWS accounts. Using DataSync to copy data between EFS file systems, you can perform one-time migrations, periodic ingest for distributed workloads, or automate replication for data protection and recovery.

## Mounting File Systems

- To mount your EFS file system on your EC2 instance, use the mount helper in the *amazon-efs-utils* package.
- You can mount your EFS file systems on your on-premises data center servers when connected to your Amazon VPC with AWS Direct Connect or VPN.
- You can use **fstab** to automatically mount your file system using the mount helper whenever the EC2 instance is mounted on reboots.

## Lifecycle Management

- You can choose from four EFS Lifecycle Management policies (14, 30, 60, or 90 days) to automatically move files into the EFS Infrequent Access (EFS IA) storage class and save up to 85% in cost.

## Monitoring File Systems

- Amazon CloudWatch Alarms
- Amazon CloudWatch Logs
- Amazon CloudWatch Events
- AWS CloudTrail Log Monitoring
- Log files on your file system

## Security

- You must have valid credentials to make EFS API requests, such as create a file system.
- You must also have permissions to create or access resources.



- When you first create the file system, there is only one root directory at /. By default, only the root user (UID 0) has read-write-execute permissions.
- Specify EC2 security groups for your EC2 instances and security groups for the EFS mount targets associated with the file system.

## Pricing

- You pay only for the storage used by your file system.
- Costs related to Provisioned Throughput are determined by the throughput values you specify.

## EFS vs EBS

- Performance Comparison

	Amazon EFS	Amazon EBS Provisioned IOPS
Per-operation latency	Low, consistent latency.	Lowest, consistent latency.
Throughput scale	10+ GB per second.	Up to 2 GB per second.

- Storage Comparison

	Amazon EFS	Amazon EBS Provisioned IOPS
Availability and durability	Data is stored redundantly across multiple AZs.	Data is stored redundantly in a single AZ.
Access	Up to thousands of EC2 instances, from multiple AZs, can connect concurrently to a file system.	A single EC2 instance in a single AZ can connect to a file system.
Use cases	Big data and analytics, media processing workflows, content management, web serving, and home directories.	Boot volumes, transactional and NoSQL databases, data warehousing, and ETL.



**Sources:**

<https://docs.aws.amazon.com/efs/latest/ug/>

<https://aws.amazon.com/efs/pricing/>

<https://aws.amazon.com/efs/faq/>

<https://aws.amazon.com/efs/features/>



## Amazon S3 Glacier

- **Long-term archival** solution optimized for infrequently used data, or "cold data."
- Glacier is a REST-based web service.
- You can store an unlimited number of archives and an unlimited amount of data.
- You cannot specify Glacier as the storage class at the time you create an object.
- It is designed to provide average annual durability of 99.99999999% for an archive. Glacier synchronously stores your data across multiple AZs before confirming a successful upload.
- To prevent corruption of data packets over the wire, Glacier uploads the checksum of the data during data upload. It compares the received checksum with the checksum of the received data and validates data authenticity with checksums during data retrieval.
- Glacier works together with **Amazon S3 lifecycle rules** to help you automate archiving of S3 data and reduce your overall storage costs. Requested archival data is copied to S3 One Zone-IA

## Data Model

- **Vault**
  - A container for storing archives.
  - Each vault resource has a unique address with form:  
`https://region-specific endpoint/account-id/vaults/vaultname`
  - You can store an unlimited number of archives in a vault.
  - Vault operations are Region specific.
- **Archive**
  - Can be any data such as a photo, video, or document and is a base unit of storage in Glacier.
  - Each archive has a unique address with form:  
`https://region-specific-endpoint/account-id/vaults/vault-name/archives/archive-id`
- **Job**
  - You can perform a select query on an archive, retrieve an archive, or get an inventory of a vault. Glacier Select runs the query in place and writes the output results to Amazon S3.
  - Select, archive retrieval, and vault inventory jobs are associated with a vault. A vault can have multiple jobs in progress at any point in time.
- **Notification Configuration**
  - Because jobs take time to complete, Glacier supports a notification mechanism to notify you when a job is complete.

## Glacier Operations

- Retrieving an archive (asynchronous operation)
- Retrieving a vault inventory (list of archives) (asynchronous operation)
- Create and delete vaults
- Get the vault description for a specific vault or for all vaults in a region



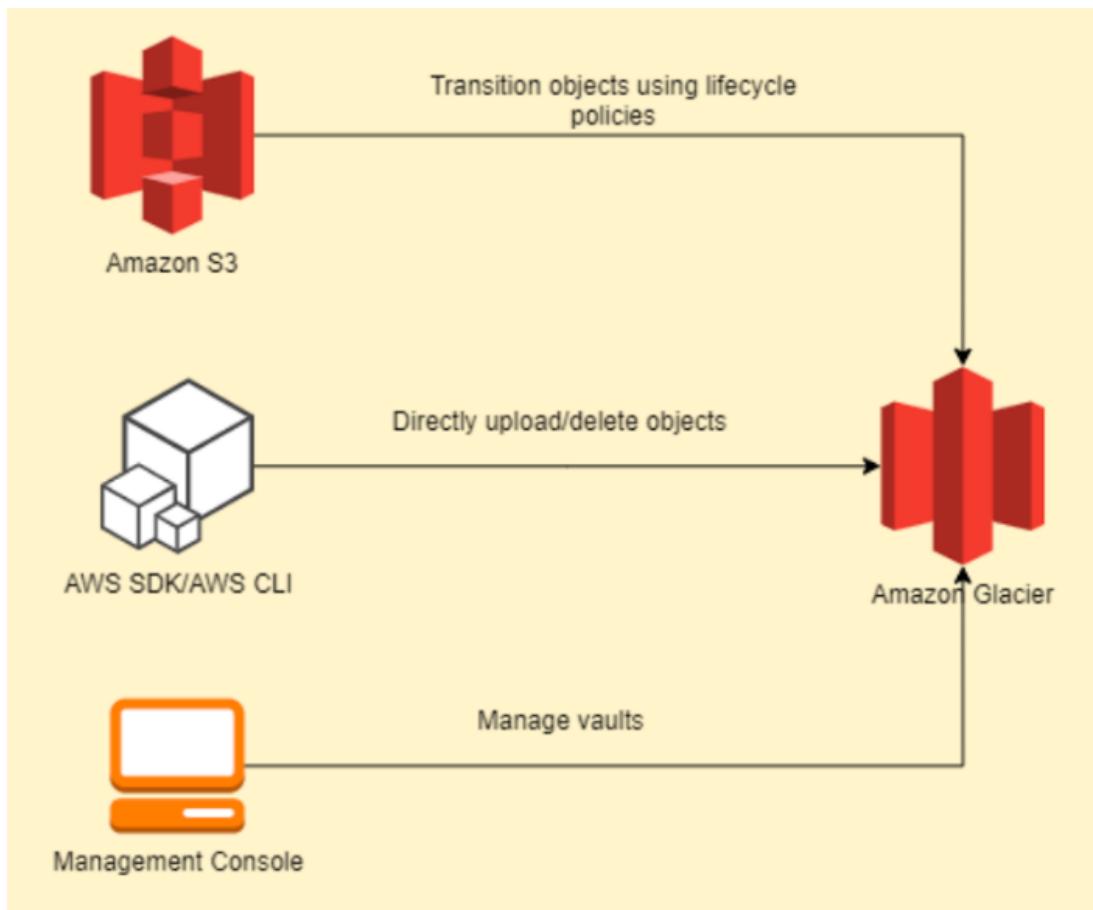
- 
- Set, retrieve, and delete a notification configuration on the vault
  - Upload and delete archives. You cannot update an existing archive.
  - Glacier jobs – select, archive-retrieval, inventory-retrieval

## Vaults

- Vault operations are region specific.
- Vault names must be unique within an account and the region in which the vault is being created.
- You can delete a vault only if there are no archives in the vault as of the last inventory that Glacier computed and there have been no writes to the vault since the last inventory.
- You can retrieve vault information such as the vault creation date, number of archives in the vault, and the total size of all the archives in the vault.
- Glacier maintains an inventory of all archives in each of your vaults for disaster recovery or occasional reconciliation. A **vault inventory** refers to the list of archives in a vault. Glacier updates the vault inventory approximately once a day. Downloading a vault inventory is an asynchronous operation.
- You can assign your own metadata to Glacier vaults in the form of **tags**. A tag is a key-value pair that you define for a vault.
- **Glacier Vault Lock** allows you to easily deploy and enforce compliance controls for individual Glacier vaults with a vault lock policy. You can specify controls such as “**write once read many**” (WORM) in a vault lock policy and lock the policy from future edits. Once locked, the policy can no longer be changed.

## Archives

- Glacier supports the following basic archive operations: upload, download, and delete. Downloading an archive is an asynchronous operation.
- You can upload an archive in a single operation or upload it in parts.
- Using the multipart upload API, you can upload large archives, up to about 10,000 x 4 GB.
- You cannot upload archives to Glacier by using the management console. Use the AWS CLI or write code to make requests, by using either the REST API directly or by using the AWS SDKs.
- You cannot delete an archive using the Amazon S3 Glacier (Glacier) management console. Glacier provides an API call that you can use to delete one archive at a time.



- After you upload an archive, you cannot update its content or its description. The only way you can update the archive content or its description is by deleting the archive and uploading another archive.
- Glacier does not support any additional metadata for the archives.

### Glacier Select

- You can perform filtering operations using simple SQL statements directly on your data in Glacier.
- You can run queries and custom analytics on your data that is stored in Glacier, without having to restore your data to a hotter tier like S3.
- When you perform select queries, Glacier provides three data access tiers:
  - **Expedited** - data accessed is typically made available within 1–5 minutes.
  - **Standard** - data accessed is typically made available within 3–5 hours.
  - **Bulk** - data accessed is typically made available within 5–12 hours.

### Glacier Data Retrieval Policies

- Set data retrieval limits and manage the data retrieval activities across your AWS account in each region.



- Three types of policies:
  - Free Tier Only - you can keep your retrievals within your daily free tier allowance and not incur any data retrieval cost.
  - Max Retrieval Rate - ensures that the peak retrieval rate from all retrieval jobs across your account in a region does not exceed the bytes-per-hour limit you set.
  - No Retrieval Limit

## Security

- Glacier encrypts your data at rest by default and supports secure data transit with SSL.
- Data stored in Amazon Glacier is immutable, meaning that after an archive is created it cannot be updated.
- Access to Glacier requires credentials that AWS can use to authenticate your requests. Those credentials must have permissions to access Glacier vaults or S3 buckets.
- Glacier requires all requests to be signed for authentication protection. To sign a request, you calculate a digital signature using a cryptographic hash function that returns a hash value that you include in the request as your signature.
- Glacier supports policies only at the vault level.
- You can attach identity-based policies to IAM identities.
- A Glacier vault is the primary resource and resource-based policies are referred to as *vault policies*.
- When activity occurs in Glacier, that activity is recorded in a CloudTrail event along with other AWS service events in *Event History*.

## Pricing

- You are charged per GB per month of storage
- You are charged for retrieval operations such as retrieve requests and amount of data retrieved depending on the data access tier - Expedited, Standard, or Bulk
- Upload requests are charged.
- You are charged for data transferred out of Glacier.
- Pricing for Glacier Select is based upon the total amount of data scanned, the amount of data returned, and the number of requests initiated.
- There is a charge if you delete data within 90 days.

## Sources:

<https://docs.aws.amazon.com/amazonglacier/latest/dev/>  
<https://aws.amazon.com/glacier/features/?nc=sn&loc=2>  
<https://aws.amazon.com/glacier/pricing/?nc=sn&loc=3>  
<https://aws.amazon.com/glacier/faqs/?nc=sn&loc=6>



## AWS Storage Gateway

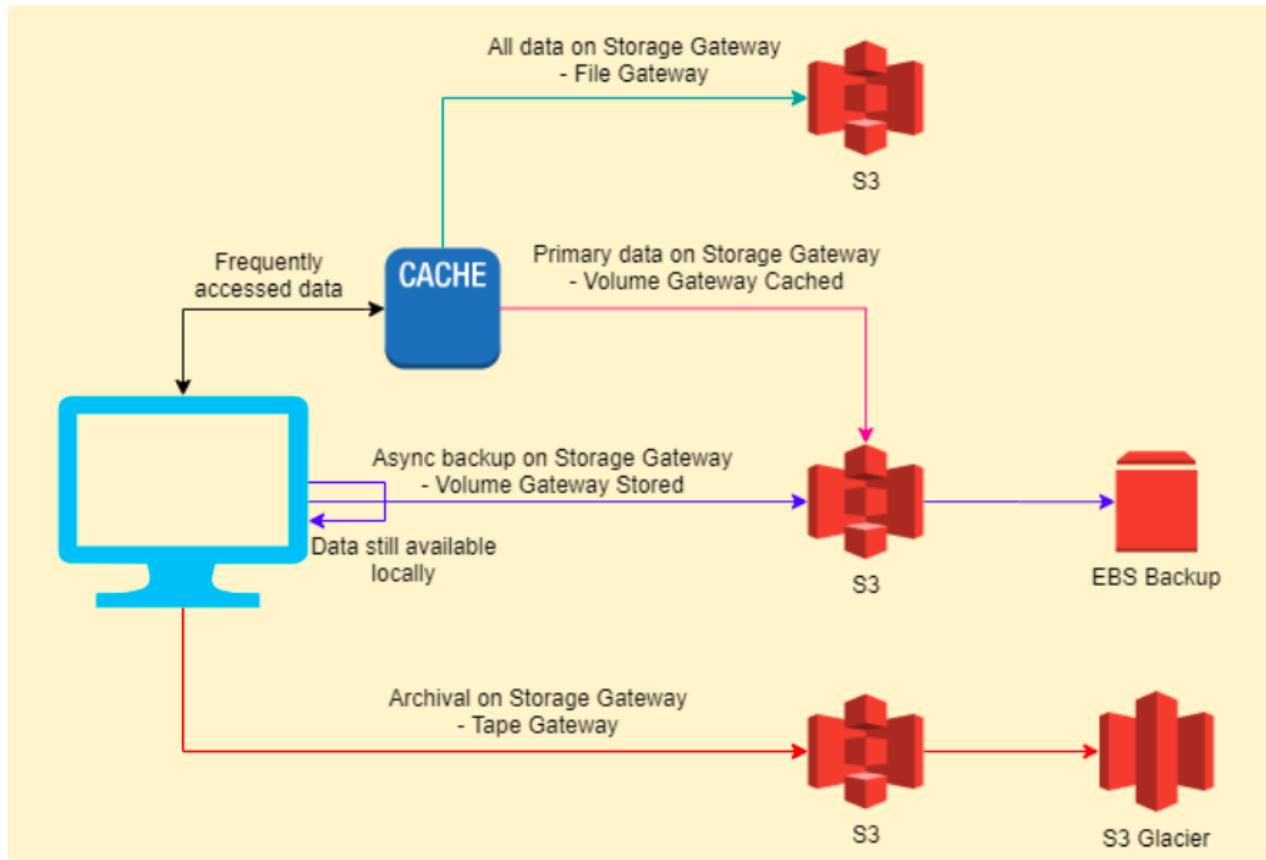
- The service enables **hybrid storage** between on-premises environments and the AWS Cloud.
- It integrates on-premises enterprise applications and workflows with Amazon's block and object cloud storage services through industry standard storage protocols.
- The service stores files as native S3 objects, archives virtual tapes in Amazon Glacier, and stores EBS Snapshots generated by the Volume Gateway with Amazon EBS.
- Storage Solutions
  - **File Gateway** - supports a file interface into S3 and combines a service and a virtual software appliance.
    - The software appliance, or gateway, is deployed into your on-premises environment as a virtual machine running on VMware ESXi or Microsoft Hyper-V hypervisor.
    - File gateway supports
      - S3 Standard
      - S3 Standard - Infrequent Access
      - S3 One Zone - IA
    - With a file gateway, you can do the following:
      - You can store and retrieve files directly using the NFS version 3 or 4.1 protocol.
      - You can store and retrieve files directly using the SMB file system version, 2 and 3 protocol.
      - You can access your data directly in S3 from any AWS Cloud application or service.
      - You can manage your S3 data using lifecycle policies, cross-region replication, and versioning.
    - File Gateway now supports Amazon S3 Object Lock, enabling write-once-read-many (WORM) file-based systems to store and access objects in Amazon S3.
    - Any modifications such as file edits, deletes or renames from the gateway's NFS or SMB clients are stored as new versions of the object, without overwriting or deleting previous versions.
  - **Volume Gateway** - provides cloud-backed storage volumes that you can mount as iSCSI devices from your on-premises application servers.
    - **Cached volumes** – you store your data in S3 and retain a copy of frequently accessed data subsets locally. Cached volumes can range from 1 GiB to 32 TiB in size and must be rounded to the nearest GiB. Each gateway configured for cached volumes can support up to 32 volumes.
    - **Stored volumes** – if you need low-latency access to your entire dataset, first configure your on-premises gateway to store all your data locally. Then asynchronously back up point-in-time snapshots of this data to S3. Stored volumes can range from 1 GiB to 16



---

TiB in size and must be rounded to the nearest GiB. Each gateway configured for stored volumes can support up to 32 volumes.

- AWS Storage Gateway customers using the Volume Gateway configuration for block storage can detach and attach volumes, from and to a Volume Gateway. You can use this feature to migrate volumes between gateways to refresh underlying server hardware, switch between virtual machine types, and move volumes to better host platforms or newer Amazon EC2 instances.
- **Tape Gateway** - archive backup data in Amazon Glacier.
  - Has a virtual tape library (VTL) interface to store data on virtual tape cartridges that you create.
  - Deploy your gateway on an EC2 instance to provision iSCSI storage volumes in AWS.
  - The AWS Storage Gateway service integrates Tape Gateway with Amazon S3 Glacier Deep Archive storage class, allowing you to store virtual tapes in the lowest-cost Amazon S3 storage class.
  - Tape Gateway also has the capability to move your virtual tapes archived in Amazon S3 Glacier to Amazon S3 Glacier Deep Archive storage class, enabling you to further reduce the monthly cost to store long-term data in the cloud by up to 75%.



### Storage Gateway Hosting Options

- As a VM containing the Storage Gateway software, run on VMware ESXi, Microsoft Hyper-V on-premises
- As a VM in VMware Cloud on AWS
- As a hardware appliance on-premises
- As an AMI in an EC2 instance
- Storage Gateway stores volume, snapshot, tape, and file data in the AWS Region in which your gateway is activated. File data is stored in the AWS Region where your S3 bucket is located. The local gateway appliance maintains a cache of recently written or read data so your applications can have low-latency access to data that is stored durably in AWS. The gateways use a **read-through and write-back** cache.

### File Gateway File Share

- As a VM You can create an NFS or SMB file share using the AWS Management Console or service API.
- After your file gateway is activated and running, you can add additional file shares and grant access to S3 buckets.
- You can use a file share to access objects in an S3 bucket that belongs to a different AWS account.



- The AWS Storage Gateway service added support for Access Control Lists (ACLs) to Server Message Block (SMB) shares on the File Gateway, helping enforce data security standards when using the gateway for storing and accessing data in Amazon Simple Storage Service (S3).

## Security

- After your file gateway is activated and running, you can add additional file shares and grant access to S3 buckets.
- You can use AWS KMS to encrypt data written to a virtual tape.
- Storage Gateway uses Challenge-Handshake Authentication Protocol (CHAP) to authenticate iSCSI and initiator connections. CHAP provides protection against playback attacks by requiring authentication to access storage volume targets.
- Authentication and access control with IAM.

## Compliance

- Storage Gateway is HIPAA eligible.
- Storage Gateway in compliance with the Payment Card Industry Data Security Standard (PCI DSS)

## Pricing

- You are charged based on the type and amount of storage you use, the requests you make, and the amount of data transferred out of AWS.
- You are charged only for the amount of data you write to the Tape Gateway tape, not the tape capacity.

## Sources:

<https://docs.aws.amazon.com/storagegateway/latest/userguide/>  
<https://aws.amazon.com/storagegateway/features/>  
<https://aws.amazon.com/storagegateway/pricing/>  
<https://aws.amazon.com/storagegateway/faqs/>



## Amazon FSx

- Amazon FSx is a fully managed third-party file system solution. It uses SSD storage to provide fast performance with low latency.
- There are two available FSx solutions available in AWS:
  - **Amazon FSx for Windows File Server**
    - A fully managed native Microsoft Windows file system with full support for the SMB protocol, Windows NTFS, and Microsoft Active Directory (AD) integration.
    - Common Use Cases
      - File systems that are accessible by multiple users, and can establish permissions at the file or folder level.
      - Application workloads that require shared file storage provided by **Windows-based file systems (NTFS)** and that use the **SMB protocol**.
      - Media workflows like media transcoding, processing, and streaming.
      - Data-intensive analytics workloads.
      - Content management and web serving applications such as IIS
    - Works with the following compute services:
      - Amazon EC2
      - Amazon Workspaces instances
      - Amazon AppStream 2.0 instances
      - VMs running in VMWare Cloud on AWS Environments
    - Works with Microsoft Active Directory (AD) to integrate your file system with your existing Windows environments.
    - Supports the use of AWS Direct Connect or AWS VPN to access your file systems from your on-premises compute instances.
    - Microsoft Windows File Share
      - A **Microsoft Windows file share** is a specific folder in your file system, including that folder's subfolders, which you make accessible to your compute instances with the Server Message Block (SMB) protocol.
      - Your file system comes with a default Windows file share, named *share*. You can create and manage as many other Windows file shares as you want by using the Windows graphical user interface (GUI) tool called Shared Folders.
      - To access your file shares, you use the Windows Map Network Drive functionality to map a drive letter on your compute instance to your Amazon FSx file share.
    - Storage
      - You can configure user storage quotas on your file systems to limit how much data storage that users can consume. After you set quotas, you can track quota status to monitor usage and see when users surpass their quotas.



- You can use **DFS (Distributed File System) Namespaces** to group file shares on multiple file systems into one common folder structure (a namespace) that you use to access the entire file dataset.
- You can reduce your data storage costs by turning on **data deduplication** for your file system. Data deduplication reduces or eliminates redundant data by storing duplicated portions of the dataset only once.
- Performance Metrics
  - Amazon FSx uses SSD storage, provides storage of up to 64 TB per file system, and throughput of up to 2GB/s per file system.
  - You can use DFS Namespaces to create shared common namespaces spanning multiple Amazon FSx file systems to scale-out storage and throughput. Each common namespace is capable of scaling out storage to hundreds of petabytes of data, and performance to tens of GB/s of throughput, and millions of IOPS.
- Migration
  - To migrate your existing file data into Amazon FSx, use Windows's Robust File Copy (RoboCopy) to copy your files (both the data and the full set of metadata like ownership and Access Control Lists) directly to Amazon FSx.
- **Amazon FSx for Lustre**
  - A high-performance file system optimized for fast processing of workloads. Lustre is a popular open-source parallel file system.
  - When linked to an S3 bucket, an FSx for Lustre file system transparently presents S3 objects as files and allows you to write results back to S3.
  - The Lustre file system provides a POSIX-compliant file system interface.
  - Also supports concurrent access to the same file or directory from thousands of compute instances.
  - You can mount your Amazon FSx for Lustre file system from an on-premises client over AWS Direct Connect or VPN, and then use parallel copy tools to import your data to your Amazon FSx for Lustre file system.
  - You can export files and their associated metadata that you have written or modified in your file system to your durable data repository on Amazon S3
  - Performance Metrics
    - Provides sub-millisecond access to your data
    - Allows you to read and write data at speeds of up to hundreds of gigabytes per second of throughput and millions of IOPS.
    - FSx for Lustre provides 200MBps of baseline throughput per TB of storage provisioned. As your file systems scales, your throughput can scale up to hundreds of GB/s.



- Amazon FSx Windows File Servers vs. Amazon EFS vs. Amazon FSx for Lustre

FSx Windows File Servers	EFS	FSx for Lustre
<p>For Windows-based applications, Amazon FSx provides fully managed Windows file servers with features and performance optimized for "lift-and-shift" business-critical application workloads including home directories (user shares), media workflows, and ERP applications. It is accessible from Windows and Linux instances via the <b>SMB</b> protocol.</p>	<p>If you have Linux-based applications, Amazon EFS is a cloud-native fully managed file system that provides simple, scalable, elastic file storage accessible from Linux instances via the <b>NFS</b> protocol.</p>	<p>For compute-intensive and fast processing workloads, like high performance computing (HPC), machine learning, EDA, and media processing, Amazon FSx for Lustre, provides a file system that's optimized for performance, with input and output stored on Amazon S3.</p>

- High Availability and Durability
  - Automatically replicates your data within an Availability Zone (AZ) it resides in (which you specify during creation).
  - Offers single AZ and multi-AZ deployment options for your Windows file-based workloads.
  - Uses the Volume Shadow Copy Service (VSS) to automatically takes highly durable, file-system consistent daily backups to S3
  - Supports restoring individual files and folders to previous versions using Windows shadow copies.
  - The default backup retention period is 7 days.
- Security
  - Supports identity-based authentication over the Server Message Block (SMB) protocol through Microsoft Active Directory.
  - Amazon FSx automatically encrypts your data at-rest using AWS KMS and in-transit using SMB Kerberos session keys.
  - Amazon FSx is ISO, PCI-DSS, and SOC compliant, and is HIPAA eligible.
  - Integrated with AWS CloudTrail that monitors and logs your API calls.
- Pricing
  - You are billed hourly for your Amazon FSx file systems, based on your configured storage capacity (priced per GB-month) and throughput capacity (priced per MBps-month).
  - You are billed hourly for your backup storage (priced per GB-month). You can enable data deduplication to automatically reduce costs associated with redundant data by storing duplicated portions of your dataset only once.

#### Sources:

<https://aws.amazon.com/fsx/windows/?nc=sn&loc=2>  
<https://docs.aws.amazon.com/fsx/latest/WindowsGuide/what-is.html>  
<https://aws.amazon.com/fsx/windows/faqs/?nc=sn&loc=7>  
<https://aws.amazon.com/fsx/lustre/?nc=sn&loc=3>  
<https://docs.aws.amazon.com/fsx/latest/LustreGuide/what-is.html>  
<https://aws.amazon.com/fsx/lustre/faqs/?nc=sn&loc=5>



## DATABASE

AWS offers purpose-built databases for all your application needs. Whether you need a Relational, Key-Value, In-memory, or any other type of data store, AWS would most likely have a database service that you can use.

Relational databases store data with predefined schemas and “relationships” between the tables, hence the “Relational” name. It is designed to support ACID (Atomicity, Consistency, Isolation, Durability) transactions with strong data consistency to maintain referential integrity. Key-value databases are suitable for storing and retrieving large volumes of data. It delivers quick response times even in large volumes of concurrent requests.

In-memory databases are primarily used for applications that require real-time access to data. It is capable of delivering data to applications in microseconds and not just in milliseconds since the data are directly stored in memory and not on disk. Aside from this, AWS also offers Document, Time Series, Ledger, and many other database types.

Database Type	Use Cases	AWS Service/s
Relational	Traditional applications, ERP, CRM, e-commerce	<a href="#">Amazon Aurora</a> <a href="#">Amazon RDS</a> <a href="#">Amazon Redshift</a>
Key-value	High-traffic web apps, e-commerce systems, gaming applications	<a href="#">Amazon DynamoDB</a>
Document	Content management, catalogs, user profiles	<a href="#">Amazon DocumentDB (with MongoDB compatibility)</a>
In-memory	Caching, session management, gaming leaderboards, geospatial applications	<a href="#">Amazon ElastiCache for Memcached</a> <a href="#">Amazon ElastiCache for Redis</a>
Wide column	High scale industrial apps for equipment maintenance, fleet management, and route optimization	<a href="#">Amazon Keyspaces (for Apache Cassandra)</a>
Graph	Fraud detection, social networking, recommendation engines	<a href="#">Amazon Neptune</a>
Time series	IoT applications, DevOps, industrial telemetry	<a href="#">Amazon Timestream</a>
Ledger	Systems of record, supply chain, registrations, banking transactions	<a href="#">Amazon QLDB</a>

Tutorials Dojo

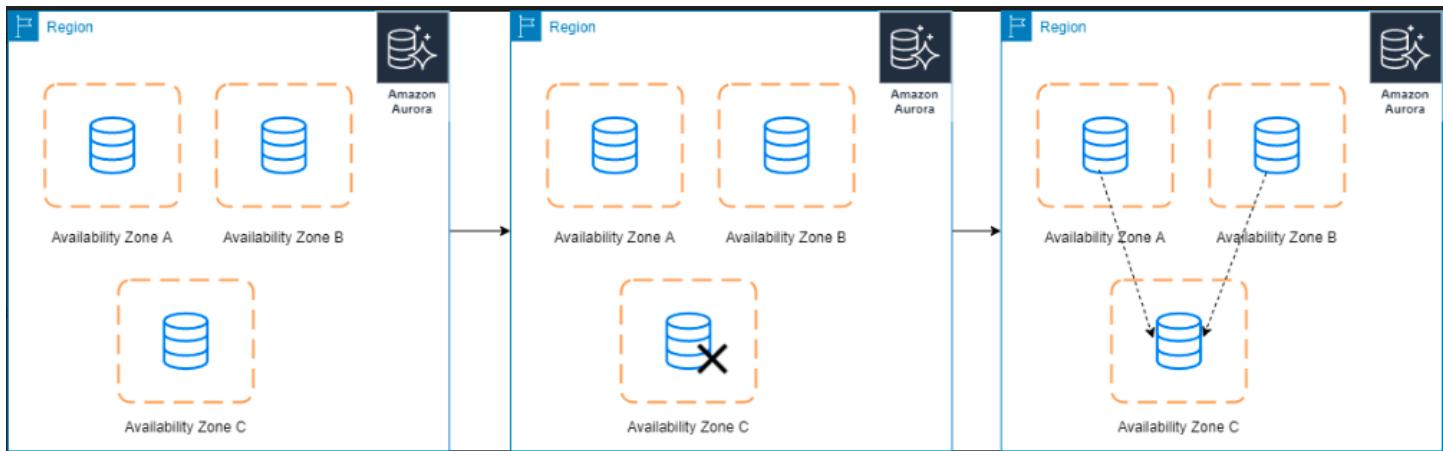


## Amazon Aurora

- A fully managed relational database engine that's compatible with **MySQL** and **PostgreSQL**.
- With some workloads, Aurora can deliver up to five times the throughput of MySQL and up to three times the throughput of PostgreSQL.
- Aurora includes a high-performance storage subsystem. The underlying storage grows automatically as needed, up to 64 terabytes. **The minimum storage is 10GB.**
- Aurora will keep your database up-to-date with the latest patches.
- Aurora supports quick, efficient cloning operations.
  - You can share your Amazon Aurora DB clusters with other AWS accounts for quick and efficient database cloning.
- Aurora is fault-tolerant and self-healing.
- DB Clusters
  - An Aurora **DB cluster** consists of one or more DB instances and a cluster volume that manages the data for those DB instances.
  - An Aurora **cluster volume** is a virtual database storage volume that spans multiple AZs, with each AZ having a copy of the DB cluster data.
  - Cluster Types:
    - **Primary DB instance** – Supports read and write operations, and performs all of the data modifications to the cluster volume. Each Aurora DB cluster has one primary DB instance.
    - **Aurora Replica** – Connects to the same storage volume as the primary DB instance and supports only read operations. Each Aurora DB cluster can have up to 15 Aurora Replicas in addition to the primary DB instance. Aurora automatically fails over to an Aurora Replica in case the primary DB instance becomes unavailable. You can specify the failover priority for Aurora Replicas. Aurora Replicas can also offload read workloads from the primary DB instance.
- Aurora Endpoints
  - **Cluster endpoint** - connects to the current primary DB instance for a DB cluster. This endpoint is the only one that can perform write operations. Each Aurora DB cluster has one cluster endpoint and one primary DB instance.
  - **Reader endpoint** - connects to one of the available Aurora Replicas for that DB cluster. Each Aurora DB cluster has one reader endpoint. The reader endpoint provides load-balancing support for read-only connections to the DB cluster. Use the reader endpoint for read operations, such as queries. You can't use the reader endpoint for write operations.
  - **Custom endpoint** - represents a set of DB instances that you choose. When you connect to the endpoint, Aurora performs load balancing and chooses one of the instances in the group to handle the connection. You define which instances this endpoint refers to, and you decide what purpose the endpoint serves.



- **Instance endpoint** - connects to a specific DB instance within an Aurora cluster. The instance endpoint provides direct control over connections to the DB cluster. The main way that you use instance endpoints is to diagnose capacity or performance issues that affect one specific instance in an Aurora cluster.
- When you connect to an Aurora cluster, the host name and port that you specify point to an intermediate handler called an *endpoint*.
- Types of Endpoints
- Storage and Reliability
  - Aurora data is stored in the cluster volume, which is designed for reliability. A cluster volume consists of copies of the data across multiple Availability Zones in a single AWS Region.
  - Aurora automatically detects failures in the disk volumes that make up the cluster volume. When a segment of a disk volume fails, Aurora immediately repairs the segment. When Aurora repairs the disk segment, it uses the data in the other volumes that make up the cluster volume to ensure that the data in the repaired segment is current.



- Aurora preloads the buffer pool with the pages for known common queries that are stored in an in-memory page cache when a database starts up after it has been shut down or restarted after a failure.
- Aurora is designed to recover from a crash almost instantaneously and continue to serve your application data without the binary log. Aurora performs crash recovery asynchronously on parallel threads, so that your database is open and available immediately after a crash.
- Amazon Aurora Auto Scaling works with Amazon CloudWatch to automatically add and remove Aurora Replicas in response to changes in performance metrics that you specify. This feature is available in the PostgreSQL-compatible edition of Aurora. There is no additional cost to use Aurora Auto Scaling beyond what you already pay for Aurora and CloudWatch alarms.
- High Availability and Fault Tolerance
  - When you create Aurora Replicas across Availability Zones, RDS automatically provisions and maintains them synchronously. The primary DB instance is synchronously replicated across



- 
- Availability Zones to Aurora Replicas to provide data redundancy, eliminate I/O freezes, and minimize latency spikes during system backups.
- An Aurora DB cluster is fault tolerant by design. If the primary instance in a DB cluster fails, Aurora automatically fails over to a new primary instance in one of two ways:
    - By promoting an existing Aurora Replica to the new primary instance
    - By creating a new primary instance
  - Aurora storage is also self-healing. Data blocks and disks are continuously scanned for errors and repaired automatically.
  - Aurora backs up your cluster volume automatically and retains restore data for the length of the backup retention period, from 1 to 35 days.
  - Aurora automatically maintains **6 copies of your data across 3 Availability Zones** and will automatically attempt to recover your database in a healthy AZ with no data loss.
  - With Aurora MySQL, you can set up cross-region Aurora Replicas using either logical or physical replication. Aurora PostgreSQL does not currently support cross-region replicas.
- Aurora Global Database (MySQL)
    - An Aurora global database spans multiple AWS Regions, enabling low latency global reads and disaster recovery from region-wide outages.
    - Consists of one primary AWS Region where your data is mastered, and one read-only, secondary AWS Region.
    - Aurora global databases use dedicated infrastructure to replicate your data.
    - Aurora global databases are available in these AWS Regions:
      - US East (N. Virginia)
      - US East (Ohio)
      - US West (Oregon)
      - EU (Ireland)
    - Aurora global databases introduce a higher level of failover capability than a default Aurora cluster.
    - An Aurora cluster can recover in less than 1 minute even in the event of a complete regional outage. This provides your application with an effective Recovery Point Objective (RPO) of 5 seconds and a Recovery Time Objective (RTO) of less than 1 minute.
  - DB Cluster Configurations
    - Aurora supports two types of instance classes
      - **Memory Optimized**
      - **Burstable Performance**
    - **Aurora Serverless** is an on-demand, autoscaling configuration for Amazon Aurora (supports both MySQL and PostgreSQL). An *Aurora Serverless DB cluster* automatically starts up, shuts down, and scales up or down capacity based on your application's needs.
      - A non-Serverless DB cluster for Aurora is called a *provisioned DB cluster*.
      - Instead of provisioning and managing database servers, you specify Aurora Capacity Units (ACUs). Each ACU is a combination of processing and memory capacity.



- You can choose to pause your Aurora Serverless DB cluster after a given amount of time with no activity. The DB cluster automatically resumes and services the connection requests after receiving requests.
- Aurora Serverless does not support fast failover, but it supports **automatic multi-AZ failover**.
- The cluster volume for an Aurora Serverless cluster is always encrypted. You can choose the encryption key, but not turn off encryption.
- You can set the following specific values:
  - **Minimum Aurora capacity unit** – Aurora Serverless can reduce capacity down to this capacity unit.
  - **Maximum Aurora capacity unit** – Aurora Serverless can increase capacity up to this capacity unit.
  - **Pause after inactivity** – The amount of time with no database traffic to scale to zero processing capacity.
- You pay by the second and only when the database is in use.
- You can share snapshots of Aurora Serverless DB clusters with other AWS accounts or publicly. You also have the ability to copy Aurora Serverless DB cluster snapshots across AWS regions.
  -
- Limitations of Aurora Serverless
  - Aurora Serverless is only available for the following:
    - Aurora with MySQL version 5.6 compatibility
    - Aurora with PostgreSQL version 10.7 compatibility
    -
  - The port number for connections must be:
    - 3306 for Aurora MySQL
    - 5432 for Aurora PostgreSQL
  - You can't give an Aurora Serverless DB cluster a public IP address. You can access an Aurora Serverless DB cluster only from within a virtual private cloud (VPC) based on the Amazon VPC service.
  - Each Aurora Serverless DB cluster requires two AWS PrivateLink endpoints. If you reach the limit for PrivateLink endpoints within your VPC, you can't create any more Aurora Serverless clusters in that VPC.
  - A DB subnet group used by Aurora Serverless can't have more than one subnet in the same Availability Zone.
  - Changes to a subnet group used by an Aurora Serverless DB cluster are not applied to the cluster.
  - Aurora Serverless doesn't support the following features:
    - Loading data from an Amazon S3 bucket
    - Saving data to an Amazon S3 bucket
    - Invoking an AWS Lambda function with an Aurora MySQL native function



- Aurora Replicas
- Backtrack
- Multi-master clusters
- Database cloning
- IAM database authentication
- Restoring a snapshot from a MySQL DB instance
- Amazon RDS Performance Insights
- When you reboot the primary instance of an Aurora DB cluster, RDS also automatically restarts all of the Aurora Replicas in that DB cluster. When you reboot the primary instance of an Aurora DB cluster, no failover occurs. When you reboot an Aurora Replica, no failover occurs.
- **Deletion protection** is enabled by default when you create a production DB cluster using the AWS Management Console. However, deletion protection is disabled by default if you create a cluster using the AWS CLI or API.
  - For Aurora MySQL, you can't delete a DB instance in a DB cluster if both of the following conditions are true:
    - The DB cluster is a Read Replica of another Aurora DB cluster.
    - The DB instance is the only instance in the DB cluster.

#### Aurora Multi Master

- The feature is available on Aurora MySQL 5.6
- Allows you to create multiple read-write instances of your Aurora database across multiple Availability Zones, which enables uptime-sensitive applications to achieve continuous write availability through instance failure.
- In the event of instance or Availability Zone failures, Aurora Multi-Master enables the Aurora database to maintain read and write availability with zero application downtime. There is no need for database failovers to resume write operations.

#### Tags

- You can use Amazon RDS tags to add metadata to your RDS resources.
- Tags can be used with IAM policies to manage access and to control what actions can be applied to the RDS resources.
- Tags can be used to track costs by grouping expenses for similarly tagged resources.

#### Monitoring

- Subscribe to **Amazon RDS events** to be notified when changes occur with a DB instance, DB cluster, DB cluster snapshot, DB parameter group, or DB security group.
- Database log files
- **RDS Enhanced Monitoring** – Look at metrics in real time for the operating system.
- **RDS Performance Insights** monitors your Amazon RDS DB instance load so that you can analyze and troubleshoot your database performance.
- Use CloudWatch Metrics, Alarms and Logs

#### Security

- Use IAM to control access.



- To control which devices and EC2 instances can open connections to the endpoint and port of the DB instance for Aurora DB clusters in a VPC, you use a VPC security group.
- You can make endpoint and port connections using Transport Layer Security (TLS) / Secure Sockets Layer (SSL). In addition, firewall rules can control whether devices running at your company can open connections to a DB instance.
- Use RDS encryption to secure your RDS instances and snapshots at rest.
- You can authenticate to your DB cluster using AWS IAM database authentication. IAM database authentication works with Aurora MySQL and Aurora PostgreSQL. With this authentication method, you don't need to use a password when you connect to a DB cluster. Instead, you use an *authentication token*, which is a unique string of characters that Amazon Aurora generates on request.

#### Aurora for MySQL

- Performance Enhancements
  - Push-Button Compute Scaling
  - Storage Auto-Scaling
  - Low-Latency Read Replicas
  - Serverless Configuration
  - Custom Database Endpoints
  - *Fast insert* accelerates parallel inserts sorted by primary key.
  - **Aurora MySQL parallel query** is an optimization that parallelizes some of the I/O and computation involved in processing data-intensive queries.
  - You can use the *high-performance Advanced Auditing* feature in Aurora MySQL to audit database activity. To do so, you enable the collection of audit logs by setting several DB cluster parameters.
- Scaling
  - Instance scaling - scale your Aurora DB cluster by modifying the DB instance class for each DB instance in the DB cluster.
  - Read scaling - as your read traffic increases, you can create additional Aurora Replicas and connect to them directly to distribute the read load for your DB cluster.



Feature	Amazon Aurora Replicas	MySQL Replicas
Number of Replicas	Up to 15	Up to 5
Replication type	Asynchronous (milliseconds)	Asynchronous (seconds)
Performance impact on primary	Low	High
Act as failover target	Yes (no data loss)	Yes (potentially minutes of data loss)
Automated failover	Yes	No
Support for user-defined replication delay	No	Yes
Support for different data or schema vs. primary	No	Yes

 Tutorials Dojo

- Aurora for PostgreSQL
  - Performance Enhancements
    - Push-button Compute Scaling
    - Storage Auto-Scaling
    - Low-Latency Read Replicas
    - Custom Database Endpoints
  - Scaling
    - Instance scaling
    - Read scaling
  - Amazon Aurora PostgreSQL now supports logical replication. With logical replication, you can replicate data changes from your Aurora PostgreSQL database to other databases using native PostgreSQL replication slots, or data replication tools such as the AWS Database Migration Service.
  - Rebooting the primary instance of an Amazon Aurora DB cluster also automatically reboots the Aurora Replicas for that DB cluster, in order to re-establish an entry point that guarantees read/write consistency across the DB cluster.



- You can import data (supported by the PostgreSQL COPY command) stored in an Amazon S3 bucket into a PostgreSQL table.
- Pricing
  - You are charged for DB instance hours, I/O requests, Backup storage and Data transfer.
  - You can purchase **On-Demand Instances** and pay by the hour for the DB instance hours that you use, or **Reserved Instances** to reserve a DB instance for a one-year or three-year term and receive a significant discount compared to the on-demand DB instance pricing.

**Sources:**

<https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/>  
<https://aws.amazon.com/rds/aurora/details/mysql-details/>  
<https://aws.amazon.com/rds/aurora/details/postgresql-details/>  
<https://aws.amazon.com/rds/aurora/global-database/>  
<https://aws.amazon.com/rds/aurora/parallel-query/>  
<https://aws.amazon.com/rds/aurora/serverless/>  
<https://aws.amazon.com/rds/aurora/pricing/>  
<https://aws.amazon.com/rds/aurora/faqs/>



## Amazon DynamoDB

- NoSQL database service that provides fast and predictable performance with seamless scalability.
- Offers encryption at rest.
- You can create database tables that can store and retrieve any amount of data, and serve any level of request traffic.
- You can scale up or scale down your tables' throughput capacity without downtime or performance degradation, and use the AWS Management Console to monitor resource utilization and performance metrics.
- Provides on-demand backup capability as well as enable point-in-time recovery for your DynamoDB tables. With point-in-time recovery, you can restore that table to any point in time during the **last 35 days**.
- All of your data is stored in partitions, backed by solid state disks (SSDs) and automatically replicated across multiple AZs in an AWS region, providing built-in high availability and data durability.
- You can create tables that are automatically replicated across two or more AWS Regions, with full support for multi-master writes.
- AWS now specifies the IP address ranges for Amazon DynamoDB endpoints. You can use these IP address ranges in your routing and firewall policies to control outbound application traffic. You can also use these ranges to control outbound traffic for applications in your Amazon Virtual Private Cloud, behind AWS Virtual Private Network or AWS Direct Connect.

## Core Components

- **Tables** - a collection of items
  - DynamoDB stores data in a table, which is a collection of data.
  - Are schemaless.
  - There is an initial limit of 256 tables per region.
- **Items** - a collection of attributes
  - DynamoDB uses **primary keys** to uniquely identify each item in a table and **secondary indexes** to provide more querying flexibility.
  - Each table contains zero or more items.
- **Attributes** - a fundamental data element
  - DynamoDB supports nested attributes up to 32 levels deep.
- **Primary Key** - uniquely identifies each item in the table, so that no two items can have the same key. Must be scalar.
  - **Partition key** - a simple primary key, composed of one attribute.
  - **Partition key and sort key (composite primary key)** - composed of two attributes.
  - DynamoDB uses the partition key value as input to an internal hash function. The output from the hash function determines the partition in which the item will be stored. All items with the same partition key are stored together, in sorted order by sort key value. If no sort key is used, no two items can have the same partition key value.



- **Secondary Indexes** - lets you query the data in the table using an alternate key, in addition to queries against the primary key.
  - You can create one or more secondary indexes on a table.
  - Two kinds of indexes:
    - **Global secondary index** – An index with a partition key and sort key that can be different from those on the table.

Table settings

Default settings provide the fastest way to get started with your table. You can modify these default settings now or after your table has been created.

Use default settings

Secondary indexes

Add index

Primary key\* Partition key  
 String

Add sort key

Index name\* user-index

Projected attributes All

Create as Local Secondary Index

Cancel **Add index**

Read/write capacity

Select on-demand if you want to provision throughput instead. Provisioned throughput costs if you exceed the provisioned amount. See the Developer Guide to learn more.

Read/write capacity mode

provisioned to save page and DynamoDB

- **Local secondary index** – An index that has the same partition key as the table, but a different sort key.



Table settings

Default settings provide the fastest way to get started with your table. You can modify these default settings now or after your table has been created.

Use default settings

Secondary indexes

Add index

Primary key\* Partition key

String

Add sort key

String

Index name\* id-age-index

Projected attributes All

Create as Local Secondary Index

Cancel **Add index**

Read/write capacity

Select on-demand if you want to pay only for the throughput you use. Or provisioned if you want to control your costs by specifying the amount of throughput you need. Learn more in the Developer Guide to learn how to choose the right model for your application.

Read/write capacity mode

Provisioned capacity

provisioned to save page and DynamoDB

- You can define up to 20 global secondary indexes and 5 local secondary indexes per table.
- **DynamoDB Streams** - an optional feature that captures data modification events in DynamoDB tables.
  - The naming convention for DynamoDB Streams endpoints is `streams.dynamodb..amazonaws.com`
  - Each event is represented by a *stream record*, and captures the following events:
    - A new item is added to the table: captures an image of the entire item, including all of its attributes.
    - An item is updated: captures the "before" and "after" image of any attributes that were modified in the item.
    - An item is deleted from the table: captures an image of the entire item before it was deleted.
  - Each stream record also contains the name of the table, the event timestamp, and other metadata.
  - Stream records are organized into groups, or **shards**. Each shard acts as a container for multiple stream records, and contains information required for accessing and iterating through these records.
  - Stream records have a lifetime of 24 hours; after that, they are automatically removed from the stream.



- You can use DynamoDB Streams together with AWS Lambda to create a *trigger*, which is a code that executes automatically whenever an event of interest appears in a stream.
- DynamoDB Streams enables powerful solutions such as data replication within and across Regions, materialized views of data in DynamoDB tables, data analysis using Kinesis materialized views, and much more.

## Data Types for Attributes

- **Scalar Types** – A scalar type can represent exactly one value. The scalar types are number, string, binary, Boolean, and null. Primary keys should be scalar types.
- **Document Types** – A document type can represent a complex structure with nested attributes—such as you would find in a JSON document. The document types are list and map.
- **Set Types** – A set type can represent multiple scalar values. The set types are string set, number set, and binary set.

## Other Notes:

- When you read data from a DynamoDB table, the response might not reflect the results of a recently completed write operation. The response might include some stale data, but you should **eventually have consistent reads**.
- When you request a **strongly consistent read**, DynamoDB returns a response with the most up-to-date data, reflecting the updates from all prior write operations that were successful. A strongly consistent read might not be available if there is a network delay or outage.
- DynamoDB does not support strongly consistent reads across AWS regions
- When you create a table or index in DynamoDB, you must specify your throughput capacity requirements for read and write activity in terms of:
  - One **read capacity unit** represents one strongly consistent read per second, or two eventually consistent reads per second, for an item up to 4 KB in size. If you need to read an item that is larger than 4 KB, DynamoDB will need to consume additional read capacity units.
  - One **write capacity unit** represents one write per second for an item up to 1 KB in size. If you need to write an item that is larger than 1 KB, DynamoDB will need to consume additional write capacity units.
- *Throttling* prevents your application from consuming too many capacity units. DynamoDB can throttle read or write requests that exceed the throughput settings for a table, and can also throttle read requests exceeds for an index.
- When a request is throttled, it fails with an **HTTP 400** code (Bad Request) and a *ProvisionedThroughputExceededException*.



## Throughput Management

- Provisioned throughput - manually defined maximum amount of capacity that an application can consume from a table or index. If your application exceeds your provisioned throughput settings, it is subject to request throttling. Free tier eligible.
  - DynamoDB auto scaling
    - Define a range (upper and lower limits) for **read and write capacity units**, and define a target utilization percentage within that range.
    - A table or a global secondary index can increase its **provisioned read and write capacity** to handle sudden increases in traffic, without request throttling.
    - DynamoDB auto scaling can decrease the throughput when the workload decreases so that you don't pay for unused provisioned capacity.
  - Reserved capacity - with reserved capacity, you pay a one-time upfront fee and commit to a minimum usage level over a period of time, for cost-saving solutions.



## Read/write capacity mode

Select on-demand if you want to pay only for the read and writes you perform, with no capacity planning required. Select provisioned to save on throughput costs if you can reliably estimate your application's throughput requirements. See the [DynamoDB pricing page](#) and [DynamoDB Developer Guide](#) to learn more.

Read/write capacity mode can be changed later.

- Provisioned (free-tier eligible)
- On-demand

## Provisioned capacity

	Read capacity units	Write capacity units
Table	5	5
Estimated cost	\$2.91 / month ( <a href="#">Capacity calculator</a> )	

## Auto Scaling

<input checked="" type="checkbox"/> Read capacity	<input checked="" type="checkbox"/> Write capacity
Target utilization	70 %
Minimum provisioned capacity	5 units
Maximum provisioned capacity	40000 units
<input checked="" type="checkbox"/> Apply same settings to global secondary indexes	

- Amazon DynamoDB on-demand is a flexible capacity mode for DynamoDB capable of serving thousands of requests per second without capacity planning. When you choose on-demand capacity mode, DynamoDB instantly accommodates your workloads as they ramp up or down to any previously reached traffic level. If a workload's traffic level hits a new peak, DynamoDB adapts rapidly to accommodate the workload. DynamoDB on-demand offers simple pay-per-request pricing for read and write requests so that you only pay for what you use, making it easy to balance costs and performance.



### Read/write capacity mode

Select on-demand if you want to pay only for the reads and writes you perform, with no capacity planning required. Select provisioned to save on throughput costs if you can reliably estimate your application's throughput requirements. See the [DynamoDB pricing page](#) and [DynamoDB Developer Guide](#) to learn more.

Read/write capacity mode can be changed later.

- Provisioned (free-tier eligible)  
 On-demand

### Provisioned capacity

Not applicable because read/write capacity mode is on-demand.

### Auto Scaling

Not applicable because read/write capacity mode is on-demand.

## Capacity Unit Consumption

- CUC for Reads - strongly consistent read request consumes one read capacity unit, while an eventually consistent read request consumes 0.5 of a read capacity unit.
  - GetItem - reads a single item from a table.
  - BatchGetItem - reads up to 100 items, from one or more tables.
  - Query - reads multiple items that have the same partition key value.
  - Scan - reads all of the items in a table
- CUC for Writes
  - PutItem - writes a single item to a table.
  - UpdateItem - modifies a single item in the table.
  - DeleteItem - removes a single item from a table.
  - BatchWriteItem - writes up to 25 items to one or more tables.

## DynamoDB Auto Scaling

- When you use the AWS Management Console to create a new table, DynamoDB auto scaling is enabled for that table by default.
- Uses the AWS Application Auto Scaling service to dynamically adjust provisioned throughput capacity on your behalf, in response to actual traffic patterns.
- You create a *scaling policy* for a table or a global secondary index. The scaling policy specifies whether you want to scale read capacity or write capacity (or both), and the minimum and maximum provisioned capacity unit settings for the table or index. The scaling policy also contains a *target utilization*, which is the percentage of consumed provisioned throughput at a point in time.
- DynamoDB auto scaling doesn't prevent you from manually modifying provisioned throughput settings.



- If you enable DynamoDB auto scaling for a table that has one or more global secondary indexes, AWS highly recommends that you also apply auto scaling uniformly to those indexes.

## Tagging

- Tags can help you:
  - Quickly identify a resource based on the tags you've assigned to it.
  - See AWS bills broken down by tags.
- Each DynamoDB table can have only one tag with the same key. If you try to add an existing tag (same key), the existing tag value will be updated to the new value.
- Maximum number of tags per resource: 50

## DynamoDB Items

- You can use the *UpdateItem* operation to implement an **atomic counter** - a numeric attribute that is incremented, unconditionally, without interfering with other write requests.
- DynamoDB optionally supports conditional writes for these operations: *PutItem*, *UpdateItem*, *DeleteItem*. A conditional write will succeed only if the item attributes meet one or more expected conditions.
- Conditional writes can be *idempotent* if the conditional check is on the same attribute that is being updated. DynamoDB performs a given write request only if certain attribute values in the item match what you expect them to be at the time of the request.
- Expressions
  - To get only a few attributes of an item, use a **projection expression**.
  - An **expression attribute name** is a placeholder that you use in an expression, as an alternative to an actual attribute name. An expression attribute name must begin with a #, and be followed by one or more alphanumeric characters.
  - **Expression attribute values** are substitutes for the actual values that you want to compare – values that you might not know until runtime. An expression attribute value must begin with a :, and be followed by one or more alphanumeric characters.
  - For *PutItem*, *UpdateItem* and *DeleteItem* operations, you can specify a **condition expression** to determine which items should be modified. If the condition expression evaluates to true, the operation succeeds; otherwise, the operation fails.
  - An **update expression** specifies how *UpdateItem* will modify the attributes of an item—for example, setting a scalar value, or removing elements from a list or a map.

## Time To Live (TTL)

- Allows you to define when items in a table expire so that they can be automatically deleted from the database.

## DynamoDB Queries



- The *Query* operation finds items based on primary key values. You can query any table or secondary index that has a composite primary key (a partition key and a sort key).

TestTables Close

Overview Items Metrics Alarms Capacity Indexes Global Tables Backups More ▾

Create item Actions ▾

Scan: [Table] TestTables: id ▾ Viewing 0 to 0 items

Query [Table] TestTables: id

Partition key id String = 1 Add filter

Sort Ascending Descending

Attributes All Projected

Start search Cancel changes

id i

- A key condition expression is a search criteria that determines the items to be read from the table or index.
- You must specify the partition key name and value as an equality condition.
- You can optionally provide a second condition for the sort key. The sort key condition must use one of the following comparison operators: `=`, `<`, `<=`, `>`, `>=`, `BETWEEN`, `AND`
- A single *Query* operation can retrieve a maximum of 1 MB of data.
- For further refining of Query results, you can optionally provide a **filter expression**, to determine which items within the Query results should be returned to you. All of the other results are discarded.
- The *Query* operation allows you to limit the number of items that it returns in the result by setting the **Limit** parameter to the maximum number of items that you want.
- DynamoDB paginates the results from *Query* operations, where *Query* results are divided into "pages" of data that are 1 MB in size (or less).
- ScannedCount** is the number of items that matched the key condition expression, before a filter expression (if present) was applied.
- Count** is the number of items that remain, after a filter expression (if present) was applied.

## DynamoDB Scans



- A Scan operation reads every item in a table or a secondary index. By default, a Scan operation returns all of the data attributes for every item in the table or index.

The screenshot shows the AWS DynamoDB console for the 'TestTables' table. The 'Items' tab is selected. A 'Scan' operation is being performed on the 'TestTables: id' table. A filter is applied with the condition 'name = daphne'. A callout box at the bottom left contains the following text: 'An item consists of one or more attributes. Each attribute consists of a name, a data type, and a value. When you read or write an item, the only attributes that are required are those that make up the primary key.' It also includes a link to 'More info'.

- Scan always returns a result set. If no matching items are found, the result set will be empty.
- A single Scan request can retrieve a maximum of 1 MB of data.
- You can optionally provide a filter expression.
- You can limit the number of items that is returned in the result.
- DynamoDB paginates the results from Scan operations.
- ScannedCount is the number of items evaluated, before any ScanFilter is applied.
- Count is the number of items that remain, after a filter expression (if present) was applied.
- A Scan operation performs eventually consistent reads, by default.
- By default, the Scan operation processes data sequentially.

## On-Demand Backup and Restore

- You can use IAM to restrict DynamoDB backup and restore actions for some resources.
- All backup and restore actions are captured and recorded in AWS CloudTrail.
- Backups
  - Each time you create an on-demand backup, the entire table data is backed up.



- All backups and restores in DynamoDB work without consuming any provisioned throughput on the table.
- DynamoDB backups do not guarantee causal consistency across items; however, the skew between updates in a backup is usually much less than a second.
- Backup and restore works only in the **same AWS Region** as the source table.
- Included in the backup are:
  - Database data
  - Global secondary indexes
  - Local secondary indexes
  - Streams
  - Provisioned read and write capacity
- While a backup is in progress, you can't do the following:
  - Pause or cancel the backup operation.
  - Delete the source table of the backup.
  - Disable backups on a table if a backup for that table is in progress.
- Restore
  - You cannot overwrite an existing table during a restore operation.
  - You restore backups to a new table.
  - For tables with even data distribution across your primary keys, the restore time is proportional to the largest single partition by item count and not the overall table size.
  - If your source table contains data with significant skew, the time to restore may increase.

## DynamoDB Transactions

- Amazon DynamoDB transactions simplify the developer experience of making coordinated, all-or-nothing changes to multiple items both within and across tables.
- Transactions provide atomicity, consistency, isolation, and durability (ACID) in DynamoDB, helping you to maintain data correctness in your applications.
- You can group multiple Put, Update, Delete, and ConditionCheck actions. You can then submit the actions as a single TransactWriteItems operation that either succeeds or fails as a unit.
- You can group and submit multiple Get actions as a single TransactGetItems operation.
- Amazon DynamoDB supports up to 25 unique items and 4 MB of data per transactional request.

## Global Tables

- Global tables provide a solution for deploying a multi-region, multi-master database, without having to build and maintain your own replication solution.
- You specify the AWS regions where you want the table to be available. DynamoDB performs all tasks to create identical tables in these regions, and propagate ongoing data changes to all of them.



Global Tables enable you to use DynamoDB as a fully-managed, multi-region, multi-master database. [Learn more](#)

IAM role: **AWSServiceRoleForDynamoDBReplication** ⓘ  
Automatically created on your behalf.

Global Table regions

Global table version: 2019.11.21

Create a replica table in another region. [Learn more](#)

Region Name	Status	Read capacity units	Write capacity units	Auto Scaling	Endpoint
Asia Pacific (Tokyo)	Active	5	5	READ_AND_WRITE	dynamodb.ap-northeast-1.amazonaws.com
US East (N. Virginia)	Active	5	5	READ_AND_WRITE	dynamodb.us-east-1.amazonaws.com

- Replica Table (Replica, for short)
  - A single DynamoDB table that functions as a part of a global table.
  - Each replica stores the same set of data items.
  - Any given global table can only have one replica table per region.
  - You can add new or delete replicas from global tables.
- To ensure eventual consistency, DynamoDB global tables use a “last writer wins” reconciliation between concurrent updates, where DynamoDB makes a best effort to determine the last writer.
- If a single AWS region becomes isolated or degraded, your application can redirect to a different region and perform reads and writes against a different replica table. DynamoDB also keeps track of any writes that have been performed, but have not yet been propagated to all of the replica tables.
- Requirements for adding a new replica table
  - The table must have the same partition key as all of the other replicas.
  - The table must have the same write capacity management settings specified.
  - The table must have the same name as all of the other replicas.
  - The table must have DynamoDB Streams enabled, with the stream containing both the new and the old images of the item.
  - None of the replica tables in the global table can contain any data.
- If global secondary indexes are specified, then the following conditions must also be met:
  - The global secondary indexes must have the same name.



- The global secondary indexes must have the same partition key and sort key (if present).

## Security

- Encryption
  - Encrypts your data at rest using an AWS Key Management Service (AWS KMS) managed encryption key for DynamoDB.
  - Encryption at rest can be enabled only when you are creating a new DynamoDB table.
  - After encryption at rest is enabled, it can't be disabled.
  - Uses AES-256 encryption.
  - The following are encrypted:
    - DynamoDB base tables
    - Local secondary indexes
    - Global secondary indexes
  - Authentication and Access Control
    - Access to DynamoDB requires credentials.
    - Aside from valid credentials, you also need to have permissions to create or access DynamoDB resources.
    - Types of Identities
      - **AWS account root user**
      - **IAM user**
      - **IAM role**
  - You can create indexes and streams only in the context of an existing DynamoDB table, referred to as *subresources*.
  - Resources and subresources have unique Amazon Resource Names (**ARNs**) associated with them.
  - A *permissions policy* describes who has access to what.
    - Identity-based Policies
      - Attach a permissions policy to a user or a group in your account
      - Attach a permissions policy to a role (grant cross-account permissions)
    - Policy Elements
      - Resource - use an ARN to identify the resource that the policy applies to.
      - Action - use action keywords to identify resource operations that you want to allow or deny.
      - Effect - specify the effect, either allow or deny, when the user requests the specific action.
      - Principal - the user that the policy is attached to is the implicit principal.
  - Web Identity Federation - Customers can sign in to an identity provider and then obtain temporary security credentials from AWS Security Token Service (AWS STS).

## Monitoring



- Automated tools:
  - **Amazon CloudWatch Alarms** – Watch a single metric over a time period that you specify, and perform one or more actions based on the value of the metric relative to a given threshold over a number of time periods.
  - **Amazon CloudWatch Logs** – Monitor, store, and access your log files from AWS CloudTrail or other sources.
  - **Amazon CloudWatch Events** – Match events and route them to one or more target functions or streams to make changes, capture state information, and take corrective action.
  - **AWS CloudTrail Log Monitoring** – Share log files between accounts, monitor CloudTrail log files in real time by sending them to CloudWatch Logs, write log processing applications in Java, and validate that your log files have not changed after delivery by CloudTrail.
- Using the information collected by CloudTrail, you can determine the request that was made to DynamoDB, the IP address from which the request was made, who made the request, when it was made, and additional details.

**DynamoDB Accelerator (DAX)** delivers microsecond response times for accessing eventually consistent data.

### Best Practices

- Know the Differences Between Relational Data Design and NoSQL

Relational database systems (RDBMS)	NoSQL database
In RDBMS, data can be queried flexibly, but queries are relatively expensive and don't scale well in high-traffic situations.	In a NoSQL database such as DynamoDB, data can be queried efficiently in a limited number of ways, outside of which queries can be expensive and slow.
In RDBMS, you design for flexibility without worrying about implementation details or performance. Query optimization generally doesn't affect schema design, but normalization is very important.	In DynamoDB, you design your schema specifically to make the most common and important queries as fast and as inexpensive as possible. Your data structures are tailored to the specific requirements of your business use cases.
For an RDBMS, you can go ahead and create a normalized data model without thinking about access patterns. You can then extend it later when new questions and query requirements arise. You can organize each type of data into its own table.	For DynamoDB, by contrast, you shouldn't start designing your schema until you know the questions it will need to answer. Understanding the business problems and the application use cases up front is essential.  You should maintain as few tables as possible in a DynamoDB application. Most well designed applications require <b>only one</b> table.



It is important to understand three fundamental properties of your application's access patterns:

1. Data size: Knowing how much data will be stored and requested at one time will help determine the most effective way to partition the data.
2. Data shape: Instead of reshaping data when a query is processed, a NoSQL database organizes data so that its shape in the database corresponds with what will be queried.
3. Data velocity: DynamoDB scales by increasing the number of physical partitions that are available to process queries, and by efficiently distributing data across those partitions. Knowing in advance what the peak query loads might be helps determine how to partition data to best use I/O capacity.

- Design and Use Partition Keys Effectively

- DynamoDB provides some flexibility in your per-partition throughput provisioning by providing **burst capacity**.
- To better accommodate uneven access patterns, **DynamoDB adaptive capacity** enables your application to continue reading and writing to 'hot' partitions without being throttled, by automatically increasing throughput capacity for partitions that receive more traffic.
- Amazon DynamoDB now applies adaptive capacity in real time in response to changing application traffic patterns, which helps you maintain uninterrupted performance indefinitely, even for imbalanced workloads. In addition, instant adaptive capacity helps you provision read and write throughput more efficiently instead of overprovisioning to accommodate uneven data access patterns. Instant adaptive capacity is on by default at no additional cost for all DynamoDB tables and global secondary indexes.
- The optimal usage of a table's provisioned throughput depends not only on the workload patterns of individual items, but also on the partition-key design. In general, you will use your provisioned throughput more efficiently as the ratio of partition key values accessed to the total number of partition key values increases.
- Structure the primary key elements to avoid one heavily requested partition key value that slows overall performance.
- Distribute loads more evenly across a partition key space by adding a random number to the end of the partition key values. Then you randomize the writes across the larger space.
- A randomizing strategy can greatly improve write throughput, but it's difficult to read a specific item because you don't know which suffix value was used when writing the item. Instead of



using a random number to distribute the items among partitions, use a number that you can calculate based upon something that you want to query on.

- Distribute write activity efficiently during data upload by using the sort key to load items from each partition key value, keeping more DynamoDB servers busy simultaneously and improving your throughput performance.
- Use Sort Keys to Organize Data
  - Well-designed sort keys gather related information together in one place where it can be queried efficiently.
  - Composite sort keys let you define hierarchical (one-to-many) relationships in your data that you can query at any level of the hierarchy.
- Use indexes efficiently by keeping the number of indexes to a minimum and avoid indexing tables that experience heavy write activity.
- Choose Projections Carefully.
- Optimize Frequent Queries to Avoid Fetches.
- Be Aware of Item-Collection Size Limits When Creating Local Secondary Indexes.
- For Querying and Scanning Data
  - Performance considerations for scans
  - Avoiding sudden spikes in read activity
  - Taking advantage of parallel scans

## Pricing

- DynamoDB charges per GB of disk space that your table consumes. The first 25 GB consumed per month is free.
- DynamoDB charges for Provisioned Throughput --- WCU and RCU, Reserved Capacity and Data Transfer Out.
- You should round up to the nearest KB when estimating how many capacity units to provision.
- There are additional charges for DAX, Global Tables, On-demand Backups (per GB), Continuous backups and point-in-time recovery (per GB), Table Restorations (per GB), and Streams (read request units).

## Sources:

<https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/Introduction.html?shortFooter=true>  
<https://aws.amazon.com/dynamodb/faqs/>



## Amazon ElastiCache

- ElastiCache is a distributed **in-memory cache** environments in the AWS Cloud.
- ElastiCache works with both the **Redis** and **Memcached** engines.
- Components
  - ElastiCache Nodes
    - A **node** is a fixed-size chunk of secure, network-attached RAM. A node can exist in isolation from or in some relationship to other nodes.
    - Every node within a cluster is the same instance type and runs the same cache engine. Each cache node has its own Domain Name Service (DNS) name and port.
- If a maintenance event is scheduled for a given week, it will be initiated and completed at some point during the 60 minute maintenance window you specify.
- ElastiCache can be used for storing session state.
- ElastiCache Redis
  - Existing applications that use Redis can use ElastiCache with almost no modification.
  - Features
    - Automatic detection and recovery from cache node failures.
    - Multi-AZ with automatic failover of a failed primary cluster to a read replica in Redis clusters that support replication.
    - Redis (cluster mode enabled) supports partitioning your data across up to 250 shards.
    - Redis supports in-transit and at-rest encryption with authentication so you can build HIPAA-compliant applications.
    - Flexible Availability Zone placement of nodes and clusters for increased fault tolerance.
    - Data is persistent.
    - Can be used as a datastore.
    - Not multi-threaded.
    - Amazon ElastiCache for Redis supports self-service updates, which allows you to apply service updates at the time of your choosing and track the progress in real-time.
  - Cache data if:
    - It is slow or expensive to acquire when compared to cache retrieval.
    - It is accessed with sufficient frequency.
    - It is relatively static, or if rapidly changing, staleness is not a significant issue.
  - **Redis sorted sets** guarantee both uniqueness and element ordering. Each time a new element is added to the sorted set it's reranked in real time. It's then added to the set in its appropriate numeric position.
  - In the **Redis publish/subscribe** paradigm, you send a message to a specific channel not knowing who, if anyone, receives it. Recipients of the message are those who are subscribed to the channel.
  - **Redis hashes** are hashes that map string names to string values.
  - Components



- **Redis Shard** - a grouping of one to six related nodes. A Redis (cluster mode disabled) cluster always has one shard. A Redis (cluster mode enabled) cluster can have 1–90 shards.
  - A *multiple node shard* implements replication by having one read/write primary node and 1–5 replica nodes.
  - If there is more than one node in a shard, the shard supports replication with one node being the read/write primary node and the others read-only replica nodes.
- **Redis Cluster** - a logical grouping of one or more ElastiCache for Redis Shards. Data is partitioned across the shards in a Redis (cluster mode enabled) cluster.
- For improved fault tolerance, have at least two nodes in a Redis cluster and enabling **Multi-AZ with automatic failover**.
- Replica nodes use asynchronous replication mechanisms to keep synchronized with the primary node.
- If any primary has no replicas and the primary fails, you lose all that primary's data.
- You can use backup and restore to **migrate** to Redis (cluster mode enabled) and resize your Redis (cluster mode enabled).
- Redis (cluster mode disabled) vs Redis (cluster mode enabled)

	Redis (cluster mode disabled)	Redis (cluster mode enabled)
Shards (node groups)	1	1–90
Replicas for each shard (node group)	0–5	0–5
Data partitioning	No	Yes
Add/Delete replicas	Yes	Yes
Add/Delete node groups	No	No
Supports scale up	Yes	No
Supports engine upgrades	Yes	Yes
Promote replica to primary	Yes	No
Multi-AZ with automatic failover	Yes, with at least 1 replica. Optional. On by default.	Required
Backup/Restore	Yes	Yes

Tutorials Dojo



- You can vertically scale up or scale down your sharded Redis Cluster on demand. Amazon ElastiCache resizes your cluster by changing the node type, while the cluster continues to stay online and serve incoming requests.
- You can set up automatic snapshots or initiate manual backups, and then seed new ElastiCache for Redis clusters. You can also export your snapshots to an S3 bucket of your choice for disaster recovery, analysis or cross-region backup and restore.
- Endpoints
  - **Single Node Redis (cluster mode disabled)** Endpoints - used to connect to the cluster for both reads and writes.
  - **Multi-Node Redis (cluster mode disabled)** Endpoints - use the primary endpoint for all writes to the cluster. The read endpoint points to your read replicas.
  - **Redis (cluster mode enabled)** Endpoints - has a single configuration endpoint. By connecting to the configuration endpoint, your application is able to discover the primary and read endpoints for each shard in the cluster.
- Parameter Groups
  - **Cache parameter group** is a named collection of engine-specific parameters that you can apply to a cluster.
  - Parameters are used to control memory usage, eviction policies, item sizes, and more.
- Redis Security
  - ElastiCache for Redis node access is restricted to applications running on whitelisted EC2 instances. You can control access of your cluster by using subnet groups or security groups. By default, network access to your clusters is turned off.
  - By default, all new ElastiCache for Redis clusters are launched in a VPC environment. Use subnet groups to grant cluster access from Amazon EC2 instances running on specific subnets.
  - ElastiCache for Redis supports TLS and in-place encryption for nodes running specified versions of the ElastiCache for Redis engine.
  - You can use your own customer managed customer master keys (CMKs) in AWS Key Management Service to encrypt data at rest in ElastiCache for Redis.
- Redis Backups
  - A point-in-time copy of a Redis cluster.
  - Backups consist of all the data in a cluster plus some metadata.
- ElastiCache Memcached
  - Features
    - Automatic detection and recovery from cache node failures.
    - Automatic discovery of nodes within a cluster enabled for automatic discovery, so that no changes need to be made to your application when you add or remove nodes.
    - Flexible Availability Zone placement of nodes and clusters.
    - **ElastiCache Auto Discovery** feature for Memcached lets your applications identify all of the nodes in a cache cluster and connect to them.



- ElastiCache node access is restricted to applications running on whitelisted EC2 instances. You can control the instances that can access your cluster by using subnet groups or security groups.
- It is not persistent.
- Supports large nodes with multiple cores or threads.
- Does not support multi-AZ failover or replication
- Does not support snapshots
- Components
  - **Memcached cluster** - a logical grouping of one or more ElastiCache Nodes. Data is partitioned across the nodes in a Memcached cluster.
    - Memcached supports up to 100 nodes per customer for each Region with each cluster having 1–20 nodes.
    - When you partition your data, use *consistent hashing*.
  - **Endpoint** - the unique address your application uses to connect to an ElastiCache node or cluster.
    - Each node in a Memcached cluster has its own endpoint.
    - The cluster also has an endpoint called the *configuration endpoint*.
  - **ElastiCache parameter group** - a named collection of engine-specific parameters that you can apply to a cluster. Parameters are used to control memory usage, eviction policies, item sizes, and more.
  - ElastiCache allows you to control access to your clusters using **security groups**. By default, network access to your clusters is turned off.
  - A **subnet group** is a collection of subnets that you can designate for your clusters running in a VPC environment. If you create a cluster in a VPC, then you must specify a *cache subnet group*. ElastiCache uses that cache subnet group to choose a subnet and IP addresses within that subnet to associate with your cache nodes.
- Mitigating Failures
  - Node Failures
    - Spread your cached data over more nodes. Because Memcached does not support replication, a node failure will always result in some data loss from your cluster.
  - Availability Zone Failure
    - Locate your nodes in as many Availability Zones as possible. In the unlikely event of an AZ failure, you will lose the data cached in that AZ, not the data cached in the other AZs.
- ElastiCache uses DNS entries to allow client applications to locate servers (nodes). The DNS name for a node remains constant, but the IP address of a node can change over time.
- Caching Strategies



- **Lazy Loading** - a caching strategy that loads data into the cache only when necessary.
  - Only requested data is cached.
  - Node failures are not fatal.
  - There is a cache miss penalty.
  - Stale data.
- **Write Through** - adds data or updates data in the cache whenever data is written to the database.
  - Data in the cache is never stale.
  - Write penalty vs. Read penalty. Every write involves two trips: A write to the cache and a write to the database.
  - Missing data.
  - Cache churn.
- By adding a time to live (TTL) value to each write, we are able to enjoy the advantages of each strategy and largely avoid cluttering up the cache with superfluous data.
- Monitoring
  - The service continuously monitors the health of your instances. In case a node experiences failure or a prolonged degradation in performance, ElastiCache will automatically restart the node and associated processes.
  - ElastiCache provides both host-level metrics and metrics that are specific to the cache engine software. These metrics are measured and published for each Cache node in 60-second intervals.
  - Monitor events with **ElastiCache Events**. When significant events happen on a cache cluster, including failure to add a node, success in adding a node, the modification of a security group, and others, ElastiCache sends a notification to a specific SNS topic.
  - Monitor costs with tags.
- Redis VS Memcached
  - Memcached is designed for **simplicity** while Redis offers a **rich set of features** that make it effective for a wide range of use cases.



	Memcached	Redis (cluster mode disabled)	Redis (cluster mode enabled)
Data types	string, objects	2.8.x - string, sets, sorted sets, lists, hashes, bitmaps, hyperloglog	3.2.x and later - string, sets, sorted sets, lists, hashes, bitmaps, hyperloglog, geospatial indexes
		string, sets, sorted sets, lists, hashes, bitmaps, hyperloglog, geospatial indexes	
Data partitioning	Yes	No	Yes
Cluster is modifiable	Yes	Yes	3.2.10 and later - Limited
Online resharding	No	No	3.2.10 and later
Encryption	No	3.2.6, 4.0.10 and later	3.2.6, 4.0.10 and later
Multi-threaded	Yes	No	No
Node type upgrade	No	Yes	No
Engine upgrading	Yes	Yes	Yes
High availability (replication)	No	Yes	Yes
Automatic failover	No	Optional	Required
Pub/Sub capabilities	No	Yes	Yes
Sorted sets	No	Yes	Yes
Backup and restore	No	Yes	Yes
Geospatial indexing	No	2.8.x - No	Yes
		3.2.x and later - Yes	



- Pricing

- With on-demand nodes you pay only for the resources you consume by the hour without any long-term commitments.
- With Reserved Nodes, you can make a low, one-time, up-front payment for each node you wish to reserve for a 1 or 3 year term. In return, you receive a significant discount off the ongoing hourly usage rate for the Node(s) you reserve.
- ElastiCache provides storage space for one snapshot free of charge for each active ElastiCache for Redis cluster. Additional backup storage is charged.
- EC2 Regional Data Transfer charges apply when transferring data between an EC2 instance and an ElastiCache Node in different Availability Zones of the same Region.

**Sources:**

<https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/>

<https://aws.amazon.com/elasticache/redis-details/>

<https://docs.aws.amazon.com/AmazonElastiCache/latest/mem-ug/>

<https://aws.amazon.com/elasticache/redis-vs-memcached/>

<https://aws.amazon.com/elasticache/features/>

<https://aws.amazon.com/elasticache/pricing/>



## Amazon Relational Database Service (RDS)

- Industry-standard relational database
- RDS manages backups, software patching, automatic failure detection, and recovery.
- You can have automated backups performed when you need them, or manually create your own backup snapshot. You can use these backups to restore a database.
- Supports **Aurora, MySQL, MariaDB, PostgreSQL, Oracle, Microsoft SQL Server**.
- You can get high availability with a primary instance and a synchronous secondary instance that you can fail over to when problems occur. You can also use MySQL, MariaDB, or PostgreSQL Read Replicas to increase read scaling.
- Basic building block of RDS is the **DB instance**, which is an isolated database environment in the cloud.
- You can have up to 40 Amazon RDS DB instances.
- Each DB instance runs a **DB engine**.
- You can select the computation and memory capacity of a DB instance, determined by its **DB instance class**. If your needs change over time, you can change DB instances.
- Each DB instance has minimum and maximum storage requirements depending on the storage type and the database engine it supports.
- You can run your DB instance in several AZs, an option called a **Multi-AZ deployment**. Amazon automatically provisions and maintains a secondary standby DB instance in a different AZ. Your primary DB instance is synchronously replicated across AZs to the secondary instance to provide data redundancy, failover support, eliminate I/O freezes, and minimize latency spikes during system backups.

## DB Engines

- MySQL DB instance:
  - You're limited to 10,000 tables if you are either using Provisioned IOPS or General Purpose storage and the DB instance is 200 GiB or larger in size.
  - You're limited to 1000 tables if you are either using Standard or General Purpose storage and the DB instance is less than 200 GiB in size.
  - MySQL database instances can be launched with up to 64 TiB of storage and provisioned I/O performance of up to 80,000 IOPS.
  - The Point-In-Time Restore and snapshot restore features of Amazon RDS for MySQL require a crash-recoverable storage engine and are supported for the InnoDB storage engine only.
  - InnoDB is the recommended and supported storage engine for MySQL DB instances on Amazon RDS.
  - The **database name** is the name of a database hosted in your DB instance. Databases hosted by the same DB instance must have a unique name within that instance.
  - You can now enforce password policies in your Amazon RDS for MySQL databases using the MySQL *validate\_password* plugin. This improves the security of your databases by defining minimum password length, required characters, and other rules.



- MariaDB instance:
  - The point-in-time restore and snapshot restore features of Amazon RDS for MariaDB require a crash-recoverable storage engine.
  - MariaDB database instances can be launched with up to 64 TiB of storage and provisioned I/O performance of up to 80,000 IOPS.
  - The **database name** is the name of a database hosted in your DB instance. Databases hosted by the same DB instance must have a unique name within that instance.
- PostgreSQL instance:
  - You can improve performance with PostgreSQL on Amazon RDS when loading data into a DB instance and when using the PostgreSQL autovacuum feature.
  - PostgreSQL database instances can be launched with up to 64 TiB of storage and provisioned I/O performance of up to 80,000 IOPS.
  - The **database name** is the unique name of a database hosted in your DB instance, and is not required when creating a DB instance.
- Microsoft SQL Server instance:
  - Up to 10 can be SQL Server DB instances under the "License Included" model. You can have 40 DB instances for SQL Server under the "BYOL" licensing model.
  - The maximum number of databases supported on a DB instance depends on the instance class type and the availability mode—Single-AZ, Multi-AZ Database Mirroring (DBM), or Multi-AZ Availability Groups (AGs). The Microsoft SQL Server system databases don't count toward this limit.

Instance Class Type	Single-AZ	Multi-AZ with DBM	Multi-AZ with Always On AGs
db.*.micro to db.*.medium	30	N/A	N/A
db.*.large	30	30	30
db.*.xlarge to db.*.16xlarge	100	50	75
db.*.24xlarge	100	50	100

- For Multi-AZ enabled instances:
  - Use Amazon RDS DB events to monitor failovers.
  - If your application caches DNS values, set time to live (TTL) to less than 30 seconds.
  - AWS recommends to NOT enable Simple Recover, Offline, or Read-only modes because they turn off transaction logging, which is required for Multi-AZ.
  - Test to determine how long it takes for your DB instance to failover.
  - Deploy your applications in all Availability Zones.
- **Database name** is not a supported parameter.



- Oracle instance:
  - Up to 10 can be Oracle instances under the "License Included" model. You can have 40 DB instances for Oracle under the "BYOL" licensing model.
  - **Database name** is used to set the value of ORACLE\_SID, which must be supplied when connecting to the Oracle RDS instance.
  - You can now create Amazon RDS for Oracle database instances with up to 64 TiB of storage (32 TiB originally) and provisioned I/O performance of up to 80,000 IOPS.
- DB Instance:
  - Class Types
    - Standard
    - Memory Optimized
    - Burstable Performance
  - You can change the CPU and memory available to a DB instance by changing its DB instance class. Specify the following processor features to optimize your DB instance for specific workloads or business needs:
    - Number of CPU cores
    - Threads per core
  - Endpoint: rds.<region>.amazonaws.com
  - Storage
    - Amazon RDS for MySQL, MariaDB, PostgreSQL, Oracle, and Microsoft SQL Server use Amazon EBS volumes for database and log storage.
    - Storage types :
      - General Purpose SSD (gp2)
        - MySQL, MariaDB, Oracle, and PostgreSQL DB instances: 20 GiB–64 TiB storage size
        - SQL Server for Enterprise, Standard, Web, and Express editions: 20 GiB–16 TiB storage size
      - Provisioned IOPS SSD (io1)

Database Engine	Range of Provisioned IOPS	Range of Storage
MariaDB	1,000–80,000	100 GiB–64 TiB
SQL Server, Enterprise and Standard editions	1000–32,000 or 64,000 for Nitro-based m5 instance types	20 GiB–16 TiB
SQL Server, Web and Express editions	1000–32,000 or 64,000 for Nitro-based m5 instance types	100 GiB–16 TiB
MySQL	1,000–80,000	100 GiB–64 TiB



Oracle	1,000–80,000	100 GiB–64 TiB
PostgreSQL	1,000–80,000	100 GiB–64 TiB

- For production OLTP use cases, use **Multi-AZ deployments** for enhanced fault tolerance with Provisioned IOPS storage for fast and predictable performance.
  - You can use PIOPS storage with Read Replicas for MySQL, MariaDB or PostgreSQL.
- Magnetic
  - Doesn't allow you to scale storage when using the SQL Server database engine.
  - Doesn't support elastic volumes.
  - Limited to a maximum size of 3 TiB.
  - Limited to a maximum of 1,000 IOPS.
- Instance Lifecycle - includes creating, modifying, maintaining and upgrading, performing backups and restores, rebooting, and deleting the instance.
  - You can't stop an **Amazon RDS for SQL Server DB** instance in a Multi-AZ configuration.
  - You can stop a DB instance for up to seven days. If you do not manually start your DB instance after seven days, your DB instance is automatically started.
  - You can't stop a DB instance that has a Read Replica, or that is a Read Replica.
  - You can't modify a stopped DB instance.
  - To delete a DB instance, you must specify the name of the instance and whether to take a final DB snapshot of the instance.
  - You can enable **deletion protection** so that users can't delete a database. Deletion protection is disabled by default.

## RDS Storage Auto Scaling

- RDS Storage Auto Scaling automatically scales storage capacity in response to growing database workloads, with zero downtime.
- Amazon RDS for MariaDB, Amazon RDS for MySQL, Amazon RDS for PostgreSQL, Amazon RDS for SQL Server and Amazon RDS for Oracle support RDS Storage Auto Scaling.
- RDS Storage Auto Scaling continuously monitors actual storage consumption, and scales capacity up automatically when actual utilization approaches provisioned storage capacity.
- Auto Scaling works with new and existing database instances.

## Security

- Security Groups



- **DB Security Groups** - controls access to a DB instance that is not in a VPC. By default, network access is turned off to a DB instance. This SG is for the EC2-Classic platform.
- **VPC Security Groups** - controls access to a DB instance inside a VPC. This SG is for the EC2-VPC platform.
- **EC2 Security Groups** - controls access to an EC2 instance and can be used with a DB instance.
- Practices
  - Assign an individual **IAM** account to each person who manages RDS resources. Do not use AWS root credentials to manage RDS resources.
  - Grant each user the minimum set of permissions required to perform his or her duties.
  - Use IAM groups to effectively manage permissions for multiple users.
  - Rotate your IAM credentials regularly.
  - Use **security groups** to control what IP addresses or Amazon EC2 instances can connect to your databases on a DB instance.
  - Run your DB instance in an Amazon Virtual Private Cloud (**VPC**) for the greatest possible network access control.
  - Use **Secure Socket Layer (SSL) connections** with DB instances running the MySQL, MariaDB, PostgreSQL, Oracle, or Microsoft SQL Server database engines.
  - Use RDS encryption to secure your RDS instances and snapshots at rest.
  - Use the security features of your DB engine to control who can log in to the databases on a DB instance.
- A *resource owner* is the AWS account that created a resource. That is, the resource owner is the AWS account of the *principal entity* (the root account, an IAM user, or an IAM role) that authenticates the request that creates the resource.
- A *permissions policy* describes who has access to what. Policies attached to an IAM identity are *identity-based policies* (IAM policies) and policies attached to a resource are *resource-based policies*. Amazon RDS supports only identity-based policies (IAM policies).
- MySQL and PostgreSQL both support **IAM database authentication**.
- Encryption
  - At rest and in-transit.
  - Manage keys used for encrypted DB instances using the AWS KMS. KMS encryption keys are specific to the region that they are created in.
  - RDS encryption is currently available for all database engines and storage types. RDS encryption is available for most DB instance classes.
  - You can't have an encrypted Read Replica of an unencrypted DB instance or an unencrypted Read Replica of an encrypted DB instance.
  - You can't restore an unencrypted backup or snapshot to an encrypted DB instance.
  - You can use **SSL** from your application to encrypt a connection to a DB instance running MySQL, MariaDB, SQL Server, Oracle, or PostgreSQL.
- Amazon RDS supports the following scenarios for accessing a DB instance in a VPC:



DB Instance	Accessed By
In a VPC	An EC2 Instance in the Same VPC
	An EC2 Instance in a Different VPC
	An EC2 Instance Not in a VPC
	A Client Application Through the Internet
Not in a VPC	An EC2 Instance in a VPC
	An EC2 Instance Not in a VPC
	A Client Application Through the Internet

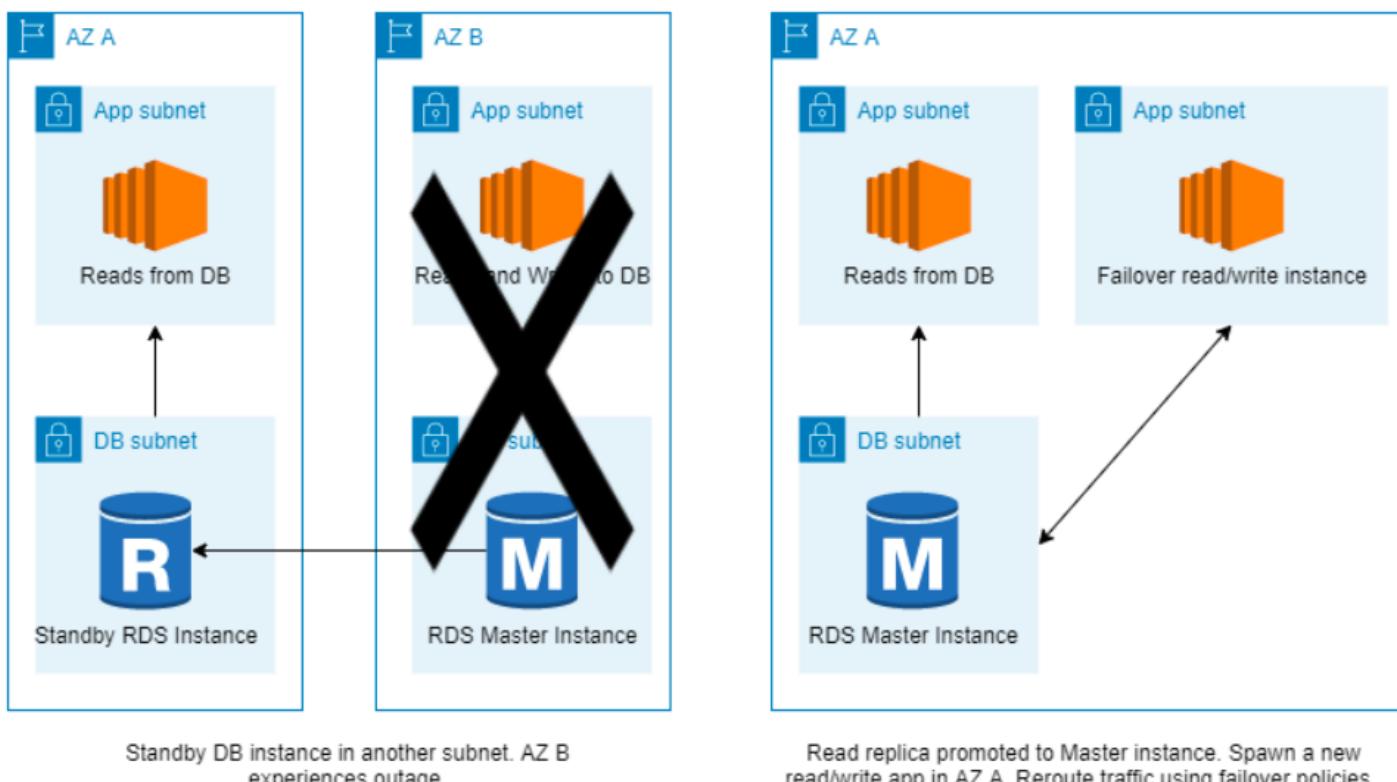
- Working with a DB Instance in a VPC
  - Your VPC must have **at least two subnets**. These subnets must be in two **different Availability Zones** in the region where you want to deploy your DB instance.
  - If you want your DB instance in the VPC to be publicly accessible, you must enable the VPC attributes *DNS hostnames* and *DNS resolution*.
  - Your VPC must have a DB subnet group that you create.
  - Your VPC must have a VPC security group that allows access to the DB instance.
  - The CIDR blocks in each of your subnets must be large enough to accommodate spare IP addresses for Amazon RDS to use during maintenance activities, including failover and compute scaling.
  - When an option group is assigned to a DB instance, it is linked to the supported platform the DB instance is on, either VPC or EC2-Classic.
  - If you restore a DB instance into a different VPC or onto a different platform, you must either assign the default option group to the DB instance, assign an option group that is linked to that VPC or platform, or create a new option group and assign it to the DB instance.

## Tagging

- An RDS tag is a **name-value pair** that you define and associate with an RDS resource. The name is referred to as the key. Supplying a value for the key is optional.
- All Amazon RDS resources can be tagged.
- Use tags to organize your AWS bill to reflect your own cost structure.
- A tag set can contain as many as 10 tags, or it can be empty.

## High Availability using Multi-AZ

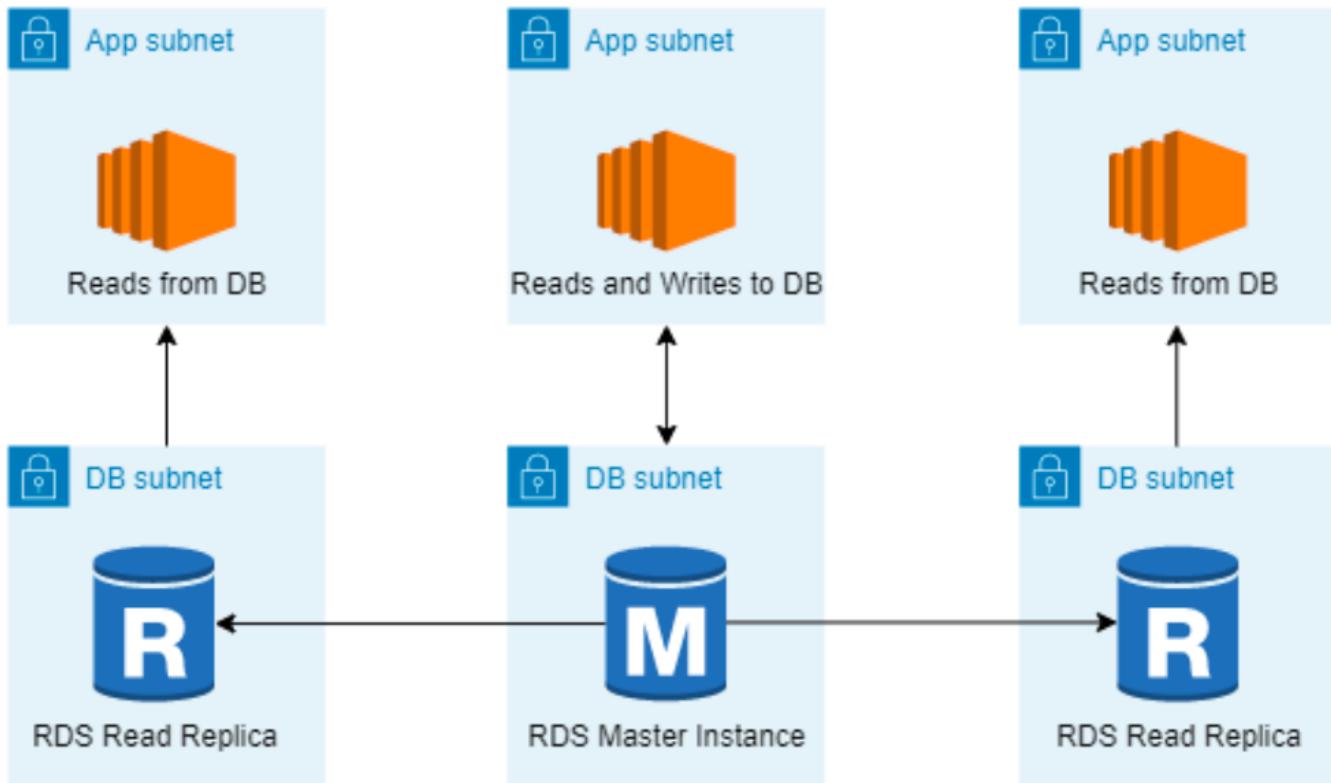
- Multi-AZ deployments for **Oracle, PostgreSQL, MySQL, and MariaDB** DB instances use **Amazon's failover technology**. **SQL Server** DB instances use **SQL Server Mirroring**.
- **Amazon RDS for SQL Server** offers **Always On Availability Groups** for the Multi-AZ configuration in all AWS Regions
- You can modify a DB instance in a Single-AZ deployment to a Multi-AZ deployment.
- The primary DB instance switches over automatically to the standby replica if any of the following conditions occur:
  - An Availability Zone outage
  - The primary DB instance fails
  - The DB instance's server type is changed
  - The operating system of the DB instance is undergoing software patching
  - A manual failover of the DB instance was initiated using **Reboot with failover**



## Read Replicas

- Updates made to the source DB instance are asynchronously copied to the Read Replica.

- You can reduce the load on your source DB instance by routing read queries from your applications to the Read Replica.



- You can elastically scale out beyond the capacity constraints of a single DB instance for read-heavy database workloads.
- You can create a Read Replica that has a different storage type from the source DB instance.
- A source DB instance can have cross-region Read Replicas in multiple regions. Due to the limit on the number of access control list (ACL) entries for a VPC, cannot have more than five cross-region Read Replica instances.
- PostgreSQL does **physical replication**. MySQL and MariaDB do **logical replication**.
- You can create a manual snapshot of a PostgreSQL Read Replica, but you can't enable automatic backups. You can enable automatic backups on a MySQL or MariaDB Read Replica.
- When creating a Read Replica, there are a few things to consider.
  - Enable automatic backups on the source DB instance by setting the backup retention period to a value other than 0.
  - Automatic backups are supported only for Read Replicas running MySQL 5.6 and later.
- Oracle supports Read Replicas with Active Data Guard for customers who are using the **Bring Your Own License model with Oracle Database Enterprise Edition and have licensed the Active Data Guard Option**. This feature allows RDS Oracle customers to offload their read workload from the primary DB Instance, and also supports the scalability of read workloads over a farm of up to **five read replicas**.



- You can promote a Read Replica to become a standalone DB instance.

Multi-AZ Deployments	Read Replicas
Synchronous replication - highly durable	Asynchronous replication - highly scalable
Only database engine on primarily instance is active	All read replicas are accesible and can be used for read scaling
Automated backups are taken from standby	No backups configured by default
Always span two Availability Zones within a single Region	Can be within an Availability Zone, Cross-AZ, or Cross-Region
Database engine version upgrades happen on primary	Database engine version upgrade is independent from source instance
Automatic failover to standby when a problem is detected	Can be manually promoted to a standalone database instance



## Backups and Restores

- Your DB instance must be in the **ACTIVE state** for automated backups to occur. Automated backups and automated snapshots don't occur while a copy is executing in the same region for the same DB instance.
- The first snapshot of a DB instance contains the data for the full DB instance. Subsequent snapshots of the same DB instance are incremental.
- The default backup retention period is one day if you create the DB instance using the RDS API or the AWS CLI, or seven days if you used the AWS Console.
- Manual snapshot limits are limited to 100 per region.
- You can copy a snapshot within the same AWS Region, you can copy a snapshot across AWS Regions, and you can copy a snapshot across AWS accounts.
- When you restore a DB instance to a point in time, the default DB parameter and default DB security group is applied to the new DB instance.
- When you restore an Oracle DB instance to a point in time, you can specify a different Oracle DB engine, license model, and DBName to be used.
- When you restore a SQL Server DB instance to a point in time, each database within that instance is restored to a point in time within 1 second of each other database within the instance.
- You can retain Amazon RDS automated backups (system snapshots and transaction logs) when you delete a database instance.

## Monitoring



- Amazon CloudWatch
- RDS Events
  - An Amazon RDS event is created when the reboot is completed.
  - Be notified when changes occur with a DB instance, DB snapshot, DB parameter group, or DB security group.
  - Uses the Amazon Simple Notification Service (SNS) to provide notification when an Amazon RDS event occurs.
- Database log files
- Using enhanced monitoring to identify operating system issues for the following:
  - MariaDB
  - Microsoft SQL Server
  - MySQL version 5.5 or later
  - Oracle
  - PostgreSQL
- Enhanced monitoring is available for all DB instance classes except for db.m1.small. Enhanced Monitoring is available in all regions except for AWS GovCloud (US).
- Enhanced monitoring metrics include:
  - IOPS - the number of I/O operations completed each second.
  - Latency - the elapsed time between the submission of an I/O request and its completion.
  - Throughput - the number of bytes each second that are transferred to or from disk.
  - Queue Depth - the number of I/O requests in the queue waiting to be serviced.
- CloudWatch gathers metrics about CPU utilization **from the hypervisor** for a DB instance, and Enhanced Monitoring gathers its metrics **from an agent** on the instance.
- Instance Status - indicates the health of the instance. Here are some statuses:

DB Instance Status	Billed	Description
available	Billed	The instance is healthy and available.
backing-up	Billed	The instance is currently being backed up.
creating	Not billed	The instance is being created. The instance is inaccessible while it is being created.
deleting	Not billed	The instance is being deleted.
failed	Not billed	The instance has failed and Amazon RDS can't recover it. Perform a point-in-time restore to the latest restorable time of the instance to recover the data.



maintenance	Billed	Amazon RDS is applying a maintenance update to the DB instance. This status is used for instance-level maintenance that RDS schedules well in advance.
rebooting	Billed	The instance is being rebooted because of a customer request or an Amazon RDS process that requires the rebooting of the instance.
starting	Billed for storage	The DB instance is starting.
stopped	Billed for storage	The DB instance is stopped.
stopping	Billed for storage	The DB instance is being stopped.
storage-full	Billed	The instance has reached its storage capacity allocation. This is a critical status, and we recommend that you fix this issue immediately. To do so, scale up your storage by modifying the DB instance. To avoid this situation, set Amazon CloudWatch alarms to warn you when storage space is getting low.

- RDS **Performance Insights** monitors your DB instance load so that you can analyze and troubleshoot your database performance. You can visualize the database load and filter the load by waits, SQL statements, hosts, or users.
- CloudTrail captures all API calls for RDS as events.

## Pricing

- With Amazon RDS, you pay only for the RDS instances that are active.
- The data transferred for cross-region replication incurs RDS data transfer charges.
- Instances are billed for DB instance hours (per second), Storage (per GiB per month), I/O requests (per 1 million requests per month), Provisioned IOPS (per IOPS per month), Backup storage (per GiB per month), and Data transfer (per GB).
  - Amazon RDS is billed in one-second increments for database instances and attached storage. Pricing is still listed on a per-hour basis, but bills are now calculated down to the second and show usage in decimal form. There is a 10 minute minimum charge when an instance is created, restored or started.
- RDS purchasing options:
  - **On-Demand Instances** – Pay by the hour for the DB instance hours that you use.



- **Reserved Instances** – Reserve a DB instance for a one-year or three-year term and receive a significant discount compared to the on-demand DB instance pricing.
- You are charged for using Enhanced Monitoring.
- Amazon RDS is now billed in one-second increments for database instances and attached storage. Pricing is still listed on a per-hour basis, but bills are now calculated down to the second and show usage in decimal form. There is a 10 minute minimum charge when an instance is created, restored or started.

## Best Practices

- Monitor your memory, CPU, and storage usage.
- Scale up your DB instance when you are approaching storage capacity limits.
- Enable automatic backups and set the backup window to occur during the daily low in write IOPS.
- If your database workload requires more I/O than you have provisioned, recovery after a failover or database failure will be slow. Increase the I/O capacity of a DB instance by:
  - Migrate to a DB instance class with High I/O capacity.
  - Convert from standard storage to either General Purpose or Provisioned IOPS storage, depending on how much of an increase you need.
  - If you convert to Provisioned IOPS storage, make sure you also use a DB instance class that is optimized for Provisioned IOPS..
  - If you are already using Provisioned IOPS storage, provision additional throughput capacity.
- If your client application is caching the Domain Name Service (DNS) data of your DB instances, set a time-to-live (TTL) value of less than 30 seconds.
- Test failover for your DB instance.

## Sources:

<https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/>

<https://aws.amazon.com/rds/features/>

<https://aws.amazon.com/rds/pricing/>

<https://aws.amazon.com/rds/faqs/>



## Amazon Redshift

- A fully managed, **petabyte-scale data warehouse** service.
- Redshift extends data warehouse queries to your data lake. You can run analytic queries against petabytes of data stored locally in Redshift, and directly against exabytes of data stored in S3.
- RedShift is an OLAP type of DB.
- Currently, Redshift only supports Single-AZ deployments.
- Features
  - Redshift uses **columnar storage**, data compression, and zone maps to reduce the amount of I/O needed to perform queries.
  - It uses a **massively parallel processing** data warehouse architecture to parallelize and distribute SQL operations.
  - Redshift uses machine learning to deliver high throughput based on your workloads.
  - Redshift uses **result caching** to deliver sub-second response times for repeat queries.
  - Redshift automatically and continuously backs up your data to S3. It can asynchronously replicate your snapshots to S3 in another region for disaster recovery.
- Components
  - **Cluster** - a set of **nodes**, which consists of a leader node and one or more compute nodes.
    - Redshift creates one database when you provision a cluster. This is the database you use to load data and run queries on your data.
    - You can scale the cluster in or out by adding or removing nodes. Additionally, you can scale the cluster up or down by specifying a different node type.
    - Redshift assigns a 30-minute maintenance window at random from an 8-hour block of time per region, occurring on a random day of the week. During these maintenance windows, your cluster is not available for normal operations.
    - Redshift supports both the EC2-VPC and EC2-Classic platforms to launch a cluster. You create a **cluster subnet group** if you are provisioning your cluster in your VPC, which allows you to specify a set of subnets in your VPC.
  - **Redshift Nodes**
    - The *leader node* receives queries from client applications, parses the queries, and develops query execution plans. It then coordinates the parallel execution of these plans with the compute nodes and aggregates the intermediate results from these nodes. Finally, it returns the results back to the client applications.
    - *Compute nodes* execute the query execution plans and transmit data among themselves to serve these queries. The intermediate results are sent to the leader node for aggregation before being sent back to the client applications.
    - Node Type
      - **Dense storage (DS)** node type - for large data workloads and use hard disk drive (HDD) storage.



- **Dense compute (DC)** node types - optimized for performance-intensive workloads. Uses SSD storage.
- **Parameter Groups** - a group of parameters that apply to all of the databases that you create in the cluster. The default parameter group has preset values for each of its parameters, and it cannot be modified.
- Database Querying Options
  - Connect to your cluster and run queries on the AWS Management Console with the Query Editor.
  - Connect to your cluster through a SQL client tool using standard ODBC and JDBC connections.
- Enhanced VPC Routing
  - By using Enhanced VPC Routing, you can use VPC features to manage the flow of data between your cluster and other resources.
  - You can also use VPC flow logs to monitor *COPY* and *UNLOAD* traffic.
- RedShift Spectrum
  - Enables you to run queries against exabytes of data in S3 without having to load or transform any data.
  - Redshift Spectrum doesn't use Enhanced VPC Routing.
  - If you store data in a columnar format, Redshift Spectrum scans only the columns needed by your query, rather than processing entire rows.
  - If you compress your data using one of Redshift Spectrum's supported compression algorithms, less data is scanned.
- Cluster Snapshots
  - Point-in-time backups of a cluster. There are two types of snapshots: automated and manual. Snapshots are stored in S3 using SSL.
  - Redshift periodically takes incremental snapshots of your data every 8 hours or 5 GB per node of data change.
  - Redshift provides free storage for snapshots that is equal to the storage capacity of your cluster until you delete the cluster. After you reach the free snapshot storage limit, you are charged for any additional storage at the normal rate.
  - Automated snapshots are enabled by default when you create a cluster. These snapshots are deleted at the end of a retention period, which is one day, but you can modify it. You cannot delete an automated snapshot manually.
  - By default, manual snapshots are retained indefinitely, even after you delete your cluster.
  - You can share an existing manual snapshot with other AWS accounts by authorizing access to the snapshot.
- Monitoring
  - Use the *database audit logging* feature to track information about authentication attempts, connections, disconnections, changes to database user definitions, and queries run in the database. The logs are stored in S3 buckets.
  - Redshift tracks events and retains information about them for a period of several weeks in your AWS account.



- Redshift provides performance metrics and data so that you can track the health and performance of your clusters and databases. It uses CloudWatch metrics to monitor the physical aspects of the cluster, such as CPU utilization, latency, and throughput.
- *Query/Load performance data* helps you monitor database activity and performance.
- When you create a cluster, you can optionally configure a CloudWatch alarm to monitor the average percentage of disk space that is used across all of the nodes in your cluster, referred to as the *default disk space alarm*.
- Security
  - By default, an Amazon Redshift cluster is only accessible to the AWS account that creates the cluster.
  - Use IAM to create user accounts and manage permissions for those accounts to control cluster operations.
  - If you are using the EC2-Classic platform for your Redshift cluster, you must use Redshift security groups.
  - If you are using the EC2-VPC platform for your Redshift cluster, you must use VPC security groups.
  - When you provision the cluster, you can optionally choose to encrypt the cluster for additional security. Encryption is an immutable property of the cluster.
  - Snapshots created from the encrypted cluster are also encrypted.
- Pricing
  - You pay an hourly rate based on the type and number of nodes in your cluster.
  - You pay for the number of bytes scanned by RedShift Spectrum
  - You can reserve instances by committing to using Redshift for a 1 or 3 year term and save costs.

**Sources:**

<https://docs.aws.amazon.com/redshift/latest/mgmt/>

<https://aws.amazon.com/redshift/features/>

<https://aws.amazon.com/redshift/pricing/>

<https://aws.amazon.com/redshift/faqs/>



## NETWORKING AND CONTENT DELIVERY

### Amazon API Gateway

- Enables developers to create, publish, maintain, monitor, and secure APIs at any scale.
- This is a HIPAA eligible service.
- Allows creating, deploying, and managing a RESTful API to expose backend HTTP endpoints, Lambda functions, or other AWS services.

The screenshot shows the AWS Lambda console interface. On the left, there's a sidebar with options like APIs, Custom Domain Names, VPC Links, API: test-API, Resources, Stages, Authorizers, Gateway Responses, Models, Resource Policy, Documentation, and Dashboard. The 'Resources' section is currently selected. In the main area, the path 'APIs > test-API (nmp7hrmcl1) > Resources > /test (10ny9q) > GET' is displayed. A sub-menu on the right shows '/test' with 'ANY' and 'GET' options, where 'GET' is highlighted. The main panel is titled '/test - GET - Setup' and contains the following configuration:

- Integration type:** Radio buttons for Lambda Function (selected), HTTP, Mock, AWS Service, and VPC Link.
- Use Lambda Proxy integration:** A checkbox that is unchecked.
- Lambda Region:** A dropdown menu set to 'us-east-1'.
- Lambda Function:** An input field containing 'samplelambdafunction'.
- Use Default Timeout:** A checkbox that is checked.

A blue 'Save' button is located in the bottom right corner of the main panel.

- Together with Lambda, API Gateway forms the app-facing part of the AWS serverless infrastructure.
- Concepts
  - **API deployment** - a point-in-time snapshot of your API Gateway API resources and methods. To be available for clients to use, the deployment must be associated with one or more API stages.
  - **API endpoints** - host names APIs in API Gateway, which are deployed to a specific region and of the format: *rest-api-id.execute-api.region.amazonaws.com*
  - **API key** - An alphanumeric string that API Gateway uses to identify an app developer who uses your API.
  - **API stage** - A logical reference to a lifecycle state of your API. API stages are identified by API ID and stage name.
  - **Model** - Data schema specifying the data structure of a request or response payload.
  - **Private API** - An API that is exposed through interface VPC endpoints and isolated from the public internet
  - **Private integration** - An API Gateway integration type for a client to access resources inside a customer's VPC through a private API endpoint without exposing the resources to the public internet.



- **Proxy integration** - You can set up a proxy integration as an HTTP proxy integration type or a Lambda proxy integration type.
  - For the HTTP proxy integration, API Gateway passes the entire request and response between the frontend and an HTTP backend.
  - For the Lambda proxy integration, API Gateway sends the entire request as an input to a backend Lambda function.
- **Usage plan** - Provides selected API clients with access to one or more deployed APIs. You can use a usage plan to configure throttling and quota limits, which are enforced on individual client API keys.
- API Endpoint Types
  - **Edge-optimized API endpoint**: The default host name of an API Gateway API that is deployed to the specified region while using a CloudFront distribution to facilitate client access typically from across AWS regions. API requests are routed to the nearest CloudFront Point of Presence.
  - **Regional API endpoint**: The host name of an API that is deployed to the specified region and intended to serve clients, such as EC2 instances, in the same AWS region. API requests are targeted directly to the region-specific API Gateway without going through any CloudFront distribution.
    - You can apply latency-based routing on regional endpoints to deploy an API to multiple regions using the same regional API endpoint configuration, set the same custom domain name for each deployed API, and configure latency-based DNS records in Route 53 to route client requests to the region that has the lowest latency.
  - **Private API endpoint**: Allows a client to securely access private API resources inside a VPC. Private APIs are isolated from the public Internet, and they can only be accessed using VPC endpoints for API Gateway that have been granted access.
- Features
  - API Gateway can execute Lambda code in your account, start Step Functions state machines, or make calls to Elastic Beanstalk, EC2, or web services outside of AWS with publicly accessible HTTP endpoints.
  - API Gateway helps you define plans that meter and restrict third-party developer access to your APIs.
  - API Gateway helps you manage traffic to your backend systems by allowing you to set throttling rules based on the number of requests per second for each HTTP method in your APIs.
  - You can set up a cache with customizable keys and time-to-live in seconds for your API data to avoid hitting your backend services for each request.
  - API Gateway lets you run multiple versions of the same API simultaneously with **API Lifecycle**.
  - After you build, test, and deploy your APIs, you can package them in an API Gateway usage plan and sell the plan as a Software as a Service (SaaS) product through AWS Marketplace.
  - API Gateway offers the ability to create, update, and delete documentation associated with each portion of your API, such as methods and resources.
- All of the APIs created expose **HTTPS endpoints only**. API Gateway does not support unencrypted (HTTP) endpoints.



- Monitoring
  - API Gateway console is integrated with CloudWatch, so you get backend performance metrics such as API calls, latency, and error rates.
  - You can set up custom alarms on API Gateway APIs.
  - API Gateway can also log API execution errors to CloudWatch Logs.
- Security
  - To authorize and verify API requests to AWS services, API Gateway can help you leverage signature version 4. Using signature version 4 authentication, you can use IAM and access policies to authorize access to your APIs and all your other AWS resources.
  - You can enable AWS WAF for your APIs in Amazon API Gateway, making it easier to protect your APIs against common web exploits such as SQL injection and Cross-Site Scripting (XSS).
- Pricing
  - You pay only for the API calls you receive and the amount of data transferred out.
  - API Gateway also provides optional data caching charged at an hourly rate that varies based on the cache size you select.

**Sources:**

<https://docs.aws.amazon.com/apigateway/latest/developerguide/>

<https://aws.amazon.com/api-gateway/features/>

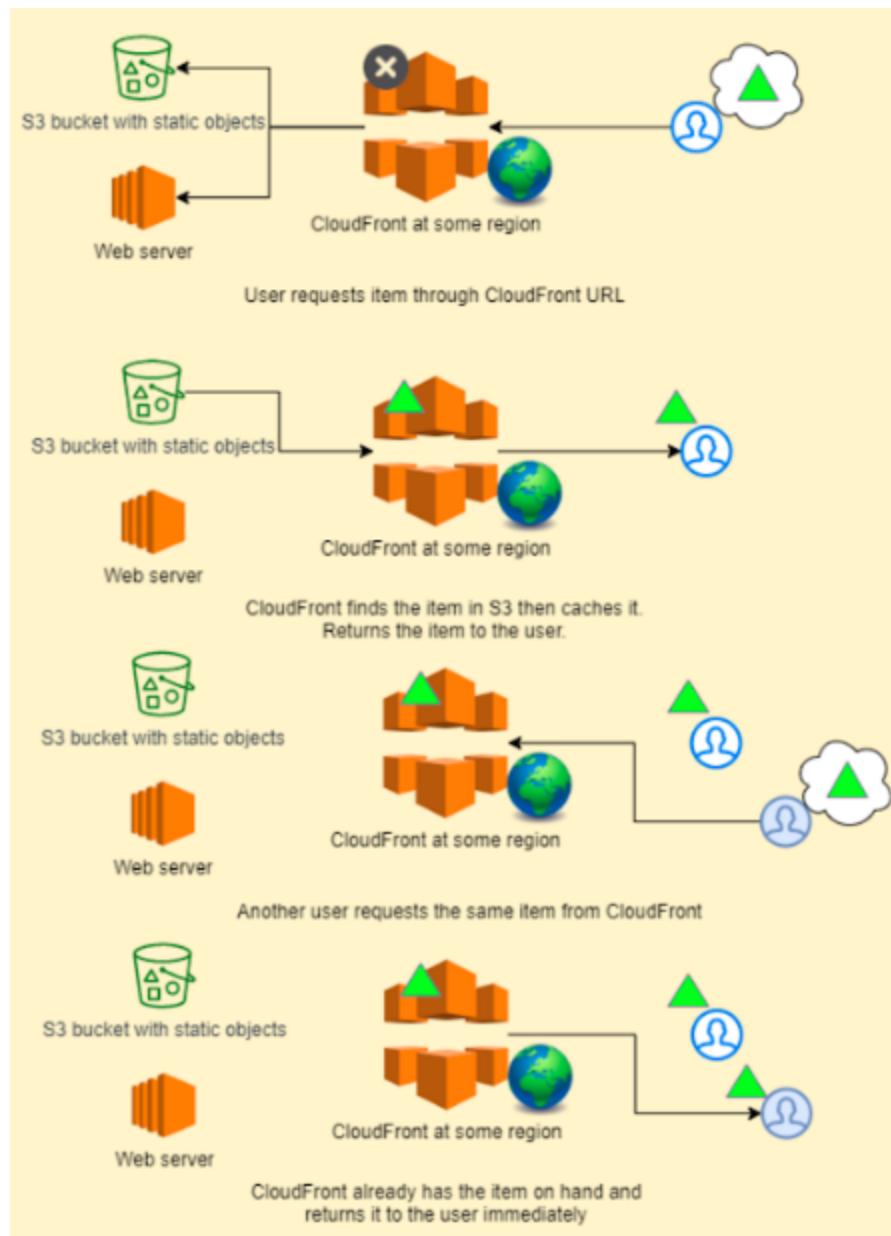
<https://aws.amazon.com/api-gateway/pricing/>

<https://aws.amazon.com/api-gateway/faqs/>



## Amazon CloudFront

- A web service that speeds up distribution of your static and dynamic web content to your users. A Content Delivery Network (CDN) service.
- It 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, 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
- How CloudFront Delivers Content
  - You specify **origin servers**, like an S3 bucket or your own HTTP server, from which CloudFront gets your files which will then be distributed from CloudFront edge locations all over the world.
  - Upload your files to your origin servers. Your files, also known as **objects**.
  - Create a **CloudFront distribution**, which tells CloudFront which origin servers to get your files from when users request the files through your web site or application. At the same time, you specify details such as whether you want CloudFront to log all requests and whether you want the distribution to be enabled as soon as it's created.
  - CloudFront assigns a domain name to your new distribution that you can see in the CloudFront console.
  - CloudFront sends your distribution's configuration (but not your content) to all of its **edge locations**—collections of servers in geographically dispersed data centers where CloudFront caches copies of your objects.



- CloudFront supports the **WebSocket protocol** as well as the **HTTP protocol** with the following HTTP methods:
  - GET
  - HEAD
  - POST
  - PUT
  - DELETE
  - OPTIONS
  - PATCH.



- Using **Lambda@Edge** with CloudFront enables a variety of ways to customize the content that CloudFront delivers. It can help you configure your CloudFront distribution to serve private content from your own custom origin, as an option to using signed URLs or signed cookies.(See AWS Compute Services Lambda Lambda@Edge)
- CloudFront also has **regional edge caches** that bring more of your content closer to your viewers, even when the content is not popular enough to stay at a CloudFront edge location, to help improve performance for that content.
- You can use a zone apex name on CloudFront
- CloudFront supports wildcard CNAME
- Different CloudFront Origins
  - **Using S3 buckets for your origin** - you place any objects that you want CloudFront to deliver in an S3 bucket.
  - **Using S3 buckets configured as website endpoints for your origin**
  - **Using an mediastore container or an mediapackage channel for your origin** - you can set up an S3 bucket that is configured as a MediaStore container, or create a channel and endpoints with MediaPackage. Then you create and configure a distribution in CloudFront to stream the video.
  - **Using EC2 or other custom origins** - A custom origin is an HTTP server, for example, a web server.
  - **Using CloudFront Origin Groups for origin failover** - use origin failover to designate a primary origin for CloudFront plus a second origin that CloudFront automatically switches to when the primary origin returns specific HTTP status code failure responses.
- Objects are cached for 24 hours by default. You can invalidate files in CloudFront edge caches even before they expire.
- You can configure CloudFront to automatically compress files of certain types and serve the compressed files when viewer requests include *Accept-Encoding: gzip* in the request header.
- CloudFront can cache different versions of your content based on the values of query string parameters.
- CloudFront Distributions
  - You create a **CloudFront distribution** to tell CloudFront where you want content to be delivered from, and the details about how to track and manage content delivery.
  - You create a distribution and choose the configuration settings you want:
    - Your content origin—that is, the Amazon S3 bucket, MediaPackage channel, or HTTP server from which CloudFront gets the files to distribute. You can specify any combination of up to 25 S3 buckets, channels, and/or HTTP servers as your origins.
    - Access—whether you want the files to be available to everyone or restrict access to some users.
    - Security—whether you want CloudFront to require users to use HTTPS to access your content.
    - Cookie or query-string forwarding—whether you want CloudFront to forward cookies or query strings to your origin.



- Geo-restrictions—whether you want CloudFront to prevent users in selected countries from accessing your content.
- Access logs—whether you want CloudFront to create access logs that show viewer activity.
- You can use distributions to serve the following content over HTTP or HTTPS:
  - Static and dynamic download content.
  - Video on demand in different formats, such as Apple HTTP Live Streaming (HLS) and Microsoft Smooth Streaming.
  - A live event, such as a meeting, conference, or concert, in real time.
- Values that you specify when you create or update a distribution
  - Delivery Method - Web or RTMP.
  - Origin Settings - information about one or more locations where you store the original versions of your web content.
  - Cache Behavior Settings - lets you configure a variety of CloudFront functionality for a given URL path pattern for files on your website.
  - Custom Error Pages and Error Caching
  - Restrictions - if you need to prevent users in selected countries from accessing your content, you can configure your CloudFront distribution either to allow users in a **whitelist** of specified countries to access your content or to not allow users in a **blacklist** of specified countries to access your content.
- Cache Behavior Settings
  - The functionality that you can configure for each cache behavior includes:
    - The path pattern.
    - If you have configured multiple origins for your CloudFront distribution, which origin you want CloudFront to forward your requests to.
    - Whether to forward query strings to your origin.
    - Whether accessing the specified files requires signed URLs.
    - Whether to require users to use HTTPS to access those files.
    - The minimum amount of time that those files stay in the CloudFront cache regardless of the value of any Cache-Control headers that your origin adds to the files.



## Default Cache Behavior Settings

Path Pattern	Default (*)	<a href="#">i</a>
Viewer Protocol Policy	<input checked="" type="radio"/> HTTP and HTTPS <input type="radio"/> Redirect HTTP to HTTPS <input type="radio"/> HTTPS Only	<a href="#">i</a>
Allowed HTTP Methods	<input checked="" type="radio"/> GET, HEAD <input type="radio"/> GET, HEAD, OPTIONS <input type="radio"/> GET, HEAD, OPTIONS, PUT, POST, PATCH, DELETE	<a href="#">i</a>
Field-level Encryption Config	<a href="#">▼</a>	<a href="#">i</a>
Cached HTTP Methods	GET, HEAD (Cached by default)	<a href="#">i</a>
Cache Based on Selected Request Headers	<a href="#">None (Improves Caching) ▾</a>	<a href="#">i</a>
<a href="#">Learn More</a>		
Object Caching	<input checked="" type="radio"/> Use Origin Cache Headers <input type="radio"/> Customize	<a href="#">i</a>
<a href="#">Learn More</a>		
Minimum TTL	0	<a href="#">i</a>
Maximum TTL	31536000	<a href="#">i</a>
Default TTL	86400	<a href="#">i</a>
Forward Cookies	<a href="#">None (Improves Caching) ▾</a>	<a href="#">i</a>
Query String Forwarding and Caching	<a href="#">None (Improves Caching) ▾</a>	<a href="#">i</a>
Smooth Streaming	<input checked="" type="radio"/> Yes <input type="radio"/> No	<a href="#">i</a>
Restrict Viewer Access (Use Signed URLs or Signed Cookies)	<input checked="" type="radio"/> Yes <input type="radio"/> No	<a href="#">i</a>
Compress Objects Automatically	<input checked="" type="radio"/> Yes <input type="radio"/> No	<a href="#">i</a>
<a href="#">Learn More</a>		

- After creating your CloudFront distribution, you can invalidate its cached items by creating an invalidation request.



The screenshot shows the AWS CloudFront Distributions page for distribution ID E1PEKC2PSRCHZG. The 'Invalidations' tab is selected. A modal dialog box titled 'Create Invalidation' is open, prompting for 'Object Paths'. The 'Distribution ID' field is populated with E1PEKC2PSRCHZG. Below it, examples of object paths are listed: 'images/image1.jpg', '/Images/Image1.jpg', '/Images/\*', and '/Images/image\*'. At the bottom right of the dialog is a blue 'Invalidate' button.

- Price Class
  - Choose the price class that corresponds with the maximum price that you want to pay for CloudFront service. By default, CloudFront serves your objects from edge locations in all CloudFront regions.
- Performance and Availability
  - CloudFront also allows you to set up multiple origins to enable redundancy with **Origin Failover**. To set up origin failover, you must have a distribution with at least two origins. Next, you create an origin group for your distribution that includes the two origins, setting one as the primary. Finally, you define a cache behavior in which you specify the origin group as your origin.
    - The two origins in the origin group can be any combination of the following: AWS origins, like Amazon S3 buckets or Amazon EC2 instances, or custom origins, like your own HTTP web server.
    - When you create the origin group, you configure CloudFront to failover to the second origin for GET, HEAD, and OPTIONS HTTP methods when the primary origin returns specific status codes that you configure.
  - CloudFront is optimized for both dynamic and static content, providing extensive flexibility for optimizing cache behavior, coupled with network-layer optimizations for latency and throughput.
- Using HTTPS with CloudFront



- You can choose HTTPS settings both for communication between viewers and CloudFront, and between CloudFront and your origin.
- If you want your viewers to use HTTPS and to use alternate domain names for your files, you need to choose one of the following options for how CloudFront serves HTTPS requests:
  - Use a dedicated IP address in each edge location
  - Use Server Name Indication (SNI)
- Monitoring
  - The billing report is a high-level view of all of the activity for the AWS services that you're using, including CloudFront.
  - The usage report is a summary of activity for a service such as CloudFront, aggregated by hour, day, or month. It also includes usage charts that provide a graphical representation of your CloudFront usage.
  - CloudFront console includes a variety of reports based on the data in CloudFront access logs:
    - CloudFront Cache Statistics Reports
    - CloudFront Popular Objects Report
    - CloudFront Top Referrers Report
    - CloudFront Usage Reports
    - CloudFront Viewers Reports
  - You can use AWS Config to record configuration changes for CloudFront distribution settings changes.
  - CloudFront integrates with Amazon CloudWatch metrics so that you can monitor your website or application.
  - Capture API requests with AWS CloudTrail. CloudFront is a global service. To view CloudFront requests in CloudTrail logs, you must update an existing trail to include global services.
- Security
  - CloudFront, AWS Shield, AWS WAF, and Route 53 work seamlessly together to create a flexible, layered security perimeter against multiple types of attacks including network and application layer DDoS attacks.
  - You can deliver your content, APIs or applications via SSL/TLS, and advanced SSL features are enabled automatically.
  - Through geo-restriction capability, you can prevent users in specific geographic locations from accessing content that you're distributing through CloudFront.
  - With **Origin Access Identity** feature, you can restrict access to an S3 bucket to only be accessible from CloudFront.
  - **Field-Level Encryption** is a feature of CloudFront that allows you to securely upload user-submitted data such as credit card numbers to your origin servers.
- Pricing
  - Charge for storage in an S3 bucket.
  - Charge for serving objects from edge locations.
  - Charge for submitting data to your origin.
    - Data Transfer Out



- HTTP/HTTPS Requests
- Invalidation Requests,
- Dedicated IP Custom SSL certificates associated with a CloudFront distribution.
- You also incur a surcharge for HTTPS requests, and an additional surcharge for requests that also have field-level encryption enabled.
- Compliance
  - CloudFront has been validated as being compliant with Payment Card Industry (PCI) Data Security Standard (DSS).
  - CloudFront is a HIPAA eligible service.
  - CloudFront is compliant with SOC measures.

**Sources:**

<https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide>

<https://aws.amazon.com/cloudfront/features/>

<https://aws.amazon.com/cloudfront/pricing/>

<https://aws.amazon.com/cloudfront/faqs/>



## AWS DirectConnect

- Using Direct Connect, data can now be delivered through a private network connection between AWS and your datacenter or corporate network.
- Direct Connect links your internal network to a Direct Connect location over a standard Ethernet fiber-optic cable. One end of the cable is connected to your router, the other to a Direct Connect router. With this connection, you can create *virtual interfaces* directly to public AWS services or to Amazon VPC.
- 1Gbps and 10Gbps ports are available.
- Supports hosted connection capacities of 1, 2, 5 and 10 Gbps. 1, 2, 5 and 10 Gbps hosted connections will provide customers with higher capacities that were previously only available via dedicated connections.
- Amazon Direct Connect also supports AWS Transit Gateway, aside from configuring Site-to-Site VPN connections. With this feature, customers can connect thousands of Amazon VPCs in multiple AWS Regions to their on-premises networks using 1/2/5/10 Gbps AWS Direct Connect connections.
- Beneficial Use Cases
  - When transferring large data sets.
  - When developing and using applications that use real-time data feeds.
  - When building hybrid environments that satisfy regulatory requirements requiring the use of private connectivity.
- Setting Up Methods

Port speed	Method
1 Gbps or higher	Connect directly to an AWS device from your router at an AWS Direct Connect location.
1 Gbps or higher	Work with a partner in the AWS Partner Network or a network provider to connect a router from your data center, office, or colocation environment to an AWS Direct Connect location. The network provider does not have to be a member of the APN to connect you.
Less than 1 Gbps	Work with a partner in the AWS Partner Network who can create a hosted connection for you. Sign up for AWS and then follow the instructions to accept your hosted connection.

- Components



- **Connections** - Create a connection in an AWS Direct Connect location to establish a network connection from your premises to an AWS Region. From Direct Connect you can connect to all AZs within the region.
- **Virtual interfaces** - Create a virtual interface to enable access to AWS services. A public virtual interface enables access to public services, such as S3. A private virtual interface enables access to your VPC.
- To access public resources in a remote Region, you must set up a public virtual interface and establish a **Border Gateway Protocol** session.
- You can create a **Direct Connect gateway** in any public Region. Use it to connect your Direct Connect connection over a private virtual interface to VPCs in your account that are located in different Regions.
- To provide for failover, request and configure two dedicated connections to AWS. These connections can terminate on one or two routers in your network. There are different configuration choices available:
  - **Active/Active (BGP multipath)** - This is the default configuration, where both connections are active. If one connection becomes unavailable, all traffic is routed through the other connection.
  - **Active/Passive (failover)** - One connection is handling traffic, and the other is on standby. If the active connection becomes unavailable, all traffic is routed through the passive connection.
- **Autonomous System numbers (ASN)** are used to identify networks that present a clearly defined external routing policy to the Internet.
- Cross Connects
  - After you have downloaded your Letter of Authorization and Connecting Facility Assignment (LOA-CFA), you must complete your cross-network connection, also known as a **cross connect**.
  - If you already have equipment located in a Direct Connect location, contact the appropriate provider to complete the cross connect.
  - If you do not already have equipment located in a Direct Connect location, you can work with one of the partners in the AWS Partner Network to help you to connect to an AWS Direct Connect location.
- Virtual Interfaces
  - You must create a virtual interface to begin using your Direct Connect connection.
  - You can configure multiple virtual interfaces on a single AWS Direct Connect connection.
    - For private virtual interfaces, you need **one private virtual interface for each VPC** to connect to from the AWS Direct Connect connection, or you can use a **AWS Direct Connect gateway**.
  - Prerequisites
    - Connection: The Direct Connect connection or link aggregation group for which you are creating the virtual interface.
    - Virtual interface name: A name for the virtual interface.
    - Virtual interface owner
    - (Private virtual interface only) Connection to
    - VLAN: A unique virtual local area network tag that's not already in use on your connection.



- Address family: Whether the BGP peering session will be over IPv4 or IPv6.
- Peer IP addresses: A virtual interface can support a BGP peering session for IPv4, IPv6, or one of each (dual-stack). You cannot create multiple BGP sessions for the same IP addressing family on the same virtual interface
- BGP information: A public or private Border Gateway Protocol Autonomous System Number for your side of the BGP session, and an MD5 BGP authentication key.
- (Public virtual interface only) Prefixes you want to advertise: Public IPv4 routes or IPv6 routes to advertise over BGP. You must advertise at least one prefix using BGP.
- (Private virtual interface only) Jumbo frames
- Link Aggregation Groups (LAG)
  - A logical interface that uses the Link Aggregation Control Protocol to aggregate multiple connections at a single Direct Connect endpoint, allowing you to treat them as a single, managed connection.
  - All connections in the LAG must use the same bandwidth.
  - You can have a maximum of four connections in a LAG. Each connection in the LAG counts towards your overall connection limit for the Region.
  - All connections in the LAG must terminate at the same Direct Connect endpoint.
  - Can aggregate up to 4 Direct Connect ports into a single connection using LAG.
  - All connections in a LAG operate in Active/Active mode.
  - It will only be available for dedicated 1G and 10G connections.
- Direct Connect Gateways
  - Use a Direct Connect gateway to connect your Direct Connect connection over a private virtual interface to one or more VPCs in your account that are located in the same or different Regions.
  - It is a globally available resource.
  - Direct Connect gateway also enables you to connect between your on-premises networks and Amazon Virtual Private Cloud (Amazon VPC) in any commercial AWS Region except in China regions.
  - Prior to multi-account support, you could only associate Amazon VPCs with a Direct Connect gateway in the same AWS account. With the launch of multi-account support for Direct Connect gateway, you can associate up to 10 Amazon VPCs from multiple accounts with a Direct Connect gateway. The VPCs must be owned by AWS Accounts that belong to the same AWS payer account ID.
- Security
  - Use IAM for controlling access.
- Monitoring
  - You can optionally assign tags to your Direct Connect resources to categorize or manage them. A tag consists of a key and an optional value, both of which you define.
  - CloudTrail captures all API calls for AWS Direct Connect as events.
  - Set up CloudWatch alarms to monitor metrics.
- Pricing



- You pay only for the network ports you use and the data you transfer over the connection.
- Pricing is per port-hour consumed for each port type. Data transfer out over AWS Direct Connect is charged per GB. Data transfer IN is \$0.00 per GB in all locations.

**Sources:**

<https://docs.aws.amazon.com/directconnect/latest/UserGuide>

<https://aws.amazon.com/directconnect/features/>

<https://aws.amazon.com/directconnect/pricing/>

<https://aws.amazon.com/directconnect/faqs/>



## AWS Elastic Load Balancing

- Distributes incoming application or network traffic across multiple targets, such as **EC2 instances, containers (ECS), and IP addresses**, in multiple Availability Zones.
- When you create a load balancer, you must specify one public subnet from at least two Availability Zones. You can specify only one public subnet per Availability Zone.

### General features

- Accepts incoming traffic from clients and routes requests to its registered targets.
- Monitors the health of its registered targets and routes traffic only to healthy targets.
- Enable deletion protection to prevent your load balancer from being deleted accidentally. Disabled by default.
- Deleting ELB won't delete the instances registered to it.
- **Cross Zone Load Balancing** - when enabled, each load balancer node distributes traffic across the registered targets in all enabled AZs.
- Supports SSL Offloading which is a feature that allows the ELB to bypass the SSL termination by removing the SSL-based encryption from the incoming traffic.

### Three Types of Load Balancers

- **Application Load Balancer**
  - Functions at the application layer, the **seventh layer** of the Open Systems Interconnection (OSI) model.
  - Allows HTTP and HTTPS.
  - At least 2 subnets must be specified when creating this type of load balancer.
  - Components:
    - A *load balancer* serves as the single point of contact for clients.
    - A *listener* checks for connection requests from clients. You must define a default rule for each listener that specifies a target group, condition, and priority.
    - *Target group* routes requests to one or more registered targets. You can register a target with multiple target groups, and configure health checks on a per target group basis.
  - Benefits
    - Support for path-based and host-based routing.
    - Support for routing requests to multiple applications on a single EC2 instance.
    - Support for registering targets by IP address, including targets outside the VPC for the load balancer.
    - Support for containerized applications.
    - Support for monitoring the health of each service independently.
  - Cross-zone load balancing is always enabled.



- If you specify targets using an **instance ID**, traffic is routed to instances using the primary private IP address specified in the primary network interface for the instance. If you specify targets using **IP addresses**, you can route traffic to an instance using any private IP address from one or more network interfaces.
- You can also specify Lambda functions as targets to serve HTTP(S) requests.
- Supports load balancer-generated cookies only for sticky sessions.
- HTTP/2 Support
- WebSockets Support
- Monitoring:
  - CloudWatch metrics - retrieve statistics about data points for your load balancers and targets as an ordered set of time-series data, known as *metrics*.
  - Access logs - capture detailed information about the requests made to your load balancer and store them as log files in S3.
  - Request tracing - track HTTP requests.
  - CloudTrail logs - capture detailed information about the calls made to the Elastic Load Balancing API and store them as log files in S3.
- **Network Load Balancer**
  - Functions at the **fourth layer** of the Open Systems Interconnection (OSI) model. Uses TCP and UDP connections.
  - At least 1 subnet must be specified when creating this type of load balancer, but the recommended number is 2.
  - Components:
    - A load balancer serves as the single point of contact for clients.
    - A listener checks for connection requests from clients.
    - A target group routes requests to one or more registered targets. You can register a target with multiple target groups. You can configure health checks on a per target group basis.
  - Benefits
    - Ability to handle volatile workloads and scale to millions of requests per second.
    - Support for static IP addresses for the load balancer, or assign one Elastic IP address per subnet enabled for the load balancer.
    - Support for registering targets by IP address.
    - Support for routing requests to multiple applications on a single EC2 instance (register each instance or IP address with the same target group using multiple ports).
    - Support for containerized applications.
    - Support for monitoring the health of each service independently.
  - Cross-zone load balancing is disabled by default.
  - If you specify targets using an **instance ID**, the source IP addresses of the clients are preserved and provided to your applications. If you specify targets by **IP address**, the source IP addresses are the private IP addresses of the load balancer nodes.



- Network Load Balancers support connections from clients over inter-region VPC peering, AWS managed VPN, and third-party VPN solutions.
- You can deploy services that rely on the UDP protocol, such as Authentication and Authorization, Logging, DNS, and IoT, behind a Network Load Balancer
- Offers multi-protocol listeners, allowing you to run applications such as DNS that rely on both TCP and UDP protocols on the same port behind a Network Load Balancer.
- You CANNOT enable or disable Availability Zones for a Network Load Balancer after you create it.
- Network Load Balancers use Proxy Protocol version 2 to send additional connection information such as the source and destination.
- Preserves the client side source IP allowing the back-end to see the IP address of the client. This can then be used by applications for further processing.
- Automatically provides a static IP per Availability Zone (subnet) that can be used by applications as the front-end IP of the load balancer.
- Zonal Isolation
- In the event that your Network load balancer is unresponsive, integration with Route 53 will remove the unavailable load balancer IP address from service and direct traffic to an alternate Network Load Balancer in another region.
- Supports TLS termination on Network Load Balancers. Additionally, Network Load Balancers preserve the source IP of the clients to the back-end applications, while terminating TLS on the load balancer.
- Monitoring:
  - CloudWatch metrics - retrieve statistics about data points for your load balancers and targets as an ordered set of time-series data, known as *metrics*.
  - VPC Flow Logs - capture detailed information about the traffic going to and from your Network Load Balancer.
  - CloudTrail logs - capture detailed information about the calls made to the Elastic Load Balancing API and store them as log files in Amazon S3.

- **Classic Load Balancer**

- Distributes incoming application traffic across multiple EC2 instances in multiple Availability Zones.
- For use with EC2 classic only. Register instances with the load balancer. AWS recommends using Application or Network load balancers instead.
- To ensure that your registered instances are able to handle the request load in each AZ, keep approximately the same number of instances in each AZ registered with the load balancer.
- Benefits
  - Support for EC2-Classic
  - Support for TCP and SSL listeners
  - Support for sticky sessions using application-generated cookies



- An **Internet-facing load balancer** has a publicly resolvable DNS name, so it can route requests from clients over the Internet to the EC2 instances that are registered with the load balancer. Classic load balancers are always Internet-facing.
- Monitoring:
  - CloudWatch metrics - retrieve statistics about ELB-published data points as an ordered set of time-series data, known as *metrics*.
  - Access logs - capture detailed information for requests made to your load balancer and stores them as log files in the S3 bucket that you specify.
  - CloudTrail logs - keep track of the calls made to the Elastic Load Balancing API by or on behalf of your AWS account
- **HTTP Headers**
  - Application Load Balancers and Classic Load Balancers support **X-Forwarded-For**, **X-Forwarded-Proto**, and **X-Forwarded-Port** headers.
- Choose whether to make an **internal load balancer** or an **Internet-facing load balancer**. Classic Load Balancer in EC2-Classic must be an Internet-facing load balancer.
  - The nodes of an Internet-facing load balancer have public IP addresses.
  - The nodes of an internal load balancer have only private IP addresses.
- Public DNS name format for your load balancers
  - EC2-VPC : *name-1234567890.region.elb.amazonaws.com* (supports IPv4 addresses only)
  - EC2-Classic: (support both IPv4 and IPv6 addresses)
    - *name-123456789.region.elb.amazonaws.com*
    - *ipv6.name-123456789.region.elb.amazonaws.com*
    - *dualstack.name-123456789.region.elb.amazonaws.com*
- **Load Balancer States**
  - **Provisioning** - The load balancer is being set up.
  - **Active** - The load balancer is fully set up and ready to route traffic.
  - **Failed** - The load balancer could not be set up.
- By default, ELB idle timeout value to **60 seconds**. If a target doesn't send data at least every 60 seconds while the request is in flight, the load balancer can close the front-end connection. For back-end connections, enable the *HTTP keep-alive* option for your EC2 instances.
- You can register each EC2 instance or IP address with the same target group multiple times using different ports, which enables the load balancer to route requests to microservices.
- **Listeners** define the port and protocol to listen on.
- **Listener rules** determine how the load balancer routes requests to the targets in one or more target groups. You can add rules that specify different target groups based on the content of the request. If no rules are found, the default rule will be followed. Parts are:
  - Rule priority
  - Rule action
  - Rule conditions
- **Slow Start Mode** gives targets time to warm up before the load balancer sends them a full share of requests.



- **Sticky sessions** route requests to the same target in a target group. You enable sticky sessions at the target group level. You can also set the duration for the stickiness of the load balancer-generated cookie, in seconds.
- **Health checks** verify the status of your targets. The statuses for a registered target are:

Value	Description
initial	The load balancer is in the process of registering the target or performing the initial health checks on the target.
healthy	The target is healthy.
unhealthy	The target did not respond to a health check or failed the health check.
unused	The target is not registered with a target group, the target group is not used in a listener rule for the load balancer, or the target is in an Availability Zone that is not enabled for the load balancer.
draining	The target is deregistering and connection draining is in process.

## Security, Authentication and Access Control

- Use IAM Policies to grant permissions
  - Resource-level permissions
  - Security groups that control the traffic allowed to and from your load balancer.
- Recommended rules for internet-facing load balancer:

Inbound	
Source	Port Range
0.0.0.0/0	listener
Outbound	
Destination	Port Range
instance security group	instance listener
instance security group	health check



---

For internal load balancer:

Inbound	
Source	Port Range
VPC CIDR	<i>listener</i>
Outbound	
Destination	Port Range
<i>instance security group</i>	<i>instance listener</i>
<i>instance security group</i>	<i>health check</i>

## Summary of Features



Feature	Application Load Balancer	Network Load Balancer	Classic Load Balancer
Protocols	HTTP HTTPS	TCP	TCP, SSL/TLS, HTTP HTTPS
Platforms	VPC	VPC	EC2-Classic, VPC
Healthchecks	✓	✓	✓
Cloudwatch Metrics	✓	✓	✓
Logging	✓	✓	✓
Zonal Failover	✓	✓	✓
Connection Draining (deregistration delay)	✓	✓	✓
Load Balancing to multiple ports on the same instance	✓	✓	
Web sockets	✓	✓	
IP addresses as targets	✓	✓	
Load balancer deletion protection	✓	✓	
Path-based routing	✓		
Host-based routing	✓		
Native HTTP/2	✓		
Configurable idle connection timeout	✓		✓
Cross-zone load balancing	✓	✓	✓
SSL offloading	✓		✓
Server Name Indication (SNI)	✓		
Sticky sessions	✓		✓
Back-end server encryption	✓		✓
Static IP		✓	
Elastic IP address		✓	
Preserve Source IP address		✓	
Resource-based IAM permissions	✓	✓	✓
Tag-based IAM permissions	✓	✓	
Slow start	✓		
User authentication	✓		
Redirects	✓		
Fixed response	✓		



## Pricing

- You are charged for each hour or partial hour that an Application Load Balancer is running and the number of Load Balancer Capacity Units (LCU) used per hour.
- You are charged for each hour or partial hour that a Network Load Balancer is running and the number of Load Balancer Capacity Units (LCU) used by Network Load Balancer per hour.
- You are charged for each hour or partial hour that a Classic Load Balancer is running and for each GB of data transferred through your load balancer.

## Sources:

<https://docs.aws.amazon.com/elasticloadbalancing/latest/application/introduction.html>  
<https://docs.aws.amazon.com/elasticloadbalancing/latest/network/introduction.html>  
<https://docs.aws.amazon.com/elasticloadbalancing/latest/classic/introduction.html>  
<https://aws.amazon.com/elasticloadbalancing/features/>  
<https://aws.amazon.com/elasticloadbalancing/pricing/?nc=sn&loc=3>



## Amazon Route 53

- A highly available and scalable Domain Name System (DNS) web service used for domain registration, DNS routing, and health checking.

### Key Features

- Resolver
- Traffic flow
- Latency based routing
- Geo DNS
- Private DNS for Amazon VPC
- DNS Failover
- Health Checks and Monitoring
- Domain Registration
- CloudFront and S3 Zone Apex Support
- Amazon ELB Integration

### Domain Registration

- Choose a domain name and confirm that it's available, then register the domain name with Route 53. The service automatically makes itself the DNS service for the domain by doing the following:
  - Creates a hosted zone that has the same name as your domain.
  - Assigns a set of four name servers to the hosted zone. When someone uses a browser to access your website, such as www.example.com, these name servers tell the browser where to find your resources, such as a web server or an S3 bucket.
  - Gets the name servers from the hosted zone and adds them to the domain.
- If you already registered a domain name with another registrar, you can choose to transfer the domain registration to Route 53.

### Routing Internet Traffic to your Website or Web Application

- Use the Route 53 console to register a domain name and configure Route 53 to route internet traffic to your website or web application.
- After you register your domain name, Route 53 automatically creates a **public hosted zone** that has the same name as the domain.
- To route traffic to your resources, you create **records**, also known as *resource record sets*, in your hosted zone.
- You can create special Route 53 records, called **alias records**, that route traffic to S3 buckets, CloudFront distributions, and other AWS resources.
- Each record includes information about how you want to route traffic for your domain, such as:



- Name - name of the record corresponds with the domain name or subdomain name that you want Route 53 to route traffic for.
- Type - determines the type of resource that you want traffic to be routed to.
- Value

## Route 53 Health Checks

- Create a health check and specify values that define how you want the health check to work, such as:
  - The IP address or domain name of the endpoint that you want Route 53 to monitor.
  - The protocol that you want Route 53 to use to perform the check: HTTP, HTTPS, or TCP.
  - The **request interval** you want Route 53 to send a request to the endpoint.
  - How many consecutive times the endpoint must fail to respond to requests before Route 53 considers it unhealthy. This is the **failure threshold**.
- You can configure a health check to check the health of one or more other health checks.
- You can configure a health check to check the status of a CloudWatch alarm so that you can be notified on the basis of a broad range of criteria.

## Know the following Concepts

- Domain Registration Concepts - domain name, domain registrar, domain registry, domain reseller, top-level domain
- DNS Concepts
  - **Alias record** - a type of record that you can create to route traffic to AWS resources.
  - DNS query
  - DNS resolver
  - Domain Name System (DNS)
  - Private DNS
  - **Hosted zone** - a container for records, which includes information about how to route traffic for a domain and all of its subdomains.
  - **Name servers** - servers in the DNS that help to translate domain names into the IP addresses that computers use to communicate with one another.
  - **Record (DNS record)** - an object in a hosted zone that you use to define how you want to route traffic for the domain or a subdomain.
  - **Routing policy**
  - **Subdomain**
  - Time to live (TTL)
- Health Checking Concepts
  - **DNS failover** - a method for routing traffic away from unhealthy resources and to healthy resources.
  - endpoint
  - health check



## Routing Policies

The screenshot shows the AWS Route 53 console. On the left, there's a list of existing record sets for the domain 'mytest.com'. One entry is for the NS record type, and another is for the SOA record type. On the right, a 'Create Record Set' dialog box is open. It includes fields for 'Name' (containing 'during'), 'Type' (set to 'A – IPv4 address'), and 'Value' (a placeholder box). Below these are options for 'Alias' (set to 'No') and 'TTL (Seconds)' (set to 300). A dropdown menu for 'Routing Policy' is open, showing 'Simple' as the selected option, along with other options like 'Weighted', 'Latency', 'Failover', 'Geolocation', and 'Multivalue Answer'. At the bottom right of the dialog is a 'Create' button.

- **Simple routing policy** – route internet traffic to a single resource that performs a given function for your domain. You can't create multiple records that have the same name and type, but you can specify multiple values in the same record, such as multiple IP addresses.
- **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.
  - You can also optionally choose to route more traffic or less to a given resource by specifying a value, known as a **bias**. A bias expands or shrinks the size of the geographic region from which traffic is routed to a resource.
  - The effect of changing the bias for your resources depends on a number of factors, including the following:
    - The number of resources that you have.
    - How close the resources are to one another.
    - The number of users that you have near the border area between geographic regions.
- **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.
- When you register a domain or transfer domain registration to Route 53, it configures the domain to renew automatically. The automatic renewal period is typically one year, although the registries for some top-level domains (TLDs) have longer renewal periods.
- When you register a domain with Route 53, it creates a hosted zone that has the same name as the domain, assigns four name servers to the hosted zone, and updates the domain to use those name servers.

## Hosted Zones

- Route 53 automatically creates the Name Server (NS) and Start of Authority (SOA) records for the hosted zones.
- Route 53 creates a set of 4 unique name servers (a delegation set) within each hosted zone.
- Public hosted zone - route internet traffic to your resources
- Private hosted zone - route traffic within an Amazon VPC. You create a private hosted zone, and specify the VPCs that you want to associate with the hosted zone.
  - To use private hosted zones, you must set the following VPC settings to true:
    - enableDnsHostnames
    - enableDnsSupport
  - In a private hosted zone, you can associate Route 53 health checks only with weighted and failover records.
  - You can use the following routing policies when you create records in a private hosted zone:
    - Simple
    - Failover
    - Multivalue answer
    - Weighted

## Records

- Create records in a hosted zone. Records define where you want to route traffic for each domain name or subdomain name. The name of each record in a hosted zone must end with the name of the hosted zone.
- Alias Records
  - Route 53 **alias records** provide a Route 53-specific extension to DNS functionality. Alias records let you route traffic to selected AWS resources. They also let you route traffic from one record in a hosted zone to another record.
  - You can create an alias record at the top node of a DNS namespace, also known as the zone apex.
- CNAME Record



- You cannot create an alias record at the top node of a DNS namespace using a CNAME record.
- Alias records vs CNAME records

CNAME Records	Alias Records
You can't create a CNAME record at the zone apex.	You can create an alias record at the zone apex. Alias records must have the same type as the record you're routing traffic to.
Route 53 charges for CNAME queries.	Route 53 doesn't charge for alias queries to AWS resources.
A CNAME record redirects queries for a domain name regardless of record type.	Route 53 responds to a DNS query only when the name and type of the alias record matches the name and type in the query.
A CNAME record can point to any DNS record that is hosted anywhere.	An alias record can only point to selected AWS resources or to another record in the hosted zone that you're creating the alias record in.
A CNAME record appears as a CNAME record in response to dig or Name Server (NS) lookup queries.	An alias record appears as the record type that you specified when you created the record, such as A or AAAA.

## Supported DNS Record Types

- **A Record Type** - the value for an A record is an IPv4 address in dotted decimal notation.
- **AAAA Record Type** - the value for a AAAA record is an IPv6 address in colon-separated hexadecimal format.
- **CAA Record Type** - lets you specify which certificate authorities (CAs) are allowed to issue certificates for a domain or subdomain.
- **CNAME Record Type** - a CNAME Value element is the same format as a domain name.
- **MX Record Type** - each value for an MX record actually contains two values, *priority* and *domain name*.
- **NAPTR Record Type**
- **NS Record Type** - identifies the name servers for the hosted zone. The value for an NS record is the domain name of a name server.
- **PTR Record Type** - is the same format as a domain name.
- **SOA Record Type** - provides information about a domain and the corresponding Amazon Route 53 hosted zone.
- **SPF Record Type**
- **SRV Record Type**
- **TXT Record Type**



## DNS Domain Name Format

- Names of domains, hosted zones, and records consist of a series of labels separated by dots, which can be up to 63 bytes long. The total length of a domain name cannot exceed 255 bytes, including the dots.
- You can create hosted zones and records that include \* in the name.

## Using Traffic Flow to Route DNS Traffic

- You use the visual editor to create a **traffic policy**. A **traffic policy** includes information about the routing configuration that you want to create:
  - the routing policies that you want to use
  - resources that you want to route DNS traffic to, such as the IP address of each EC2 instance and the domain name of each ELB load balancer.
- Create a policy record where you specify the hosted zone in which you want to create the configuration that you defined in your traffic policy. It's also where you specify the DNS name that you want to associate the configuration with.

## Route 53 Resolvers

- Resolver answers DNS queries for VPC domain names such as domain names for EC2 instances or ELB load balancers, and performs recursive lookups against public name servers for all other domain names.
- DNS resolvers on your network can forward DNS queries to Resolver in a specified VPC. You can also configure Resolver to forward queries that it receives from EC2 instances in your VPCs to DNS resolvers on your network.
- Resolver is **regional**.
- An **inbound endpoint** specifies the VPC that queries pass through on the way from your network to Resolver.
- To forward DNS queries that originate on EC2 instances in one or more VPCs to your network, you create an **outbound endpoint** and one or more rules.

## Route 53 Health Checks and DNS Failover



Step 1: Configure health check

Step 2: Get notified when health check fails

## Configure health check

Route 53 health checks let you track the health status of your resources, such as web servers or mail servers, and take action when an outage occurs.

Name: test

What to monitor: Endpoint

Monitor an endpoint

Multiple Route 53 health checkers will try to establish a TCP connection with the following resource to determine whether it's healthy. [Learn more](#)

Specify endpoint by: IP address

Protocol: HTTP

IP address: 127.0.0.1

Host name: mytest.com

Port: 80

Path: /images

Advanced configuration

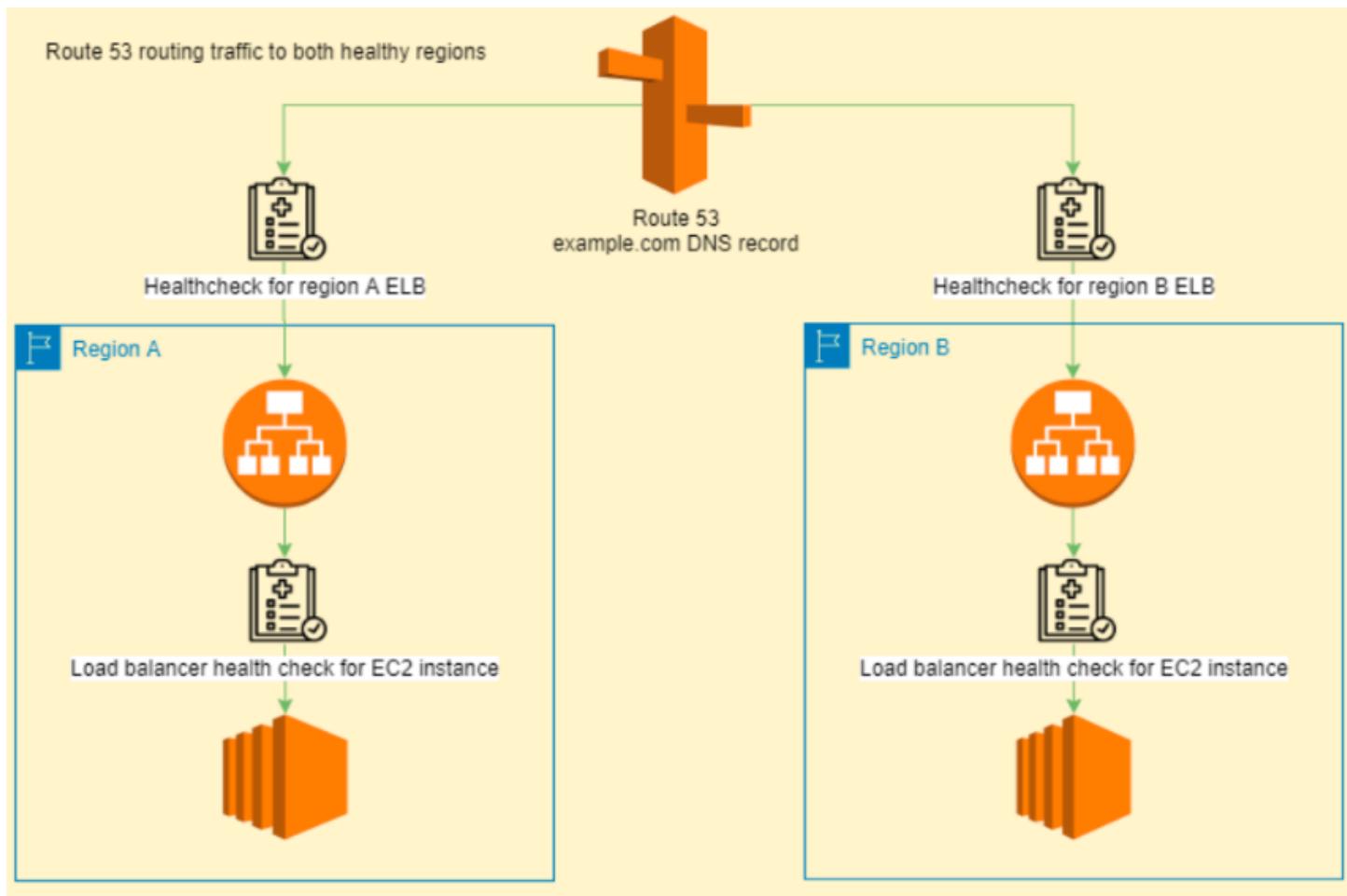
URL: http://127.0.0.1:80/

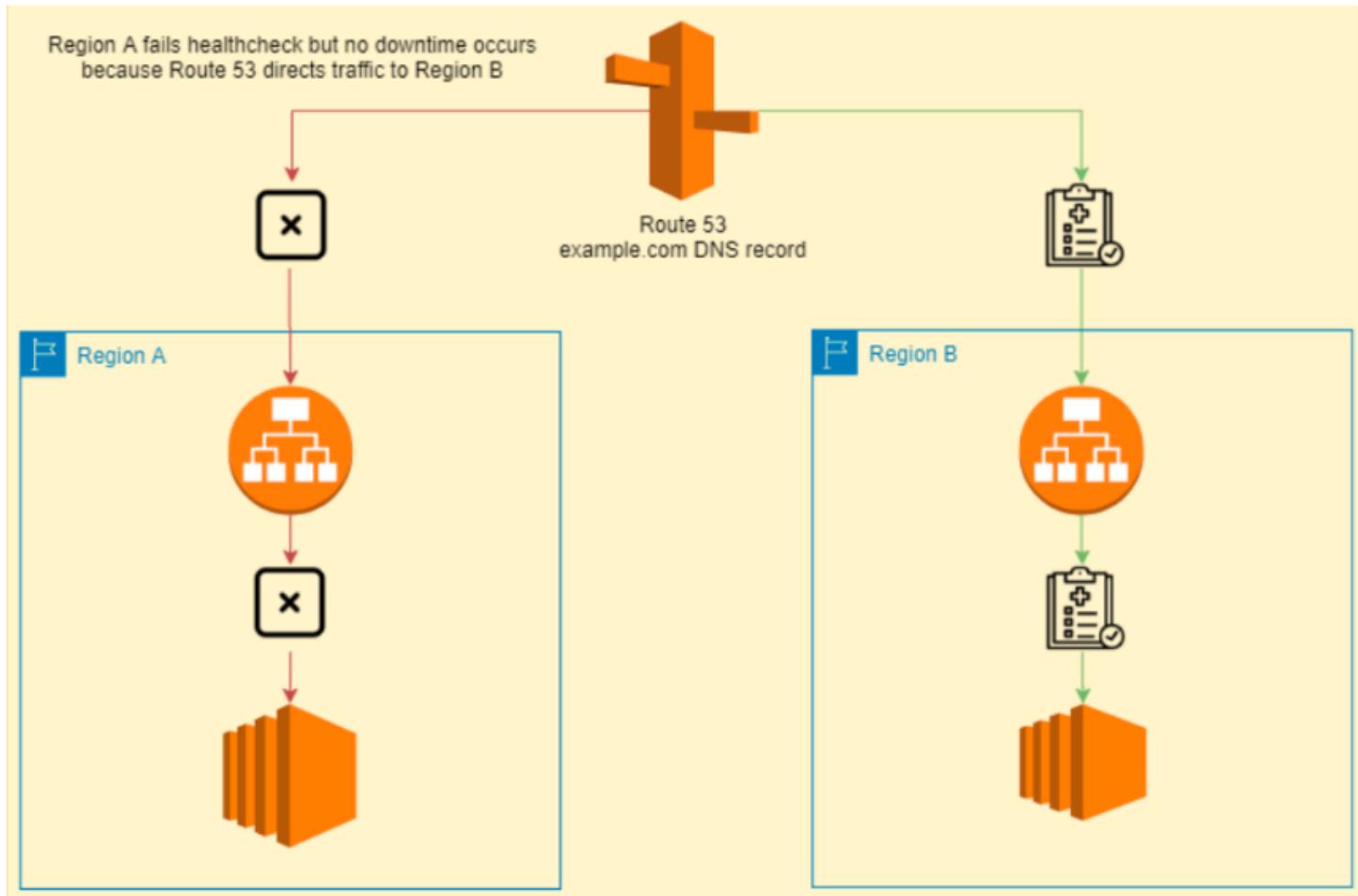
Health check type: Basic - no additional options selected ([View Pricing](#))

- Each health check that you create can monitor one of the following:
  - The health of a specified resource, such as a web server
  - The status of other health checks
  - The status of an Amazon CloudWatch alarm
- Each health checker evaluates the health of the endpoint based on two values:
  - Response time
  - Whether the endpoint responds to a number of consecutive health checks that you specify (the failure threshold)
- Types of health checks
  - **HTTP and HTTPS health checks** – Route 53 must be able to establish a TCP connection with the endpoint within four seconds. In addition, the endpoint must respond with an HTTP status code of 2xx or 3xx within two seconds after connecting.
  - **TCP health checks** – Route 53 must be able to establish a TCP connection with the endpoint within ten seconds.
  - **HTTP and HTTPS health checks with string matching** – Route 53 must be able to establish a TCP connection with the endpoint within four seconds, and the endpoint must respond with an HTTP status code of 2xx or 3xx within two seconds after connecting. After a Route 53 health

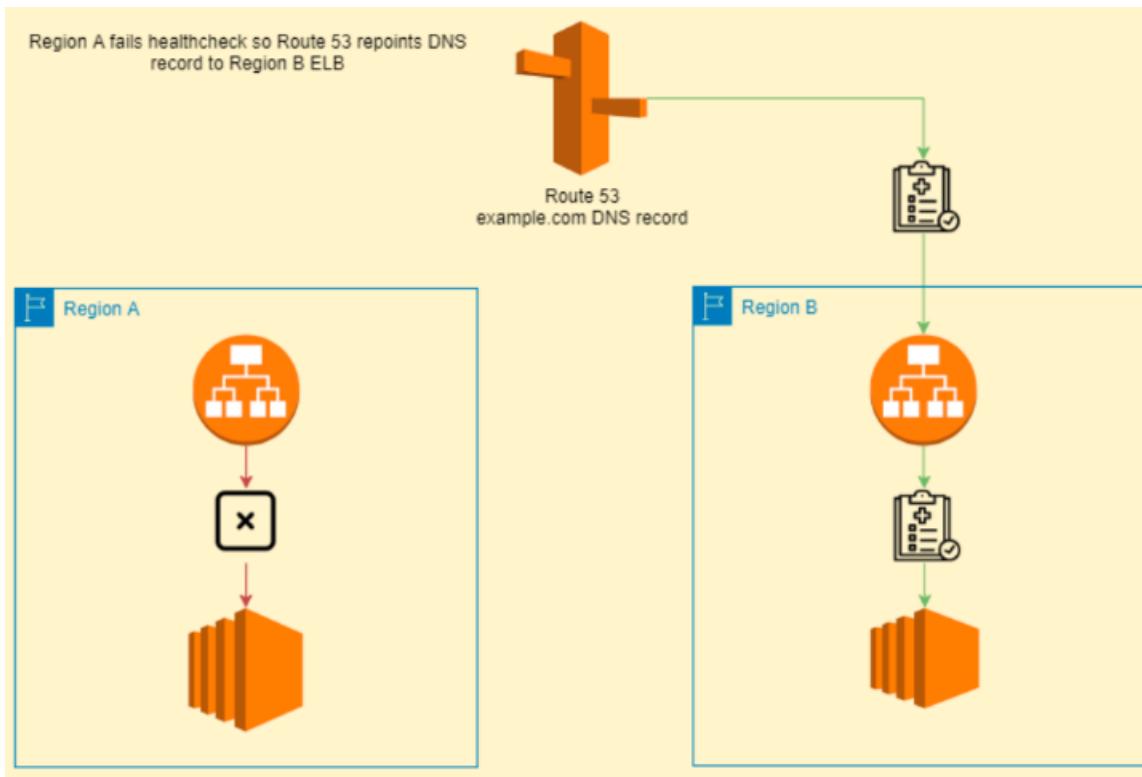
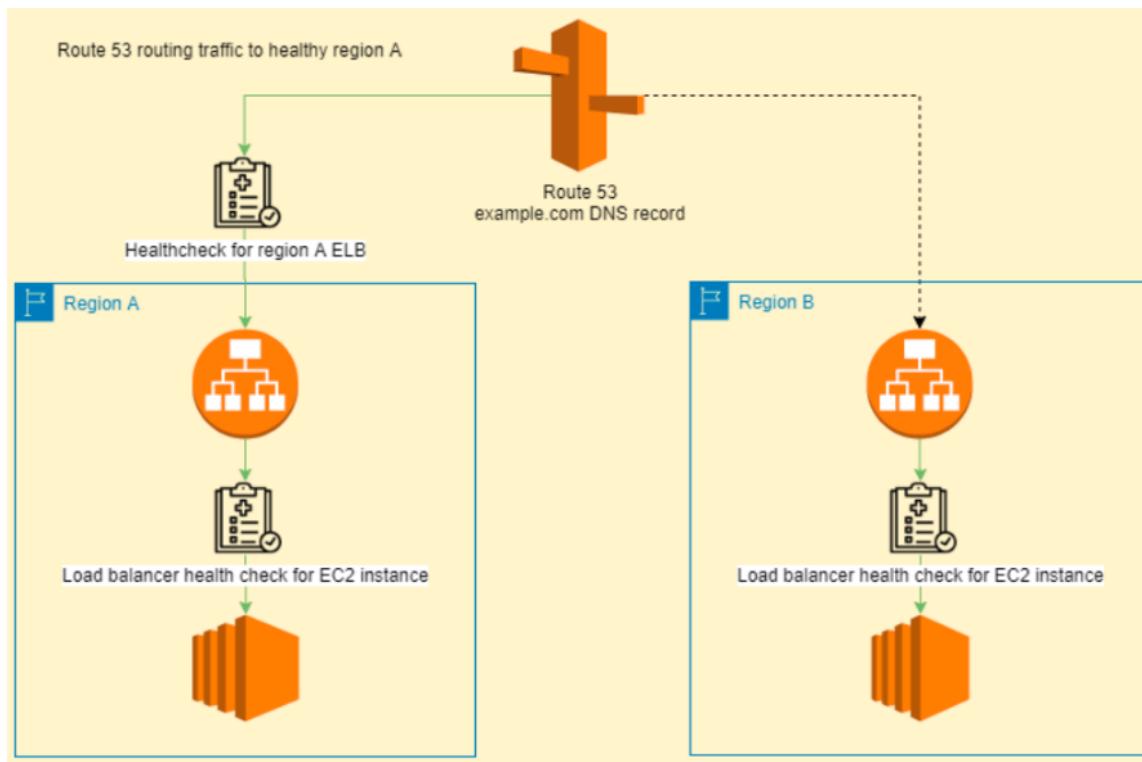
checker receives the HTTP status code, it must receive the response body from the endpoint within the next two seconds.

- Health-checking features to route traffic only to the healthy resources:
  - Check the health of EC2 instances and other resources (non-alias records)
  - Evaluate the health of an AWS resource (alias records)
- Two types of failover configurations
  - **Active-Active Failover** - all the records that have the same name, the same type, and the same routing policy are active unless Route 53 considers them unhealthy. Use this failover configuration when you want all of your resources to be available the majority of the time.





- **Active-Passive Failover** - use this failover configuration when you want a primary resource or group of resources to be available the majority of the time and you want a secondary resource or group of resources to be on standby in case all the primary resources become unavailable.





- When responding to queries, Route 53 includes only the healthy primary resources.
- To create an **active-passive failover configuration** with one primary record and one secondary record, you just create the records and specify **Failover for the routing policy**.
- To configure **active-passive failover** with multiple resources for the primary or secondary record, create records with the same name, type, and routing policy for your primary resources. If you're using AWS resources that you can create alias records for, specify **Yes** for **Evaluate Target Health**.
- You can also use weighted records for active-passive failover, with caveats.
- You can configure Amazon Route 53 to log information about the queries that Route 53 receives. Query logging is available only for public hosted zones.

## Authentication and Access Control

- Authenticate with IAM before allowing to perform any operation on Route 53 resources.
- Every AWS resource is owned by an AWS account, and permissions to create or access a resource are governed by permissions policies. A *permissions policy* specifies who has access to what.

## Monitoring

- The Route 53 dashboard provides detailed information about the status of your domain registrations, including:
  - Status of new domain registrations
  - Status of domain transfers to Route 53
  - List of domains that are approaching the expiration date
- You can use Amazon CloudWatch metrics to see the number of DNS queries served for each of your Route 53 public hosted zones. With these metrics, you can see at a glance the activity level of each hosted zone to monitor changes in traffic.
- You can monitor your resources by creating Route 53 health checks, which use CloudWatch to collect and process raw data into readable, near real-time metrics.
- Log API calls with CloudTrail

## Pricing

- A hosted zone is charged at the time it's created and on the first day of each subsequent month. To allow testing, a hosted zone that is deleted within 12 hours of creation is not charged, however, any queries on that hosted zone will still incur charges.
- Billion queries / month
- Queries to Alias records are provided at no additional cost to current Route 53 customers when the records are mapped to the following AWS resource types:
  - Elastic Load Balancers
  - Amazon CloudFront distributions
  - AWS Elastic Beanstalk environments
  - Amazon S3 buckets that are configured as website endpoints



- Traffic flow policy record / month
- Pricing for domain names varies by Top Level Domain (TLD)

**Sources:**

[<https://aws.amazon.com/route53/features/>](https://docs.aws.amazon.com/Route53/latest/DeveloperGuide>Welcome.html</a></p></div><div data-bbox=)

<https://aws.amazon.com/route53/pricing/>



## Amazon VPC

- Create a virtual network in the cloud dedicated to your AWS account where you can launch AWS resources
- Amazon VPC is the networking layer of Amazon EC2
- A VPC spans all the Availability Zones in the region. After creating a VPC, you can add one or more subnets in each Availability Zone.

## Key Concepts

- A **virtual private cloud** (VPC) allows you to specify an IP address range for the VPC, add subnets, associate security groups, and configure route tables.
- A **subnet** is a range of IP addresses in your VPC. You can launch AWS resources into a specified subnet. Use a **public subnet** for resources that must be connected to the internet, and a **private subnet** for resources that won't be connected to the internet.
- To protect the AWS resources in each subnet, use **security groups** and **network access control lists (ACL)**.
- Expand your VPC by adding secondary IP ranges.

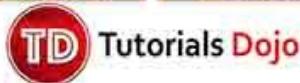
## EC2-VPC vs EC2-Classic

EC2-VPC		vs	EC2-Classic
✓	Assign static private IPv4 addresses to your instances that persist across starts and stops	X	
✓	Optionally associate an IPv6 CIDR block to your VPC and assign IPv6 addresses to your instances	X	
✓	Assign multiple IP addresses to your instances	X	
✓	Define network interfaces, and attach one or more network interfaces to your instances	X	
✓	Change security group membership for your instances while they're running	X	
✓	Control the outbound traffic from your instances (egress filtering) in addition to controlling the inbound traffic to them (ingress filtering)	X	
✓	Add an additional layer of access control to your instances in the form of network access control lists (ACL)	X	
✓	Run your instances on single-tenant hardware	X	



## Default vs Non-Default VPC

Default	Non-Default VPC
If your account supports the EC2-VPC platform only, it comes with a default VPC that has a default subnet in each Availability Zone.	You can create your own non-default VPC, and configure it as you need. Subnets that you create in your non-default VPC and additional subnets that you create in your default VPC are called non-default subnets.
Your default VPC includes an internet gateway, which allows your instances to communicate with the internet, and each default subnet is a public subnet.	Instances can communicate with each other, but can't access the internet. You can enable internet access for an instance launched into a non-default subnet by attaching an internet gateway and associating an Elastic IP address with the instance.
Each instance that you launch into a default subnet has a private IPv4 address and a public IPv4 address.	By default, each instance that you launch into a non-default subnet has a private IPv4 address, but no public IPv4 address, unless you specifically assign one at launch, or you modify the subnet's public IP address attribute.
To allow an instance in your VPC to initiate outbound connections to the internet but prevent unsolicited inbound connections from the internet, you can use a network address translation (NAT) device for IPv4 traffic.	To allow an instance in your VPC to initiate outbound connections to the internet but prevent unsolicited inbound connections from the internet, you can use a network address translation (NAT) device for IPv4 traffic.
You can optionally associate an Amazon-provided IPv6 CIDR block with your VPC and assign IPv6 addresses to your instances. IPv6 traffic is separate from IPv4 traffic; your route tables must include separate routes for IPv6 traffic.	You can optionally associate an Amazon-provided IPv6 CIDR block with your VPC and assign IPv6 addresses to your instances. IPv6 traffic is separate from IPv4 traffic; your route tables must include separate routes for IPv6 traffic.



## Accessing a Corporate or Home Network

- You can optionally connect your VPC to your own corporate data center using an **IPsec AWS managed VPN connection**, making the AWS Cloud an extension of your data center.
- A **VPN connection** consists of:
  - a **virtual private gateway** (which is the VPN concentrator on the Amazon side of the VPN connection) attached to your VPC.
  - a **customer gateway** (which is a physical device or software appliance on your side of the VPN connection) located in your data center.
  - A diagram of the connection
- **AWS Site-to-Site Virtual Private Network (VPN)** connections can be moved from a virtual private gateway to an **AWS Transit Gateway** without having to make any changes on your customer gateway.



Transit Gateways enable you to easily scale connectivity across thousands of Amazon VPCs, AWS accounts, and on-premises networks.

- **AWS PrivateLink** enables you to privately connect your VPC to supported AWS services, services hosted by other AWS accounts (VPC endpoint services), and supported AWS Marketplace partner services. You do not require an internet gateway, NAT device, public IP address, AWS Direct Connect connection, or VPN connection to communicate with the service. Traffic between your VPC and the service does not leave the Amazon network.
- AWS PrivateLink-Supported Services:
  - Amazon API Gateway
  - Amazon AppStream 2.0
  - AWS App Mesh
  - Application Auto Scaling
  - Amazon Athena
  - AWS Auto Scaling
  - Amazon Cloud Directory
  - AWS CloudFormation
  - AWS CloudTrail
  - Amazon CloudWatch
  - Amazon CloudWatch Events
  - Amazon CloudWatch Logs
  - AWS CodeBuild
  - AWS CodeCommit
  - AWS CodePipeline
  - AWS Config
  - AWS DataSync
  - Amazon EC2 API
  - Amazon EC2 Auto Scaling
  - Amazon Elastic File System
  - Elastic Load Balancing
  - Amazon Elastic Container Registry
  - Amazon Elastic Container Service
  - AWS Glue
  - AWS Key Management Service
  - Amazon Kinesis Data Firehose
  - Amazon Kinesis Data Streams
  - Amazon Rekognition
  - Amazon SageMaker and Amazon SageMaker Runtime
  - Amazon SageMaker Notebook
  - AWS Secrets Manager
  - AWS Security Token Service
  - AWS Server Migration Service
  - AWS Service Catalog
  - Amazon SNS
  - Amazon SQS
  - AWS Systems Manager
  - AWS Storage Gateway
  - AWS Transfer for SFTP
  - Amazon WorkSpaces
  - Endpoint services hosted by other AWS accounts
  - Supported AWS Marketplace partner services
- You can create a **VPC peering connection** between your VPCs, or with a VPC in another AWS account, and enable routing of traffic between the VPCs using private IP addresses. You cannot create a VPC peering connection between VPCs that have overlapping CIDR blocks.
- Applications in an Amazon VPC can securely access AWS PrivateLink endpoints across VPC peering connections. The support of VPC peering by AWS PrivateLink makes it possible for customers to



---

privately connect to a service even if that service's endpoint resides in a different Amazon VPC that is connected using VPC peering.

- AWS PrivateLink endpoints can now be accessed across both intra- and inter-region VPC peering connections.

## VPC Use Case Scenarios

- VPC with a Single Public Subnet
- VPC with Public and Private Subnets (NAT)
- VPC with Public and Private Subnets and AWS Managed VPN Access
- VPC with a Private Subnet Only and AWS Managed VPN Access

## Subnets

- When you create a VPC, you must specify a range of IPv4 addresses for the VPC in the form of a Classless Inter-Domain Routing (CIDR) block (example: 10.0.0.0/16). This is the **primary CIDR block** for your VPC.
- You can add one or more subnets in each Availability Zone of your VPC's region.
- You specify the CIDR block for a subnet, which is a subset of the VPC CIDR block.
- A CIDR block must not overlap with any existing CIDR block that's associated with the VPC.
- Types of Subnets
  - Public Subnet - has an internet gateway
  - Private Subnet - doesn't have an internet gateway
  - VPN-only Subnet - has a virtual private gateway instead
- IPv4 CIDR block size should be between a /16 netmask (65,536 IP addresses) and /28 netmask (16 IP addresses).
- The **first four IP addresses and the last IP address in each subnet CIDR block** are **NOT available** for you to use, and cannot be assigned to an instance.
- You cannot increase or decrease the size of an existing CIDR block.
- When you associate a CIDR block with your VPC, a route is automatically added to your VPC route tables to enable routing within the VPC (the destination is the CIDR block and the target is *local*).
- You have a limit on the number of CIDR blocks you can associate with a VPC and the number of routes you can add to a route table.
- The following rules apply when you add IPv4 CIDR blocks to a VPC that's part of a **VPC peering connection**:
  - If the VPC peering connection is active, you can add CIDR blocks to a VPC provided they do not overlap with a CIDR block of the peer VPC.
  - If the VPC peering connection is pending-acceptance, the owner of the requester VPC cannot add any CIDR block to the VPC. Either the owner of the accepter VPC must accept the peering connection, or the owner of the requester VPC must delete the VPC peering connection request, add the CIDR block, and then request a new VPC peering connection.



- If the VPC peering connection is pending-acceptance, the owner of the accepter VPC can add CIDR blocks to the VPC. If a secondary CIDR block overlaps with a CIDR block of the requester VPC, the VPC peering connection request fails and cannot be accepted.
- If you're using AWS Direct Connect to connect to multiple VPCs through a direct connect gateway, the VPCs that are associated with the direct connect gateway must not have overlapping CIDR blocks.
- The CIDR block is ready for you to use when it's in the **associated** state.
- You can disassociate a CIDR block that you've associated with your VPC; however, you cannot disassociate the primary CIDR block.

## Subnet Routing

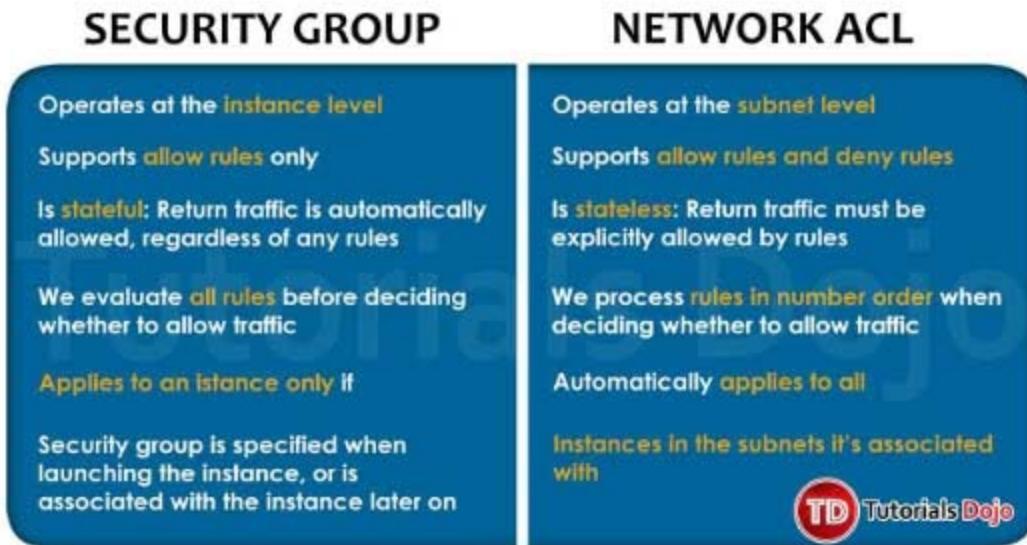
- Each subnet must be associated with a **route table**, which specifies the allowed routes for **outbound traffic** leaving the subnet.
- Every subnet that you create is automatically associated with the main route table for the VPC.
- You can change the association, and you can change the contents of the main route table.
- You can allow an instance in your VPC to initiate outbound connections to the internet over IPv4 but prevent unsolicited inbound connections from the internet using a **NAT gateway or NAT instance**.
- To initiate outbound-only communication to the internet over IPv6, you can use an egress-only internet gateway.

## Subnet Security

- Security Groups – control inbound and outbound traffic for your instances
  - You can associate one or more (up to five) security groups to an instance in your VPC.
  - If you don't specify a security group, the instance automatically belongs to the default security group.
  - When you create a security group, it has no inbound rules. By default, it includes an outbound rule that allows all outbound traffic.
  - Security groups are associated with network interfaces.
- Network Access Control Lists – control inbound and outbound traffic for your subnets
  - Each subnet in your VPC must be associated with a network ACL. If none is associated, automatically associated with the default network ACL.
  - You can associate a network ACL with multiple subnets; however, a subnet can be associated with only one network ACL at a time.
  - A network ACL contains a numbered list of rules that is evaluated in order, starting with the lowest numbered rule, to determine whether traffic is allowed in or out of any subnet associated with the network ACL.
  - The default network ACL is configured to **allow all traffic to flow in and out** of the subnets to which it is associated.
- Flow logs – capture information about the IP traffic going to and from network interfaces in your VPC that is published to CloudWatch Logs.



- VPC Flow Logs can be sent directly to an Amazon S3 bucket which allows you to retrieve and analyze these logs yourself.
- Amazon security groups and network ACLs don't filter traffic to or from link-local addresses or AWS-reserved IPv4 addresses. Flow logs do not capture IP traffic to or from these addresses.



- Diagram of security groups and NACLs in a VPC

## VPC Networking Components

- Network Interfaces
  - a virtual network interface that can include:
    - a primary private IPv4 address
    - one or more secondary private IPv4 addresses
    - one Elastic IP address per private IPv4 address
    - one public IPv4 address, which can be auto-assigned to the network interface for eth0 when you launch an instance
    - one or more IPv6 addresses
    - one or more security groups
    - a MAC address
    - a source/destination check flag
    - a description
  - Network interfaces can be attached and detached from instances, however, you cannot detach a primary network interface.
- Route Tables



- contains a set of rules, called *routes*, that are used to determine where network traffic is directed.
- A subnet can only be associated with one route table at a time, but you can associate multiple subnets with the same route table.
- You cannot delete the main route table, but you can replace the main route table with a custom table that you've created.
- You must update the route table for any subnet that uses gateways or connections.
- Uses the most specific route in your route table that matches the traffic to determine how to route the traffic (longest prefix match).
- Internet Gateways
  - Allows communication between instances in your VPC and the internet.
  - Imposes no availability risks or bandwidth constraints on your network traffic.
  - Provides a target in your VPC route tables for internet-routable traffic, and performs network address translation for instances that have been assigned public IPv4 addresses.
  - The following table provides an overview of whether your VPC automatically comes with the components required for internet access over IPv4 or IPv6.
  - To enable access to or from the Internet for instances in a VPC subnet, you must do the following:
    - Attach an Internet Gateway to your VPC
    - Ensure that your subnet's route table points to the Internet Gateway.
    - Ensure that instances in your subnet have a globally unique IP address (public IPv4 address, Elastic IP address, or IPv6 address).
    - Ensure that your network access control and security group rules allow the relevant traffic to flow to and from your instance

	Default VPC	Non-default VPC
Internet gateway	Yes	Yes, if you created the VPC using the first or second option in the VPC wizard. Otherwise, you must manually create and attach the internet gateway.
Route table with route to internet gateway for IPv4 traffic (0.0.0.0/0)	Yes	Yes, if you created the VPC using the first or second option in the VPC wizard. Otherwise, you must manually create the route table and add the route.
Route table with route to internet gateway for IPv6 traffic (::/0)	No	Yes, if you created the VPC using the first or second option in the VPC wizard, and if you specified the option to associate an IPv6 CIDR block with the VPC. Otherwise, you must manually create the route table and add the route.



Public IPv4 address automatically assigned to instance launched into subnet	Yes (default subnet)	No (non-default subnet)
IPv6 address automatically assigned to instance launched into subnet	No (default subnet)	No (non-default subnet)

- Egress-Only Internet Gateways
  - VPC component that allows outbound communication over IPv6 from instances in your VPC to the Internet, and prevents the Internet from initiating an IPv6 connection with your instances.
  - An egress-only Internet gateway is stateful.
  - You cannot associate a security group with an egress-only Internet gateway.
  - You can use a network ACL to control the traffic to and from the subnet for which the egress-only Internet gateway routes traffic.
- NAT
  - Enable instances in a private subnet to connect to the internet or other AWS services, but prevent the internet from initiating connections with the instances.
  - NAT Gateways
    - You must specify the **public subnet** in which the NAT gateway should reside.
    - You must specify an **Elastic IP address** to associate with the NAT gateway when you create it.
    - Each NAT gateway is created in a specific Availability Zone and implemented with redundancy in that zone.
    - Deleting a NAT gateway disassociates its Elastic IP address, but does not release the address from your account.
    - A NAT gateway supports the following protocols: TCP, UDP, and ICMP.
    - You cannot associate a security group with a NAT gateway.
    - A NAT gateway can support up to 55,000 simultaneous connections to each unique destination.
    - A NAT gateway cannot send traffic over VPC endpoints, VPN connections, AWS Direct Connect, or VPC peering connections.
  - NAT Instance vs NAT Gateways



## Tutorials Dojo

Attribute	NAT gateway	NAT instance
Availability	Highly available. NAT gateways in each Availability Zone are implemented with redundancy. Create a NAT gateway in each Availability Zone to ensure zone-independent architecture.	Use a script to manage failover between instances
Bandwidth	Can scale up to 45 Gbps.	Depends on the bandwidth of the instance type
Maintenance	Manage by AWS	Manage by you.
Performance	Software is optimized for handling NAT traffic	A generic Amazon Linux AMI that's configured to perform NAT
Cost	Charged depending on the number of NAT gateways you use, duration of usage, and amount of data that you send through the NAT gateways.	Charged depending on the number of NAT instances that you use, duration of usage, and instance type and size.
Type and size	Uniform offering; you don't need to decide on the type or size.	Choose a suitable instance type and size, according to your predicted workload
Public IP addresses	Choose the Elastic IP address to associate with a NAT gateway at creation.	Use an elastic IP address or a public IP address with a NAT instance. You can change the public IP address at any time by associating a new elastic IP address with the instance.
Private IP addresses	Automatically selected from the subnet's IP address range when you create the gateway.	Assign a specific private IP address from the subnet's IP address range when you launch the instance.
Security groups	Cannot be associated with a NAT gateway	Associate with your NAT instance and the resources behind your NAT instance to control inbound and outbound traffic.
Network ACLs	Use a network ACL to control the traffic to and from the subnet in which your NAT gateway resides.	Use a network ACL to control the traffic to and from the subnet in which your NAT instance resides.
Flow logs	Use flow logs to capture the traffic.	Use flow logs to capture the traffic.
Port Forwarding	Not supported.	Manually customize the configuration to support port forwarding.
Bastion Servers	Not supported.	Use as a bastion server.
Traffic Metrics	Monitor your NAT gateway using CloudWatch Metrics.	View Cloudwatch metrics for the instance.
Timeout Behavior	When a connection times out, a NAT gateway returns an RST packet to any resources behind the NAT gateway that attempt to continue the connection (it does not send a FIN packet).	When a connection times out, a NAT instance sends a FIN packet to resources behind the NAT instance to close the connection.
IP Fragmentation	Supports forwarding of IP fragmented packets for the UDP protocol. Does not support fragmentation for the TCP and ICMP protocols. Fragmented packets for these protocols will get dropped.	Supports reassembly of IP fragmented packets for the UDP, TCP, and ICMP protocols.

- DHCP Options Sets

- **Dynamic Host Configuration Protocol (DHCP)** provides a standard for passing configuration information to hosts on a TCP/IP network.
- You can assign your own domain name to your instances, and use up to four of your own DNS servers by specifying a special set of DHCP options to use with the VPC.
- Creating a VPC automatically creates a set of DHCP options, which are `domain-name-servers=AmazonProvidedDNS`, and `domain-name=domain-name-for-your-region`, and associates them with the VPC.



- After you create a set of DHCP options, you can't modify them. Create a new set and associate a different set of DHCP options with your VPC, or use no DHCP options at all.
- DNS
  - AWS provides instances launched in a default VPC with public and private DNS hostnames that correspond to the public IPv4 and private IPv4 addresses for the instance.
  - AWS provides instances launched in a non-default VPC with private DNS hostname and possibly a public DNS hostname, depending on the DNS attributes you specify for the VPC and if your instance has a public IPv4 address.
  - Set VPC attributes `enableDnsHostnames` and `enableDnsSupport` to true so that your instances receive a public DNS hostname and Amazon-provided DNS server can resolve Amazon-provided private DNS hostnames.
    - If you use custom DNS domain names defined in a private hosted zone in Route 53, the `enableDnsHostnames` and `enableDnsSupport` attributes must be set to true.
- VPC Peering
  - A networking connection between two VPCs that enables you to route traffic between them privately. Instances in either VPC can communicate with each other as if they are within the same network.
- Elastic IP Addresses
  - A **static, public IPv4 address**.
  - You can associate an Elastic IP address with any instance or network interface for any VPC in your account.
  - You can mask the failure of an instance by rapidly remapping the address to another instance in your VPC.
  - Your Elastic IP addresses remain associated with your AWS account until you explicitly release them.
  - AWS imposes a small hourly charge when EIPs aren't associated with a running instance, or when they are associated with a stopped instance or an unattached network interface.
  - You're limited to five Elastic IP addresses.
- VPC Endpoints
  - Privately connect your VPC to supported AWS services and VPC endpoint services powered by PrivateLink without requiring an internet gateway, NAT device, VPN connection, or AWS Direct Connect connection.
  - Endpoints are virtual devices.
  - Two Types
    - **Interface Endpoints**
      - An elastic network interface with a private IP address that serves as an entry point for traffic destined to a supported service.
      - Can be accessed through AWS VPN connections or AWS Direct Connect connections, through intra-region VPC peering connections from Nitro instances, and through inter-region VPC peering connections from any type of instance.



- For each interface endpoint, you can choose only one subnet per Availability Zone. Endpoints are supported within the same region only.
  - Interface endpoints do not support the use of endpoint policies.
  - An interface endpoint supports IPv4 TCP traffic only.
- **Gateway Endpoints**
    - A gateway that is a target for a specified route in your route table, used for traffic destined to a supported AWS service.
    - You can create multiple endpoints in a single VPC, for example, to multiple services. You can also create multiple endpoints for a single service, and use different route tables to enforce different access policies from different subnets to the same service.
    - You can modify the endpoint policy that's attached to your endpoint, and add or remove the route tables that are used by the endpoint.
    - Endpoints are supported within the same region only. You cannot create an endpoint between a VPC and a service in a different region.
    - Endpoints support IPv4 traffic only.
    - You must enable DNS resolution in your VPC, or if you're using your own DNS server, ensure that DNS requests to the required service (such as S3) are resolved correctly to the IP addresses maintained by AWS.
    - You can create your own application in your VPC and configure it as an AWS PrivateLink-powered service (referred to as an *endpoint service*). You are the *service provider*, and the AWS principals that create connections to your service are *service consumers*.
  - ClassicLink
    - Allows you to link an EC2-Classic instance to a VPC in your account, within the same region. This allows you to associate the VPC security groups with the EC2-Classic instance, enabling communication between your EC2-Classic instance and instances in your VPC using private IPv4 addresses.

## VPN Connections

VPN connectivity option	Description
AWS managed VPN	You can create an IPsec VPN connection between your VPC and your remote network. On the AWS side of the VPN connection, a <i>virtual private gateway</i> provides two VPN endpoints (tunnels) for automatic failover. You configure your <i>customer gateway</i> on the remote side of the VPN connection.
AWS VPN CloudHub	If you have more than one remote network, you can create multiple AWS managed VPN connections via your virtual private gateway to enable communication between these networks.



Third party software VPN appliance	You can create a VPN connection to your remote network by using an Amazon EC2 instance in your VPC that's running a third party software VPN appliance. AWS does not provide or maintain third party software VPN appliances; however, you can choose from a range of products provided by partners and open source communities.
AWS Direct Connect	You can also use AWS Direct Connect to create a dedicated private connection from a remote network to your VPC. You can combine this connection with an AWS managed VPN connection to create an IPsec-encrypted connection.

- Specify a private Autonomous System Number (ASN) for the virtual private gateway. If you don't specify an ASN, the virtual private gateway is created with the default ASN (64512). You cannot change the ASN after you've created the virtual private gateway.
- When you create a VPN connection, you must:
  - Specify the type of routing that you plan to use (static or dynamic)
  - Update the route table for your subnet
- If your VPN device supports Border Gateway Protocol (BGP), specify **dynamic routing** when you configure your VPN connection. If your device does not support BGP, specify **static routing**.
- VPG uses path selection to determine how to route traffic to your remote network. Longest prefix match applies.
- Each VPN connection has two tunnels, with each tunnel using a unique virtual private gateway public IP address. It is important to configure both tunnels for redundancy.

## Pricing

- Charged for VPN Connection-hour
- Charged for each "NAT Gateway-hour" that your NAT gateway is provisioned and available.
- Data processing charges apply for each Gigabyte processed through the NAT gateway regardless of the traffic's source or destination.
- You also incur standard AWS data transfer charges for all data transferred via the NAT gateway.
- Charges for unused or inactive Elastic IPs.

## Sources:

<https://docs.aws.amazon.com/vpc/latest/userguide/what-is-amazon-vpc.html>  
<https://aws.amazon.com/vpc/details/>  
<https://aws.amazon.com/vpc/pricing/>  
<https://aws.amazon.com/vpc/faqs/>



## VPC Peering

- A networking connection between **two VPCs** that enables you to route traffic between them privately using private IPv4 addresses or IPv6 addresses. Instances in either VPC can communicate with each other as if they are within the same network.
- You can create a VPC peering connection between your own VPCs, with a VPC in another AWS account, or with a VPC in a different AWS Region (also called Inter-Region VPC Peering).
- A VPC peering connection is neither a gateway nor a AWS Site-to-Site 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.
- VPC Peering does **NOT support edge-to-edge routing**. You can create multiple VPC peering connections for each VPC that you own, but transitive peering relationships are not supported.
- Establishing A Peering Connection
  - The owner of the *requester* VPC sends a request to the owner of the *accepter* VPC to create the VPC peering connection. The *accepter* VPC cannot have a CIDR block that overlaps with the *requester* VPC's CIDR block.
  - To enable the flow of traffic between the VPCs using private IP addresses, the owner of each VPC in the VPC peering connection must manually add a route to one or more of their VPC route tables that points to the IP address range of the other VPC (the peer VPC).
  - Update the security group rules that are associated with your instance to ensure that traffic to and from the peer VPC is not restricted.
  - By default, if instances on either side of a VPC peering connection address each other using a public DNS hostname, the hostname resolves to the instance's public IP address. To change this behavior, enable DNS hostname resolution for your VPC connection. This will allow the DNS hostname to resolve to the instance's private IP address.
- Limitations
  - You cannot create a VPC peering connection between VPCs that have matching or overlapping IPv4 or IPv6 CIDR blocks.
  - You cannot have more than one VPC peering connection between the same two VPCs at the same time.
  - **Unicast reverse path forwarding** in VPC peering connections is not supported.
  - If the VPCs are in the same region, you can enable the resources on either side of a VPC peering connection to communicate with each other over IPv6.
  - For inter-region peering, you cannot create a security group rule that references a peer VPC security group. Communication over IPv6 is not supported as well.

### Sources:

<https://docs.aws.amazon.com/vpc/latest/userguide/vpc-peering.html>  
<https://docs.aws.amazon.com/vpc/latest/peering/what-is-vpc-peering.html>



## AWS Global Accelerator

- A service that uses the AWS Global Network to improve availability and performance of your applications to your local and global users.
- It provides static IP addresses that act as a fixed entry point to your application endpoints in a single or multiple AWS Regions, such as your Application Load Balancers, Network Load Balancers or Amazon EC2 instances.
- AWS Global Accelerator continually monitors the health of your application endpoints and will detect an unhealthy endpoint and redirect traffic to healthy endpoints in less than 1 minute.
- Concepts
  - An **accelerator** is the resource you create to direct traffic to optimal endpoints over the AWS global network.
  - **Network zones** are isolated units with their own set of physical infrastructure and service IP addresses from a unique IP subnet.
  - AWS Global Accelerator provides you with a set of **two static IP addresses** that are anycast from the AWS edge network. It also assigns a default **Domain Name System (DNS) name** to your accelerator, similar to `a1234567890abcdef.awsglobalaccelerator.com`, that points to the static IP addresses.
  - A **listener** processes inbound connections from clients to Global Accelerator, based on the port (or port range) and protocol that you configure. Global Accelerator supports both TCP and UDP protocols.
  - Each **endpoint group** is associated with a specific AWS Region. Endpoint groups include one or more endpoints in the Region.
  - **Endpoints** can be Network Load Balancers, Application Load Balancers, EC2 instances, or Elastic IP addresses.
- Benefits
  - **Instant regional failover** - AWS Global Accelerator automatically checks the health of your applications and routes user traffic only to healthy application endpoints. If the health status changes or you make configuration updates, AWS Global Accelerator reacts instantaneously to route your users to the next available endpoint. Monitors the health of your application endpoints by using TCP, HTTP, and HTTPS health checks.
  - **High availability** - AWS Global Accelerator has a fault-isolating design that increases the availability of your application. When you create an accelerator, you are allocated two IPv4 static IP addresses that are serviced by independent network zones.
  - **No variability around clients that cache IP addresses** - You do not have to rely on the IP address caching settings of client devices. Change propagation of a configuration update only takes a matter of seconds.
  - **Improved performance** - AWS Global Accelerator ingresses traffic from the edge location that is closest to your end clients through anycast static IP addresses. Then traffic traverses the AWS global network, which optimizes the path to your application that is running in an AWS Region.



AWS Global Accelerator chooses the optimal AWS Region based on the geography of end clients, which reduces first-byte latency and improves performance.

- **Easy manageability** - The static IP addresses provided by AWS Global Accelerator are fixed and provide a single entry point to your applications. This lets you easily move your endpoints between Availability Zones or between AWS Regions, without having to update your DNS configuration or client-facing applications.
- **Fine-grained control** - AWS Global Accelerator lets you set a traffic dial for your regional endpoint groups, to dial traffic up or down for a specific AWS Region when you conduct performance testing or application updates. In addition, if you have stateful applications, you can choose to direct all requests from a user to the same endpoint, regardless of the source port and protocol, to maintain client affinity. These features give you fine-grained control. By default, the traffic dial is set to 100% for all regional endpoint groups.
- Steps in deploying an accelerator
  1. Create an accelerator
  2. Configure endpoint groups
  3. Register endpoints for endpoint groups
- Global Accelerator vs CloudFront - Both services integrate with AWS Shield for DDoS protection.

	AWS Global Accelerator	Amazon CloudFront
Purpose	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.	CloudFront improves performance for both cacheable content (such as images and videos) and dynamic content (such as API acceleration and dynamic site delivery).
Use Cases	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.	CloudFront is best used for HTTP use cases, as well as securing access over your endpoints (S3, ELB, EC2). CloudFront also offers several options for streaming your media to global viewers—both pre-recorded files and live events. Lastly, using Lambda@Edge can help you configure your CloudFront distribution to serve private content from your own custom origin, as an option to using signed URLs or signed cookies.

- Security
  - SOC, PCI, HIPAA, GDPR, and ISO compliant.
  - AWS Global Accelerator Flow Logs provide detailed records about traffic that flows through an accelerator to an endpoint.
  - As a managed service, AWS Global Accelerator is protected by the AWS global network security procedures.
- Pricing
  - You are charged for each accelerator that is provisioned and the amount of traffic in the dominant direction that flows through the accelerator.
  - For every accelerator that is provisioned (both enabled and disabled), you are charged a fixed hourly fee and an incremental charge over your standard Data Transfer rates per GB, also called a Data Transfer-Premium fee.

#### Sources:

<https://aws.amazon.com/global-accelerator/>  
<https://aws.amazon.com/global-accelerator/faqs/>



## AWS Transit Gateway

- A networking service that uses a hub and spoke model to enable customers to connect their on-premises data centers and their Amazon Virtual Private Clouds (VPCs) to a single gateway.
- With this service, customers only have to create and manage a single connection from the central gateway into each on-premises data center, remote office, or VPC across your network.
- If a new VPC is created, it is automatically connected to the Transit Gateway and will also be available to every other network that is also connected to the Transit Gateway.

### Features:

- **Inter-region peering**
  - Transit Gateway leverages the AWS global network to allow customers to route traffic across AWS Regions.
  - Inter-region peering provides an easy and cost-effective way to replicate data for geographic redundancy or to share resources between AWS Regions.
- **Multicast**
  - Enables customers to have fine-grain control on who can consume and produce multicast traffic.
  - It allows you to easily create and manage multicast groups in the cloud instead of the time-consuming task of deploying and managing legacy hardware on-premises.
  - This multicast solution is also scalable so the customers can simultaneously distribute a stream of content to multiple subscribers.
- **Automated Provisioning**
  - Customers can automatically identify the Site-to-Site VPN connections and the on-premises resources with which they are associated using AWS Transit Gateway.
  - Using the Transit Gateway Network Manager, you can also manually define your on-premises network.

### Source:

<https://aws.amazon.com/transit-gateway/>



## SECURITY AND IDENTITY

### AWS Identity and Access Management (IAM)

- Control who is authenticated (signed in) and authorized (has permissions) to use resources.
- AWS account **root user** is a single sign-in identity that has complete access to all AWS services and resources in the account.
- **Features**
  - You can grant other people permission to administer and use resources in your AWS account without having to share your password or access key.
  - You can grant different permissions to different people for different resources.
  - You can use IAM features to securely provide credentials for applications that run on EC2 instances which provide permissions for your applications to access other AWS resources.
  - You can add two-factor authentication to your account and to individual users for extra security.
  - You can allow users to use **identity federation** to get temporary access to your AWS account.
  - You receive AWS CloudTrail log records that include information about **IAM identities** who made requests for resources in your account.
  - You use an **access key** (an access key ID and secret access key) to make programmatic requests to AWS. An Access Key ID and Secret Access Key can only be uniquely generated once and must be regenerated if lost.
  - IAM has been validated as being compliant with Payment Card Industry (PCI) Data Security Standard (DSS).
  - IAM is *eventually consistent*. IAM achieves high availability by replicating data across multiple servers within Amazon's data centers around the world.
  - IAM and AWS Security Token Service (STS) are offered at no additional charge.
  - Your unique account sign-in page URL:  
[https://My\\_AWS\\_Account\\_ID.signin.aws.amazon.com/console/](https://My_AWS_Account_ID.signin.aws.amazon.com/console/)
  - You can use IAM tags to add custom attributes to an IAM user or role using a tag key-value pair.
- **Infrastructure Elements**
  - **Principal**
    - An entity that can make a request for an action or operation on an AWS resource. Users, roles, federated users, and applications are all AWS principals.
    - Your AWS account root user is your *first principal*.
  - **Request**
    - When a principal tries to use the AWS Management Console, the AWS API, or the AWS CLI, that principal sends a *request* to AWS.
    - Requests includes the following information:
      - **Actions or operations** – the actions or operations that the principal wants to perform.



- **Resources** – the AWS resource object upon which the actions or operations are performed.
- **Principal** – the user, role, federated user, or application that sent the request. Information about the principal includes the policies that are associated with that principal.
- **Environment data** – information about the IP address, user agent, SSL enabled status, or the time of day.
- **Resource data** – data related to the resource that is being requested.
- **Authentication**
  - To authenticate from the console as a user, you must sign in with your user name and password.
  - To authenticate from the API or AWS CLI, you must provide your access key and secret key.
- **Authorization**
  - AWS uses values from the *request context* to check for policies that apply to the request. It then uses the policies to determine whether to allow or deny the request.
  - Policies types can be categorized as *permissions policies* or *permissions boundaries*.
    - *Permissions policies* define the permissions for the object to which they're attached. These include identity-based policies, resource-based policies, and ACLs.
    - *Permissions boundary* is an advanced feature that allows you to use policies to limit the maximum permissions that a principal can have.
  - To provide your users with permissions to access the AWS resources in their own account, you need **identity-based policies**.
  - **Resource-based policies** are for granting cross-account access.
  - Evaluation logic rules for policies:
    - By default, **all requests are denied**.
    - An *explicit allow* in a permissions policy overrides this default.
    - A *permissions boundary* overrides the allow. If there is a permissions boundary that applies, that boundary must allow the request. Otherwise, it is implicitly denied.
    - An explicit deny in any policy overrides any allows.
- **Actions or Operations**
  - Operations are defined by a service, and include things that you can do to a resource, such as viewing, creating, editing, and deleting that resource.
- **Resource**
  - An object that exists within a service. The service defines a set of actions that can be performed on each resource.
- **Users**
  - **IAM Users**



- Instead of sharing your root user credentials with others, you can create individual **IAM users** within your account that correspond to users in your organization. IAM users are not separate accounts; they are users within your account.
- Each user can have its own password for access to the AWS Management Console. You can also create an individual access key for each user so that the user can make programmatic requests to work with resources in your account.
- By default, a brand new IAM user has **NO permissions** to do anything.
- Users are global entities.
- **Federated Users**
  - If the users in your organization already have a way to be authenticated, you can federate those user identities into AWS.
- **IAM Groups**
  - An IAM group is a collection of IAM users.
  - You can organize IAM users into IAM groups and attach access control policies to a group.
  - A user can belong to multiple groups.
  - Groups cannot belong to other groups.
  - Groups do not have security credentials, and cannot access web services directly.
- **IAM Role**
  - A role does not have any credentials associated with it.
  - An IAM user can assume a role to temporarily take on different permissions for a specific task. A role can be assigned to a federated user who signs in by using an external identity provider instead of IAM.
  - **AWS service role** is a role that a service assumes to perform actions in your account on your behalf. This service role must include all the permissions required for the service to access the AWS resources that it needs.
    - **AWS service role for an EC2 instance** is a special type of service role that a service assumes to launch an EC2 instance that runs your application. This role is assigned to the EC2 instance when it is launched.
    - **AWS service-linked role** is a unique type of service role that is linked directly to an AWS service. Service-linked roles are predefined by the service and include all the permissions that the service requires to call other AWS services on your behalf.
    - An instance profile is a container for an IAM role that you can use to pass role information to an EC2 instance when the instance starts.
- Users or groups can have multiple policies attached to them that grant different permissions.



When to Create IAM User	When to Create an IAM Role
You created an AWS account and you're the only person who works in your account.	You're creating an application that runs on an Amazon EC2 instance and that application makes requests to AWS.
Other people in your group need to work in your AWS account, and your group is using no other identity mechanism.	You're creating an app that runs on a mobile phone and that makes requests to AWS.
You want to use the command-line interface to work with AWS.	Users in your company are authenticated in your corporate network and want to be able to use AWS without having to sign in again (federate into AWS)



- Policies

- Most permission policies are JSON policy documents.
- The IAM console includes *policy summary tables* that describe the access level, resources, and conditions that are allowed or denied for each service in a policy.
- The *policy summary table* includes a list of services. Choose a service there to see the *service summary*.
- This *summary table* includes a list of the actions and associated permissions for the chosen service. You can choose an action from that table to view the *action summary*.
- To assign permissions to federated users, you can create an entity referred to as a **role** and define permissions for the **role**.
- **Identity-Based Policies**
  - Permissions policies that you attach to a principal or identity.
  - **Managed policies** are standalone policies that you can attach to multiple users, groups, and roles in your AWS account.
  - **Inline policies** are policies that you create and manage and that are embedded directly into a single user, group, or role.



Policies > PowerUserAccess

## Summary

Policy ARN: arn:aws:iam::aws:policy/PowerUserAccess

Description: Provides full access to AWS services and resources, but does not allow management of Users and groups.

Permissions Policy usage Policy versions Access Advisor

Policy summary

```
1 {  
2   "Version": "2012-10-17",  
3   "Statement": [  
4     {  
5       "Effect": "Allow",  
6       "NotAction": [  
7         "iam:*",  
8         "organizations:*",  
9         "account:*"  
10      ],  
11      "Resource": "*"  
12    },  
13    {  
14      "Effect": "Allow",  
15      "Action": [  
16        "iam>CreateServiceLinkedRole",  
17        "iam>DeleteServiceLinkedRole",  
18        "iam>ListRoles",  
19        "organizations:DescribeOrganization",  
20        "account>ListRegions"  
21      ],  
22      "Resource": "*"  
23    }  
24  ]  
25 }
```

### Resource-based Policies

- Permissions policies that you attach to a resource such as an Amazon S3 bucket.
- Resource-based policies are only inline policies.
- **Trust policies** - resource-based policies that are attached to a role and define which principals can assume the role.



The screenshot shows the AWS Bucket Policy editor interface. At the top, there are tabs for Overview, Properties, Permissions, Management, and Access points. The Management tab is selected, and within it, the Bucket Policy tab is active. Below these tabs, there are four buttons: Block public access, Access Control List, Bucket Policy (which is highlighted in blue), and CORS configuration. A text area titled "Bucket policy editor ARN: arn:aws:s3:::adrianprimeclerk" contains the following JSON policy:

```
1 {
2     "Version": "2012-10-17",
3     "Id": "S3PolicyId1",
4     "Statement": [
5         {
6             "Sid": "IPAllow",
7             "Effect": "Deny",
8             "Principal": "*",
9             "Action": "s3:*",
10            "Resource": "arn:aws:s3:::examplebucket/*",
11            "Condition": {
12                "NotIpAddress": {"aws:SourceIp": "54.240.143.0/24"}
13            }
14        }
15    ]
16 }
17 }
```

Below the text area, a note states: "The block public access settings turned on for this bucket prevent granting public access." To the right of the text area are three buttons: Delete, Cancel, and Save.

- **AWS Security Token Service (STS)**

- Create and provide trusted users with temporary security credentials that can control access to your AWS resources.
- Temporary security credentials are short-term and are not stored with the user but are generated dynamically and provided to the user when requested.
- By default, AWS STS is a global service with a single endpoint at <https://sts.amazonaws.com>.

- **Best Practices**

- Lock Away Your AWS Account Root User Access Keys
- Create Individual IAM Users
- Use Groups to Assign Permissions to IAM Users
- Use AWS Defined Policies to Assign Permissions Whenever Possible
- Grant Least Privilege
- Use Access Levels to Review IAM Permissions
- Configure a Strong Password Policy for Your Users
- Enable MFA for Privileged Users
- Use Roles for Applications That Run on Amazon EC2 Instances
- Use Roles to Delegate Permissions
- Do Not Share Access Keys
- Rotate Credentials Regularly
- Remove Unnecessary Credentials
- Use Policy Conditions for Extra Security
- Monitor Activity in Your AWS Account



**Sources:**

<https://docs.aws.amazon.com/IAM/latest/UserGuide/introduction.html>

<https://aws.amazon.com/iam/faqs/>



## AWS Directory Service

### For Microsoft Active Directory

- Also known as AWS Managed Microsoft AD, the service enables your directory-aware workloads and AWS resources to use managed Active Directory in the AWS Cloud.
- The service is built on actual Microsoft Active Directory and powered by Windows Server 2012 R2.
- AWS Managed Microsoft AD is your best choice if you need actual Active Directory features to support AWS applications or Windows workloads, including Amazon RDS for Microsoft SQL Server. It's also best if you want a standalone AD in the Cloud that supports Office 365 or you need an LDAP directory to support your Linux applications.
- Concepts
  - AWS Managed Microsoft AD provides multiple directory choices for customers who want to use existing Microsoft AD or Lightweight Directory Access Protocol (LDAP)-aware applications in the cloud.
  - When you create a directory, AWS Directory Service creates two domain controllers and adds the DNS service on your behalf. The domain controllers are created in different subnets in a VPC.
  - When creating a directory, you need to provide some basic information such as a fully qualified domain name (FQDN) for your directory, Administrator account name and password, and the VPC you want the directory to be attached to.
  - AWS does not provide Windows PowerShell access to directory instances, and it restricts access to directory objects, roles, and groups that require elevated privileges.
  - AWS Managed Microsoft AD does not allow direct host access to domain controllers via Telnet, Secure Shell (SSH), or Windows Remote Desktop Connection.
  - When you create an AWS Managed Microsoft AD directory, you are assigned an organizational unit (OU) and an administrative account with delegated administrative rights for the OU.
  - AWS Managed Microsoft AD directories are deployed across two Availability Zones in a region by default and connected to your Amazon VPC.
  - You cannot configure the storage, CPU, or memory parameters of your AWS Managed Microsoft AD directory.
- Active Directory Schema
  - A schema is the definition of attributes and classes that are part of a distributed directory and is similar to fields and tables in a database. Schemas include a set of rules which determine the type and format of data that can be added or included in the database.
  - Attributes, classes and objects are the basic elements that are used to build object definitions in the schema.
    - Each schema attribute, which is similar to a field in a database, has several properties that define the characteristics of the attribute.
    - The classes are analogous to tables in a database and also have several properties to be defined.
    - Each class and attribute must have an Object ID that is unique for all of your objects. Software vendors must obtain their own Object ID to ensure uniqueness.



- Some attributes are linked between two classes with forward and back links, such as groups. A group shows you the members of the group; while a member shows what groups it belongs to.
- Features
  - AWS Managed Microsoft AD is deployed in HA and across multiple Availability Zones. You can also scale out your directory by deploying additional domain controllers.
  - AWS Managed Microsoft AD runs on AWS managed infrastructure with monitoring that automatically detects and replaces domain controllers that fail.
  - Data replication and automated daily snapshots are configured for you.
  - You can integrate AWS Managed Microsoft AD easily with your existing Active Directory by using Active Directory trust relationships.
  - Allows seamless domain join for new and existing Amazon EC2 for Windows Server instances.
  - AWS Managed Microsoft AD can also provide a single directory for all kinds of workloads (EC2, RDS, WorkSpaces, etc.).
  - The service supports schema extensions that you submit to the service in the form of a LDAP Data Interchange Format (LDIF) file.
  - You can configure Amazon SNS to receive email and text messages when the status of your AWS Directory Service changes.
  - You can configure SAML 2.0-based authentication with cloud applications using AWS Directory Service.
  - You can use AWS Managed Microsoft AD as a resource forest that contains primarily computers and groups with trust relationships to your on-premises directory. This enables your users to access AWS applications and resources with their on-premises AD credentials.
- Microsoft AD Prerequisites
  - A VPC with at least two subnets. Each of the subnets must be in a different Availability Zone.
  - The necessary ports for the domain controllers that AWS Directory Service creates for you should be open to allow them to communicate with each other.
  - The VPC must have default hardware tenancy.
  - AWS Directory Service does not support using NAT with Active Directory.
- Two Editions of AWS Managed Microsoft AD
  - Both Standard Edition and Enterprise Edition can be used as your organization's primary directory to manage users, devices, and computers.
  - You also can use both editions to create resource forests and extend your on-premises AD to the AWS Cloud. Resource forests use a trust relationship with your on-premises AD to enable you to access AWS applications and resources with your on-premises AD credentials.
  - Both editions also support the creation of additional domain controllers to improve the redundancy and performance of your managed directory.
  - Unique to Standard Edition
    - Optimized to be a primary directory for small and midsize businesses with up to 5,000 employees.



- Provides you enough storage capacity to support up to approximately 30,000 directory objects, such as users, groups, and computers.
- Unique to Enterprise Edition
  - Designed to support enterprise organizations with up to approximately 500,000 directory objects.
- Seamless Domain Joins
  - Seamless domain join is a feature that allows you to join your Amazon EC2 for Windows Server instances seamlessly to a domain, at the time of launch and from the AWS Management Console. You can join instances to AWS Managed Microsoft AD that you launch in the AWS Cloud.
  - You cannot use the seamless domain join feature from the AWS Management Console for existing EC2 for Windows Server instances, but you can join existing instances to a domain using the EC2 API or by using PowerShell on the instance.
- Security and Monitoring
  - AWS Managed Microsoft AD is both HIPAA and PCI DSS compliant.
  - Manage users and devices by using native Active Directory Group Policy objects (GPOs).
  - AWS Managed Microsoft AD uses the same Kerberos-based authentication as Active Directory to deliver Single Sign-On (SSO).
  - AWS Managed Microsoft AD supports federation access for users and groups to the AWS Management Console.
  - Amazon EBS volumes used in the directory service are encrypted.
- Pricing
  - You pay only for the type and size of the managed directory that you use.
  - AWS Managed Microsoft AD allows you to use a directory in one account and share it with multiple accounts and VPCs. There is an hourly sharing charge for each additional account to which you share a directory.

### Active Directory Connector

- A proxy service that provides an easy way to connect compatible AWS applications, such as Amazon WorkSpaces, Amazon QuickSight, and Amazon EC2 for Windows Server instances, to your existing on-premises Microsoft Active Directory.
- AD Connector is your best choice when you want to use your existing on-premises directory with compatible AWS services.
- Features
  - When users log in to the AWS applications, AD Connector forwards sign-in requests to your on-premises Active Directory domain controllers for authentication.
  - You can also join your EC2 Windows instances to your on-premises Active Directory domain through AD Connector using seamless domain join.
  - AD Connector is NOT compatible with RDS SQL Server.
  - AD Connector comes in two sizes, small and large.



- You can spread application loads across multiple AD Connectors to scale to your performance needs. There are no enforced user or connection limits.
- AD Connector Prerequisites
  - You need to have a VPC with at least two subnets. Each of the subnets must be in a different Availability Zone.
  - The VPC must be connected to your existing network through a virtual private network (VPN) connection or AWS Direct Connect.
  - The VPC must have default hardware tenancy.
  - Your user accounts must have Kerberos pre-authentication enabled.

## Simple AD

- A standalone Microsoft Active Directory–compatible directory from AWS Directory Service that is powered by Samba 4.
- You can use Simple AD as a standalone directory in the cloud to support Windows workloads that need basic AD features, compatible AWS applications, or to support Linux workloads that need LDAP service.
- Features
  - Simple AD supports basic Active Directory features such as user accounts, group memberships, joining a Linux domain or Windows based EC2 instances, Kerberos-based SSO, and group policies.
  - AWS provides monitoring, daily snapshots, and recovery as part of the service.
  - Simple AD is compatible with the following AWS applications: Amazon WorkSpaces, Amazon WorkDocs, Amazon QuickSight, and Amazon WorkMail.
  - You can also sign in to the AWS Management Console with Simple AD user accounts.
  - Simple AD does NOT support multi-factor authentication, trust relationships, DNS dynamic update, schema extensions, communication over LDAPS, PowerShell AD cmdlets, or FSMO role transfer.
  - Simple AD is NOT compatible with RDS SQL Server.
  - Simple AD is available in two sizes:
    - Small - Supports up to 500 users
    - Large - Supports up to 5,000 users
- Simple AD Prerequisites
  - Your VPC should have at least two subnets. For Simple AD to install correctly, you must install your two domain controllers in separate subnets that must be in a different Availability Zone. In addition, the subnets must be in the same Classless Inter-Domain Routing (CIDR) range.
  - The necessary ports for the domain controllers that AWS Directory Service creates for you should be open to allow them to communicate with each other.
  - The VPC must have default hardware tenancy.
- When you create a directory with Simple AD, AWS Directory Service performs the following tasks on your behalf:
  - Sets up a Samba-based directory within the VPC.



- Creates a directory administrator account with the user name 'Administrator' and the specified password. You use this account to manage your directory.
- Creates a security group for the directory controllers.
- Creates an account that has domain admin privileges.
- Simple AD forwards DNS requests to the IP address of the Amazon-provided DNS servers for your VPC. These DNS servers will resolve names configured in your Route 53 private hosted zones

### Amazon Cloud Directory

- A cloud-native directory that can store hundreds of millions of application-specific objects with multiple relationships and schemas. Use Amazon Cloud Directory if you need a highly scalable directory store for your application's hierarchical data.
- You can organize directory objects into multiple hierarchies to support many organizational pivots and relationships across directory information.
- Concepts
  - A schema is a collection of facets that define what objects can be created in a directory and how they are organized.
  - A schema also enforces data integrity and interoperability.
  - A single schema can be applied to more than one directory at a time.
  - Amazon Cloud Directory supports uploading of a compliant JSON file for schema creation.
  - A directory is a schema-based data store that contains specific types of objects organized in a multi-hierarchical structure.
  - Before you can create a directory in Amazon Cloud Directory, AWS Directory Service requires that you first apply a schema to it. A directory cannot be created without a schema and typically has one schema applied to it.

### Amazon Cognito

- A user directory that adds sign-up and sign-in to your mobile app or web application using Amazon Cognito User Pools.
- Amazon Cognito supports sign-in with social identity providers, such as Facebook, Google, and Amazon, and enterprise identity providers via SAML 2.0.

### Sources:

<https://aws.amazon.com/directoryservice/features/?nc=sn&loc=2>

<https://aws.amazon.com/directoryservice/pricing/?nc=sn&loc=3>

<https://aws.amazon.com/directoryservice/faqs/?nc=sn&loc=5>

[https://docs.aws.amazon.com/clouddirectory/latest/developerguide/what\\_is\\_cloud\\_directory.html](https://docs.aws.amazon.com/clouddirectory/latest/developerguide/what_is_cloud_directory.html)

[https://docs.aws.amazon.com/directoryservice/latest/admin-guide/what\\_is.html](https://docs.aws.amazon.com/directoryservice/latest/admin-guide/what_is.html)

[https://docs.aws.amazon.com/clouddirectory/latest/developerguide/what\\_is\\_cloud\\_directory.html](https://docs.aws.amazon.com/clouddirectory/latest/developerguide/what_is_cloud_directory.html)

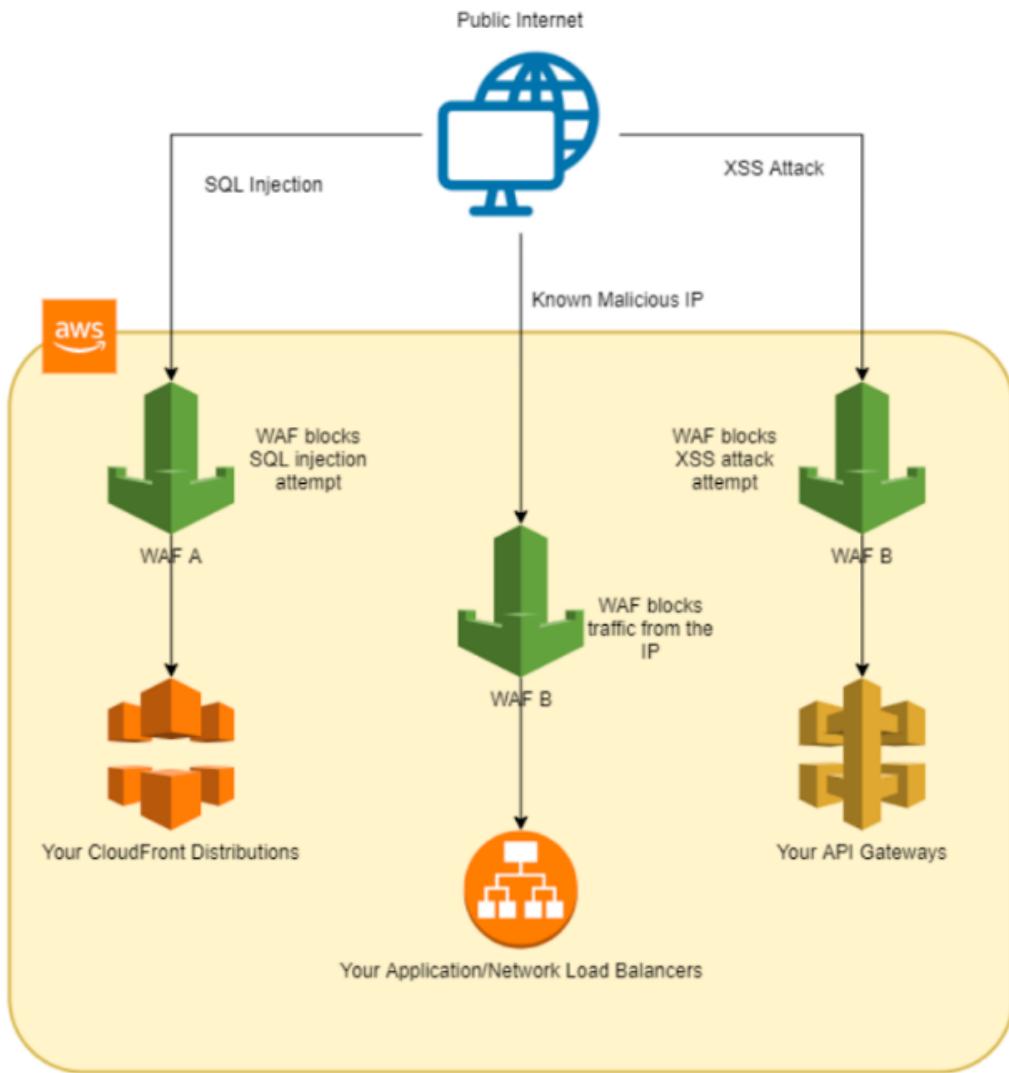


## AWS WAF

- A web application firewall that helps protect web applications from attacks by allowing you to configure rules that **allow, block, or monitor (count) web requests** based on conditions that you define.
- These conditions include:
  - IP addresses
  - HTTP headers
  - HTTP body
  - URI strings
  - SQL injection
  - cross-site scripting.

## Features

- WAF lets you create rules to filter web traffic based on conditions that include IP addresses, HTTP headers and body, or custom URIs.
- You can also create rules that block common web exploits like SQL injection and cross site scripting.
- For application layer attacks, you can use WAF to respond to incidents. You can set up proactive rules like *Rate Based Blacklisting* to automatically block bad traffic, or respond immediately to incidents as they happen.
- WAF provides real-time metrics and captures raw requests that include details about IP addresses, geo locations, URIs, User-Agent and Referers.
- **AWS WAF Security Automations** is a solution that automatically deploys a single web access control list (web ACL) with a set of AWS WAF rules designed to filter common web-based attacks. The solution supports log analysis using Amazon Athena and AWS WAF full logs.



### Conditions, Rules, and Web ACLs

- You define your conditions, combine your conditions into rules, and combine the rules into a web ACL.
- **Conditions** define the basic characteristics that you want WAF to watch for in web requests.
- You combine conditions into **rules** to precisely target the requests that you want to allow, block, or count. WAF provides two types of rules:
  - **Regular rules** - use only conditions to target specific requests.
  - **Rate-based rules** - are similar to regular rules, with a rate limit. Rate-based rules count the requests that arrive from a specified IP address every five minutes. The rule can trigger an action if the number of requests exceed the rate limit.
- **WAF Managed Rules** are an easy way to deploy pre-configured rules to protect your applications common threats like application vulnerabilities. All Managed Rules are automatically updated by AWS Marketplace security Sellers.



- After you combine your conditions into rules, you combine the rules into a **web ACL**. This is where you define an action for each rule—allow, block, or count—and a default action, which determines whether to allow or block a request that doesn't match all the conditions in any of the rules in the web ACL.

## Pricing

- WAF charges based on the number of web access control lists (web ACLs) that you create, the number of rules that you add per web ACL, and the number of web requests that you receive.

### Sources:

<https://docs.aws.amazon.com/waf/latest/developerguide>

<https://aws.amazon.com/waf/features/>

<https://aws.amazon.com/waf/pricing/>

<https://aws.amazon.com/waf/faqs/>



## Amazon Macie

- A security service that uses machine learning to automatically discover, classify, and protect sensitive data in AWS. Macie recognizes sensitive data such as personally identifiable information (PII) or intellectual property.
- Amazon Macie is currently only available in N. Virginia and Oregon region.
- Amazon Macie allows you to achieve the following:
  - Identify and protect various data types, including PII, PHI, regulatory documents, API keys, and secret keys
  - Verify compliance with automated logs that allow for instant auditing
  - Identify changes to policies and access control lists
  - Observe changes in user behavior and receive actionable alerts
  - Receive notifications when data and account credentials leave protected zones
  - Detect when large quantities of business-critical documents are shared internally and externally
- Concepts
  - An **Alert** is a notification about a potential security issue that Macie discovers. Alerts appear on the Macie console and provide a comprehensive narrative about all activity that occurred over the last 24 hours.
    - Basic alerts – Alerts that are generated by the security checks that Macie performs. There are two types of basic alerts in Macie:
      - Managed (curated by Macie) basic alerts that you can't modify. You can only enable or disable the existing managed basic alerts.
      - Custom basic alerts that you can create and modify to your exact specifications.
    - Predictive alerts – Automatic alerts based on activity in your AWS infrastructure that deviates from the established normal activity baseline. More specifically, Macie continuously monitors IAM user and role activity in your AWS infrastructure and builds a model of the normal behavior. It then looks for deviations from that normal baseline, and when it detects such activity, it generates automatic predictive alerts.
  - **Data source** is the origin or location of a set of data.
    - AWS CloudTrail event logs and errors, including Amazon S3 object-level API activity. You can't modify existing or add new CloudTrail events to the list that Macie manages. You can enable or disable the supported CloudTrail events, thus instructing Macie to either include or exclude them in its data security process.
    - Amazon S3 objects. You can integrate Macie with your S3 buckets and/or specify S3 prefixes
  - **User**, in the context of Macie, a user is the AWS Identity and Access Management (IAM) identity that makes the request.
- There are certain file formats that Macie does not support, such as wav files.
- Once Macie begins monitoring your data, it uses several **automatic content classification methods** to identify and prioritize your sensitive and critical data and to accurately assign business value to your



data. Each classification has a designated risk level between 1 and 10, with 10 being the highest risk and 1 being the lowest. These methods include:

- **Content Type Classification** - Macie uses an identifier that is embedded in the file header of your data objects. Macie can assign only one content type to an object. You can't modify existing or add new content types. You can only enable or disable any existing content types, thus enabling or disabling Macie to assign them to your objects during the classification process.
  - **File Extension Classification** - Macie offers a set of managed file extensions. Macie can assign only one file extension to an object. You can't modify existing or add new file extensions. You can enable or disable any existing file extensions, thus enabling or disabling Macie to assign them to your objects during the classification process.
  - **Theme Classification** - Object classification by theme is based on keywords that Macie searches for as it examines the contents of data objects. Macie can assign one or more themes to an object. You can't modify existing or add new themes. You can enable or disable any existing themes, thus enabling or disabling Macie to assign them to your objects during the classification process.
  - **Regex Classification** - Macie offers a set of managed regexes. Object classification by regex is based on specific data or data patterns that Macie searches for as it examines the contents of data objects. Macie can assign one or more regexes to an object. You can't modify existing or add new regexes. You can enable or disable any existing regexes, thus enabling or disabling Macie to assign them to your objects during the classification process.
  - **PII Classification** - Object classification by personally identifiable information (PII) is based on recognizing any personally identifiable artifacts based on industry standards such as NIST-80-122 and FIPS 199.
  - **Support Vector Machine-Based Classifier** - It classifies content inside your S3 objects (text, token n-grams, and character n-grams) that Macie monitors and their metadata features (document length, extension, encoding, headers) to accurately classify documents based on content.
- You can use the **Research** tab in the Macie console to construct and run queries in the query parser and conduct in-depth investigative research of your data and activity that Macie monitors.
  - If you disable Macie, the following actions occur:
    - It no longer has access to the resources in the master account and all member accounts. You must add member accounts again if you decide to reenable Macie.
    - It stops processing the resources in the master account and all member accounts. After Macie is disabled, the metadata that Macie collected while monitoring the data in your master and member accounts is deleted. Within 90 days from disabling Macie, all of this metadata is expired from the Macie system backups.
  - Pricing
    - You are charged based on the amount of content classified, and the amount of AWS CloudTrail events assessed by Amazon Macie for anomalies (both Management API activity and Amazon S3 object-level API activity).



- Amazon Macie stores the generated metadata of classified S3 objects for 30 days at no additional cost. Additional monthly fees will be incurred if you choose the optional Extended Data Retention feature.

**Sources:**

<https://aws.amazon.com/macie/>

<https://docs.aws.amazon.com/macie/latest/userguide/what-is-macie.html>

<https://aws.amazon.com/macie/faq/>

<https://www.youtube.com/watch?v=LCjX2rsQ2wA>



## AWS Secrets Manager

- A secret management service that enables you to easily rotate, manage, and retrieve database credentials, API keys, and other secrets throughout their lifecycle.
- Features
  - AWS Secrets Manager encrypts secrets at rest using encryption keys that you own and store in AWS Key Management Service [customer managed keys]. When you retrieve a secret, Secrets Manager decrypts the secret and transmits it securely over TLS to your local environment.
  - You can rotate secrets on a schedule or on demand by using the Secrets Manager console, AWS SDK, or AWS CLI.
  - Secrets Manager natively supports rotating credentials for databases hosted on Amazon RDS and Amazon DocumentDB and clusters hosted on Amazon Redshift.
  - You can extend Secrets Manager to rotate other secrets, such as credentials for Oracle databases hosted on EC2 or OAuth refresh tokens, by using custom AWS Lambda functions.
- A secret consists of a set of credentials (user name and password), and the connection details used to access a secured service.
- A secret also contains **metadata** which include:
  - Basic information includes the name of the secret, a description, and the Amazon Resource Name (ARN) to serve as a unique identifier.
  - The ARN of the AWS KMS key Secrets Manager uses to encrypt and decrypt the protected text in the secret. If you don't provide this information, Secrets Manager uses the default AWS KMS key for the account.
  - Information about how frequently to rotate the key and what Lambda function to use to perform the rotation.
  - A user-provided set of tags. You can attach tags as key-value pairs to AWS resources for organizing, logical grouping, and cost allocation.
- A secret can contain **versions**:
  - Although you typically only have one version of the secret active at a time, multiple versions can exist while you rotate a secret on the database or service. Whenever you change the secret, Secrets Manager creates a new version.
  - Each version holds a copy of the encrypted secret value.
  - Each version can have one or more *staging labels* attached identifying the stage of the secret rotation cycle.
- Supported Secrets
  - Database credentials, on-premises resource credentials, SaaS application credentials, third-party API keys, and SSH keys.
  - You can also store JSON documents.
- To retrieve secrets, you simply replace secrets in plain text in your applications with code to pull in those secrets programmatically using the Secrets Manager APIs.
- Secrets can be cached on the client side, and updated only during a secret rotation.



- During the secret rotation process, Secrets Manager tracks the older credentials, as well as the new credentials you want to start using, until the rotation completes. It tracks these different versions by using *staging labels*.
- How Secret Rotation Works
  - The rotation function contacts the secured service authentication system and creates a new set of credentials to access the database. Secrets Manager stores these new credentials as the secret text in a new version of the secret with the AWSPENDING staging label attached.
  - The rotation function then tests the AWSPENDING version of the secret to ensure that the credentials work, and grants the required level of access to the secured service.
  - If the tests succeed, the rotation function then moves the label AWSCURRENT to the new version to mark it as the default version. Then, all of the clients start using this version of the secret instead of the old version. The function also assigns the label AWSPREVIOUS to the old version. The version that had AWSPREVIOUS staging label now has no label, and therefore deprecated.
- Network Setup for Secret Rotation
  - When rotating secrets on natively supported services, Secrets Manager uses CloudFormation to build the rotation function and configure the network connection between the two.
    - If your protected database service **runs in a VPC and is not publicly accessible**, then the CloudFormation template configures the **Lambda rotation function to run in the same VPC**. The rotation function can communicate with the protected service **directly within the VPC**.
    - If you run your protected service as a **publicly accessible resource**, in a VPC or not, then the CloudFormation template configures the **Lambda rotation function not to run in a VPC**. The Lambda rotation function communicates with the protected service **through the publicly accessible connection point**.
  - By default, the Secrets Manager endpoints run on the public Internet. If you run your Lambda rotation function and protected database or service in a VPC, then you must perform one of the following steps:
    - **Add a NAT gateway to your VPC**. This enables traffic that originates in your VPC to reach the public Secrets Manager endpoint.
    - **Configure Secrets Manager service endpoints directly within your VPC**. This configures your VPC to intercept any request addressed to the public regional endpoint, and redirect the request to the private service endpoint running within your VPC.
- You can create two secrets that have different permissions
  - User Secret - can be used to connect to linked services, but it cannot be rotated. The user will have to wait for the master secret to be rotated and propagated for it to change.
  - Master Secret - has sufficient permissions to rotate secrets of linked services. This scenario is typically used when you have users that are actively using the old secret, and you do not want to break operations after you rotate the secret. You can have your users update their clients first before using the newly rotated credentials.
- Security



- By default, Secrets Manager does not write or cache the secret to persistent storage.
- By default, Secrets Manager only accepts requests from hosts that use the open standard Transport Layer Security (TLS) and Perfect Forward Secrecy.
- You can control access to the secret using AWS Identity and Access Management (IAM) policies.
- You can tag secrets individually and apply tag-based access controls.
- You can configure VPC endpoints to keep traffic between your VPC and Secrets Manager within the AWS network.
- Secrets Manager does not immediately delete secrets. Instead, Secrets Manager immediately makes the secrets inaccessible and scheduled for deletion after a recovery window of a **minimum of seven days**. Until the recovery window ends, you can recover a secret you previously deleted.
- By using the CLI, you can delete a secret without a recovery window.
- Compliance
  - Secrets Manager is HIPAA, PCI DSS and ISO compliant.
- Pricing
  - You pay based on the number of secrets stored and API calls made per month.

**Sources:**

<https://aws.amazon.com/secrets-manager/>

<https://aws.amazon.com/secrets-manager/faqs/>

<https://docs.aws.amazon.com/secretsmanager/latest/userguide/intro.html>

[Understanding AWS Secrets Manager](#)



## Amazon GuardDuty

- An intelligent threat detection service. It analyzes billions of events across your AWS accounts from AWS CloudTrail (AWS user and API activity in your accounts), Amazon VPC Flow Logs (network traffic data), and DNS Logs (name query patterns).
- GuardDuty is a regional service.
- Threat detection categories
  - **Reconnaissance** – Activity suggesting reconnaissance by an attacker, such as unusual API activity, intra-VPC port scanning, unusual patterns of failed login requests, or unblocked port probing from a known bad IP.
  - **Instance compromise** – Activity indicating an instance compromise, such as cryptocurrency mining, backdoor command and control activity, malware using domain generation algorithms, outbound denial of service activity, unusually high volume of network traffic, unusual network protocols, outbound instance communication with a known malicious IP, temporary Amazon EC2 credentials used by an external IP address, and data exfiltration using DNS.
  - **Account compromise** – Common patterns indicative of account compromise include API calls from an unusual geolocation or anonymizing proxy, attempts to disable AWS CloudTrail logging, changes that weaken the account password policy, unusual instance or infrastructure launches, infrastructure deployments in an unusual region, and API calls from known malicious IP addresses.
- Amazon GuardDuty provides three severity levels (Low, Medium, and High) to allow you to prioritize response to potential threats.
- CloudTrail Event Source
  - Currently, GuardDuty only analyzes CloudTrail management events. (Read about types of CloudTrail trails for more information)
  - GuardDuty processes all CloudTrail events that come into a region, including global events that CloudTrail sends to all regions, such as AWS IAM, AWS STS, Amazon CloudFront, and Route 53.
- VPC Flow Logs Event Source
  - VPC Flow Logs capture information about the IP traffic going to and from Amazon EC2 network interfaces in your VPC.
- DNS Logs Event Source
  - If you use AWS DNS resolvers for your EC2 instances (the default setting), then GuardDuty can access and process your request and response DNS logs through the internal AWS DNS resolvers. Using other DNS resolvers will not provide GuardDuty access to its DNS logs.
- GuardDuty vs Macie
  - Amazon GuardDuty provides broad protection of your AWS accounts, workloads, and data by helping to identify threats such as attacker reconnaissance, instance compromise, and account compromise. Amazon Macie helps you protect your data in Amazon S3 by helping you classify what data you have, the value that data has to the business, and the behavior associated with access to that data.



- GuardDuty Findings
  - GuardDuty generates **findings** when it detects unexpected and potentially malicious activity in your AWS environment. These are viewable via Console, GuardDuty CLI or API operations.
  - A Finding's summary includes:
    - **Finding type** – a concise yet readable description of the potential security issue.
    - **Severity** – a finding's assigned severity level of either High, Medium, or Low.
    - **Region** – the AWS region in which the finding was generated.
    - **Count** – the number of times GuardDuty generated the finding after you enabled GuardDuty in your AWS account.
    - **Account ID** – the ID of the AWS account in which the activity took place that prompted GuardDuty to generate this finding.
    - **Resource ID** – the ID of the AWS resource against which the activity took place that prompted GuardDuty to generate this finding.
    - **Threat list name** - the name of the threat list that includes the IP address or the domain name involved in the activity that prompted GuardDuty to generate the finding.
    - **Last seen** – the time (your local timezone if checked through console, and UTC if checked through CLI or API) at which the activity took place that prompted GuardDuty to generate this finding.
  - A finding's **Resource affected** section includes:
    - **Resource role** – a value that usually is set to **Target** because the affected resource can be a potential target of an attack.
    - **Resource type** – the type of the affected resource. This value is either **AccessKey** or **Instance**.
    - **Instance ID** – the ID of the EC2 instance involved in the activity that prompted GuardDuty to generate the finding.
    - **Port** – the port number for the connection used during the activity that prompted GuardDuty to generate the finding.
    - **Access key ID** – access key ID of the user engaged in the activity that prompted GuardDuty to generate the finding.
    - **Principal ID** – the principal ID of the user engaged in the activity that prompted GuardDuty to generate the finding.
    - **User type** – the type of user engaged in the activity that prompted GuardDuty to generate the finding.
    - **User name** – The name of the user engaged in the activity that prompted GuardDuty to generate the finding.
  - A finding's **Action** section includes:
    - **Action type** – the finding activity type. This value can be one of the following: NETWORK\_CONNECTION, AWS\_API\_CALL, PORT\_PROBE, or DNS\_REQUEST.
    - **API** – the name of the API operation that was invoked and thus prompted GuardDuty to generate this finding.
    - **Service name** – the name of the AWS service (GuardDuty) that generated the finding.



- **Connection direction** – the network connection direction observed in the activity that prompted GuardDuty to generate the finding. The values can be INBOUND, OUTBOUND, and UNKNOWN.
- **Protocol** – the network connection protocol observed in the activity that prompted GuardDuty to generate the finding.
- A finding's **Actor** section includes:
  - **Location** – location information of the IP address involved in the activity that prompted GuardDuty to generate the finding.
  - **Organization** – ISP organization information of the IP address involved in the activity that prompted GuardDuty to generate the finding.
  - **IP address** – the IP address involved in the activity that prompted GuardDuty to generate the finding.
  - **Port** – the port number involved in the activity that prompted GuardDuty to generate the finding.
  - **Domain** – the domain involved in the activity that prompted GuardDuty to generate the finding.
- A finding's **Details** section includes:
  - **ThreatPurpose** - describes the primary purpose of a threat or a potential attack. Can have the following values:
    - **Backdoor** - this value indicates that the attack has compromised an AWS resource and is capable of contacting its home command and control (C&C) server to receive further instructions for malicious activity.
    - **Behavior** - this value indicates that GuardDuty is detecting activity or activity patterns that are different from the established baseline for a particular AWS resource.
    - **Cryptocurrency** - this value indicates that GuardDuty is detecting software that is associated with cryptocurrencies.
    - **Pentest** - sometimes owners of AWS resources or their authorized representatives intentionally run tests against AWS applications to find vulnerabilities, like open security groups or access keys that are overly permissive. These pen tests are done in an attempt to identify and lock down vulnerable resources before they are discovered by attackers.
    - **Persistence** - this value indicates that a principal in your AWS environment is exhibiting behavior that is different from the established baseline. Such as a principal has no prior history of updating network configuration settings, or updating policies or permissions attached to AWS users or resources.
    - **Policy** - this value indicates that your AWS account is exhibiting behavior that goes against recommended security best practices.
    - **PrivilegeEscalation** - this value informs you that a specific principal in your AWS environment is exhibiting behavior that can be indicative of a privilege escalation attack.



- **Recon** - this value indicates that a reconnaissance attack is underway, scoping out vulnerabilities in your AWS environment by probing ports, listing users, database tables, and so on.
- **ResourceConsumption** - this value indicates that a principal in your AWS environment is exhibiting behavior that is different from the established baseline. Such as a principal has no prior history of launching EC2 instances.
- **Stealth** - this value indicates that an attack is actively trying to hide its actions and its tracks.
- **Trojan** - this value indicates that an attack is using Trojan programs that silently carry out malicious activity. Sometimes this software takes on an appearance of a legitimate program. Sometimes users accidentally run this software. Other times this software might run automatically by exploiting a vulnerability.
- **UnauthorizedAccess** - this value indicates that GuardDuty is detecting suspicious activity or a suspicious activity pattern by an unauthorized individual.
- **ResourceTypeAffected** - describes which AWS resource is identified in this finding as the potential target of an attack. Currently, only EC2 instances and principals (and their credentials) can be identified as affected resources in GuardDuty findings.
- **ThreatFamilyName** - describes the overall threat or potential malicious activity that GuardDuty is detecting.
- **ThreatFamilyVariant** - describes the specific variant of the **ThreatFamily** that GuardDuty is detecting. Attackers often slightly modify the functionality of the attack, thus creating new variants.
- **Artifact** - describes a specific resource that is owned by a tool that is used in the attack.
- You can create filters for your GuardDuty findings.
  - A *suppression rule* is a filter used to automatically archive new findings. After you create a suppression rule, new findings that match the criteria defined in the rule are automatically archived.
- GuardDuty supports exporting active findings to CloudWatch Events and, optionally, to an Amazon S3 bucket. New Active findings that GuardDuty generates are automatically exported within about 5 minutes after the finding is generated.
- Trusted IP Lists and Threat Lists
  - **Trusted IP lists** consist of IP addresses that you have **whitelisted for secure communication** with your AWS infrastructure and applications. GuardDuty does not generate findings for IP addresses on trusted IP lists.
  - At any given time, you can have only one uploaded trusted IP list per AWS account per region.
  - **Threat lists** consist of known **malicious** IP addresses. GuardDuty generates findings based on threat lists.
  - At any given time, you can have up to six uploaded threat lists per AWS account per region.
- Pricing



Pricing is based on the quantity of AWS CloudTrail Events analyzed (per 1,000,000 events) and the volume of Amazon VPC Flow Log and DNS Log data analyzed (per GB).

**Sources:**

<https://aws.amazon.com/guardduty/>

<https://aws.amazon.com/guardduty/faqs/>

<https://docs.aws.amazon.com/guardduty/latest/ug/what-is-guardduty.html>

[https://www.youtube.com/watch?time\\_continue=7&v=o2Yalsps5LY](https://www.youtube.com/watch?time_continue=7&v=o2Yalsps5LY)



## AWS Shield

- A managed Distributed Denial of Service (DDoS) protection service that safeguards applications running on AWS.

### Shield Tiers and Features

#### Standard

- All AWS customers benefit from the automatic protections of Shield Standard.
- Shield Standard provides always-on network flow monitoring which inspects incoming traffic to AWS and detect malicious traffic in real-time.
- Uses several techniques like deterministic packet filtering, and priority based traffic shaping to automatically mitigate attacks without impact to your applications.
- When you use Shield Standard with CloudFront and Route 53, you receive comprehensive availability protection against all known infrastructure attacks.

#### Advanced

- Shield Advanced provides enhanced detection, inspecting network flows and also monitoring application layer traffic to your Elastic IP address, Elastic Load Balancing, CloudFront, or Route 53 resources.
- It handle the majority of DDoS protection and mitigation responsibilities for **layer 3, layer 4, and layer 7** attacks.
- You have 24x7 access to the AWS DDoS Response Team. To contact the DDoS Response Team, customers will need the Enterprise or Business Support levels of AWS Premium Support.
- It automatically provides additional mitigation capacity to protect against larger DDoS attacks. The DDoS Response Team also applies manual mitigations for more complex and sophisticated DDoS attacks.
- It gives you complete visibility into DDoS attacks with near real-time notification via CloudWatch and detailed diagnostics on the "AWS WAF and AWS Shield" Management Console.
- Shield Advanced comes with "DDoS cost protection", a safeguard from scaling charges as a result of a DDoS attack that cause usage spikes on your AWS services. It does so by providing service credits for charges due to usage spikes.
- It is available globally on all CloudFront and Route 53 edge locations. You can also enable Shield Advanced directly on an Elastic IP or ELB in the following AWS Regions - Northern Virginia, Oregon, Ireland, Tokyo, and Northern California.
- With Shield Advanced you will be able to see the history of all incidents in the trailing 13 months.

## Pricing

- **Shield Standard** provides protection at no additional charge.



- **Shield Advanced**, however, is a paid service. It requires a 1-year subscription commitment and charges a monthly fee, plus a usage fee based on data transfer out from CloudFront, ELB, EC2, and AWS Global Accelerator.

**Sources:**

<https://aws.amazon.com/shield/features/>

<https://aws.amazon.com/shield/pricing/>

<https://aws.amazon.com/shield/faqs/>



## Amazon Inspector

- An automated security assessment service that helps you test the network accessibility of your EC2 instances and the security state of your applications running on the instances.
- Inspector uses IAM service-linked roles.

## Features

- Inspector provides an engine that analyzes system and resource configuration and monitors activity to determine what an assessment target looks like, how it behaves, and its dependent components. The combination of this telemetry provides a complete picture of the assessment target and its potential security or compliance issues.
- Inspector incorporates a built-in library of rules and reports. These include checks against best practices, common compliance standards and vulnerabilities.
- Automate security vulnerability assessments throughout your development and deployment pipeline or against static production systems.
- Inspector is an API-driven service that uses an optional agent, making it easy to deploy, manage, and automate.

## Concepts

- **Inspector Agent** - A software agent that you can install on all EC2 instances that are included in the assessment target, the security of which you want to evaluate with Inspector.
- **Assessment run** - The process of discovering potential security issues through the analysis of your assessment target's configuration and behavior against specified rules packages.
- **Assessment target** - A collection of AWS resources that work together as a unit to help you accomplish your business goals. Inspector assessment targets can consist only of EC2 instances.
- **Assessment template** - A configuration that is used during your assessment run, which includes
  - Rules packages against which you want Inspector to evaluate your assessment target,
  - The duration of the assessment run,
  - Amazon SNS topics to which you want Inspector to send notifications about assessment run states and findings,
  - Inspector-specific attributes (key-value pairs) that you can assign to findings generated by the assessment run that uses this assessment template.
  - After you create an assessment template, you can't modify it.
- **Finding** - A potential security issue discovered during the assessment run of the specified target.
- **Rule** - A security check performed during an assessment run. When a rule detects a potential security issue, Inspector generates a finding that describes the issue.
- **Rules package** - A collection of rules that corresponds to a security goal that you might have.
- **Telemetry** - EC2 instance data collected by Inspector during an assessment run and passed to the Inspector service for analysis.



- The telemetry data generated by the Inspector Agent during assessment runs is formatted in JSON files and delivered in near-real-time over TLS to Inspector, where it is encrypted with a per-assessment-run, ephemeral KMS-derived key and securely stored in an S3 bucket dedicated for the service.

## Rules Packages and Rules

- Inspector compares the behavior and the security configuration of the assessment targets to selected security *rules packages*.
- *Rules* are grouped together into distinct rules packages either by category, severity, or pricing.
- Each rule has an assigned severity level
  - **High, Medium, and Low** levels all indicate a security issue that can result in compromised information confidentiality, integrity, and availability within your assessment target.
  - The **Informational** level simply highlights a security configuration detail of your assessment target.
- The findings generated by **rules in the Network Reachability package** show whether your ports are reachable from the internet through an internet gateway, a VPC peering connection, or a VPN through a virtual gateway. These findings also highlight network configurations that allow for potentially malicious access, such as mismanaged security groups, ACLs, IGWs, and so on.

## Assessment Reports

- A document that details what is tested in the assessment run, and the results of the assessment.
- You can view the following types of assessment reports:
  - **Findings report** - this report contains the following information:
    - Executive summary of the assessment
    - EC2 instances evaluated during the assessment run
    - Rules packages included in the assessment run
    - Detailed information about each finding, including all EC2 instances that had the finding
  - **Full report** - this report contains all the information that is included in a findings report, and additionally provides the list of rules that passed on all instances in the assessment target.

## Pricing

- Pricing is based on two dimensions
  - The number of EC2 instances included in each assessment
  - The type(s) of rules package you select: host assessment rules packages and/or the network reachability rules package

## Sources:

<https://docs.aws.amazon.com/inspector/latest/userguide>



<https://aws.amazon.com/inspector/pricing/>

<https://aws.amazon.com/inspector/faqs/>



## AWS Organizations

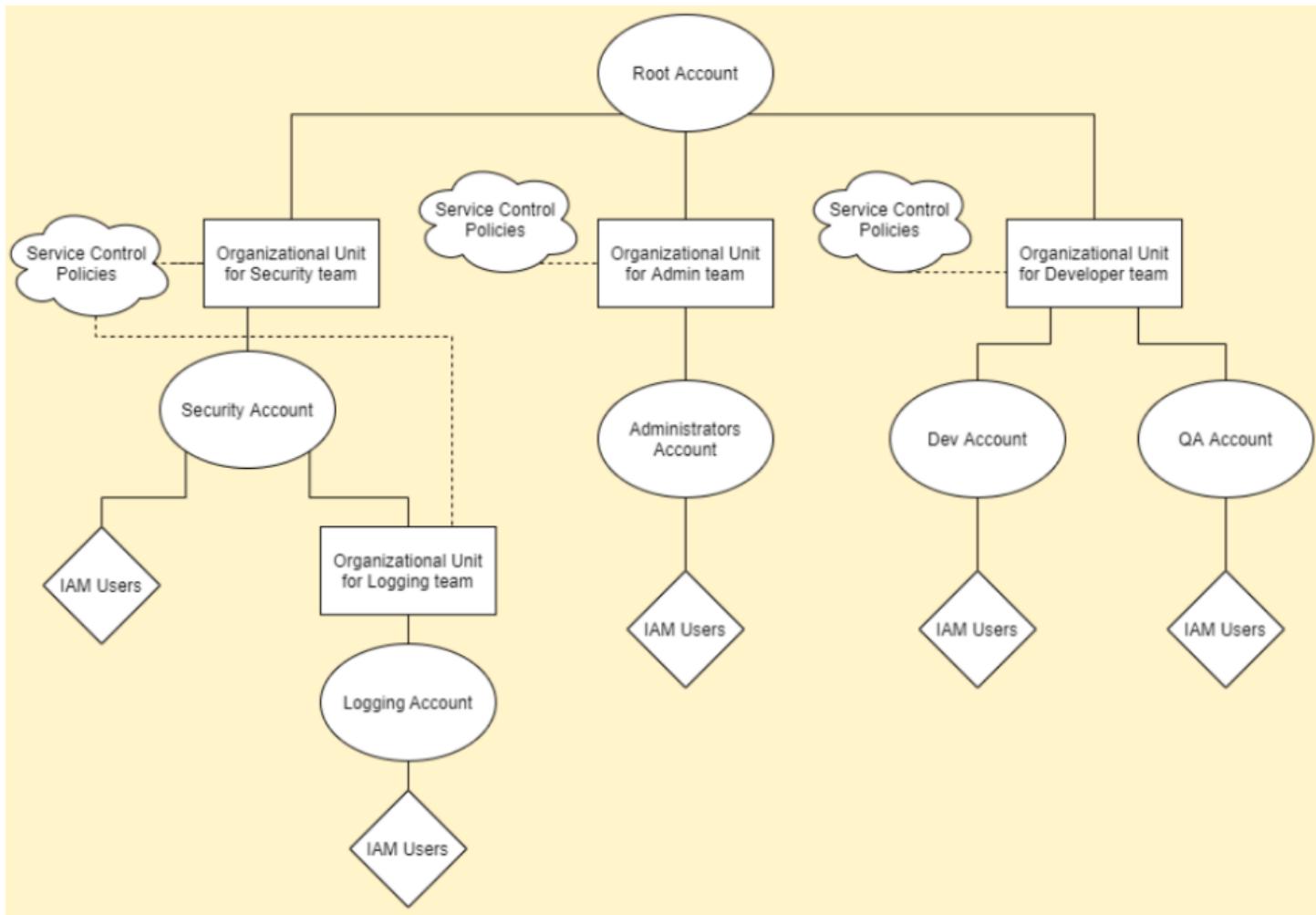
- It offers policy-based management for multiple AWS accounts.

### Features

- With Organizations, you can create groups of accounts and then apply policies to those groups.
- Organizations provides you a policy framework for multiple AWS accounts. You can apply policies to a group of accounts or all the accounts in your organization.
- AWS Organizations enables you to set up a single payment method for all the AWS accounts in your organization through **consolidated billing**. With consolidated billing, you can see a combined view of charges incurred by all your accounts, as well as take advantage of pricing benefits from aggregated usage, such as volume discounts for EC2 and S3.
- AWS Organizations, like many other AWS services, is **eventually consistent**. It achieves high availability by replicating data across multiple servers in AWS data centers within its region.

### Administrative Actions in Organizations

- Create an AWS account and add it to your organization, or add an existing AWS account to your organization.
- Organize your AWS accounts into groups called *organizational units* (OUs).
- Organize your OUs into a hierarchy that reflects your company's structure.
- Centrally manage and attach policies to the entire organization, OUs, or individual AWS accounts.



## Concepts

- An **organization** is a collection of AWS accounts that you can organize into a hierarchy and manage centrally.
- A **master account** is the AWS account you use to create your organization. You cannot change which account in your organization is the master account.
  - From the master account, you can create other accounts in your organization, invite and manage invitations for other accounts to join your organization, and remove accounts from your organization.
  - You can also attach policies to entities such as administrative roots, organizational units (OUs), or accounts within your organization.
  - The master account has the role of a payer account and is responsible for paying all charges accrued by the accounts in its organization.
- A **member account** is an AWS account, other than the master account, that is part of an organization. A member account can belong to only one organization at a time. The master account has the



---

responsibilities of a payer account and is responsible for paying all charges that are accrued by the member accounts.

- An **administrative root** is the starting point for organizing your AWS accounts. The administrative root is the top-most container in your organization's hierarchy. Under this root, you can create OUs to logically group your accounts and organize these OUs into a hierarchy that best matches your business needs.
- An **organizational unit (OU)** is a group of AWS accounts within an organization. An OU can also contain other OUs enabling you to create a hierarchy.
- A **policy** is a “document” with one or more statements that define the controls that you want to apply to a group of AWS accounts.
  - **Service control policy (SCP)** is a policy that specifies the services and actions that users and roles can use in the accounts that the SCP affects. SCPs are similar to IAM permission policies except that they don't grant any permissions. Instead, SCPs are *filters* that allow only the specified services and actions to be used in affected accounts.
- AWS Organizations has two available feature sets:
  - All organizations support **consolidated billing**, which provides basic management tools that you can use to centrally manage the accounts in your organization.
  - If you enable **all features**, you continue to get all the consolidated billing features plus a set of advanced features such as service control policies.
- You can remove an AWS account from an organization and make it into a standalone account.
- Organization Hierarchy
  - Including root and AWS accounts created in the lowest OUs, your hierarchy can be five levels deep.
  - Policies inherited through hierarchical connections in an organization.
  - Policies can be assigned at different points in the hierarchy.

## Pricing

- This service is free.

## Sources:

<https://docs.aws.amazon.com/organizations/latest/userguide/>

<https://aws.amazon.com/organizations/features/>

<https://aws.amazon.com/organizations/faqs/>



## AWS Artifact

- A self-service central repository of AWS' security and compliance reports and select online agreements.
- An **audit artifact** is a piece of evidence that demonstrates that an organization is following a documented process or meeting a specific requirement (business compliant).
- **AWS Artifact Reports** include the following:
  - ISO,
  - Service Organization Control (SOC) reports,
  - Payment Card Industry (PCI) reports,
  - and certifications that validate the implementation and operating effectiveness of AWS security controls.
- **AWS Artifacts Agreements** include
  - the Nondisclosure Agreement (NDA)
  - the Business Associate Addendum (BAA), which typically is required for companies that are subject to the HIPAA Act to ensure that protected health information (PHI) is appropriately safeguarded.
- **All AWS Accounts with AWS Artifact IAM permissions have access to AWS Artifact.** Root users and IAM users with admin permissions can download all audit artifacts available to their account by agreeing to the associated terms and conditions. You will need to grant IAM users with non-admin permissions access to AWS Artifact.
- To use organization agreements in AWS Artifact, your organization must be enabled for **all features**.
- **AWS Artifact Agreements**
  - AWS Artifact Account Agreements apply only to the individual account you used to sign into AWS.
  - AWS Artifact Organization Agreements apply to all accounts in an organization created through AWS Organizations, including the organization's master account and all member accounts. Only the master account in an organization can accept agreements in AWS Artifact Organization Agreements.
  - Master accounts and member accounts of an Organization can have AWS Artifact Account Agreements and AWS Artifact Organization Agreements of the same type in place at the same time.
  - If you have accounts in separate organizations that you want covered by an agreement, you must log in to each organization's master account and accept the relevant agreements through AWS Artifact Organization Agreements.
  - Terminating the organization agreement does not terminate the account agreement.
  - When a member account is removed from an organization (e.g. by leaving the organization, or by being removed from the organization by the master account), any organization agreements accepted on its behalf will no longer apply to that member account.
- **Business Associate Addendum (BAA)**



- You can accept the AWS BAA for your individual account, or if you are a master account in an organization, you can accept the AWS BAA on behalf of all accounts in your organization.
- Upon accepting the AWS BAA in AWS Artifact Agreements, you will instantly designate your AWS account(s) for use in connection with protected health information (PHI) and HIPAA.
- If you terminate an online BAA under the Account agreements tab in AWS Artifact, the account you used to sign into AWS will immediately cease to be a HIPAA Account, unless it was also covered by an organization BAA.
- If you are a user of a master account and terminate an online BAA in AWS Artifact, all accounts within your organization will immediately be removed as HIPAA Accounts, unless they were covered by individual account BAAs.
- If you have both an account BAA and an organization BAA in place at the same time, the terms of the organization BAA will apply instead of the terms of the account BAA.
- AWS Australian Notifiable Data Breach Addendum (ANDB Addendum)
  - Using the master account of your organization you can use the Organization agreements tab in AWS Artifact Agreements to accept an ANDB Addendum on behalf of all existing and future member accounts in your organization.
  - When both the account ANDB Addendum and organizations ANDB Addendum are accepted, the organizations ANDB Addendum will apply instead of the account ANDB Addendum.
  - If you terminate an account ANDB Addendum under the Account agreements tab in AWS Artifact, the AWS account you used to sign into AWS Artifact will not be covered by an ANDB Addendum with AWS, unless it is also covered by an organizations ANDB Addendum.
  - If you are a user of a master account and terminate an organizations ANDB Addendum within the Organization agreements tab in AWS Artifact, the AWS accounts in that AWS organization will not be covered by an ANDB Addendum with AWS, unless they are covered by an account ANDB Addendum
- Most errors you receive from AWS Artifact can be resolved by adding the necessary IAM permissions.

**Sources:**

<https://aws.amazon.com/artifact/>

<https://docs.aws.amazon.com/artifact/latest/ug/what-is-aws-artifact.html>

<https://aws.amazon.com/artifact/faq/>



## MIGRATION

### AWS Snowball Edge

- A type of Snowball device with on-board storage and compute power for select AWS capabilities. It can undertake local processing and edge-computing workloads in addition to transferring data between your local environment and the AWS Cloud.
- Has on-board S3-compatible storage and compute to support running Lambda functions and EC2 instances.
- Options for device configurations
  - **Storage optimized** - this option has the most storage capacity at up to 80 TB of useable storage space, 24 vCPUs, and 32 GiB of memory for compute functionality. You can transfer up to **100 TB** with a single Snowball Edge Storage Optimized device.
  - **Compute optimized** - this option has the most compute functionality with 52 vCPUs, 208 GiB of memory, and 7.68 TB of dedicated NVMe SSD storage for instances. This option also comes with 42 TB of additional storage space.
  - Compute Optimized with GPU - identical to the compute optimized option, save for an installed GPU, equivalent to the one available in the P3 Amazon EC2 instance type.
- Features
  - Network adapters with transfer speeds of up to 100 GB/second.
  - Encryption is enforced, protecting your data at rest and in physical transit.
  - You can import or export data between your local environments and S3.
  - Snowball Edge devices come with an on-board LCD display that can be used to manage network connections and get service status information.
  - You can cluster Snowball Edge devices for local storage and compute jobs to achieve 99.999 percent data durability across 5–10 devices, and to locally grow and shrink storage on demand.
  - You can use the file interface to read and write data to a Snowball Edge device through a file share or Network File System (NFS) mount point.
  - You can write Python-language Lambda functions and associate them with S3 buckets when you create a Snowball Edge device job. Each function triggers whenever there's a local S3 *PUT* object action executed on the associated bucket on the appliance.
  - Snowball Edge devices have S3 and EC2 compatible endpoints available, enabling programmatic use cases.
  - Customers who need to run Amazon EC2 workloads on AWS Snowball Edge devices can attach multiple persistent block storage volumes to their Amazon EC2 instances.
  - For latency-sensitive applications such as machine learning, you can deploy a **performance-optimized SSD volume (sbp1)**. Performance optimized volumes on the Snowball Edge Compute Optimized device use NVMe SSD, and on the Snowball Edge Storage Optimized device they use SATA SSD. Alternatively, you can use capacity-optimized **HDD volumes (sbg1)** on any Snowball Edge.



- Snowball vs Snowball Edge
  - (View AWS Migration CheatSheet: Snowball: Snowball vs Snowball Edge section)
- Job Types
  - **Import To S3** - transfer of 80 TB or less of your local data copied onto a single device, and then moved into S3.
    - Snowball Edge devices and jobs have a one-to-one relationship. Each job has exactly one device associated with it. If you need to import more data, you can create new import jobs or clone existing ones.
  - **Export From S3** - transfer of any amount of data located in S3, copied onto any number of Snowball Edge devices, and then move one Snowball Edge device at a time into your on-premises data destination.
    - When you create an export job, it's split into job parts. Each job part is no more than 100 TB in size, and each job part has exactly one Snowball Edge device associated with it.
  - **Local Compute and Storage Only** - these jobs involve one Snowball Edge device, or multiple devices used in a cluster. This job type is only for local use.
    - A **cluster job** is for those workloads that require increased data durability and storage capacity. Clusters have anywhere from 5 to 10 Snowball Edge devices, called **nodes**.
    - A cluster offers increased durability and increased storage over a standalone Snowball Edge for local storage and compute.
- Recommendations
  - Files should be in a static state while being written to the device.
  - The *Job created* status is the only status in which you can cancel a job. When a job changes to a different status, it can't be canceled.
  - All files transferred to a Snowball be no smaller than 1 MB in size.
  - Perform multiple write operations at one time by running each command from multiple terminal windows on a computer with a network connection to a single Snowball Edge device.
  - Transfer small files in batches.
- Security
  - All data transferred to a device is protected by SSL encryption over the network.
  - To protect data at rest, Snowball Edge uses server side-encryption.
  - Access to Snowball Edge requires credentials that AWS can use to authenticate your requests. Those credentials must have permissions to access AWS resources, such as an Amazon S3 bucket or an AWS Lambda function.
- Pricing
  - You are charged depending on the type of Snowball Edge machine you choose (storage optimized or compute optimized).
  - You are charged based on what you select for your term of use: On demand, a 1-year commitment, or a 3-year commitment.
  - For on-demand use, you pay a service fee per data transfer job, which includes 10 days of on-site Snowball Edge device usage. Shipping days, including the day the device is received and



the day it is shipped back to AWS, are not counted toward the 10 days. If the device is kept for more than 10 days, you will incur an additional fee for each day beyond 10 days.

- There is a one-time setup fee per job ordered through the console.

**Sources:**

<https://aws.amazon.com/snowball-edge/features/>

<https://aws.amazon.com/snowball-edge/pricing/>

<https://aws.amazon.com/snowball-edge/faqs/>



## AWS Snowmobile

- An **exabyte-scale** data transfer service used to move extremely large amounts of data to AWS. You can transfer up to 100PB per Snowmobile.
- Snowmobile will be returned to your designated AWS region where your data will be uploaded into the AWS storage services you have selected, such as S3 or Glacier.
- Snowmobile uses multiple layers of security to help protect your data including dedicated security personnel:
  - GPS tracking, alarm monitoring
  - 24/7 video surveillance
  - an optional escort security vehicle while in transit
  - All data is encrypted with 256-bit encryption keys you manage through the AWS Key Management Service and designed for security and full chain-of-custody of your data.
- Snowmobile pricing is based on the amount of data stored on the truck per month.

### Sources:

<https://aws.amazon.com/snowmobile/faqs/>  
<https://aws.amazon.com/snowmobile/pricing/>



## AWS DataSync

- An online data transfer service that simplifies, automates, and accelerates copying large amounts of data to and from AWS storage services over the internet or AWS Direct Connect.
- DataSync can copy data between:
  - Network File System (NFS) or Server Message Block (SMB) file servers,
  - Amazon Simple Storage Service (Amazon S3) buckets,
  - Amazon Elastic File System (Amazon EFS) file systems,
  - Amazon FSx for Windows File Server file systems
- How It Works
  1. Deploy an agent - Deploy a DataSync agent and associate it to your AWS account via the Management Console or API. The agent will be used to access your NFS server or SMB file share to read data from it or write data to it.
  2. Create a data transfer task - Create a task by specifying the location of your data source and destination, and any options you want to use to configure the transfer, such as the desired task schedule.
  3. Start the transfer - Start the task and monitor data movement in the console or with Amazon CloudWatch.

### Concepts

- Agent – A virtual machine used to read data from or write data to an on-premises location.
- Location – Any source or destination location used in the data transfer.
- Task – A task includes two locations (source and destination), and also the configuration of how to transfer the data from one location to the other. Configuration settings can include options such as how to treat metadata, deleted files, and copy permission.
- Task execution – An individual run of a task, which includes options such as start time, end time, bytes written, and status. A task execution has five transition phases and two terminal statuses, as shown in the following diagram. If the VerifyMode option is not enabled, a terminal status occurs after the TRANSFERRING phase. Otherwise, it occurs after the VERIFYING phase.

- Features
  - The service employs an AWS-designed transfer protocol—decoupled from storage protocol—to speed data movement. The protocol performs optimizations on how, when, and what data is sent over the network.
  - A single DataSync agent is capable of saturating a 10 Gbps network link.
  - DataSync auto-scales cloud resources to support higher-volume transfers, and makes it easy to add agents on-premises.
  - All of your data is encrypted in transit with TLS. DataSync supports using default encryption for S3 buckets using Amazon S3-Managed Encryption Keys (SSE-S3), and Amazon EFS file system encryption of data at rest.



- DataSync supports storing data directly into S3 Standard, S3 Intelligent-Tiering, S3 Standard-Infrequent Access (S3 Standard-IA), S3 One Zone-Infrequent Access (S3 One Zone-IA), Amazon S3 Glacier, and S3 Glacier Deep Archive.
  - You can use AWS DataSync to copy files into EFS and configure EFS Lifecycle Management to migrate files that have not been accessed for a set period of time to the Infrequent Access (IA) storage class.
  - DataSync ensures that your data arrives intact by performing integrity checks both in transit and at rest.
  - You can specify an exclude filter, an include filter, or both, to limit which files, folders, or objects get transferred each time a task runs.
  - Task scheduling enables you to configure periodically executing a task, to detect and copy changes from your source storage system to the destination.
  - DataSync supports VPC endpoints (powered by AWS PrivateLink) in order to move files directly into your Amazon VPC.
- Use Cases
    - Data migration to Amazon S3, Amazon EFS, or Amazon FSx for Windows File Server.
    - Data processing for hybrid workloads. If you have on-premises systems generating or using data that needs to move into or out of AWS for processing, you can use DataSync to accelerate and schedule the transfers.
    - If you have large amounts of cold data stored in expensive on-premises storage systems, you can move this data directly to durable and secure long-term storage such as Amazon S3 Glacier or Amazon S3 Glacier Deep Archive.
    - If you have large Network Attached Storage (NAS) systems with important files that need to be protected, you can replicate them into S3 using DataSync.
  - DataSync Agent
    - Agents need to be activated first using an activation key entered in the AWS console, before you can start using them. You must activate your agent in the same region where your S3 or EFS source/destination resides.
    - You run DataSync on-premises as a virtual machine (VM).
    - DataSync provides an Amazon Machine Image (AMI) that contains the DataSync VM image when running in an EC2 instance.
    - The agent VM requires access to some endpoints to communicate with AWS. You must configure your firewall settings to allow these connections.
    - You can have more than one DataSync Agent running.
  - AWS DataSync vs AWS CLI tools
    - AWS DataSync fully automates and accelerates moving large active datasets to AWS, up to 10 times faster than command line tools.
    - DataSync uses a purpose-built network protocol and scale-out architecture to transfer data.
    - DataSync fully automates the data transfer. It comes with retry and network resiliency mechanisms, network optimizations, built-in task scheduling, and CloudWatch monitoring that provides granular visibility into the transfer process.



- DataSync performs data integrity verification both during the transfer and at the end of the transfer.
- DataSync provides end to end security, and integrates directly with AWS storage services.
- AWS DataSync vs Snowball/Snowball Edge
  - AWS DataSync is ideal for online data transfers. AWS Snowball/ Snowball Edge is suitable for offline data transfers, for customers who are bandwidth constrained, or transferring data from remote, disconnected, or austere environments.
- AWS DataSync vs AWS Storage Gateway File Gateway
  - Use AWS DataSync to migrate existing data to Amazon S3, and then use the File Gateway to retain access to the migrated data and for ongoing updates from your on-premises file-based applications.
- AWS DataSync vs Amazon S3 Transfer Acceleration
  - If your applications are already integrated with the Amazon S3 API, and you want higher throughput for transferring large files to S3, you can use S3 Transfer Acceleration. If not, you may use AWS DataSync.
- AWS DataSync vs AWS Transfer for SFTP
  - If you currently use SFTP to exchange data with third parties, you may use AWS Transfer for SFTP to transfer directly these data.
  - If you want an accelerated and automated data transfer between NFS servers, SMB file shares, Amazon S3, Amazon EFS, and Amazon FSx for Windows File Server, you can use AWS DataSync.
- Pricing
  - You pay for the amount of data that you copy. Your costs are based on a flat per-gigabyte fee for the use of network acceleration technology, managed cloud infrastructure, data validation, and automation capabilities in DataSync.
  - You are charged standard request, storage, and data transfer rates to read to and write from AWS services, such as Amazon S3, Amazon EFS, Amazon FSx for Windows File Server, and AWS Key Management Service (KMS).
  - When copying data from AWS to an on-premises storage system, you pay for AWS Data Transfer at your standard rate. You are also charged standard rates for Amazon CloudWatch Logs, Amazon CloudWatch Events, and Amazon CloudWatch Metrics.
  - You will be billed by AWS PrivateLink for interface VPC endpoints that you create to manage and control the traffic between your agent(s) and the DataSync service over AWS PrivateLink.

**Sources:**

<https://aws.amazon.com/datasync/>

<https://docs.aws.amazon.com/datasync/latest/userguide/what-is-datasync.html>

<https://aws.amazon.com/datasync/faqs/>



## MANAGEMENT

### AWS Auto Scaling

- Configure automatic scaling for the AWS resources quickly through a scaling plan that uses **dynamic scaling** and **predictive scaling**.
- Optimize for availability, for cost, or a balance of both.
- Scaling in means decreasing the size of a group while scaling out means increasing the size of a group.
- Useful for
  - Cyclical traffic such as high use of resources during regular business hours and low use of resources overnight
  - On and off traffic patterns, such as batch processing, testing, or periodic analysis
  - Variable traffic patterns, such as software for marketing campaigns with periods of spiky growth
- It is a region specific service. Currently, AWS Auto Scaling is only available in US East (Northern Virginia), US East (Ohio), US West (Oregon), EU (Ireland), and Asia Pacific (Singapore) public AWS regions.
- Features
  - Launch or terminate EC2 instances in an Auto Scaling group.
  - Launch or terminate instances from an EC2 Spot Fleet request, or automatically replace instances that get interrupted for price or capacity reasons.
  - Adjust the ECS service desired count up or down in response to load variations.
  - Enable a DynamoDB table or a global secondary index to increase or decrease its provisioned read and write capacity to handle increases in traffic without throttling.
  - Dynamically adjust the number of Aurora read replicas provisioned for an Aurora DB cluster to handle changes in active connections or workload.
  - Use **Dynamic Scaling** to add and remove capacity for resources to maintain resource utilization at the specified target value.
  - Use **Predictive Scaling** to forecast your future load demands by analyzing your historical records for a metric. It also allows you to schedule scaling actions that proactively add and remove resource capacity to reflect the load forecast, and control maximum capacity behavior. Only available for EC2 Auto Scaling groups.
  - AWS Auto Scaling scans your environment and automatically discovers the scalable cloud resources underlying your application, so you don't have to manually identify these resources one by one through individual service interfaces.
  - You can suspend and resume any of your AWS Application Auto Scaling actions.
- Amazon EC2 Auto Scaling
  - Ensuring you have the correct number of EC2 instances available to handle your application load using **Auto Scaling Groups**.
  - An **Auto Scaling group** contains a collection of EC2 instances that share similar characteristics and are treated as a logical grouping for the purposes of instance scaling and management.



- You specify the minimum, maximum and desired number of instances in each Auto Scaling group.
- Key Components

Groups	Your EC2 instances are organized into <i>groups</i> so that they are treated as a logical unit for scaling and management. When you create a group, you can specify its minimum, maximum, and desired number of EC2 instances.
Launch configurations	Your group uses a <i>launch configuration</i> as a template for its EC2 instances. When you create a launch configuration, you can specify information such as the AMI ID, instance type, key pair, security groups, and block device mapping for your instances.
Scaling options	How to scale your Auto Scaling groups.

- You can add a **lifecycle hook** to your Auto Scaling group to perform custom actions when instances launch or terminate.
- Scaling Options
  - Scale to maintain current instance levels at all times
  - Manual Scaling
  - Scale based on a schedule
  - Scale based on a demand
- Scaling Policy Types
  - **Target tracking scaling**—Increase or decrease the current capacity of the group based on a target value for a specific metric.
  - **Step scaling**—Increase or decrease the current capacity of the group based on a set of scaling adjustments, known as step adjustments, that vary based on the size of the alarm breach.
  - **Simple scaling**—Increase or decrease the current capacity of the group based on a single scaling adjustment.
- The **cooldown period** is a configurable setting that helps ensure to not launch or terminate additional instances before previous scaling activities take effect.
  - EC2 Auto Scaling supports cooldown periods when using simple scaling policies, but not when using target tracking policies, step scaling policies, or scheduled scaling.



The screenshot shows the AWS Auto Scaling Groups console. At the top, there is a table with one row for 'testasg-adrian'. The columns include Name, Launch Configuration, Instances, Desired, Min, Max, Availability Zones, Default Cooldown, and Health Check Grace. The 'Instances' column shows 0 for both min and max, and the availability zones listed are us-east-1d, us-east-1e, us-e... with a cooldown of 300 seconds and a grace period of 300 seconds. Below the table, it says 'Auto Scaling Group: testasg-adrian'. A navigation bar below the table has tabs for Details, Activity History, Scaling Policies, Instances, Monitoring, Notifications, Tags, Scheduled Actions, and Lifecycle Hooks. The 'Scaling Policies' tab is selected and highlighted in orange. A modal window titled 'Create Scaling policy' is open in the center. It contains fields for 'Name' (with a placeholder box), 'Metric type' (set to 'Average CPU Utilization'), 'Target value' (with a placeholder box), 'Instances need' (set to '300 seconds to warm up after scaling'), and a 'Disable scale-in' checkbox which is unchecked. At the bottom of the modal, there are two links: 'Create a simple scaling policy' and 'Create a scaling policy with steps'.

- Amazon EC2 Auto Scaling marks an instance as unhealthy if the instance is in a state other than *running*, the system status is *impaired*, or Elastic Load Balancing reports that the instance failed the health checks.
- Termination of Instances
  - When you configure automatic scale in, you must decide which instances should terminate first and set up a **termination policy**. You can also use **instance protection** to prevent specific instances from being terminated during automatic scale in.
  - Default Termination Policy
  - Custom Termination Policies
    - *OldestInstance* - Terminate the oldest instance in the group.
    - *NewestInstance* - Terminate the newest instance in the group.
    - *OldestLaunchConfiguration* - Terminate instances that have the oldest launch configuration.
    - *ClosestToNextInstanceHour* - Terminate instances that are closest to the next billing hour.



Edit details - testasg

testpublicsubnet2 | us-east-1 |  
subnet-0d5054a867c8469db(10.1.2.0/24) |  
testpublicsubnet2 | us-east-1e

Classic Load Balancers (i) [ ]

Target Groups (i) [ ]

Health Check Type (i) EC2

Health Check Grace Period (i) 300

Instance Protection (i) [ ]

Termination Policies (i) Default X |  
OldestInstance

Suspended Processes (i)  
OldestLaunchConfiguration  
NewestInstance  
ClosestToNextInstanceHour  
AllocationStrategy  
OldestLaunchTemplate

Max Instance Lifetime (i)

Placement Groups (i)

Default Cooldown (i) 300

Cancel Save

You can create **launch templates** that specifies instance configuration information when you launch EC2 instances, and allows you to have multiple versions of a template.

A **launch configuration** is an instance configuration template that an Auto Scaling group uses to launch EC2 instances, and you specify information for the instances.

- You can specify your launch configuration with multiple Auto Scaling groups.
- You can only specify one launch configuration for an Auto Scaling group at a time, and you can't modify a launch configuration after you've created it.
- When you create a VPC, by default its tenancy attribute is set to *default*. You can launch instances with a tenancy value of *dedicated* so that they run as single-tenancy instances. Otherwise, they run as shared-tenancy instances by default.
- If you set the tenancy attribute of a VPC to *dedicated*, all instances launched in the VPC run as single-tenancy instances.



- When you create a launch configuration, the default value for the instance placement tenancy is *null* and the instance tenancy is controlled by the tenancy attribute of the VPC.

Launch Configuration Tenancy	VPC Tenancy = default	VPC Tenancy = dedicated
not specified	shared-tenancy instance	Dedicated Instance
default	shared-tenancy instance	Dedicated Instance
dedicated	Dedicated Instance	Dedicated Instance

- If you are launching the instances in your Auto Scaling group in EC2-Classic, you can link them to a VPC using *ClassicLink*.
- Application Auto Scaling
    - Allows you to configure automatic scaling for the following resources:
      - Amazon ECS services
      - Spot Fleet requests
      - Amazon EMR clusters
      - AppStream 2.0 fleets
      - DynamoDB tables and global secondary indexes
      - Aurora replicas
      - Amazon SageMaker endpoint variants
      - Custom resources provided by your own applications or services.
    - Features
      - Target tracking scaling**—Scale a resource based on a target value for a specific CloudWatch metric.
      - Step scaling**—Scale a resource based on a set of scaling adjustments that vary based on the size of the alarm breach.
      - Scheduled scaling**—Scale a resource based on the date and time.
    - Target tracking scaling
      - You can have multiple target tracking scaling policies for a scalable target, provided that each of them uses a different metric.
      - You can also optionally disable the scale-in portion of a target tracking scaling policy.
    - Step scaling
      - Increase or decrease the current capacity of a scalable target based on a set of scaling adjustments, known as **step adjustments**, that vary based on the size of the alarm breach.
    - Scheduled scaling



- Scale your application in response to predictable load changes by creating *scheduled actions*, which tell Application Auto Scaling to perform scaling activities at specific times.
  - The *scale out cooldown period* is the amount of time, in seconds, after a scale out activity completes before another scale out activity can start.
  - The *scale in cooldown period* is the amount of time, in seconds, after a scale in activity completes before another scale in activity can start.
- You can attach one or more classic ELBs to your existing Auto Scaling Groups. The ELBs must be in the same region.
- Auto Scaling rebalances by launching new EC2 instances in the AZs that have fewer instances first, only then will it start terminating instances in AZs that had more instances
- Monitoring
  - **Health checks** - identifies any instances that are unhealthy
    - Amazon EC2 status checks (default)
    - Elastic Load Balancing health checks
    - Custom health checks.
  - Auto scaling does not perform health checks on instances in the **standby** state. Standby state can be used for performing updates/changes/troubleshooting without health checks being performed or replacement instances being launched.
  - **CloudWatch metrics** - enables you to retrieve statistics about Auto Scaling-published data points as an ordered set of time-series data, known as *metrics*. You can use these metrics to verify that your system is performing as expected.
  - **CloudWatch Events** - Auto Scaling can submit events to CloudWatch Events when your Auto Scaling groups launch or terminate instances, or when a lifecycle action occurs.
  - **SNS notifications** - Auto Scaling can send Amazon SNS notifications when your Auto Scaling groups launch or terminate instances.
  - CloudTrail logs - enables you to keep track of the calls made to the Auto Scaling API by or on behalf of your AWS account, and stores the information in log files in an S3 bucket that you specify.
- Security
  - Use IAM to help secure your resources by controlling who can perform AWS Auto Scaling actions.
  - By default, a brand new IAM user has NO permissions to do anything. To grant permissions to call Auto Scaling actions, you attach an IAM policy to the IAM users or groups that require the permissions it grants.

#### Sources:

<https://docs.aws.amazon.com/autoscaling/plans/userguide/what-is-aws-auto-scaling.html>

<https://aws.amazon.com/autoscaling/features/>

<https://docs.aws.amazon.com/autoscaling/ec2/userguide/what-is-amazon-ec2-auto-scaling.html>



<https://docs.aws.amazon.com/autoscaling/application/userguide/what-is-application-auto-scaling.html>

<https://aws.amazon.com/autoscaling/pricing/>

<https://aws.amazon.com/autoscaling/faqs/>



## AWS CloudFormation

- A service that gives developers and businesses an easy way to create a collection of related AWS resources and provision them in an orderly and predictable fashion.

### Features

- CloudFormation allows you to model your entire infrastructure in a text file called a **template**. You can use JSON or YAML to describe what AWS resources you want to create and configure. If you want to design visually, you can use *AWS CloudFormation Designer*.
- CloudFormation automates the provisioning and updating of your infrastructure in a safe and controlled manner. You can use **Rollback Triggers** to specify the CloudWatch alarm that CloudFormation should monitor during the stack creation and update process. If any of the alarms are breached, CloudFormation rolls back the entire stack operation to a previous deployed state.
- **CloudFormation Change Sets** allow you to preview how proposed changes to a stack might impact your running resources.
- **AWS StackSets** lets you provision a common set of AWS resources across multiple accounts and regions with a single CloudFormation template. StackSets takes care of automatically and safely provisioning, updating, or deleting stacks in multiple accounts and across multiple regions.
- CloudFormation enables you to build custom extensions to your stack template using AWS Lambda.

### CloudFormation vs Elastic Beanstalk

- Elastic Beanstalk provides an **environment** to easily **deploy and run** applications in the cloud.
- CloudFormation is a convenient **provisioning mechanism** for a broad range of AWS resources.

### Concepts

- **Templates**
  - A JSON or YAML formatted text file.
  - CloudFormation uses these templates as blueprints for building your AWS resources.
- **Stacks**
  - Manage related resources as a single unit.
  - All the resources in a stack are defined by the stack's CloudFormation template.
- **Change Sets**
  - Before updating your stack and making changes to your resources, you can generate a change set, which is a summary of your proposed changes.
  - Change sets allow you to see how your changes might impact your running resources, especially for critical resources, before implementing them.
- With AWS CloudFormation and AWS CodePipeline, you can use continuous delivery to automatically build and test changes to your CloudFormation templates before promoting them to production stacks.



- CloudFormation artifacts can include a stack template file, a template configuration file, or both. AWS CodePipeline uses these artifacts to work with CloudFormation stacks and change sets.
  - **Stack Template File** - defines the resources that CloudFormation provisions and configures. You can use YAML or JSON-formatted templates.
  - **Template Configuration File** - a JSON-formatted text file that can specify template parameter values, a stack policy, and tags. Use these configuration files to specify parameter values or a stack policy for a stack.
- Through the AWS PrivateLink, you can use CloudFormation APIs inside of your Amazon VPC and route data between your VPC and CloudFormation entirely within the AWS network.

## Stacks

- If a resource cannot be created, CloudFormation rolls the stack back and automatically deletes any resources that were created. If a resource cannot be deleted, any remaining resources are retained until the stack can be successfully deleted.
- Stack update methods
  - Direct update
  - Creating and executing change sets
- Drift detection enables you to detect whether a stack's actual configuration differs, or has drifted, from its expected configuration. Use CloudFormation to detect drift on an entire stack, or on individual resources within the stack.
  - A resource is considered to have drifted if any of its actual property values differ from the expected property values.
  - A stack is considered to have drifted if one or more of its resources have drifted.
- To share information between stacks, export a stack's output values. Other stacks that are in the same AWS account and region can import the exported values.
- You can nest stacks.

## Templates

- Templates include several major sections. The Resources section is the only required section.



```
---
AWSTemplateFormatVersion: "version date"

Description:
  String

Metadata:
  template metadata

Parameters:
  set of parameters

Mappings:
  set of mappings

Conditions:
  set of conditions

Transform:
  set of transforms

Resources:
  set of resources

Outputs:
  set of outputs
```

- **CloudFormation Designer** is a graphic tool for creating, viewing, and modifying CloudFormation templates. You can diagram your template resources using a drag-and-drop interface, and then edit their details using the integrated JSON and YAML editor.
- Custom resources enable you to write custom provisioning logic in templates that CloudFormation runs anytime you create, update (if you changed the custom resource), or delete stacks.
- Template macros enable you to perform custom processing on templates, from simple actions like find-and-replace operations to extensive transformations of entire templates.

## StackSets

- Administrator and target accounts - An *administrator account* is the AWS account in which you create stack sets. A stack set is managed by signing in to the AWS administrator account in which it was created. A *target account* is the account into which you create, update, or delete one or more stacks in your stack set.



- Stack sets - A stack set lets you create stacks in AWS accounts across regions by using a single CloudFormation template. All the resources included in each stack are defined by the stack set's CloudFormation template. A stack set is a regional resource.
- Stack instances - A *stack instance* is a reference to a stack in a target account within a region. A stack instance can exist without a stack, and can be associated with only one stack set.
- Stack set operations - Create stack set, update stack set, delete stacks, and delete stack set.
- Tags - You can add tags during stack set creation and update operations by specifying key and value pairs.

## Monitoring

- CloudFormation is integrated with AWS CloudTrail, a service that provides a record of actions taken by a user, role, or an AWS service in CloudFormation. CloudTrail captures all API calls for CloudFormation as events, including calls from the CloudFormation console and from code calls to the CloudFormation APIs.

## Security

- You can use IAM with CloudFormation to control what users can do with AWS CloudFormation, such as whether they can view stack templates, create stacks, or delete stacks.
- A *service role* is an IAM role that allows CloudFormation to make calls to resources in a stack on your behalf. You can specify an IAM role that allows CloudFormation to create, update, or delete your stack resources.
- You can improve the security posture of your VPC by configuring CloudFormation to use an interface VPC endpoint.

## Pricing

- No additional charge for CloudFormation. You pay for AWS resources created using CloudFormation in the same manner as if you created them manually.

## Sources:

<https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/>

<https://aws.amazon.com/cloudformation/features/>

<https://aws.amazon.com/cloudformation/pricing/>

<https://aws.amazon.com/cloudformation/faqs/>



## AWS CloudTrail

- Actions taken by a user, role, or an AWS service in the AWS Management Console, AWS Command Line Interface, and AWS SDKs and APIs are recorded as **events**.
- CloudTrail is enabled on your AWS account when you create it.
- CloudTrail focuses on auditing API activity.
- View events in **Event History**, where you can view, search, and download the past 90 days of activity in your AWS account.
- **Trails**
  - Create a **CloudTrail trail** to archive, analyze, and respond to changes in your AWS resources.
  - **Types**
    - A trail that applies to **all regions** - CloudTrail records events in each region and delivers the CloudTrail event log files to an S3 bucket that you specify. This is the default option when you create a trail in the CloudTrail console.
    - A trail that applies to **one region** - CloudTrail records the events in the region that you specify only. This is the default option when you create a trail using the AWS CLI or the CloudTrail API.
  - You can create a **organization trail** that will log all events for all AWS accounts in an organization created by AWS Organizations. Organization trails must be created in the master account.
  - By default, CloudTrail event log files are encrypted using Amazon S3 server-side encryption. You can also choose to encrypt your log files with an AWS Key Management Service key.
  - You can store your log files in your S3 bucket for as long as you want, and also define S3 lifecycle rules to archive or delete log files automatically.
  - If you want notifications about log file delivery and validation, you can set up Amazon SNS notifications.
  - CloudTrail publishes log files about every five minutes.
- **Events**
  - The record of an activity in an AWS account. This activity can be an action taken by a user, role, or service that is monitorable by CloudTrail.
  - **Types**
    - **Management events**
      - Logged by default
      - Management events provide insight into management operations performed on resources in your AWS account, also known as *control plane operations*.
    - **Data events**
      - Not logged by default
      - Data events provide insight into the resource operations performed on or in a resource, also known as *data plane operations*.
      - Data events are often high-volume activities.



- For global services such as IAM, STS, CloudFront, and Route 53, events are delivered to any trail that includes global services, and are logged as occurring in US East (N. Virginia) Region.
- You can filter logs by specifying Time range and one of the following attributes: Event name, User name, Resource name, Event source, Event ID, and Resource type.
- **Monitoring**
  - Use **CloudWatch Logs** to monitor log data. CloudTrail events that are sent to CloudWatch Logs can trigger alarms according to the metric filters you define.
  - To determine whether a log file was modified, deleted, or unchanged after CloudTrail delivered it, you can use CloudTrail log file integrity validation.
- **Price**
  - The first copy of management events within each region is delivered free of charge. Additional copies of management events are charged.
  - Data events are recorded and charged only for the Lambda functions and S3 buckets you specify.
  - Once a CloudTrail trail is set up, S3 charges apply based on your usage, since CloudTrail delivers logs to an S3 bucket.

**Sources:**

<https://docs.aws.amazon.com/awscloudtrail/latest/userguide/>  
<https://aws.amazon.com/cloudtrail/features/>  
<https://aws.amazon.com/cloudtrail/pricing/>  
<https://aws.amazon.com/cloudtrail/faqs/>



## Amazon CloudWatch

- Monitoring tool for your AWS resources and applications.
- Display metrics and create alarms that watch the metrics and send notifications or automatically make changes to the resources you are monitoring when a threshold is breached.
- CloudWatch does not aggregate data across regions. Therefore, metrics are completely separate between regions.
- **CloudWatch Concepts**
  - **Namespaces** - a container for CloudWatch metrics.
    - There is no default namespace.
    - The AWS namespaces use the following naming convention: AWS/service.
  - **Metrics** - represents a time-ordered set of data points that are published to CloudWatch.
    - Exists only in the region in which they are created.
    - Cannot be deleted, but they automatically expire after 15 months if no new data is published to them.
    - As new data points come in, data older than 15 months is dropped.
    - Each metric data point must be marked with a *timestamp*. The timestamp can be up to two weeks in the past and up to two hours into the future. If you do not provide a timestamp, CloudWatch creates a timestamp for you based on the time the data point was received.
    - By default, several services provide free metrics for resources. You can also enable **detailed monitoring**, or publish your own application metrics.
    - **Metric math** enables you to query multiple CloudWatch metrics and use math expressions to create new time series based on these metrics.
    - **Important note for EC2 metrics:** CloudWatch does not collect memory utilization and disk space usage metrics right from the get go. You need to install CloudWatch Agent in your instances first to retrieve these metrics.
  - **Dimensions** - a name/value pair that uniquely identifies a metric.
    - You can assign up to 10 dimensions to a metric.
    - Whenever you add a unique dimension to one of your metrics, you are creating a new variation of that metric.
  - **Statistics** - metric data aggregations over specified periods of time.
    - Each statistic has a unit of measure. Metric data points that specify a unit of measure are aggregated separately.
    - You can specify a unit when you create a custom metric. If you do not specify a unit, CloudWatch uses *None* as the unit.
    - A *period* is the length of time associated with a specific CloudWatch statistic. The default value is 60 seconds.
    - CloudWatch aggregates statistics according to the period length that you specify when retrieving statistics.



- For large datasets, you can insert a pre-aggregated dataset called a *statistic set*.

Statistic	Description
Minimum	The lowest value observed during the specified period. You can use this value to determine low volumes of activity for your application.
Maximum	The highest value observed during the specified period. You can use this value to determine high volumes of activity for your application.
Sum	All values submitted for the matching metric added together. Useful for determining the total volume of a metric.
Average	The value of Sum / SampleCount during the specified period. By comparing this statistic with the Minimum and Maximum, you can determine the full scope of a metric and how close the average use is to the Minimum and Maximum. This comparison helps you to know when to increase or decrease your resources as needed.
SampleCount	The count (number) of data points used for the statistical calculation.
pNN.NN	The value of the specified percentile. You can specify any percentile, using up to two decimal places (for example, p95.45). Percentile statistics are not available for metrics that include any negative values.

- **Percentiles** - indicates the relative standing of a value in a dataset. Percentiles help you get a better understanding of the distribution of your metric data.
- **Alarms** - watches a single metric over a specified time period, and performs one or more specified actions, based on the value of the metric relative to a threshold over time
  - You can create an alarm for monitoring CPU usage and load balancer latency, for managing instances, and for billing alarms.
  - When an alarm is on a dashboard, it turns red when it is in the **ALARM** state.
  - Alarms invoke actions for sustained state changes only.
  - **Alarm States**
    - **OK**—The metric or expression is within the defined threshold.
    - **ALARM**—The metric or expression is outside of the defined threshold.
    - **INSUFFICIENT\_DATA**—The alarm has just started, the metric is not available, or not enough data is available for the metric to determine the alarm state.
  - When you create an alarm, you specify three settings:



- **Period** is the length of time to evaluate the metric or expression to create each individual data point for an alarm. It is expressed in seconds.
- **Evaluation Period** is the number of the most recent periods, or data points, to evaluate when determining alarm state.
- **Datapoints to Alarm** is the number of data points within the evaluation period that must be breaching to cause the alarm to go to the ALARM state. The breaching data points do not have to be consecutive, they just must all be within the last number of data points equal to **Evaluation Period**.
- For each alarm, you can specify CloudWatch to treat missing data points as any of the following:
  - *missing*—the alarm does not consider missing data points when evaluating whether to change state (default)
  - *notBreaching*—missing data points are treated as being within the threshold
  - *breaching*—missing data points are treated as breaching the threshold
  - *ignore*—the current alarm state is maintained
- You can now create tags in CloudWatch alarms that let you define policy controls for your AWS resources. This enables you to create resource level policies for your alarms.

### CloudWatch Dashboard

- Customizable home pages in the CloudWatch console that you can use to monitor your resources in a single view, even those spread across different regions.
- There is no limit on the number of CloudWatch dashboards you can create.
- All dashboards are **global**, not region-specific.
- You can add, remove, resize, move, edit or rename a graph. You can metrics manually in a graph.

### CloudWatch Events

- Deliver near real-time stream of system events that describe changes in AWS resources.
- Events respond to these operational changes and take corrective action as necessary, by sending messages to respond to the environment, activating functions, making changes, and capturing state information.
- Concepts
  - **Events** - indicates a change in your AWS environment.
  - **Targets** - processes events.
  - **Rules** - matches incoming events and routes them to targets for processing.

### CloudWatch Logs

- Features
  - Monitor logs from EC2 instances in real-time
  - Monitor CloudTrail logged events
  - By default, logs are kept indefinitely and never expire
  - Archive log data
  - Log Route 53 DNS queries
- **CloudWatch Logs Insights** enables you to interactively search and analyze your log data in CloudWatch Logs using queries.



- **CloudWatch Vended logs** are logs that are natively published by AWS services on behalf of the customer. **VPC Flow logs** is the first Vended log type that will benefit from this tiered model.

### CloudWatch Agent

- Collect more logs and system-level metrics from EC2 instances and your on-premises servers.
- Needs to be installed.

### Authentication and Access Control

- Use IAM users or roles for authenticating who can access
- Use Dashboard Permissions, IAM identity-based policies, and service-linked roles for managing access control.
- A *permissions policy* describes who has access to what.
  - Identity-Based Policies
  - Resource-Based Policies
- There are no CloudWatch Amazon Resource Names (ARNs) for you to use in an IAM policy. Use an \* (asterisk) instead as the resource when writing a policy to control access to CloudWatch actions.

### Pricing

- You are charged for the number of metrics you have per month
- You are charged per 1000 metrics requested using CloudWatch API calls
- You are charged per dashboard per month
- You are charged per alarm metric (Standard Resolution and High Resolution)
- You are charged per GB of collected, archived and analyzed log data
- There is no Data Transfer IN charge, only Data Transfer Out.
- You are charged per million custom events and per million cross-account events
- Logs Insights is priced per query and charges based on the amount of ingested log data scanned by the query.

### Sources:

<https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/>

<https://aws.amazon.com/cloudwatch/features/>

<https://aws.amazon.com/cloudwatch/pricing/>

<https://aws.amazon.com/cloudwatch/faqs/>



## AWS OpsWorks

- A configuration management service that helps you configure and operate applications in a cloud enterprise by using **Puppet** or **Chef**.
- AWS OpsWorks Stacks and AWS OpsWorks for Chef Automate (1 and 2) let you use Chef cookbooks and solutions for configuration management, while OpsWorks for Puppet Enterprise lets you configure a Puppet Enterprise master server in AWS.
- **Limits**
  - Configuration management servers per region: 5
  - Automated backups per configuration management server: 30
  - Manual backups per configuration management server: 10

## OpsWorks for Puppet Enterprise

- Provides a fully-managed Puppet master, a suite of automation tools that enable you to inspect, deliver, operate, and future-proof your applications, and access to a user interface that lets you view information about your nodes and Puppet activities.
- Does not support all regions.
- Uses puppet-agent software.
- **Features**
  - AWS manages the Puppet master server running on an EC2 instance. You retain control over the underlying resources running your Puppet master.
  - You can choose the weekly maintenance window during which OpsWorks for Puppet Enterprise will automatically install updates.
  - Monitors the health of your Puppet master during update windows and automatically rolls back changes if issues are detected.
  - You can configure automatic backups for your Puppet master and store them in an S3 bucket in your account.
  - You can register new nodes to your Puppet master by inserting a user-data script, provided in the *OpsWorks for Puppet Enterprise StarterKit*, into your Auto Scaling groups.
  - Puppet uses SSL and a certification approval process when communicating to ensure that the Puppet master responds only to requests made by trusted users.
- Deleting a server also deletes its events, logs, and any modules that were stored on the server. Supporting resources are also deleted, along with all automated backups.
- **Pricing**
  - You are charged based on the number of nodes (servers running the Puppet agent) connected to your Puppet master and the time those nodes are running on an hourly rate, and you also pay for the underlying EC2 instance running your Puppet master.

## OpsWorks for Chef Automate



- Lets you create AWS-managed Chef servers that include Chef Automate premium features, and use the Chef DK and other Chef tooling to manage them.
- AWS OpsWorks for Chef Automate supports Chef Automate 2.
- Uses chef-client.
- **Features**
  - You can use Chef to manage both Amazon EC2 instances and on-premises servers running Linux or Windows.
  - You receive the full Chef Automate platform which includes premium features that you can use with Chef server, like Chef Workflow, Chef Visibility, and Chef Compliance.
  - You provision a managed Chef server running on an EC2 instance in your account. You retain control over the underlying resources running your Chef server and you can use Knife to SSH into your Chef server instance at any time.
  - You can set a weekly maintenance window during which OpsWorks for Chef Automate will automatically install updates.
  - You can configure automatic backups for your Chef server and is stored in an S3 bucket.
  - You can register new nodes to your Chef server by inserting user-data code snippets provided by OpsWorks for Chef Automate into your Auto Scaling groups.
  - Chef uses SSL to ensure that the Chef server responds only to requests made by trusted users. The Chef server and Chef client use bidirectional validation of identity when communicating with each other.
- Deleting a server also deletes its events, logs, and any cookbooks that were stored on the server. Supporting resources are deleted also, along with all automated backups.
- **Pricing**
  - You are charged based on the number of nodes connected to your Chef server and the time those nodes are running, and you also pay for the underlying EC2 instance running your Chef server.

## OpsWorks Stacks

- Provides a simple and flexible way to create and manage stacks and applications.
- **Stacks** are groups of AWS resources that constitute a full-stack application. By default, you can create up to 40 Stacks, and each stack can hold up to 40 layers, 40 instances, and 40 apps.
- You can create stacks that help you manage cloud resources in specialized groups called **layers**. A layer represents a set of EC2 instances that serve a particular purpose, such as serving applications or hosting a database server. Layers depend on Chef recipes to handle tasks such as installing packages on instances, deploying apps, and running scripts.
- OpsWorks Stacks does NOT require or create Chef servers.
- **Features**
  - You can deploy EC2 instances from template configurations, including EBS volume creation.



- You can configure the software on your instances on-demand or automatically based on lifecycle events, from bootstrapping the base OS image into a working server to modifying running services to reflect changes.
  - OpsWorks Stacks can auto heal your stack. If an instance fails in your stack, OpsWorks Stacks can replace it with a new one.
  - You can adapt the number of running instances to match your load, with time-based or load-based auto scaling.
  - You can use OpsWorks Stacks to configure and manage both Linux and Windows EC2 instances.
  - You can use AWS OpsWorks Stacks to deploy, manage, and scale your application on any Linux server such as EC2 instances or servers running in your own data center.
- **Instance Types**
    - **24/7 instances** are started manually and run until you stop them.
    - **Time-based instances** are run by OpsWorks Stacks on a specified daily and weekly schedule. They allow your stack to automatically adjust the number of instances to accommodate predictable usage patterns.
    - **Load-based instances** are automatically started and stopped by OpsWorks Stacks, based on specified load metrics, such as CPU utilization. They allow your stack to automatically adjust the number of instances to accommodate variations in incoming traffic.
      - Load-based instances are available only for Linux-based stacks.
  - **Lifecycle Events**
    - You can run recipes manually, but OpsWorks Stacks also lets you automate the process by supporting a set of five lifecycle events:
      - **Setup** occurs on a new instance after it successfully boots.
      - **Configure** occurs on all of the stack's instances when an instance enters or leaves the online state.
      - **Deploy** occurs when you deploy an app.
      - **Undeploy** occurs when you delete an app.
      - **Shutdown** occurs when you stop an instance.
  - **Monitoring**
    - OpsWorks Stacks sends all of your resource metrics to CloudWatch.
    - Logs are available for each action performed on your instances.
    - CloudTrail logs all API calls made to OpsWorks.
  - **Security**
    - Grant IAM users access to specific stacks, making management of multi-user environments easier.
    - You can also set user-specific permissions for actions on each stack, allowing you to decide who can deploy new application versions or create new resources.
    - Each EC2 instance has one or more associated **security groups** that govern the instance's network traffic. A security group has one or more rules, each of which specifies a particular category of allowed traffic.



- **Pricing**

- You pay for AWS resources created using OpsWorks Stacks in the same manner as if you created them manually.

**Sources:**

<https://aws.amazon.com/opsworks/chefautomate/features>  
<https://aws.amazon.com/opsworks/chefautomate/pricing>  
<https://aws.amazon.com/opsworks/chefautomate/faqs>  
<https://aws.amazon.com/opsworks/puppetenterprise/feature>  
<https://aws.amazon.com/opsworks/puppetenterprise/pricing>  
<https://aws.amazon.com/opsworks/puppetenterprise/faqs>  
<https://aws.amazon.com/opsworks/stacks/features>  
<https://aws.amazon.com/opsworks/stacks/pricing>  
<https://aws.amazon.com/opsworks/stacks/faqs>



## AWS Config

- A fully managed service that provides you with an AWS resource inventory, configuration history, and configuration change notifications to enable security and governance.

## Features

- Multi-account, multi-region data aggregation gives you an enterprise-wide view of your **Config rule** compliance status, and you can associate your AWS organization to quickly add your accounts.
- Provides you pre-built rules to evaluate your AWS resource configurations and configuration changes, or create your own custom rules in AWS Lambda that define your internal best practices and guidelines for resource configurations.
- **Config records** details of changes to your AWS resources to provide you with a configuration history, and automatically deliver it to an S3 bucket you specify.
- Receive a notification whenever a resource is created, modified, or deleted.
- Config enables you to record software configuration changes within your EC2 instances and servers running on-premises, as well as servers and Virtual Machines in environments provided by other cloud providers. You gain visibility into:
  - operating system configurations
  - system-level updates
  - installed applications
  - network configuration
- Config can provide you with a **configuration snapshot** - a point-in-time capture of all your resources and their configurations.
- Config discovers, maps, and tracks AWS resource relationships in your account.  
Ex. EC2 instances and associated security groups

## Concepts

### Configuration History

- A collection of the configuration items for a given resource over any time period, containing information such as when the resource was first created, how the resource has been configured over the last month, etc.
- Config automatically delivers a configuration history file for each resource type that is being recorded to an S3 bucket that you specify.
- A configuration history file is sent every six hours for each resource type that Config records.

### Configuration item

- A record of the configuration of a resource in your AWS account. Config creates a configuration item whenever it detects a change to a resource type that it is recording.
- The components of a configuration item include metadata, attributes, relationships, current configuration, and related events.



### Configuration Recorder

- Stores the configurations of the supported resources in your account as configuration items.
- By default, the configuration recorder records all supported resources in the region where Config is running. You can create a customized configuration recorder that records only the resource types that you specify.
- You can also have Config record supported types of *global resources* which are IAM users, groups, roles, and customer managed policies.

### Configuration Snapshot

- A complete picture of the resources that are being recorded and their configurations.
- Stored in an S3 bucket that you specify.

### Configuration Stream

- An automatically updated list of all configuration items for the resources that Config is recording.
- Helpful for observing configuration changes as they occur so that you can spot potential problems, generating notifications if certain resources are changed, or updating external systems that need to reflect the configuration of your AWS resources.

### Configuration Item

- The configuration of a resource at a given point-in-time. A CI consists of 5 sections:
  - Basic information about the resource that is common across different resource types.
  - Configuration data specific to the resource.
  - Map of relationships with other resources.
  - CloudTrail event IDs that are related to this state.
  - Metadata that helps you identify information about the CI, such as the version of this CI, and when this CI was captured.

### Resource Relationship

- Config discovers AWS resources in your account and then creates a map of relationships between AWS resources.

### Config rule

- Represents your desired configuration settings for specific AWS resources or for an entire AWS account.
- Provides customizable, predefined rules. If a resource violates a rule, Config flags the resource and the rule as noncompliant, and notifies you through Amazon SNS.
- Evaluates your resources either **in response to configuration changes** or **periodically**.



required-tags

**Description**: Checks whether your resources have the tags that you specify. For example, you can check whether your EC2 instances have the 'CostCenter' tag. Separate multiple values with commas.

**Trigger type**: Configuration changes

**Scope of changes**: Resources

**Resource types**:

- ACM Certificate
- AutoScaling AutoScalingGroup
- CloudFormation Stack
- CodeBuild Project
- DynamoDB Table
- EC2 CustomerGateway
- EC2 Instance
- EC2 InternetGateway
- EC2 NetworkAcl
- EC2 NetworkInterface
- EC2 RouteTable
- EC2 SecurityGroup
- EC2 Subnet
- EC2 Volume
- EC2 VPC
- EC2 VPNCConnection
- EC2 VPNGateway
- ElasticLoadBalancing LoadBalancer
- ElasticLoadBalancingV2 LoadBalancer
- RDS DBInstance
- RDS DBSecurityGroup
- RDS DBSnapshot
- RDS DBSubnetGroup
- RDS EventSubscription
- Redshift Cluster
- Redshift ClusterParameterGroup
- Redshift ClusterSecurityGroup
- Redshift ClusterSnapshot
- Redshift ClusterSubnetGroup
- S3 Bucket

**Auto remediation**: Off

**Config rule ARN**: arn:aws:config:us-east-1:██ config-rule/config-rule-2hxini

**Parameters**:

- tag1Key: Name
- tag1Value: null
- tag2Key: Owner
- tag2Value: null
- tag3Key: Environment
- tag3Value: null
- tag4Key: null
- tag4Value: null
- tag5Key: null
- tag5Value: null
- tag6Key: null
- tag6Value: null

**Overall rule status**: Not available

Choose resources in scope

Resources in scope represent those resources where this rule is being applied to and their compliance status.

**Compliance status**:

- Noncompliant

**Action buttons**:

- Remove exceptions
- Resource actions
- Remediate

Resource ID	Resource type	Resource compliance status	Action status
-------------	---------------	----------------------------	---------------

- Config deletes data older than your specified retention period. The default period is 7 years.
- Multi-Account Multi-Region Data Aggregation
  - An aggregator collects configuration and compliance data from the following:
    - Multiple accounts and multiple regions.
    - Single account and multiple regions.
    - An organization in AWS Organizations and all the accounts in that organization.

## Monitoring

- Use Amazon SNS to send you notifications every time a supported AWS resource is created, updated, or otherwise modified as a result of user API activity.
- Use Amazon CloudWatch Events to detect and react to changes in the status of AWS Config events.
- Use AWS CloudTrail to capture API calls to Config.

## Security

- Use IAM to create individual users for anyone who needs access to Config and grant different permissions to each IAM user.

## Compliances



- ISO
- PCI DSS
- HIPAA

## Pricing

- You are charged based on the number of configuration items recorded and on the number of AWS Config rules evaluations recorded, instead of the number of active rules in your account per region.. You are charged only once for recording the configuration item.

## Sources:

<https://docs.aws.amazon.com/config/latest/developerguide>

<https://aws.amazon.com/config/features/>

<https://aws.amazon.com/config/pricing/>

<https://aws.amazon.com/config/faq/>



## AWS Systems Manager

- Allows you to centralize operational data from multiple AWS services and automate tasks across your AWS resources.

### Features

- Create logical groups of resources such as applications, different layers of an application stack, or production versus development environments.
- You can select a resource group and view its recent API activity, resource configuration changes, related notifications, operational alerts, software inventory, and patch compliance status.
- Collects information about your instances and the software installed on them.
- Allows you to safely automate common and repetitive IT operations and management tasks across AWS resources.
- Provides a browser-based interactive shell and CLI for managing Windows and Linux EC2 instances, without the need to open inbound ports, manage SSH keys, or use bastion hosts. Administrators can grant and revoke access to instances through a central location by using IAM policies.
- Helps ensure that your software is up-to-date and meets your compliance policies.
- Lets you schedule windows of time to run administrative and maintenance tasks across your instances.

**SSM Agent** is the tool that processes Systems Manager requests and configures your machine as specified in the request. SSM Agent must be installed on each instance you want to use with Systems Manager. On newer AMIs and instance types, SSM Agent is installed by default. On older versions, you must install it manually.

### Capabilities

- **Automation**
  - Allows you to safely automate common and repetitive IT operations and management tasks across AWS resources
  - A **step** is defined as an initiated action performed in the Automation execution on a per-target basis. You can execute the entire Systems Manager automation document in one action or choose to execute one step at a time.
  - Concepts
    - **Automation document** - defines the Automation workflow.
    - **Automation action** - the Automation workflow includes one or more steps. Each step is associated with a particular action or plugin. The action determines the inputs, behavior, and outputs of the step.
    - **Automation queue** - if you attempt to run more than 25 Automations simultaneously, Systems Manager adds the additional executions to a queue and displays a status of *Pending*. When an Automation reaches a terminal state, the first execution in the queue starts.



- You can schedule Systems Manager automation document execution.
- **Resource Groups**
  - A collection of AWS resources that are all in the same AWS region, and that match criteria provided in a query.
  - Use Systems Manager tools such as *Automation* to simplify management tasks on your groups of resources. You can also use groups as the basis for viewing monitoring and configuration *insights* in Systems Manager.
- **Built-in Insights**
  - Show detailed information about a single, selected resource group.
  - Includes recent API calls through CloudTrail, recent configuration changes through Config, Instance software inventory listings, instance patch compliance views, and instance configuration compliance views.
- **Systems Manager Activation**
  - Enable hybrid and cross-cloud management. You can register any server, whether physical or virtual to be managed by Systems Manager.
- **Inventory Manager**
  - Automates the process of collecting software inventory from managed instances.
  - You specify the type of metadata to collect, the instances from where the metadata should be collected, and a schedule for metadata collection.
- **Configuration Compliance**
  - Scans your fleet of managed instances for patch compliance and configuration inconsistencies.
  - View compliance history and change tracking for Patch Manager patching data and State Manager associations by using AWS Config.
  - Customize Systems Manager Compliance to create your own compliance types.
- **Run Command**
  - Remotely and securely manage the configuration of your managed instances at scale.
  - **Managed Instances** - any EC2 instance or on-premises server or virtual machine in your hybrid environment that is configured for Systems Manager.
- **Session Manager**
  - Manage your EC2 instances through an interactive one-click browser-based shell or through the AWS CLI.
  - Makes it easy to comply with corporate policies that require controlled access to instances, strict security practices, and fully auditable logs with instance access details, while still providing end users with simple one-click cross-platform access to your Amazon EC2 instances.
  - You can use AWS Systems Manager Session Manager to tunnel SSH (Secure Shell) and SCP (Secure Copy) traffic between a client and a server.
- **Distributor**
  - Lets you package your own software or find AWS-provided agent software packages to install on Systems Manager managed instances.
  - After you create a package in Distributor, which creates an Systems Manager document, you can install the package in one of the following ways.



- One time by using Systems Manager Run Command.
  - On a schedule by using Systems Manager State Manager.
- **Patch Manager**
    - Automate the process of patching your managed instances.
    - Enables you to scan instances for missing patches and apply missing patches individually or to large groups of instances by using EC2 instance tags.
    - For security patches, Patch Manager uses *patch baselines* that include rules for auto-approving patches within days of their release, as well as a list of approved and rejected patches.
    - You can now use AWS Systems Manager Patch Manager to select and apply Microsoft application patches automatically across your Amazon EC2 or on-premises instances.
  - **Maintenance Window**
    - Set up recurring schedules for managed instances to execute administrative tasks like installing patches and updates without interrupting business-critical operations.
    - Supports running four types of tasks:
      - Systems Manager Run Command commands
      - Systems Manager Automation workflows
      - AWS Lambda functions
      - AWS Step Functions tasks
  - **Systems Manager Document (SSM)**
    - Defines the actions that Systems Manager performs.
    - Types of SSM Documents

Type	Use with	Details
Command document	Run Command, State Manager	Run Command uses command documents to execute commands. State Manager uses command documents to apply a configuration. These actions can be run on one or more targets at any point during the lifecycle of an instance.
Policy document	State Manager	Policy documents enforce a policy on your targets. If the policy document is removed, the policy action no longer happens.
Automation document	Automation	Use automation documents when performing common maintenance and deployment tasks such as creating or updating an AMI.
Package document	Distributor	In Distributor, a package is represented by a Systems Manager document. A package document includes attached ZIP archive files that contain software or assets to install on managed instances. Creating a package in Distributor creates the package document.



- Can be in JSON or YAML.
- You can create and save different versions of documents. You can then specify a default version for each document.
- If you want to customize the steps and actions in a document, you can create your own.
- You can tag your documents to help you quickly identify one or more documents based on the tags you've assigned to them.

### State Manager

- A service that automates the process of keeping your EC2 and hybrid infrastructure in a state that you define.
- A *State Manager association* is a configuration that is assigned to your managed instances. The configuration defines the state that you want to maintain on your instances. The association also specifies actions to take when applying the configuration.

### Parameter Store

- Provides secure, hierarchical storage for configuration data and secrets management.
- You can store values as plain text or encrypted data.
- Parameters work with Systems Manager capabilities such as Run Command, State Manager, and Automation.

### OpsCenter

- OpsCenter helps you view, investigate, and resolve operational issues related to your environment from a central location.
- OpsCenter complements existing case management systems by enabling integrations via Amazon Simple Notification Service (SNS) and public AWS SDKs. By aggregating information from AWS Config, AWS CloudTrail logs, resource descriptions, and Amazon CloudWatch Events, OpsCenter helps you reduce the mean time to resolution (MTTR) of incidents, alarms, and operational tasks.

### Monitoring

- SSM Agent writes information about executions, scheduled actions, errors, and health statuses to log files on each instance. For more efficient instance monitoring, you can configure either SSM Agent itself or the CloudWatch Agent to send this log data to CloudWatch Logs.
- Using CloudWatch Logs, you can monitor log data in real-time, search and filter log data by creating one or more metric filters, and archive and retrieve historical data when you need it.
- Log System Manager API calls with CloudTrail.

### Security

- Systems Managers is linked directly to IAM for access controls.



## Pricing

- For your own packages, you pay only for what you use. Upon transferring a package into Distributor, you will be charged based on the size and duration of storage for that package, the number of Get and Describe API calls made, and the amount of out-of-Region and on-premises data transfer out of Distributor for those packages.
- You are charged based on the number and type of Automation steps.

### Sources:

<https://docs.aws.amazon.com/systems-manager/latest/userguide>

<https://aws.amazon.com/systems-manager/features/>

<https://aws.amazon.com/systems-manager/pricing/>

<https://aws.amazon.com/systems-manager/faq/>



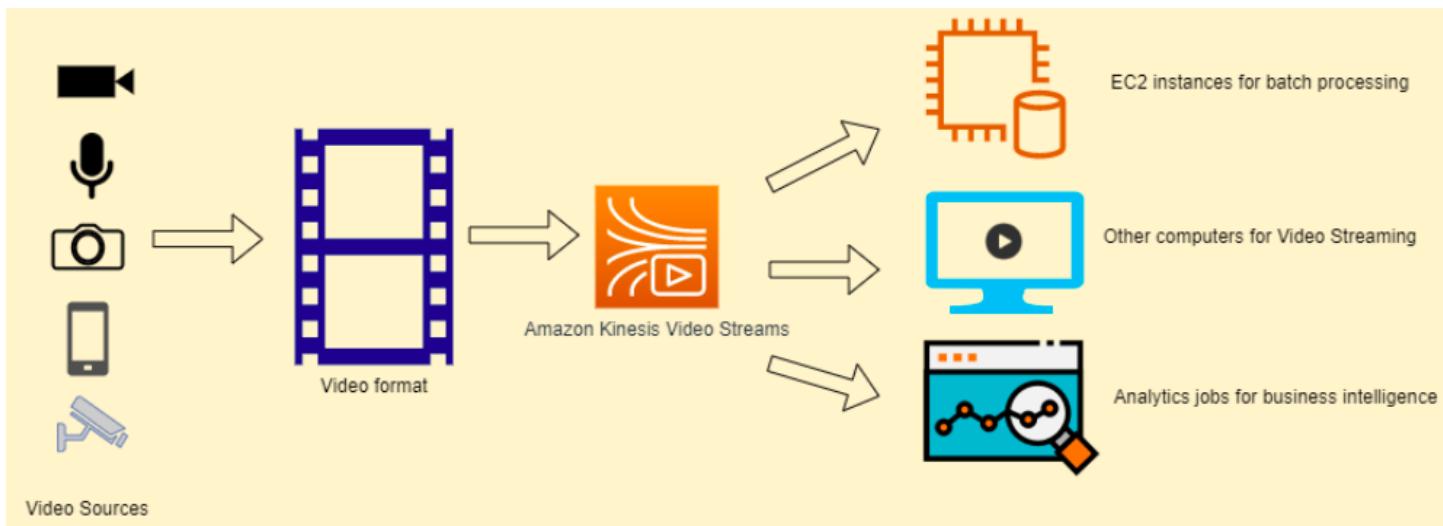
## ANALYTICS

### Amazon Kinesis

- For your own packages
- Makes it easy to collect, process, and analyze real-time, streaming data.
- Kinesis can ingest real-time data such as video, audio, application logs, website clickstreams, and IoT telemetry data for machine learning, analytics, and other applications.

### Kinesis Video Streams

- A fully managed AWS service that you can use to stream live video from devices to the AWS Cloud, or build applications for real-time video processing or batch-oriented video analytics.
- **Benefit**
  - You can connect and stream from millions of devices.
  - You can configure your Kinesis video stream to durably store media data for custom retention periods. Kinesis Video Streams also generates an index over the stored data based on producer-generated or service-side timestamps.
  - Kinesis Video Streams is serverless, so there is no infrastructure to set up or manage.
  - You can build real-time and batch applications on data streams.
  - Kinesis Video Streams enforces Transport Layer Security (TLS)-based encryption on data streaming from devices, and encrypts all data at rest using AWS KMS.
- **Components**
  - **Producer** – Any source that puts data into a Kinesis video stream.
  - **Kinesis video stream** – A resource that enables you to transport live video data, optionally store it, and make the data available for consumption both in real time and on a batch or ad hoc basis.
    - **Time-encoded data** is any data in which the records are in a time series, and each record is related to its previous and next records.
    - A **fragment** is a self-contained sequence of frames. The frames belonging to a fragment should have no dependency on any frames from other fragments.
    - Upon receiving the data from a producer, Kinesis Video Streams stores incoming media data as **chunks**. Each chunk consists of the actual media fragment, a copy of media metadata sent by the producer, and the Kinesis Video Streams-specific metadata such as the fragment number, and server-side and producer-side timestamps.
  - **Consumer** – Gets data, such as fragments and frames, from a Kinesis video stream to view, process, or analyze it. Generally these consumers are called Kinesis Video Streams applications.



- **Kinesis Video Streams provides**

- APIs for you to create and manage streams and read or write media data to and from a stream.
- A console that supports live and video-on-demand playback.
- A set of producer libraries that you can use in your application code to extract data from your media sources and upload to your Kinesis video stream.

- **Video Playbacks**

- You can view a Kinesis video stream using either
  - **HTTP Live Streaming (HLS)** - You can use HLS for live playback.
  - **GetMedia API** - You use the GetMedia API to build your own applications to process Kinesis video streams. *GetMedia* is a real-time API with low latency.

- **Metadata**

- Metadata is a mutable key-value pair. You can use it to describe the content of the fragment, embed associated sensor readings that need to be transferred along with the actual fragment, or meet other custom needs.
- There are two modes in which the metadata can be embedded with fragments in a stream:
  - Nonpersistent: You can affix metadata on an ad hoc basis to fragments in a stream, based on business-specific criteria that have occurred.
  - Persistent: You can affix metadata to successive, consecutive fragments in a stream based on a continuing need.

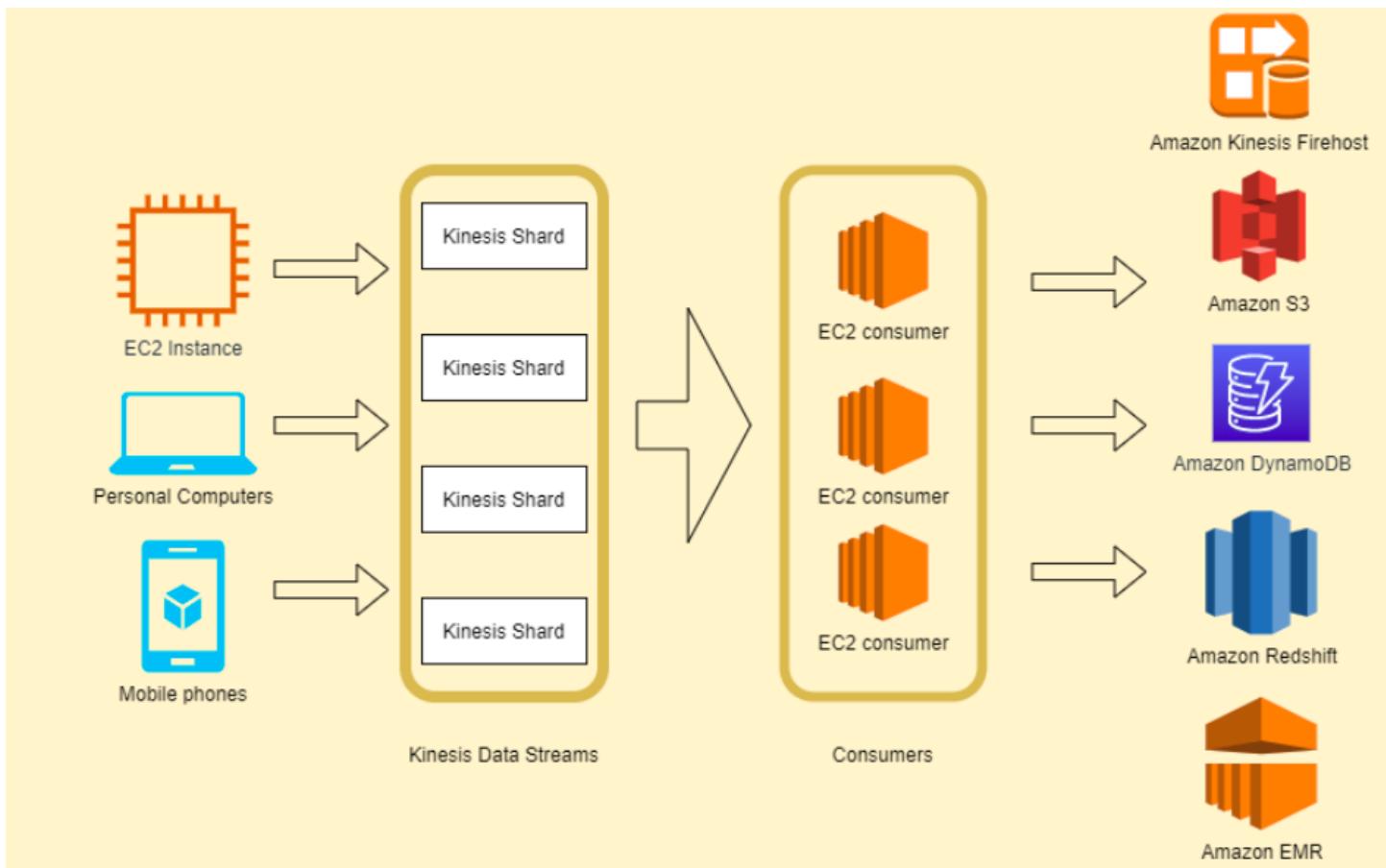
- **Pricing**

- You pay only for the volume of data you ingest, store, and consume through the service.

## Kinesis Data Stream



- A massively scalable, highly durable data ingestion and processing service optimized for streaming data. You can configure hundreds of thousands of data producers to continuously put data into a Kinesis data stream.
- **Concepts**
  - **Data Producer** - An application that typically emits data records as they are generated to a Kinesis data stream. Data producers assign partition keys to records. Partition keys ultimately determine which shard ingests the data record for a data stream.
  - **Data Consumer** - A distributed Kinesis application or AWS service retrieving data from all shards in a stream as it is generated. Most data consumers are retrieving the most recent data in a shard, enabling real-time analytics or handling of data.
  - **Data Stream** - A logical grouping of shards. There are no bounds on the number of shards within a data stream. A data stream will retain data for **24 hours, or up to 7 days** when extended retention is enabled.
  - **Shard** - The base throughput unit of a Kinesis data stream.
    - A shard is an append-only log and a unit of streaming capability. A shard contains an ordered sequence of records ordered by arrival time.
    - Add or remove shards from your stream dynamically as your data throughput changes.
    - One shard can ingest up to 1000 data records per second, or 1MB/sec. Add more shards to increase your ingestion capability.
    - When consumers use **enhanced fan-out**, one shard provides 1MB/sec data input and 2MB/sec data output for each data consumer registered to use enhanced fan-out.
    - When consumers do **not use enhanced fan-out**, a shard provides 1MB/sec of input and 2MB/sec of data output, and this output is shared with any consumer not using enhanced fan-out.
    - You will specify the number of shards needed when you create a stream and can change the quantity at any time.



- **Data Record**
  - A record is the unit of data stored in a Kinesis stream. A record is composed of a sequence number, partition key, and data blob.
  - A data blob is the data of interest your data producer adds to a stream. The maximum size of a data blob is 1 MB.
- **Partition Key**
  - A partition key is typically a meaningful identifier, such as a user ID or timestamp. It is specified by your data producer while putting data into a Kinesis data stream, and useful for consumers as they can use the partition key to replay or build a history associated with the partition key.
  - The partition key is also used to segregate and route data records to different shards of a stream.
- **Sequence Number**
  - A sequence number is a unique identifier for each data record. Sequence number is assigned by Kinesis Data Streams when a data producer calls *PutRecord* or *PutRecords* API to add data to a Kinesis data stream.



- **Amazon Kinesis Agent** is a pre-built Java application that offers an easy way to collect and send data to your Amazon Kinesis data stream.
- **Monitoring**
  - You can monitor shard-level metrics in Kinesis Data Streams.
  - You can monitor your data streams in Amazon Kinesis Data Streams using CloudWatch, Kinesis Agent, Kinesis libraries.
  - Log API calls with CloudTrail.
- **Security**
  - Kinesis Data Streams can automatically encrypt sensitive data as a producer enters it into a stream. Kinesis Data Streams uses AWS KMS master keys for encryption.
  - Use IAM for managing access controls.
  - You can use an interface VPC endpoint to keep traffic between your Amazon VPC and Kinesis Data Streams from leaving the Amazon network.
- **Pricing**
  - You are charged for each shard at an hourly rate.
  - PUT Payload Unit is charged with a per million PUT Payload Units rate.
  - When consumers use enhanced fan-out, they incur hourly charges per consumer-shard hour and per GB of data retrieved.
  - You are charged for an additional rate on each shard hour incurred by your data stream once you enable extended data retention.

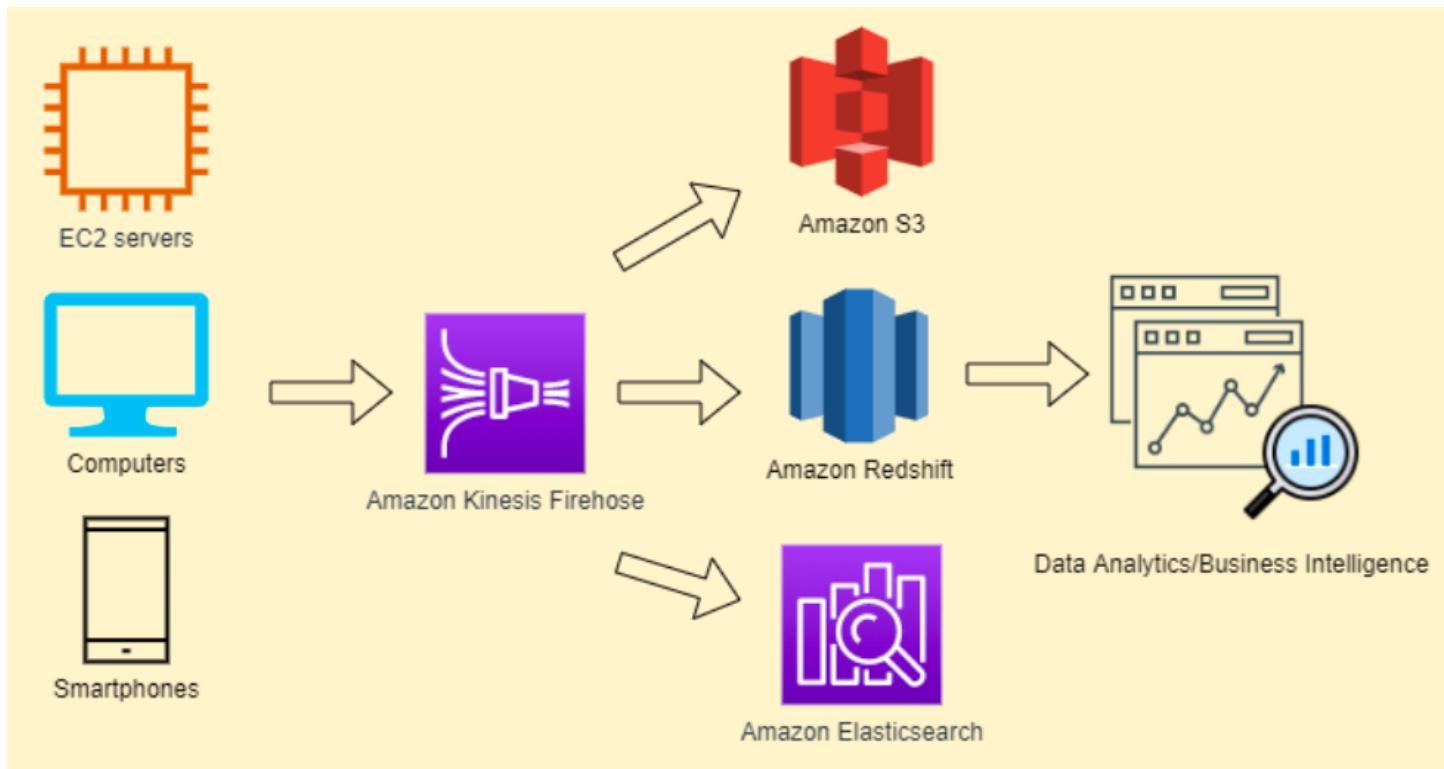
## Kinesis Data Firehose

- The easiest way to load streaming data into data stores and analytics tools.
- It is a fully managed service that automatically scales to match the throughput of your data.
- It can also batch, compress, and encrypt the data before loading it.
- **Features**
  - It can capture, transform, and load streaming data into S3, Redshift, Elasticsearch Service, and Splunk, enabling near real-time analytics with existing business intelligence tools and dashboards being used today.
  - You can specify a batch size or batch interval to control how quickly data is uploaded to destinations. Additionally, you can specify if data should be compressed.
  - Once launched, your delivery streams automatically scale up and down to handle gigabytes per second or more of input data rate, and maintain data latency at levels you specify for the stream.
  - Kinesis Data Firehose can convert the format of incoming data from JSON to Parquet or ORC formats before storing the data in S3.
  - You can configure Kinesis Data Firehose to prepare your streaming data before it is loaded to data stores. Kinesis Data Firehose provides pre-built Lambda blueprints for converting common

data sources such as Apache logs and system logs to JSON and CSV formats. You can use these pre-built blueprints without any change, or customize them further, or write your own custom functions.

- **Concepts**

- **Kinesis Data Firehose Delivery Stream** - The underlying entity of Kinesis Data Firehose. You use Kinesis Data Firehose by creating a Kinesis Data Firehose delivery stream and then sending data to it.
- **Record** - The data of interest that your data producer sends to a Kinesis Data Firehose delivery stream. A record can be as large as 1,000 KB.
- **Data Producer** - Producers send records to Kinesis Data Firehose delivery streams.
- **Buffer Size and Buffer Interval** - Kinesis Data Firehose buffers incoming streaming data to a certain size or for a certain period of time before delivering it to destinations. Buffer Size is in MBs and Buffer Interval is in seconds.



- **Stream Sources**

- You can send data to your Kinesis Data Firehose Delivery stream using different types of sources:
  - a Kinesis data stream,
  - the Kinesis Agent,
  - or the Kinesis Data Firehose API using the AWS SDK.
- You can also use CloudWatch Logs, CloudWatch Events, or AWS IoT as your data source.



- Some AWS services can only send messages and events to a Kinesis Data Firehose delivery stream that is in the same Region.

- **Data Delivery and Transformation**

- Kinesis Data Firehose can invoke your Lambda function to transform incoming source data and deliver the transformed data to destinations.
- Kinesis Data Firehose buffers incoming data up to 3 MB by default.
- If your Lambda function invocation fails because of a network timeout or because you've reached the Lambda invocation limit, Kinesis Data Firehose retries the invocation three times by default.
- Kinesis Data Firehose can convert the format of your input data from JSON to Apache Parquet or Apache ORC before storing the data in S3. Parquet and ORC are columnar data formats that save space and enable faster queries compared to row-oriented formats like JSON.
- Data delivery format:
  - **For data delivery to S3**, Kinesis Data Firehose concatenates multiple incoming records based on buffering configuration of your delivery stream. It then delivers the records to S3 as an S3 object.
  - **For data delivery to Redshift**, Kinesis Data Firehose first delivers incoming data to your S3 bucket in the format described earlier. Kinesis Data Firehose then issues an Redshift COPY command to load the data from your S3 bucket to your Redshift cluster.
  - **For data delivery to ElasticSearch**, Kinesis Data Firehose buffers incoming records based on buffering configuration of your delivery stream. It then generates an Elasticsearch bulk request to index multiple records to your Elasticsearch cluster.
  - **For data delivery to Splunk**, Kinesis Data Firehose concatenates the bytes that you send.
- Data delivery frequency
  - The frequency of data delivery to S3 is determined by the **S3 Buffer size** and **Buffer interval** value that you configured for your delivery stream.
  - The frequency of data COPY operations from S3 to Redshift is determined by how fast your Redshift cluster can finish the COPY command.
  - The frequency of data delivery to ElasticSearch is determined by the **Elasticsearch Buffer size** and **Buffer interval** values that you configured for your delivery stream.
  - Kinesis Data Firehose buffers incoming data before delivering it to Splunk. The buffer size is 5 MB, and the buffer interval is 60 seconds.

- **Monitoring**

- Kinesis Data Firehose exposes several metrics through the console, as well as CloudWatch for monitoring.
- Kinesis Agent publishes custom CloudWatch metrics, and helps assess whether the agent is healthy, submitting data into Kinesis Data Firehose as specified, and consuming the appropriate amount of CPU and memory resources on the data producer.
- Log API calls with CloudTrail.

- **Security**



- Kinesis Data Firehose provides you the option to have your data automatically encrypted after it is uploaded to the destination.
- Manage resource access with IAM.
- **Pricing**
  - You pay only for the volume of data you transmit through the service. You are billed for the volume of data ingested into Kinesis Data Firehose, and if applicable, for data format conversion to Apache Parquet or ORC.

## Kinesis Data Analytics

- Analyze streaming data, gain actionable insights, and respond to your business and customer needs in real time. You can quickly build SQL queries and Java applications using built-in templates and operators for common processing functions to organize, transform, aggregate, and analyze data at any scale.
- **General Features**
  - Kinesis Data Analytics is **serverless** and takes care of everything required to continuously run your application.
  - Kinesis Data Analytics elastically scales applications to keep up with any volume of data in the incoming data stream.
  - Kinesis Data Analytics delivers sub-second processing latencies so you can generate real-time alerts, dashboards, and actionable insights.
- **SQL Features**
  - Kinesis Data Analytics supports standard ANSI SQL.
  - Kinesis Data Analytics integrates with Kinesis Data Streams and Kinesis Data Firehose so that you can readily ingest streaming data.
  - SQL applications in Kinesis Data Analytics support two types of inputs:
    - A **streaming data source** is continuously generated data that is read into your application for processing.
    - A **reference data source** is static data that your application uses to enrich data coming in from streaming sources.
  - Kinesis Data Analytics provides an easy-to-use schema editor to discover and edit the structure of the input data. The wizard automatically recognizes standard data formats such as JSON and CSV.
  - Kinesis Data Analytics offers functions optimized for stream processing so that you can easily perform advanced analytics such as anomaly detection and top-K analysis on your streaming data.
- **Java Features**



- Kinesis Data Analytics includes open source libraries based on Apache Flink. Apache Flink is an open source framework and engine for building highly available and accurate streaming applications.
- You can use the Kinesis Data Analytics Java libraries to integrate with multiple AWS services.
- You can create and delete durable application backups through a simple API call. You can immediately restore your applications from the latest backup after a disruption, or you can restore your application to an earlier version.
- Java applications in Kinesis Data Analytics enable you to build applications whose processed records affect the results exactly once, referred to as **exactly once processing**.
- The service stores previous and in-progress computations, or state, in running application storage. State is always encrypted and incrementally saved in running application storage.
- An **application** is the primary resource in Kinesis Data Analytics. Kinesis data analytics applications continuously read and process streaming data in real time.
  - You write application code using SQL to process the incoming streaming data and produce output. Then, Kinesis Data Analytics writes the output to a configured destination.
  - You can also process and analyze streaming data using Java.
- **Components**
  - Input is the streaming source for your application. In the input configuration, you map the streaming source to an in-application data stream(s).
  - **Application code** is a series of SQL statements that process input and produce output.
  - You can create one or more in-application streams to store the **output**. You can then optionally configure an application output to persist data from specific in-application streams to an external destination.
- An **in-application data stream** is an entity that continuously stores data in your application for you to perform processing.
- Kinesis Data Analytics provisions capacity in the form of **Kinesis Processing Units (KPU)**. A single KPU provides you with the memory (4 GB) and corresponding computing and networking.
- For Java applications using Apache Flink, you build your application locally, and then make your application code available to the service by uploading it to an S3 bucket.
- Kinesis Data Analytics for Java applications provides your application 50 GB of running application storage per Kinesis Processing Unit. Kinesis Data Analytics scales storage with your application.
- Running application storage is used for **saving application state using checkpoints**. It is also accessible to your application code to use as **temporary disk for caching data** or any other purpose.
- **Pricing**
  - You are charged an hourly rate based on the average number of Kinesis Processing Units (or KPUs) used to run your stream processing application.
  - For Java applications, you are charged a single additional KPU per application for application orchestration. Java applications are also charged for running application storage and durable application backups.



**Sources:**

<https://aws.amazon.com/kinesis/>  
<https://docs.aws.amazon.com/kinesisvideostreams/latest/dg/>  
<https://aws.amazon.com/kinesis/video-streams/features/>  
<https://aws.amazon.com/kinesis/video-streams/pricing/>  
<https://aws.amazon.com/kinesis/video-streams/faqs/>  
<https://docs.aws.amazon.comstreams/latest/dev/>  
<https://aws.amazon.com/kinesis/data-streams/pricing/>  
<https://aws.amazon.com/kinesis/data-streams/faqs/>  
<https://docs.aws.amazon.com/firehose/latest/dev/>  
<https://aws.amazon.com/kinesis/data-firehose/pricing/>  
<https://aws.amazon.com/kinesis/data-firehose/faqs/>  
<https://docs.aws.amazon.com/kinesisanalytics/latest/dev>  
<https://docs.aws.amazon.com/kinesisanalytics/latest/java>  
<https://aws.amazon.com/kinesis/data-analytics/features/>  
<https://aws.amazon.com/kinesis/data-analytics/pricing/>  
<https://aws.amazon.com/kinesis/data-analytics/faqs/>



## Amazon Athena

- An interactive query service that makes it easy to analyze data **directly** in S3 using standard SQL.

### Features

- Athena is **serverless**.
- Has a built-in query editor.
- Uses *Presto*, an open source, distributed SQL query engine optimized for low latency, ad hoc analysis of data.
- Athena supports a wide variety of data formats such as CSV, JSON, ORC, Avro, or Parquet.
- Athena automatically executes queries in **parallel**, so that you get query results in seconds, even on large datasets.
- Athena uses Amazon S3 as its underlying data store, making your data highly available and durable.
- Athena integrates with Amazon QuickSight for easy data visualization.
- Athena integrates out-of-the-box with AWS Glue.

Athena uses a managed **Data Catalog** to store information and schemas about the databases and tables that you create for your data stored in S3.

### Partitioning

- By partitioning your data, you can restrict the amount of data scanned by each query, thus improving performance and reducing cost.
- Athena leverages *Hive* for partitioning data.
- You can partition your data by any key.

### Queries

- You can query geospatial data.
- You can query different kinds of logs as your datasets.
- Athena stores query results in S3.
- Athena retains query history for 45 days.
- Athena does not support user-defined functions, *INSERT INTO* statements, and stored procedures.
- Athena supports both simple data types such as INTEGER, DOUBLE, VARCHAR and complex data types such as MAPS, ARRAY and STRUCT.
- Athena supports querying data in Amazon S3 Requester Pays buckets.

### Security

- Control access to your data by using IAM policies, access control lists, and S3 bucket policies.



- If the files in the target S3 bucket is encrypted, you can perform queries on the encrypted data itself.

## Pricing

- You pay only for the queries that you run. You are charged based on the amount of data scanned by each query.
- You are not charged for failed queries.
- You can get significant cost savings and performance gains by compressing, partitioning, or converting your data to a columnar format, because each of those operations reduces the amount of data that Athena needs to scan to execute a query.

## Sources:

<https://docs.aws.amazon.com/athena/latest/ug/>

<https://aws.amazon.com/athena/features>

<https://aws.amazon.com/athena/pricing>

<https://aws.amazon.com/athena/faqs>



## Amazon Elasticsearch

- Amazon ES lets you search, analyze, and visualize your data **in real-time**. This service manages the capacity, scaling, patching, and administration of your Elasticsearch clusters for you, while still giving you direct access to the Elasticsearch APIs.
- The service offers open-source Elasticsearch APIs, managed Kibana, and integrations with Logstash and other AWS Services. This combination is often coined as the **ELK Stack**.
- Concepts
  - An Amazon ES **domain** is synonymous with an Elasticsearch cluster. Domains are clusters with the settings, instance types, instance counts, and storage resources that you specify.

A domain is the collection of resources needed to run Elasticsearch. The domain name will be part of your domain endpoint.

Elasticsearch domain name  The name must start with a lowercase letter and must be between 3 and 28 characters. Valid characters are a-z (lowercase only), 0-9, and - (hyphen).

### Data nodes

Select an instance type that corresponds to the compute, memory, and storage needs of your application. Consider the size of your Elasticsearch indices, number of shards and replicas, type of queries, and volume of requests. [Learn more](#)

Availability zones  3-AZ Recommended for production workloads with higher availability requirements.  
 2-AZ Suitable for production workloads.

Instance type  r5.large.elasticsearch instance type needs EBS storage.

Number of nodes  For three Availability Zones, we recommend instances in multiples of three for equal distribution across the Availability Zones.



### Data nodes storage

Choose a storage type for your data nodes. If you choose the EBS storage type, multiply the EBS storage size per node by the number of data nodes in your cluster to calculate the total storage available to your cluster. Storage settings do not apply to any dedicated master nodes in the cluster.

Data nodes storage type  ⓘ

EBS volume type\*  ⓘ

EBS storage size per node\*  ⓘ  
Total cluster size will be 30 GiB (EBS volume size x Instance count).

### Dedicated master nodes

Dedicated master nodes improve the stability of your domain. For production domains, we recommend three.

Dedicated master nodes  Enable ⓘ

Instance type  ⓘ

Number of master nodes  ⓘ

- You can create multiple Elasticsearch indices within the same domain. Elasticsearch automatically distributes the indices and any associated replicas between the instances allocated to the domain.
- Amazon ES uses a **blue/green deployment process** when updating domains. Blue/green typically refers to the practice of running two production environments, one live and one idle, and switching the two as you make software changes.
- When new service software becomes available, you can request an update to your domain and benefit from new features immediately. Some updates are required, and others are optional.
- If you take no action on required updates, AWS updates the service software automatically after a certain timeframe (typically two weeks).
- The alerting feature notifies you when data from one or more Elasticsearch indices meets certain conditions, such as receiving HTTP 503 errors.
- New SQL support enables you to query your domain using the familiar SQL syntax without compromising on Elasticsearch's full-text search and scoring capabilities. With SQL support, you can query your data using aggregations, group by, and where clauses to investigate your data.
- Storage
  - You can choose between local on-instance storage (up to 3 PB) or Amazon EBS volumes to store your Elasticsearch indices.
  - You can build data durability for your Amazon Elasticsearch domain through automated and manual snapshots. By default, the service will automatically create daily snapshots of each



domain and retain them for 14 days. The automated snapshots are stored free of charge in Amazon S3, while the manual snapshots will incur standard Amazon S3 usage charges.

- Snapshots backup a cluster's data and state, includes cluster settings, node information, index settings, and shard allocation.
- You cannot use automated snapshots to migrate to new domains. For migrations, you must use manual snapshots stored in your S3 bucket.
- Data Ingestion
  - Easily ingest structured and unstructured data into your Amazon Elasticsearch domain with **Logstash**, an open-source data pipeline that helps you process logs and other event data.
  - You can also ingest data into your Amazon Elasticsearch domain using Amazon Kinesis Firehose, AWS IoT, or Amazon CloudWatch Logs.
  - You can get faster and better insights into your data using **Kibana**, an open-source analytics and visualization platform. Kibana is automatically deployed with your Amazon Elasticsearch Service domain.
  - You can load streaming data from the following sources using AWS Lambda event handlers:
    - Amazon S3
    - Amazon Kinesis Data Streams and Data Firehose
    - Amazon DynamoDB
    - Amazon CloudWatch
    - AWS IoT
  - Amazon ES exposes three Elasticsearch logs through CloudWatch Logs:
    - error logs
    - search slow logs - These logs help fine tune the performance of any kind of search operation on Elasticsearch.
    - index slow logs - These logs provide insights into the indexing process and can be used to fine-tune the index setup.
  - Indexing
    - Before you can search data, you must index it. **Indexing** is the method by which search engines organize data for fast retrieval.
    - In Elasticsearch, the basic unit of data is a **JSON document**.
    - Within an index, Elasticsearch organizes documents into **types** (arbitrary data categories that you define) and identifies them using a unique ID.
  - Kibana and Logstash
    - Kibana is a popular open source visualization tool designed to work with Elasticsearch.
    - The URL is `elasticsearch-domain-endpoint/_plugin/kibana/`.
    - You can configure your own Kibana instance aside from using the default provided Kibana.
    - Amazon ES uses **Amazon Cognito** to offer username and password protection for Kibana. (Optional feature)



- Logstash provides a convenient way to use the bulk API to upload data into your Amazon ES domain with the S3 plugin. The service also supports all other standard Logstash input plugins that are provided by Elasticsearch.
- Amazon ES also supports two Logstash output plugins:
  - standard Elasticsearch plugin
  - *logstash-output-amazon-es* plugin, which signs and exports Logstash events to Amazon ES.
- Security
  - Amazon ES is HIPAA eligible and compliant with PCI DSS, SOC and ISO standards.
  - You can securely connect your applications to your managed Elasticsearch environment from your VPC or via the public Internet, configuring network access using VPC security groups or IP-based access policies.
  - Securely authenticate your users and control access using Amazon Cognito and AWS IAM.
  - Has built-in encryption of data at-rest and in-transit to protect your data both when it is stored in your domain or in automated snapshots, and when it is transferred between nodes in your domain.
  - You can create three types of policies to control access to domains:
    - Resource-based Policies - attached to domains. These policies specify which actions a principal can perform on the domain's subresources, which include Elasticsearch indices and APIs.
    - Identity-based policies - attached to IAM users or roles.
    - IP-based Policies - restrict access to a domain to one or more IP addresses or CIDR blocks.
  - Placing an Amazon ES domain within a VPC enables secure communication between Amazon ES and other services within the VPC without the need for an internet gateway, NAT device, or VPN connection. All traffic remains securely within the AWS Cloud.
- High Availability
  - You can deploy your Amazon ES instances across multiple AZs (**up to three**). If you enable replicas for your indexes, the shards will automatically be distributed such that you have cross-zone replication.
  - If one or more instances in an AZ are unreachable or not functional, Amazon ES automatically tries to bring up new instances in the same AZ to replace the affected instances.
  - When enabling Multi-AZ, you should create at least one replica for each index in your cluster. Without replicas, Amazon ES can't distribute copies of your data to other Availability Zones.
  - Even if you select two Availability Zones when configuring your domain, Amazon ES automatically distributes dedicated master nodes **across three Availability Zones**. This distribution helps prevent cluster downtime if a zone experiences a service disruption. It



- 
- also assists in electing a new master node through a quorum between the two remaining nodes.
- Amazon Elasticsearch Service has increased its snapshot frequency from daily to hourly and are retained for 14 days at no extra charge.
  - Limitations
    - You can either launch your domain within a VPC or use a public endpoint, but not in both.
    - You cannot switch your domain from a VPC to a public endpoint and vice versa.
    - You can't launch your domain within a VPC that uses dedicated tenancy.
    - You cannot move your domain to a different VPC that where it was initially launched in.
    - To access the default installation of Kibana for a domain that resides within a VPC, users must have access to the VPC.
  - Pricing
    - You pay for each hour of use of an EC2 instance and for the cumulative size of any EBS storage volumes attached to your instances.
    - You can use Reserved Instances to reduce long term cost on your EC2 instances.
  - Use Cases
    - Log Analytics
    - Real-Time Application Monitoring
    - Security Analytics
    - Full Text Search
    - Clickstream Analytics

#### Sources:

<https://aws.amazon.com/elasticsearch-service/features/?nc=sn&loc=2>

<https://docs.aws.amazon.com/elasticsearch-service/latest/developerguide/what-is-amazon-elasticsearch-service.html>



## Amazon Quicksight

Amazon QuickSight is a cloud-powered business analytics service that makes it easy to build visualizations, perform ad-hoc analysis, and quickly get business insights from their data, anytime, on any device.

- Features
  - Provides **ML Insights** for discovering hidden trends and outliers, identify key business drivers, and perform powerful what-if analysis and forecasting.
  - Has a wide library of visualizations, charts, and tables; You can add interactive features like drill-downs and filters, and perform automatic data refreshes to build interactive dashboards.
  - Allows you to schedule automatic email-based reports, so you can get key insights delivered to your inbox.
  - QuickSight allows users to connect to data sources, create/edit datasets, create visual analyses, invite co-workers to collaborate on analyses, and publish dashboards and reports.
  - Has a super-fast, parallel, in-memory, calculation engine (**SPICE**), allowing you to achieve blazing fast performance at scale.
  - Allows you to directly connect to and import data from a wide variety of cloud and on-premises data sources.
- SPICE
  - Uses a combination of columnar storage, in-memory technologies.
  - Data in SPICE is persisted until it is explicitly deleted by the user.
  - SPICE also automatically replicates data for high availability and enables QuickSight to scale easily.
- Concepts
  - A **QuickSight Author** is a user who can connect to data sources (within AWS or outside), create interactive dashboards using advanced QuickSight capabilities, and publish dashboards with other users in the account.
  - A **QuickSight Reader** is a user who uses interactive dashboards. Readers can log in via QuickSight username/password, SAML portal or AD auth, view shared dashboards, filter data, drill down to details or export data as a CSV file.
    - Readers can be easily upgraded to authors via the QuickSight user management options.
    - Readers with pay-per-session pricing only exist in Enterprise Edition. Standard Edition accounts can be easily upgraded to Enterprise.
  - A **QuickSight Admin** is a user who can manage QuickSight users and account-level preferences, as well as purchase SPICE capacity and annual subscriptions for the account.
    - Admins have all QuickSight authoring capabilities.
    - Admins can also upgrade Standard Edition accounts to Enterprise Edition.
    - QuickSight Authors and Readers can be upgraded to Admins at any time.
  - A **QuickSight Reader session** has a 30-minute duration and is renewed at 30-minute intervals. The session starts with a user-initiated action (login, dashboard load, page refresh, drill-down or filtering).



- **Dashboards** are a collection of visualizations, tables, and other visual displays arranged and visible together.
- **Stories** are guided tours through specific views of an analysis. They are used to convey key points, a thought process, or the evolution of an analysis for collaboration.
- Data Management
  - *Data preparation* is the process of transforming raw data for use in an analysis.
  - You can upload XLSX, CSV, TSV, CLF, XLF data files directly from Amazon QuickSight website, or to an Amazon S3 bucket and point Quicksight to the bucket.
  - You can also connect Amazon QuickSight to an Amazon EC2 or on-premises database.
- Data Visualization and Analysis
  - A *visual*, also known as a data visualization, is a graphical representation of a data set using a type of diagram, chart, graph, or table. All visuals begin in AutoGraph mode, which automatically selects a visualization based on the fields you select.
  - A *data analysis* is the basic workspace for creating and interacting with visuals, which are graphical representations of your data. Each analysis contains a collection of visuals that you assemble and arrange for your purposes.
  - To create a visualization, start by selecting the data fields you want to analyze, or drag the fields directly on to the visual canvas, or a combination of both actions. Amazon QuickSight will automatically select the appropriate visualization to display based on the data you've selected.
  - Amazon QuickSight has a feature called **AutoGraph** that allows it to select the most appropriate visualizations based on the properties of the data, such as cardinality and data type.
  - You can perform typical arithmetic and comparison functions; conditional functions such as if,then; and date, numeric, and string calculations.
- Machine Learning Insights
  - Using machine learning and natural language capabilities, Amazon QuickSight Enterprise Edition launches you into **forecasting** and **decision-making**.
  - You can select from a list of customized context-sensitive narratives, called *auto-narratives*, and add them to your analysis. In addition to choosing auto-narratives, you can choose to view forecasts, anomalies, and factors contributing to these.
  - Major features
    - ML-powered anomaly detection - continuously analyze all your data to detect anomalies.
    - ML-powered forecasting - forecast key business metrics.
    - Auto-narratives - build rich dashboards with embedded narratives to tell the story of your data in plain language.
- Security
  - Offers role-based access control, Active Directory integration, CloudTrail auditing, single sign-on, private VPC subnets, and data backup.
  - FedRamp, HIPAA, PCI PSS, ISO, and SOC compliant.



- Row-level security enables QuickSight dataset owners to control access to data at row granularity based on permissions associated with the user interacting with the data.
- Pricing
  - Quicksight has a pay-per-session model for dashboard readers, users who consume dashboards others have created.

**Sources:**

<https://aws.amazon.com/quicksight/resources/faqs/>



## Amazon EMR

- A managed cluster platform that simplifies running big data frameworks, such as Apache Hadoop and Apache Spark, on AWS to process and analyze vast amounts of data.
- You can process data for analytics purposes and business intelligence workloads using EMR together with Apache Hive and Apache Pig.
- You can use EMR to transform and move large amounts of data into and out of other AWS data stores and databases.

## Features

- EMR Notebooks provide a managed environment, based on Jupyter Notebooks, to help users prepare and visualize data, collaborate with peers, build applications, and perform interactive analysis using EMR clusters.
- EMR enables you to quickly and easily provision as much capacity as you need, and automatically or manually add and remove capacity.
- You can leverage multiple data stores, including S3, the Hadoop Distributed File System (HDFS), and DynamoDB.
- EMR supports powerful and proven *Hadoop* tools such as *Hive*, *Pig*, *HBase*, and *Impala*. Additionally, it can run distributed computing frameworks besides *Hadoop MapReduce* such as *Spark* or *Presto* using bootstrap actions. You can also use *Hue* and *Zeppelin* as GUIs for interacting with applications on your cluster.

## Components

- **Clusters** - A collection of EC2 instances. You can create two types of clusters:
  - a **transient cluster** that auto-terminates after steps complete.
  - a **long-running cluster** that continues to run until you terminate it deliberately.
- **Nodes** - Each EC2 instance in a cluster is called a node.
- **Node Type** - Each node has a role within the cluster, referred to as the node type. The node types are:
  - **Master node**: A node that manages the cluster by running software components to coordinate the distribution of data and tasks among other nodes for processing. The master node tracks the status of tasks and monitors the health of the cluster. Every cluster has a master node, and it's possible to create a single-node cluster with only the master node. Does not support automatic failover.
  - **Core node**: A node with software components that run tasks and store data in the Hadoop Distributed File System (HDFS) on your cluster. Multi-node clusters have at least one core node. EMR is fault tolerant for slave failures and continues job execution if a slave node goes down.
  - **Task node**: A node with software components that only runs tasks and does not store data in HDFS. Task nodes are optional.



## EMR Architecture

- **Storage** - this layer includes the different file systems that are used with your cluster.
  - Hadoop Distributed File System (HDFS) - a distributed, scalable file system for Hadoop.
    - HDFS distributes the data it stores across instances in the cluster, storing multiple copies of data on different instances to ensure that no data is lost if an individual instance fails.
    - HDFS is ephemeral storage.
    - HDFS is useful for caching intermediate results during MapReduce processing or for workloads that have significant random I/O.
  - EMR File System (EMRFS) - With EMRFS, EMR extends Hadoop to directly be able to access data stored in S3 as if it were a file system.
  - Local File System - refers to a locally connected disk.
    - Each EC2 node in your cluster comes with a pre-configured instance store, which persists only on the lifetime of the EC2 instance.
- **Cluster Resource Management** - this layer is responsible for managing cluster resources and scheduling the jobs for processing data.
  - By default, Amazon EMR uses YARN, which is a component introduced in Apache Hadoop 2.0 to centrally manage cluster resources for multiple data-processing frameworks.
  - EMR has an agent on each node that administers YARN components, keeps the cluster healthy, and communicates with EMR.
- **Data Processing Frameworks** - this layer is the engine used to process and analyze data.
  - Hadoop MapReduce - an open-source programming model for distributed computing.
  - Apache Spark - a cluster framework and programming model for processing big data workloads.



Create Cluster - Quick Options [Go to advanced options](#)

**General Configuration**

Cluster name  [?](#)

Logging [?](#)

Launch mode  Cluster [?](#)  Step execution [?](#)

**Software configuration**

Release  [?](#)

Applications  Core Hadoop: Hadoop 2.8.5 with Ganglia 3.7.2, Hive 2.3.6, Hue 4.4.0, Mahout 0.13.0, Pig 0.17.0, and Tez 0.9.2

HBase: HBase 1.4.10 with Ganglia 3.7.2, Hadoop 2.8.5, Hive 2.3.6, Hue 4.4.0, Phoenix 4.14.3, and ZooKeeper 3.4.14

Presto: Presto 0.227 with Hadoop 2.8.5 HDFS and Hive 2.3.6 Metastore

Spark: Spark 2.4.4 on Hadoop 2.8.5 YARN with Ganglia 3.7.2 and Zeppelin 0.8.2

Use AWS Glue Data Catalog for table metadata [?](#)

**Hardware configuration**

Instance type  [?](#) The selected instance type adds 64 GiB of GP2 EBS storage per instance by default. [Learn more](#) [?](#)

Number of Instances  (1 master and 2 core nodes)

**Security and access**

EC2 key pair  [?](#) Learn how to create an EC2 key pair.

Permissions  Default  Custom  
Use default IAM roles. If roles are not present, they will be automatically created for you with managed policies for automatic policy updates.

EMR role [EMR\\_DefaultRole](#) [?](#) [?](#)

EC2 instance profile [EMR\\_EC2\\_DefaultRole](#) [?](#) [?](#)

[Cancel](#) [Create cluster](#)

## Data Processing

- Ways to process data in your EMR cluster:
  - **Submitting Jobs Directly to Applications** - Submit jobs and interact directly with the software that is installed in your EMR cluster. To do this, you connect to the master node over a secure connection and access the interfaces and tools that are available for the software that runs directly on your cluster.
  - **Running Steps to Process Data** - Submit one or more ordered steps to an EMR cluster. Each step is a unit of work that contains instructions to manipulate data for processing by software installed on the cluster.



## Scaling

- There are two main options for adding or removing capacity:
  - **Deploy multiple clusters:** If you need more capacity, you can easily launch a new cluster and terminate it when you no longer need it. There is no limit to how many clusters you can have.
  - **Resize a running cluster:** You may want to scale out a cluster to temporarily add more processing power to the cluster, or scale in your cluster to save on costs when you have idle capacity. When adding instances to your cluster, EMR can now start utilizing provisioned capacity as soon it becomes available. When scaling in, EMR will proactively choose idle nodes to reduce impact on running jobs.

## Deployment

- Choose the instance size and type that best suits the processing needs for your cluster
  - Batch processing
  - Low-latency queries
  - Streaming data
  - Large data storage
- In addition to the standard software and applications that are available for installation on your cluster, you can use bootstrap actions to install custom software.

## EMR Notebooks

- A serverless Jupyter notebook.
- An EMR cluster is required to execute the code and queries within an EMR notebook, but the notebook is not locked to the cluster.
- Runs Apache Spark.

## Managing Clusters

- A **cluster step** is a user-defined unit of processing, mapping roughly to one algorithm that manipulates the data.
- When creating a cluster, typically you should select the Region where your data is located.
- You can connect to the master node only while the cluster is running. When the cluster terminates, the EC2 instance acting as the master node is terminated and is no longer available.
- By default, the *ElasticMapReduce-master* security group does not permit inbound SSH access.
- You can set termination protection on a cluster.
- You can adjust the number of EC2 instances available to an EMR cluster automatically or manually in response to workloads that have varying demands.
- The EMR File System (EMRFS) is an implementation of HDFS that all EMR clusters use for reading and writing regular files from EMR directly to S3. It provides the convenience of storing persistent data in S3 for use with Hadoop while also providing features like consistent view and data encryption.



- You can add tags to your clusters.
- You can launch an EMR cluster with three master nodes to enable high availability for EMR applications. Amazon EMR automatically fails over to a standby master node if the primary master node fails or if critical processes.

## High Availability

- You can launch an EMR cluster with three master nodes and support high availability for HBase clusters on EMR. Amazon EMR automatically fails over to a standby master node if the primary master node fails or if critical processes such as Resource Manager or Name Node crash.
- In the event of a failover, Amazon EMR automatically replaces the failed master node with a new master node with the same configuration and boot-strap actions.

## Monitoring

- EMR integrates with CloudTrail to log information about requests made by or on behalf of your AWS account.
- You can monitor and interact with your cluster by forming a secure connection between your remote computer and the master node by using SSH.
- EMR provides the ability to archive log files in S3 so you can store logs and troubleshoot issues even after your cluster terminates.
- EMR also provides an optional debugging tool.
- EMR integrates with CloudWatch to track performance metrics for the cluster and jobs within the cluster.

## Security

- EMR integrates with IAM to manage permissions. You define permissions using IAM policies, which you attach to IAM users or IAM groups. The permissions that you define in the policy determine the actions that those users or members of the group can perform and the resources that they can access.
- EMR uses IAM roles for the EMR service itself and the EC2 instance profile for the instances. These roles grant permissions for the service and instances to access other AWS services on your behalf. There is a default role for the EMR service and a default role for the EC2 instance profile.
- EMR uses security groups to control inbound and outbound traffic to your EC2 instances. When you launch your cluster, EMR uses a security group for your master instance and a security group to be shared by your core/task instances.
- EMR supports optional S3 server-side and client-side encryption with EMRFS to help protect the data that you store in S3.
- EMR supports launching clusters in a VPC.
- EMR release version 5.10.0 and later supports Kerberos, which is a network authentication protocol

## Pricing



- You pay a per-second rate for every second for each node you use, with a one-minute minimum.
- The EMR price is in addition to the EC2 price (the price for the underlying servers) and EBS price (if attaching EBS volumes).

**Sources:**

<https://docs.aws.amazon.com/emr/latest/ManagementGuide>

<https://aws.amazon.com/emr/features>

<https://aws.amazon.com/emr/pricing>

<https://aws.amazon.com/emr/faqs>



## APPLICATION

### Amazon SQS

- A hosted queue that lets you integrate and decouple distributed software systems and components.
- SQS supports both **standard** and **FIFO queues**.
- SQS uses pull based (polling) not push based
- Users can access Amazon SQS from their VPC using VPC endpoints, without using public IPs, and without needing to traverse the public internet. VPC endpoints for Amazon SQS are powered by AWS PrivateLink.
- **Benefits**
  - You control who can send messages to and receive messages from an SQS queue.
  - Supports server-side encryption.
  - SQS stores messages on multiple servers for durability.
  - SQS uses redundant infrastructure to provide highly-concurrent access to messages and high availability for producing and consuming messages.
  - SQS can scale to process each buffered request and handle any load increases or spikes independently.
  - SQS locks your messages during processing, so that multiple producers can send and multiple consumers can receive messages at the same time.
- **Types of Queues**



Standard Queue	FIFO Queue
Available in all regions	Available in the US East (N. Virginia), US East (Ohio) US West (Oregon), EU (Ireland), Asia Pacific (Sydney), and Asia Pacific (Tokyo) regions.
<b>Unlimited Throughput</b> - Standard queues support a nearly unlimited number of transactions per second (TPS) per action.	<b>High Throughput</b> - By default, FIFO queues support up to 3,000 messages per second with batching. (Can request a limit increase). FIFO queues support up to 300 messages per second (300 send, receive, or delete operations per second) without batching.
<b>At-Least-Once Delivery</b> - A message is delivered at least once, but occasionally more than one copy of a message is delivered.	<b>Exactly-Once Processing</b> - A message is delivered once and remains available until a consumer processes and deletes it. Duplicates aren't introduced into the queue.
<b>Best-Effort Ordering</b> - Occasionally, messages might be delivered in an order different from which they were sent.	<b>First-in-First-Out Delivery</b> - The order in which messages are sent and received is strictly preserved.
Send data between applications when the throughput is important.	Send data between applications when the order of events is important.



- You can include structured metadata (such as timestamps, geospatial data, signatures, and identifiers) with messages using **message attributes**.
- **Message timers** let you specify an initial invisibility period for a message added to a queue. The default (minimum) invisibility period for a message is 0 seconds. The maximum is 15 minutes.
- SQS doesn't automatically delete a message after receiving it for you, in case you don't successfully receive the message.
- You can subscribe one or more SQS queues to an Amazon SNS topic from a list of topics available for the selected queue.
- You can configure an existing SQS queue to trigger an AWS Lambda function when new messages arrive in a queue.
  - Your queue and Lambda function must be in the same AWS Region.
  - FIFO queues don't support Lambda function triggers.



- You can associate only one queue with one or more Lambda functions.
- You can't associate an encrypted queue that uses an AWS managed Customer Master Key for SQS with a Lambda function in a different AWS account.
- You can delete all the messages in your queue by **purguing** them.
- **Long polling** helps reduce the cost by eliminating the number of empty responses and false empty responses. While the regular **short polling** returns immediately, even if the message queue being polled is empty, long polling doesn't return a response until a message arrives in the message queue, or the long poll times out.
- To prevent other consumers from processing a message redundantly, SQS sets a **visibility timeout**, a period of time SQS prevents other consumers from receiving and processing the message. The default visibility timeout for a message is 30 seconds. The minimum is 0 seconds. The maximum is 12 hours.
- SQS supports **dead-letter queues**, which other queues can target for messages that can't be processed successfully.
- **Delay queues** let you postpone the delivery of new messages to a queue for a number of seconds.
- **Basic SQS Architecture**
  - Main Parts
    - The components of your distributed system
    - The queue
    - The messages
  - Standard Queues
    - Default queue type.
    - Makes a best effort to preserve the order of messages.
    - Stores copies of your messages on multiple servers for redundancy and high availability.
    - Consumes messages using **short polling** (default) - take a subset of SQS servers (based on a weighted random distribution) and returns messages from only those servers.
  - FIFO Queues
    - The order in which messages are sent and received is strictly preserved and a message is delivered once and remains available until a consumer processes and deletes it.
    - Duplicates aren't introduced into the queue.
    - FIFO queues support **message groups** that allow multiple ordered message groups within a single queue.
  - When you create a new queue, you must specify a *queue name* unique for your AWS account and region. This becomes your queue url.  
<https://sqs.region.amazonaws.com/accountnumber/queuename>
  - Each message receives a system-assigned *message ID* for identifying messages.
  - Every time you receive a message from a queue, you receive a *receipt handle* for that message.
- You can use **cost allocation tags** to organize your AWS bill to reflect your own cost structure.
- Send, receive, or delete messages in **batches** of up to 10 messages or 256KB.
- **Best Practices**



- Extend the message's visibility timeout to the maximum time it takes to process and delete the message. If you don't know how long it takes to process a message, as long as your consumer still works on the message, keep extending the visibility timeout .
- Using the appropriate polling mode.
- Configure a dead-letter queue to capture problematic messages.
- To avoid inconsistent message processing by standard queues, avoid setting the number of maximum receives to 1 when you configure a dead-letter queue.
- Don't create reply queues per message. Instead, create reply queues on startup, per producer, and use a correlation ID message attribute to map replies to requests. Don't let your producers share reply queues.
- Reduce cost by batching message actions.
- **Monitoring, Logging, and Automating**
  - Monitor SQS queues using CloudWatch
  - Log SQS API Calls Using AWS CloudTrail
  - Automate notifications from AWS Services to SQS using CloudWatch Events
- **Security**
  - Use IAM for user authentication.
  - SQS has its own resource-based permissions system that uses policies written in the same language used for IAM policies.
  - Protect data using Server-Side Encryption and AWS KMS.
- **Pricing**
  - You are charged per 1 million SQS requests. Price depends on the type of queue being used.  
Requests include:
    - API Actions
    - FIFO Requests
    - A single request of 1 to 10 messages, up to a maximum total payload of 256 KB
    - Each 64 KB chunk of a payload is billed as 1 request
    - Interaction with Amazon S3
    - Interaction with AWS KMS
  - Data transfer out of SQS per TB/month after consuming 1 GB for that month

#### Sources:

- <https://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDeveloperGuide>
- <https://aws.amazon.com/sqs/features/>
- <https://aws.amazon.com/sqs/pricing/>
- <https://aws.amazon.com/sqs/faqs/>



## Amazon SNS

- A web service that makes it easy to set up, operate, and send notifications from the cloud. SNS follows the “**publish-subscribe**” (**pub-sub**) messaging paradigm, with notifications being delivered to clients using a “**push**” mechanism rather than to periodically check or “**poll**” for new information and updates.

## Features

- SNS is an **event-driven** computing hub that has native integration with a wide variety of AWS event sources (including EC2, S3, and RDS) and AWS event destinations (including SQS, and Lambda).
  - **Event-driven computing** is a model in which subscriber services automatically perform work in response to events triggered by publisher services. It can automate workflows while decoupling the services that collectively and independently work to fulfil these workflows.
- **Message filtering** allows a subscriber to create a filter policy, so that it only gets the notifications it is interested in.
- **Message fanout** occurs when a message is sent to a topic and then replicated and pushed to multiple endpoints. Fanout provides asynchronous event notifications, which in turn allows for parallel processing.
- **SNS mobile notifications** allows you to fanout mobile push notifications to iOS, Android, Fire OS, Windows and Baidu-based devices. You can also use SNS to fanout text messages (SMS) to 200+ countries and fanout email messages (SMTP).
- **Application and system alerts** are notifications, triggered by predefined thresholds, sent to specified users by SMS and/or email.
- **Push email** and **text messaging** are two ways to transmit messages to individuals or groups via email and/or SMS.
- SNS provides durable storage of all messages that it receives. When SNS receives your *Publish* request, it stores multiple copies of your message to disk. Before SNS confirms to you that it received your request, it stores the message in multiple Availability Zones within your chosen AWS Region.
- SNS allows you to set a TTL (Time to Live) value for each message. When the TTL expires for a given message that was not delivered and read by an end user, the message is deleted.

SNS provides simple APIs and easy integration with applications.

## Publishers and Subscribers

- Publishers communicate asynchronously with subscribers by producing and sending a message to a topic, which is a logical access point and communication channel.
- Subscribers consume or receive the message or notification over one of the supported protocols when they are subscribed to the topic.
- Publishers create topics to send messages, while subscribers subscribe to topics to receive messages.



- 
- SNS does not support forwarding messages to SQS FIFO queues. You can use SNS to forward messages to standard queues.

## SNS Topics

- Instead of including a specific destination address in each message, a publisher sends a message to a **topic**. SNS matches the topic to a list of subscribers who have subscribed to that topic, and delivers the message to each of those subscribers.
- Each topic has a unique name that identifies the SNS endpoint for publishers to post messages and subscribers to register for notifications.
- A topic can support subscriptions and notification deliveries over multiple transports.

The SNS service will attempt to deliver messages from the publisher in the order they were published into the topic, so no guarantee.

SNS also logs the delivery status of notification messages sent to topics with the following SNS endpoints:

- Application
- HTTP
- Lambda
- SQS

## Message Attributes

- Amazon SNS supports delivery of *message attributes*. Message attributes allow you to provide structured metadata items (such as time stamps, geospatial data, signatures, and identifiers) about the message. Message attributes are optional and separate from, but sent along with, the message body.
- You can use message attributes to help structure the push notification message for mobile endpoints. The attributes are not delivered to the mobile endpoint, as they are when sending messages with message attributes to SQS endpoints.
- You can also use message attributes to make your messages filterable with *subscription filter policies*. You apply filter policies to topic subscriptions.
- Message attributes contain a name, type, and value that must not be empty or null. The message body should not be empty or null also.

## Message Filtering

- A filter policy is a simple JSON object.
- By default, a subscriber of an SNS topic receives every message published to the topic. The filter policy contains attributes that define which messages the subscriber receives.

## Raw Message Delivery



- By default, messages are delivered encoded in JSON that provides metadata about the message and topic.
- You can send large payload messages using AWS SDK that supports AWS Signature Version 4 signing.
- You can also enable raw message delivery for messages delivered to either SQS endpoints or HTTP/S endpoints.

## System to System Messaging

- When a message is published to an SNS topic that has a **Lambda function** subscribed to it, the Lambda function is invoked with the payload of the published message. The Lambda function receives the message payload as an input parameter and can manipulate the information in the message, publish the message to other SNS topics, or send the message to other AWS services.
- When you subscribe a **SQS queue** to a SNS topic, you can publish a message to the topic and SNS sends a SQS message to the subscribed queue. The SQS message contains the subject and message that were published to the topic along with metadata about the message in a JSON document.
- When you subscribe an **HTTP/s endpoint** to a topic, you can publish a notification to the topic and SNS sends an HTTP POST request delivering the contents of the notification to the subscribed endpoint. When you subscribe the endpoint, you select whether SNS uses HTTP or HTTPS to send the POST request to the endpoint.

## User Notifications

- You have the ability to send **push notification messages** directly to apps on mobile devices. Push notification messages sent to a mobile endpoint can appear in the mobile app as message alerts, badge updates, or even sound alerts.
- *Direct addressing* allows you to deliver notifications directly to a single endpoint, rather than sending identical messages to all subscribers of a topic. This is useful if you want to deliver precisely targeted messages to each recipient.
- You can use SNS to send **text messages**, or SMS messages, to SMS-enabled devices. You can send a message directly to a phone number, or you can send a message to multiple phone numbers at once by subscribing those phone numbers to a topic and sending your message to the topic.
- You can use the **Delivery Status** feature to get information on the final disposition of your SMS message.
- SMS messages that are of high priority to your business should be marked as *Transactional*. This ensures that messages such as those that contain one-time passwords or PINs get delivered over routes with the highest delivery reliability.
- SMS messages that carry marketing messaging should be marked *Promotional*. Amazon SNS ensures that such messages are sent over routes that have a reasonable delivery reliability but are substantially cheaper than the most reliable routes.

## SNS Delivery Retries

---



- All messages sent to SNS are processed and delivered immediately. If a message cannot be successfully delivered on the first attempt, SNS implements a 4-phase retry policy:
  - 1) retries with no delay in between attempts
  - 2) retries with some minimum delay between attempts
  - 3) retries with some back-off model (linear or exponential)
  - 4) retries with some maximum delay between attempts

## Monitoring

- Monitoring SNS topics using CloudWatch
- Logging SNS API calls using CloudTrail

## Security

- SNS provides encrypted topics to protect your messages from unauthorized and anonymous access. The encryption takes place on the server side.
- SNS supports VPC Endpoints via AWS PrivateLink. You can use VPC Endpoints to privately publish messages to SNS topics, from a VPC, without traversing the public internet.
- Using access control policies, you have detailed control over which endpoints a topic allows, who is able to publish to a topic, and under what conditions.
- You can enable AWS X-Ray for your messages passing through Amazon SNS, making it easier to trace and analyze messages as they travel through to the downstream services.

## Pricing

- You pay based on the number of notifications you publish, the number of notifications you deliver, and any additional API calls for managing topics and subscriptions. Delivery pricing varies by endpoint type.

## Sources:

<https://docs.aws.amazon.com/sns/latest/dg>  
<https://aws.amazon.com/sns/features/>  
<https://aws.amazon.com/sns/pricing/>  
<https://aws.amazon.com/sns/faqs/>



## Amazon SWF

- A fully-managed state tracker and task coordinator in the Cloud. You create desired workflows with their associated tasks and any conditional logic you wish to apply and store them with SWF.

## Features

- SWF promotes a separation between the control flow of your background job's stepwise logic and the actual units of work that contain your unique business logic.
- SWF manages your workflow execution history and other details of your workflows across 3 availability zones.
- SWF lets you write your application components and coordination logic in any programming language and run them in the cloud or on-premises.
- SWF is highly scalable. It gives you full control over the number of workers that you run for each activity type and the number of instances that you run for a decider.
- SWF also provides the **AWS Flow Framework** to help developers use asynchronous programming in the development of their applications.

## Concepts

### Workflow

- A set of activities that carry out some objective, together with logic that coordinates the activities.
- Workflows coordinate and manage the execution of activities that can be run asynchronously across multiple computing devices and that can feature both sequential and parallel processing.
- Each workflow runs in an AWS resource called a *domain*, which controls the workflow's scope.
- An AWS account can have multiple domains, each of which can contain multiple workflows, but workflows in different domains can't interact.
- When you register an activity to a workflow, you provide information such as a name and version, and some timeout values based on how long you expect the activity to take.

### Activity Task

- An **activity task** tells an activity worker to perform its function.
- SWF stores tasks and assigns them to workers when they are ready, tracks their progress, and maintains their state, including details on their completion.
- To coordinate tasks, you write a program that gets the latest state of each task from SWF and uses it to initiate subsequent tasks.
- Activity tasks can run synchronously or asynchronously. They can be distributed across multiple computers, potentially in different geographic regions, or they can all run on the same computer.
- Activity tasks for a running workflow execution appear on the *activity task list*, which is provided when you schedule an activity in the workflow.



- If you don't specify a task list when scheduling an activity task, the task is automatically placed on the default task list.

#### Lambda task

- Executes a Lambda function instead of a traditional SWF activity.

#### Decision task

- A Decision task tells a decider that the state of the workflow execution has changed so that the decider can determine the next activity that needs to be performed. The decision task contains the current workflow history.
- SWF assigns each decision task to exactly one decider and allows only one decision task at a time to be active in a workflow execution.

#### Workflow Starter

- Any application that can initiate workflow executions.

#### Activity Worker

- An **activity worker** is a program that receives activity tasks, performs them, and provides results back.
- Implement workers to perform tasks. These workers can run either on cloud infrastructure, or on your own premises.
- Different activity workers can be written in different programming languages and run on different operating systems.
- Assigning particular tasks to particular activity workers is called *task routing*. Task routing is optional.

#### Decider

- A software program that contains the coordination logic in a workflow.
- It schedules activity tasks, provides input data to the activity workers, processes events that arrive while the workflow is in progress, and ultimately ends the workflow when the objective has been completed.
- Both activity workers and the decider receive their tasks by polling the SWF service.

#### Workflow Execution History

- The workflow execution history is composed of events, where an event represents a significant change in the state of the workflow execution.
- SWF informs the decider of the state of the workflow by including, with each decision task, a copy of the current workflow execution history.

#### Polling

- Deciders and activity workers communicate with SWF using *long polling*.



The screenshot shows the Amazon Simple Workflow Service Dashboard for a domain named 'test-swf'. A yellow warning bar at the top states: 'Note: The selected domain has been deprecated.' Below this, there's a search bar for 'Find Execution(s)'. To the right, 'Aggregated Workflow Execution Metrics' show 'No Active Executions' and 'No Closed Executions' over the last hour. Below the metrics, a section for 'Getting Started with Amazon SWF' includes a 'Launch Sample Walkthrough' button. On the left sidebar, under 'Quick Links', there are buttons for 'Start a new Workflow Execution', 'Register a new Workflow Type', and 'Register a new Activity Type'. On the right, there's a 'View Task List Backlog' section with a 'Task List Name' input field and a 'View Backlog' button. At the bottom right, there's a 'Related Resource Links' section with links to 'Service Details', 'Documentation', and 'Forums'.

## Workflow Execution

1. Write activity workers that implement the processing steps in your workflow.
2. Write a decider to implement the coordination logic of your workflow.
3. Register your activities and workflow with Amazon SWF.
4. Start your activity workers and decider.
5. Start one or more executions of your workflow. Each execution runs independently and you can provide each with its own set of input data. When an execution is started, Amazon SWF schedules the initial decision task. In response, your decider begins generating decisions which initiate activity tasks. Execution continues until your decider makes a decision to close the execution.
6. Filter and view complete details of running as well as completed executions.

SWF provides service operations that are **accessible through HTTP requests**.

## Endpoints

- To reduce latency and to store data in a location that meets your requirements, SWF provides endpoints in different regions.
- Each endpoint is completely independent. When you register an SWF domain, workflow or activity, it exists only within the region you registered it in.

## AWS Flow Framework

- An enhanced SDK for writing distributed, asynchronous programs that can run as workflows on SWF.



- 
- It is available for the Java and Ruby programming languages, and it provides classes that simplify writing complex distributed programs.

## Pricing

- You pay for workflow executions when you start them and for each 24-hour period until they are completed. The first 24 hours of workflow execution are free.
- You pay for “markers” (custom workflow execution log entries), start timers, or receive signals of additional tasks generated from a workflow.
- Data transferred between SWF and other AWS services **within a single region is free** of charge. Data transferred between SWF and other AWS services **in different regions will be charged** at Internet Data Transfer rates on both sides of the transfer. First 1 GB of transferred data is free.

## SQS vs SWF

- SQS offers reliable, highly-scalable hosted queues for storing messages while they travel between applications or microservices.
- SWF is a web service that makes it easy to coordinate work across distributed application components.
- SWF API actions are task-oriented. SQS API actions are message-oriented.
- SWF keeps track of all tasks and events in an application. SQS requires you to implement your own application-level tracking.
- SWF offers several features that facilitate application development, such as passing data between tasks, signaling, and flexibility in distributing tasks.

## Sources:

<https://docs.aws.amazon.com/amazonswf/latest/developerguide/>

<https://aws.amazon.com/swf/faqs/>

<https://aws.amazon.com/swf/pricing/>



## AWS Step Functions

- AWS Step Functions is a web service that provides **serverless orchestration** for modern applications. It enables you to coordinate the components of distributed applications and microservices using visual workflows.
- Concepts
  - Step Functions is based on the concepts of **tasks** and **state machines**.
    - A task performs work by using an activity or an AWS Lambda function, or by passing parameters to the API actions of other services.
    - A finite state machine can express an algorithm as a number of states, their relationships, and their input and output.
  - You define state machines using the **JSON-based Amazon States Language**.

Step 1  
Review Hello World example

Step 2  
Specify details

### Review Hello World example

Get started fast by running a Hello World example in 3 clicks. You can also skip and access more functionality [here](#).

**Definition**

Example code is read-only, but you can edit it after creation. In AWS Step Functions, workflows are defined using Amazon States Language. [Learn more](#)

**Export ▾** **Layout ▾**

```
1  {
2      "Comment": "A Hello World example demonstrating various state types of the Amazon States
3          Language",
4      "StartAt": "Pass",
5      "States": {
6          "Pass": {
7              "Comment": "A Pass state passes its input to its output, without performing work. Pass
8                  states are useful when constructing and debugging state machines.",
9              "Type": "Pass",
10             "Next": "Hello World example?"
11         },
12         "Hello World example?": {
13             "Comment": "A Choice state adds branching logic to a state machine. Choice rules can
14                 implement 16 different comparison operators, and can be combined using And, Or, and Not",
15             "Type": "Choice",
16             "Choices": [
17                 {
18                     "Variable": "$.IsHelloWorldExample",
19                     "BooleanEquals": true,
20                     "Next": "Yes"
21                 },
22                 {
23                     "Variable": "$.IsHelloWorldExample",
24                     "BooleanEquals": false,
25                     "Next": "No"
26                 }
27             ]
28         }
29     }
30 }
```

```
graph TD
    Start((Start)) --> Pass[Pass]
    Pass --> Choice{Hello World example?}
    Choice -- Yes --> Wait[Wait 3 sec]
    Wait --> Parallel[Hello, World]
    Parallel --> End((End))
    Choice -- No --> End
```

- A state is referred to by its *name*, which can be any string, but which must be unique within the scope of the entire state machine. An instance of a state exists until the end of its execution.
  - There are 6 types of states:
    - Task state - Do some work in your state machine. AWS Step Functions can invoke Lambda functions directly from a task state.
    - Choice state - Make a choice between branches of execution
    - Fail or Succeed state - Stop an execution with a failure or success
    - Pass state - Simply pass its input to its output or inject some fixed data



- Wait state - Provide a delay for a certain amount of time or until a specified time/date
- Parallel state - Begin parallel branches of execution
- Common features between states
  - Each state must have a *Type* field indicating what type of state it is.
  - Each state can have an optional *Comment* field to hold a human-readable comment about, or description of, the state.
  - Each state (except a Succeed or Fail state) requires a *Next* field or, alternatively, can become a terminal state by specifying an *End* field.
- **Activities** enable you to place a task in your state machine where the work is performed by an **activity worker** that can be hosted on Amazon EC2, Amazon ECS, or mobile devices.
- Activity tasks let you assign a specific step in your workflow to code running in an activity worker. Service tasks let you connect a step in your workflow to a supported AWS service.
- With **Transitions**, after executing a state, AWS Step Functions uses the value of the *Next* field to determine the next state to advance to. States can have multiple incoming transitions from other states.
- State machine data is represented by JSON text. It takes the following forms:
  - The initial input into a state machine
  - Data passed between states
  - The output from a state machine
- Individual states receive JSON as input and usually pass JSON as output to the next state.
- A **state machine execution** occurs when a state machine runs and performs its tasks. Each Step Functions state machine can have multiple simultaneous executions.
- **State machine updates** in AWS Step Functions are **eventually consistent**.
- By default, when a state reports an error, AWS Step Functions causes the execution to **fail entirely**.
  - Task and Parallel states can have a field named *Retry* and *Catch* to retry an execution or to have a fallback state.
- The Step Functions console displays a graphical view of your state machine's structure, which provides a way to visually check a state machine's logic and monitor executions.

## Features

- Using Step Functions, you define your **workflows as state machines**, which transform complex code into easy to understand statements and diagrams.
- Step Functions provides ready-made steps for your workflow called **states** that implement basic service primitives for you, which means you can remove that logic from your application. States are able to:
  - pass data to other states and microservices,
  - handle exceptions,
  - add timeouts,
  - make decisions,
  - execute multiple paths in parallel,



- 
- and more.
  - Using Step Functions **service tasks**, you can configure your Step Functions workflow to call other AWS services.
  - Step Functions can coordinate any application that can make an **HTTPS** connection, regardless of where it is hosted—Amazon EC2 instances, mobile devices, or on-premises servers.
  - AWS Step Functions coordinates your existing Lambda functions and microservices, and lets you modify them into new compositions. The tasks in your workflow can run anywhere, including on instances, containers, functions, and mobile devices.
  - Nesting your Step Functions workflows allows you to build larger, more complex workflows out of smaller, simpler workflows.
  - Step Functions keeps the logic of your application strictly separated from the implementation of your application. You can add, move, swap, and reorder steps without having to make changes to your business logic.
  - Step Functions maintains the state of your application during execution, including tracking what step of execution it is in, and storing data that is moving between the steps of your workflow. You won't have to manage state yourself with data stores or by building complex state management into all of your tasks.
  - Step Functions automatically handles errors and exceptions with **built-in try/catch and retry**, whether the task takes seconds or months to complete. You can automatically retry failed or timed-out tasks, respond differently to different types of errors, and recover gracefully by falling back to designated cleanup and recovery code.
  - Step Functions has **built-in fault tolerance and maintains service capacity across multiple Availability Zones in each region**, ensuring high availability for both the service itself and for the application workflow it operates.
  - Step Functions **automatically scales** the operations and underlying compute to run the steps of your application for you in response to changing workloads.
  - AWS Step Functions has a 99.9% SLA.
  - AWS Step Functions enables access to metadata about workflow executions so that you can easily identify related resources. For example, your workflow can retrieve its execution ID or a timestamp indicating when a task started. This makes it easier to correlate logs for faster debugging and to measure workflow performance data.
  - It also supports callback patterns. Callback patterns automate workflows for applications with human activities and custom integrations with third-party services.
  - AWS Step Functions supports workflow execution events, which make it faster and easier to build and monitor event-driven, serverless workflows. Execution event notifications can be automatically delivered when a workflow starts or completes through CloudWatch Events, reaching targets like AWS Lambda, Amazon SNS, Amazon Kinesis, or AWS Step Functions for automated response to the event.

#### How Step Functions Work

- Orchestration centrally manages a workflow by breaking it into multiple steps, adding flow logic, and tracking the inputs and outputs between the steps.



- As your applications execute, Step Functions maintains application state, tracking exactly which workflow step your application is in, and stores an event log of data that is passed between application components. That means that if networks fail or components hang, your application can pick up right where it left off.
- Amazon States Language
  - A JSON-based, structured language used to define your state machine, a collection of states, that can do work (*Task states*), determine which states to transition to next (*Choice states*), stop an execution with an error (*Fail states*), and so on.
  - When creating a state machine, only the *States* field is required in the JSON document.
- Security and Monitoring
  - AWS Step Functions can execute code and access AWS resources (such as invoking an AWS Lambda function). To maintain security, you must grant Step Functions access to those resources by using an IAM role.
  - Step Functions is a HIPAA eligible service, and is also compliant with SOC.
  - Step Functions delivers real-time diagnostics and dashboards, integrates with Amazon CloudWatch and AWS CloudTrail, and logs every execution, including overall state, failed steps, inputs, and outputs.
- Pricing
  - Step Functions counts a state transition each time a step of your workflow is executed. You are charged for the total number of state transitions across all your state machines, including retries.
- Common Use Cases
  - Step Functions can help ensure that long-running, multiple ETL jobs execute in order and complete successfully, instead of manually orchestrating those jobs or maintaining a separate application.
  - By using Step Functions to handle a few tasks in your codebase, you can approach the transformation of monolithic applications into microservices as a series of small steps.
  - You can use Step Functions to easily automate recurring tasks such as patch management, infrastructure selection, and data synchronization, and Step Functions will automatically scale, respond to timeouts, and retry failed tasks.
  - Use Step Functions to combine multiple AWS Lambda functions into responsive serverless applications and microservices, without having to write code for workflow logic, parallel processes, error handling, timeouts or retries.
  - You can also orchestrate data and services that run on Amazon EC2 instances, containers, or on-premises servers.

**Sources:**

<https://aws.amazon.com/step-functions/features/>

<https://aws.amazon.com/step-functions/pricing/>

<https://docs.aws.amazon.com/step-functions/latest/dg/welcome.html>

<https://aws.amazon.com/step-functions/faqs/>



## DESKTOP AND APP STREAMING

### Amazon WorkSpaces

- A fully-managed, secure cloud desktop service.

### Features

- You can create a standalone, managed directory for users, or you can integrate with your existing Active Directory environment so that your users can use their current credentials to obtain seamless access to corporate resources.
- WorkSpaces provides a persistent desktop.
- Offers a range of bundles that provide different hardware and software options to meet your needs. A bundle is a combination of both that image and the hardware from which a WorkSpace can be launched.
- You can bring your existing Windows 7 and Windows 10 Desktop licenses to Amazon WorkSpaces and run them on hardware that is physically dedicated to you.
- Whether you choose to launch one or many WorkSpaces, all you need to do is to choose the bundles that best meet the needs of your users, and the number of WorkSpaces that you would like to launch.
- WorkSpaces provides a high quality desktop experience that helps you meet compliance and security requirements, including HIPAA-eligibility and PCI compliance.
- WorkSpaces allows you to use your on-premises Microsoft Active Directory to manage your WorkSpaces and your end user credentials.
- WorkSpaces provides each user with access to varying amounts of persistent storage (SSD Volumes) in the AWS cloud based on the bundle selected. Data that users store on the 'user volume' attached to the WorkSpace is automatically backed up to S3 on a regular basis.
- Users can also use Amazon WorkDocs Drive.
- WorkSpaces incorporates PC-over-IP (PCoIP) technology to provide a secure experience.
- Deploy and manage applications for your WorkSpaces by using Amazon WorkSpaces Application Manager.
- By default, users can personalize their WorkSpaces with their favorite settings for items such as wallpaper, icons, shortcuts, etc. An administrator can lock down a WorkSpace.
- By default, users are configured as local administrators of their WorkSpaces.
- The user volume on a WorkSpace is backed up every 12 hours.
- Each user you provision a WorkSpace for needs to exist in a directory, but you do not have to provision a directory yourself.
- To ensure that your data is preserved, the volume sizes of either volume cannot be reduced after a WorkSpace is launched. You can increase the size of the root and user volumes attached to your WorkSpaces at any time.



## Pricing

- You are charged based on the bundle type and the number of Amazon WorkSpaces you launch.
- WorkSpaces provides the flexibility to pay monthly or hourly.
  - With monthly billing, you pay a fixed monthly fee for unlimited usage during the month, which is best for workers who use their Amazon WorkSpace full-time or as their primary desktop.
  - With hourly billing you pay a small fixed monthly fee per WorkSpace to cover infrastructure costs and storage, and a low hourly rate for each hour the WorkSpace is used during the month.

## Sources:

<https://docs.aws.amazon.com/workspaces/latest/adminguide/>

<https://aws.amazon.com/workspaces/pricing/>

<https://aws.amazon.com/workspaces/faqs/>



## COMPARISON OF AWS SERVICES

### S3 vs EBS vs EFS

TD Tutorials Dojo	S3	EBS	EFS
Type of storage	Object storage. You can store virtually any kind of data in any format.	Persistent block level storage for EC2 instances.	POSIX-compliant file storage for EC2 instances
Features	Accessible to anyone or any service with the right permissions	Deliver performance for workloads that require the lowest-latency access to data from a single EC2 instance	Has a file system interface, file system access semantics (such as strong consistency and file locking), and concurrently-accessible storage for multiple EC2 instances
Max Storage Size	Virtually unlimited	16 TiB for one volume	Unlimited system size
Max File Size	Individual Amazon S3 objects can range in size to a maximum of 5 terabytes.	Equivalent to the maximum size of your volumes	47.9 TiB for a single file
Performance (Latency)	Low, for mixed request types, and integration with CloudFront	Lowest, consistent; SSD-backed storages include the highest performance Provisioned OPS SSD and General Purpose SSD that balance price and performance.	Low, consistent; use Max I/O mode for higher performance
Performance (Throughput)	Multiple GBs per second; supports multi-part upload	Up to 2 GB per second. HDD-backed volumes include Throughput Optimized HDD for frequently accessed, throughput intensive workloads and Cold HDD for less frequently accessed data.	10+ GB per second. Bursting Throughput mode scales with the size of the file system. Provisioned Throughput mode offers higher dedicated throughput than bursting throughput.
Durability	Stored redundantly across multiple AZs; has 99.99999999% durability	Stored redundantly in a single AZ	Stored redundantly across multiple AZs
Availability	S3 Standard - 99.99% availability  S3 Standard-IA - 99.9% availability  S3 One Zone-IA - 99.5% availability.	Has 99.999% availability	No SLA. Runs in multi-AZ



TD Tutorials Dojo	S3	EBS	EFS
Scalability	Highly scalable	Manually increase/decrease your memory size. Attach and detach additional volumes to and from your EC2 instance to scale.	EFS file systems are elastic, and automatically grow and shrink as you add and remove files.
Data Accessing	One to millions of connections over the web; S3 provides a REST web services interface	Single EC2 instance in a single AZ	One to thousands of EC2 instances or on-premises servers, from multiple AZs, regions, VPCs, and accounts concurrently
Access Control	Uses bucket policies and IAM user policies. Has <i>Block Public Access</i> settings to help manage public access to resources.	IAM Policies, Roles, and Security Groups	Only resources that can access endpoints in your VPC, called a <i>mount target</i> , can access your file system; POSIX-compliant user and group-level permissions
Encryption Methods	Supports SSL endpoints using the HTTPS protocol, Client-Side and Server-Side Encryption (SSE-S3, SSE-C, SSE-KMS)	Encrypts both data-at-rest and data-in-transit through EBS encryption that uses AWS KMS CMKs.	Encrypt data at rest and in transit. Data at rest encryption uses AWS KMS. Data in-transit uses TLS 1.2.
Backup and Restoration	Use versioning or cross-region replication	All EBS volume types offer durable snapshot capabilities.	EFS-to-EFS backup solution.
Pricing	Billing prices are based on the location of your bucket. Lower costs equals lower prices. You get cheaper prices the more you use S3 storage.	You pay GB-month of provisioned storage, provisioned IOPS-month, GB-month of snapshot data stored in S3	You pay for the amount of file system storage used per month. When using the Provisioned Throughput mode you pay for the throughput you provision per month.
Use Cases	Web serving and content management, media and entertainment, backups, big data analytics, data lake	Boot volumes, transactional and NoSQL databases, data warehousing & ETL	Web serving and content management, enterprise applications, media and entertainment, home directories, database backups, developer tools, container storage, big data analytics
Service endpoint	Can be accessed within and outside a VPC (via S3 bucket URL)	Accessed within one's VPC	Accessed within one's VPC



## Amazon S3 vs Glacier

- Amazon S3 is a durable, secure, simple, and fast storage service, while Amazon S3 Glacier is used for archiving solutions.
- Use S3 if you need low latency or frequent access to your data. Use S3 Glacier for low storage cost, and you do not require millisecond access to your data.
- You have three retrieval options when it comes to Glacier, each varying in the cost and speed it retrieves an object for you. You retrieve data in milliseconds from S3.
- Both S3 and Glacier are designed for durability of 99.99999999% of objects across multiple Availability Zones.
- S3 is designed for availability of 99.99% while Glacier has no percentage provided by AWS.
- S3 can be used to host static web content, while Glacier cannot.
- In S3, users create buckets. In Glacier, users create archives and vaults.
- You can store a virtually unlimited amount of data in both S3 and Glacier.
- A single Glacier archive can contain 40TB of data.
- S3 supports Versioning.
- You can run analytics and querying on S3.
- You can configure a lifecycle policy for your S3 objects to automatically transfer them to Glacier. You can also upload objects directly to either S3 or Glacier.
- S3 Standard-IA and One Zone-IA have a minimum capacity charge per object of 128KB. Glacier's minimum is 40KB.
- Objects stored in S3 have a minimum storage duration of 30 days (except for S3 Standard). Objects that are archived to Glacier have a minimum 90 days of storage. Objects that are deleted, overwritten, or transitioned to a different storage class before the minimum duration will incur the normal usage charge plus a pro-rated request charge for the remainder of the minimum storage duration.
- Glacier has a per GB retrieval fee.
- You can transition objects from some S3 storage classes to another. Glacier objects cannot be transitioned to any S3 storage class.
- S3 (standard, standard-IA, and one zone-IA) is backed by an SLA.



## S3 Standard vs S3 Standard-IA vs S3OneZone-IA

	S3 Standard	S3 Standard-Infrequent Access (IA)	S3 One Zone-Infrequent Access (IA)
Features	General-purpose storage of frequently accessed data	For long-lived, rapid but less frequently accessed data; data is stored redundantly in multiple AZs	For long-lived, rapid but less frequently accessed data; data is stored redundantly in only one AZ of your choice
Durability	99.999999999% (11 9's)	99.999999999% (11 9's)	99.999999999% (11 9's)
Availability	99.99%	99.9%	99.5%
Availability SLA	99.9%	99%	99%
Number of Availability Zones	At least 3	At least 3	Only 1
Minimum capacity charge per object	N/A	128KB	128KB
Minimum storage duration charge	N/A	30 days	30 days
Inserting data	Directly PUT into S3 Standard	Directly PUT into S3 Standard-IA or set Lifecycle policies to transition objects from the S3 Standard to the S3 Standard-IA storage class.	Directly PUT into S3 One Zone-IA or set Lifecycle policies to transition objects from the S3 Standard to the S3 One Zone-IA storage class.
Retrieval fee	N/A	per GB retrieved	per GB retrieved
First byte latency	milliseconds	milliseconds	milliseconds
Storage transition	S3 Standard to all other S3 storage types including Glacier	S3 Standard-IA to S3 One Zone-IA or S3 Glacier	S3 One Zone-IA to S3 Glacier
Use Cases	Cloud applications, dynamic websites, content distribution, mobile and gaming applications, and big data analytics.	Ideally suited for long-term file storage, older sync and share storage, and other aging data.	For infrequently-accessed storage, like backup copies, disaster recovery copies, or other easily recreatable data.



### Additional Notes:

- Data stored in the S3 One Zone-IA storage class will be lost in the event of AZ destruction.
- S3 Standard-IA costs less than S3 Standard in terms of storage price, while still providing the same high durability, throughput, and low latency of S3 Standard.
- S3 One Zone-IA has 20% less cost than Standard-IA.
- S3 Reduced Redundancy Storage (RRS) will be deprecated soon.
- It is recommended to use multipart upload for objects larger than 100MB.



## RDS vs DynamoDB

	RDS	DynamoDB
Type of database	Managed relational (SQL) database	Fully managed key-value and document (NoSQL) database
Features	Has several database instance types for different kinds of workloads and supports six database engines - Amazon Aurora, PostgreSQL, MySQL, MariaDB, Oracle Database, and SQL Server.	Delivers single-digit millisecond performance at any scale.
Storage Size	64 TB for Aurora engine. 16 TB for MySQL, MariaDB, SQL Server, Oracle, and PostgreSQL engines.	Supports tables of virtually any size.
Number of tables per unit	Depends on the database engine	256
Performance	General Purpose Storage is an SSD-backed storage option that delivers a consistent baseline of 3 IOPS per provisioned GB with the ability to burst up to 3,000 IOPS.  Provisioned IOPS Storage is an SSD-backed storage option designed to deliver a consistent IOPS rate that you specify when creating a database instance, up to 40,000 IOPS per database instance. Amazon RDS provisions that IOPS rate for the lifetime of the database instance. Optimized for OLTP database workloads.	Single-digit millisecond read and write performance. Can handle more than 10 trillion requests per day with peaks greater than 20 million requests per second, over petabytes of storage.  DynamoDB Accelerator (DAX) is an in-memory cache that can improve the read performance of your DynamoDB tables by up to 10 times—taking the time required for reads from milliseconds to microseconds, even at millions of requests per second.  You specify the read and write throughput for each of your tables.
Availability and durability	Amazon RDS Multi-AZ deployments synchronously replicates your data to a standby instance in a different Availability Zone  Amazon RDS will automatically replace the compute instance powering your deployment in the event of a hardware failure..	DynamoDB global tables replicate your data automatically across 3 Availability Zones of your choice of AWS Regions and automatically scale capacity to accommodate your workloads.
Backups	The automated backup feature enables point-in-time recovery for your database instance.  Database snapshots are user-initiated backups of your instance stored in Amazon S3 that are kept until you explicitly delete them.	Point-in-time recovery (PITR) provides continuous backups of your DynamoDB table data, and you can restore that table to any point in time up to the second during the preceding 35 days. On-demand backup and restore allows you to create full backups of your DynamoDB tables' data for data archiving.



	RDS	DynamoDB
Scalability	<p>The Amazon Aurora engine will automatically grow the size of your database volume. The MySQL, MariaDB, SQL Server, Oracle, and PostgreSQL engines allow you to scale on-the-fly with zero downtime.</p> <p>RDS also supports storage auto scaling</p> <p>Read replicas are available in Amazon RDS for MySQL, MariaDB, and PostgreSQL as well as Amazon Aurora.</p> <p>Can scale compute and memory resources (vertically) of up to a maximum of 32 vCPUs and 244 GiB of RAM.</p>	<p>Support tables of virtually any size with horizontal scaling.</p> <p>For tables using on-demand capacity mode, DynamoDB instantly accommodates your workloads as they ramp up or down to any previously reached traffic level.</p> <p>For tables using provisioned capacity, DynamoDB delivers automatic scaling of throughput and storage based on your previously set capacity.</p>
Security	<p>Isolate your database in your own virtual network.</p> <p>Connect to your on-premises IT infrastructure using industry-standard encrypted IPsec VPNs.</p> <p>You can configure firewall settings and control network access to your database instances.</p> <p>Integrates with IAM.</p>	Integrates with IAM.
Encryption	<p>Encrypt your databases using keys you manage through AWS KMS. With encryption enabled, data stored at rest is encrypted, as are its automated backups, read replicas, and snapshots.</p> <p>Supports Transparent Data Encryption in SQL Server and Oracle.</p> <p>Supports the use of SSL to secure data in transit.</p>	DynamoDB encrypts data at rest by default using encryption keys stored in AWS KMS.
Maintenance	Amazon RDS will update databases with the latest patches. You can exert optional control over when and if your database instance is patched.	No maintenance since DynamoDB is serverless.
Pricing	<p>A monthly charge for each database instance that you launch.</p> <p>Option to reserve a DB instance for a one or three year term and receive discounts in pricing, compared to On-Demand instance pricing.</p>	<p>Charges for reading, writing, and storing data in your DynamoDB tables, along with any optional features you choose to enable.</p> <p>There are specific billing options for each of DynamoDB's capacity modes.</p>
Use Cases	Traditional applications, ERP, CRM, and e-commerce.	Internet-scale applications, real-time bidding, shopping carts, and customer preferences, content management, personalization, and mobile applications.



---

### Additional notes:

- DynamoDB has built-in support for ACID transactions.
- DynamoDB uses filter expressions because it does not support complex queries.
- Multi-AZ deployments for the MySQL, MariaDB, Oracle, and PostgreSQL engines utilize synchronous physical replication. Multi-AZ deployments for the SQL Server engine use synchronous logical replication.
- An Oracle Database is limited to one database per instance. An SQL Server is limited to 30 databases per instance. Other RDS engines have no limits.



## CloudTrail vs CloudWatch

- CloudWatch is a monitoring service for AWS resources and applications. CloudTrail is a web service that records API activity in your AWS account. They are both useful monitoring tools in AWS.
- By default, CloudWatch offers free basic monitoring for your resources, such as EC2 instances, EBS volumes, and RDS DB instances. CloudTrail is also enabled by default when you create your AWS account.
- With CloudWatch, you can collect and track metrics, collect and monitor log files, and set alarms. CloudTrail, on the other hand, logs information on who made a request, the services used, the actions performed, parameters for the actions, and the response elements returned by the AWS service. CloudTrail Logs are then stored in an S3 bucket or a CloudWatch Logs log group that you specify.
- You can enable detailed monitoring from your AWS resources to send metric data to CloudWatch more frequently, with an additional cost.
- CloudTrail delivers one free copy of management event logs for each AWS region. Management events include management operations performed on resources in your AWS account, such as when a user logs in to your account. Logging data events are charged. Data events include resource operations performed on or within the resource itself, such as S3 object-level API activity or Lambda function execution activity.
- CloudTrail helps you ensure compliance and regulatory standards.
- CloudWatch Logs reports on application logs, while CloudTrail Logs provide you specific information on what occurred in your AWS account.
- CloudWatch Events is a near real time stream of system events describing changes to your AWS resources. CloudTrail focuses more on AWS API calls made in your AWS account.
- Typically, CloudTrail delivers an event within 15 minutes of the API call. CloudWatch delivers metric data in 5 minutes periods for basic monitoring and 1 minute periods for detailed monitoring. The CloudWatch Logs Agent will send log data every five seconds by default.



## Security Group vs NACL

Security Group	Network Access Control List
Acts as a firewall for associated Amazon EC2 instances	Acts as a firewall for associated subnets
Controls both inbound and outbound traffic at the instance level	Controls both inbound and outbound traffic at the subnet level
You can secure your VPC instances using only security groups	Network ACLs are an additional layer of defense.
Supports allow rules only	Supports allow rules and deny rules
Stateful (Return traffic is automatically allowed, regardless of any rules)	Stateless (Return traffic must be explicitly allowed by rules)
Evaluates all rules before deciding whether to allow traffic	Evaluates rules in number order when deciding whether to allow traffic, starting with the lowest numbered rule.
Applies only to the instance that is associated to it	Applies to all instances in the subnet it is associated with
Has separate rules for inbound and outbound traffic	Has separate rules for inbound and outbound traffic
A newly created security group denies all inbound traffic by default	A newly created nACL denies all inbound traffic by default
A newly created security group has an outbound rule that allows all outbound traffic by default	A newly created nACL denies all outbound traffic default
Instances associated with a security group can't talk to each other unless you add rules allowing it	Each subnet in your VPC must be associated with a network ACL. If none is associated, the default nACL is selected.
Security groups are associated with network interfaces	You can associate a network ACL with multiple subnets; however, a subnet can be associated with only one network ACL at a time.





**Your VPC has a default security group with the following rules:**

1. Allow inbound traffic from instances assigned to the same security group.
2. Allow all outbound IPv4 traffic and IPv6 traffic if you have allocated an IPv6 CIDR block.

**Your VPC has a default network ACL with the following rules:**

1. Allows all inbound and outbound IPv4 traffic and, if applicable, IPv6 traffic.
2. Each network ACL also includes a non modifiable and non removable rule whose rule number is an asterisk. This rule ensures that if a packet doesn't match any of the other numbered rules, it's denied.



## EBS-SSD vs HDD

On a given volume configuration, certain I/O characteristics drive the performance behavior for your EBS volumes. SSD-backed volumes, such as General Purpose SSD (gp2) and Provisioned IOPS SSD (io1), deliver consistent performance whether an I/O operation is random or sequential. HDD-backed volumes like Throughput Optimized HDD (st1) and Cold HDD (sc1) deliver optimal performance only when I/O operations are large and sequential.

In the exam, always consider the difference between SSD and HDD as shown on the table below. This will allow you to easily eliminate specific EBS-types in the options which are not SSD or not HDD, depending on whether the question asks for a storage type which has *small, random* I/O operations or *large, sequential* I/O operations.

FEATURES	SSD Solid State Drive	HDD Hard Disk Drive
Best for workloads with:	<i>small, random</i> I/O operations	<i>large, sequential</i> I/O operations
Can be used as a bootable volume?	Yes	No
Suitable Use Cases	<ul style="list-style-type: none"><li>- Best for <b>transactional workloads</b></li><li>- Critical business applications that require sustained IOPS performance</li><li>- Large database workloads such as MongoDB, Oracle, Microsoft SQL Server and many others...</li></ul>	<ul style="list-style-type: none"><li>- Best for <b>large streaming workloads</b> requiring consistent, fast throughput at a low price</li><li>- Big data, Data warehouses, Log processing</li><li>- Throughput-oriented storage for large volumes of data that is <b>infrequently</b> accessed</li></ul>
Cost	moderate / high 	low 
Dominant Performance Attribute	IOPS	Throughput (MiB/s)





Provisioned IOPS SSD (io1) volumes are designed to meet the needs of I/O-intensive workloads, particularly database workloads, that are sensitive to storage performance and consistency. Unlike gp2, which uses a bucket and credit model to calculate performance, an io1 volume allows you to specify a consistent IOPS rate when you create the volume, and Amazon EBS delivers within 10 percent of the provisioned IOPS performance 99.9 percent of the time over a given year.

	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	<ul style="list-style-type: none"><li>• Recommended for most workloads</li><li>• System boot volumes</li><li>• Virtual desktops</li><li>• Low-latency interactive apps</li><li>• Development and test environments</li></ul>	<ul style="list-style-type: none"><li>• Critical business applications that require sustained IOPS performance, or more than 16,000 IOPS or 250 MiB/s of throughput per volume</li><li>• Large database workloads, such as:<ul style="list-style-type: none"><li>◦ MongoDB</li><li>◦ Cassandra</li><li>◦ Microsoft SQL Server</li><li>◦ MySQL</li><li>◦ PostgreSQL</li><li>◦ Oracle</li></ul></li></ul>	<ul style="list-style-type: none"><li>• Streaming workloads requiring consistent, fast throughput at a low price</li><li>• Big data</li><li>• Data warehouses</li><li>• Log processing</li><li>• Cannot be a boot volume</li></ul>	<ul style="list-style-type: none"><li>• Throughput-oriented storage for large volumes of data that is infrequently accessed</li><li>• Scenarios where the lowest storage cost is important</li><li>• Cannot be a boot volume</li></ul>
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**/Volume	16,000***	64,000***	500	250
Max. Throughput/Volume	250 MiB/s***	1,000 MiB/s†	500 MiB/s	250 MiB/s
Max. IOPS/Instance††	80,000	80,000	80,000	80,000
Max. Throughput/Instance††	1,750 MiB/s	1,750 MiB/s	1,750 MiB/s	1,750 MiB/s
Dominant Performance Attribute	IOPS	IOPS	MiB/s	MiB/s



## S3 Pre-Signed URLs vs CloudFront Signed URLs vs Origin Access Identity

S3 Pre-signed URLs	CloudFront Signed URLs	Origin Access Identity (OAI)
All S3 buckets and objects by default are <b>private</b> . Only the object owner has permission to access these objects. Pre-signed URLs use the owner's security credentials to grant others time-limited permission to download or upload objects.	You can control user access to your private content in two ways <ul style="list-style-type: none"><li>• Restrict access to files in CloudFront edge caches</li><li>• Restrict access to files in your Amazon S3 bucket (unless you've configured it as a website endpoint)</li></ul>	You can configure an S3 bucket as the origin of a CloudFront distribution. OAI prevents users from viewing your S3 files by simply using the direct URL for the file. Instead, they would need to access it through a CloudFront URL.
When creating a pre-signed URL, you (as the owner) need to provide the following: <ul style="list-style-type: none"><li>• Your security credentials</li><li>• An S3 bucket name</li><li>• An object key</li><li>• Specify the HTTP method (GET to download the object or PUT to upload an object)</li><li>• Expiration date and time of the URL.</li></ul>	You can configure CloudFront to require that users access your files using either <b>signed URLs</b> or <b>signed cookies</b> . You then develop your application either to create and distribute signed URLs to authenticated users or to send Set-Cookie headers that set signed cookies on the viewers for authenticated users.  When you create signed URLs or signed cookies to control access to your files, you can specify the following restrictions: <ul style="list-style-type: none"><li>• An expiration date and time for the URL</li><li>• (Optional) The date and time the URL becomes valid</li><li>• (Optional) The IP address or range of addresses of the computers that can be used to access your content</li></ul> You can use signed URLs or signed cookies for any CloudFront distribution, regardless of whether the origin is an Amazon S3 bucket or an HTTP server.	To require that users access your content through CloudFront URLs, you perform the following tasks: <ul style="list-style-type: none"><li>• Create a special CloudFront user called an <b>origin access identity</b>.</li><li>• Give the origin access identity permission to read the files in your bucket.</li><li>• Remove permission for anyone else to use Amazon S3 URLs to read the files (through bucket policies or ACLs).</li></ul> You cannot set OAI if your S3 bucket is configured as a website endpoint.



## CloudWatch Agent vs SSM Agent vs Custom Daemon Scripts

CloudWatch Agent	SSM Agent (AWS Systems Manager)	Custom Daemon Scripts
<p>CloudWatch agent allows you to collect more system-level metrics from your EC2 and on-premises servers than just the standard CloudWatch metrics.</p> <p>It also enables you to retrieve custom metrics from your applications or services using the <i>StatsD</i> and <i>collectd</i> protocols. <i>StatsD</i> is supported on both Linux servers and servers running Windows Server. <i>collectd</i> is supported only on Linux servers.</p> <p>You can use CloudWatch agent to collect logs from your servers and send them to CloudWatch Logs.</p> <p>Metrics collected by the CloudWatch agent are billed as custom metrics.</p> <p>You can install CloudWatch Agent using three ways:</p> <ul style="list-style-type: none"><li>• via Command Line</li><li>• via SSM Agent</li><li>• via AWS CloudFormation</li></ul>	<p>SSM Agent is Amazon software that runs on your EC2 instances and your hybrid instances that are configured for Systems Manager.</p> <p>SSM Agent processes requests from the Systems Manager service in the cloud and configures your machine as specified in the request. You can manage servers without having to log in to them using automation.</p> <p>SSM Agent sends status and execution information back to the Systems Manager service by using the <i>EC2 Messaging</i> service.</p> <p>SSM Agent runs on Amazon EC2 instances using root permissions (Linux) or SYSTEM permissions (Windows).</p> <p>CloudWatch agent replaces SSM agent in sending metric logs to CloudWatch Logs.</p>	<p>You use custom scripts (such as cron or bash scripts) if the two previously mentioned agents do not fit your needs.</p> <p>CloudWatch agent is useful for collecting system-level metrics and logs. You can create custom scripts that perform some modifications before the metrics are sent out.</p> <p>SSM Agent is also useful for automation purposes, though Systems Manager does not have a document for every case scenario. You may also have some compliance requirements that would require SSM Agent to be disabled (recall that SSM agent runs at root level permissions).</p>



## Amazon SWF vs AWS Step Function vs Amazon SQS

### Amazon Simple Workflow (SWF)

- A web service that makes it easy to coordinate work across distributed application components.
- In Amazon SWF, **tasks** represent invocations of logical steps in applications. Tasks are processed by **workers** which are programs that interact with Amazon SWF to get tasks, process them, and return their results.
- The coordination of tasks involves managing execution dependencies, scheduling, and concurrency in accordance with the logical flow of the application.

### AWS Step Functions

- A fully managed service that makes it easy to coordinate the components of distributed applications and microservices using **visual workflows**.
- You define **state machines** that describe your workflow as a series of steps, their relationships, and their inputs and outputs. State machines contain a number of **states**, each of which represents an individual step in a workflow diagram. States can perform work, make choices, pass parameters, initiate parallel execution, manage timeouts, or terminate your workflow with a success or failure.

### Amazon SQS

- A message queue service used by distributed applications to exchange messages through a polling model, and can be used to decouple sending and receiving components.
- FIFO (first-in-first-out) queues preserve the exact order in which messages are sent and received. Standard queues provide a loose-FIFO capability that attempts to preserve the order of messages.

Amazon SWF vs AWS Step Functions	AWS Step Functions vs Amazon SQS	Amazon SQS vs AWS SWF
Consider using AWS Step Functions for all your new applications, since it provides a more productive and agile approach to coordinating application components <b>using visual workflows</b> . If you require <b>external signals</b> (deciders) to intervene in your processes, or you would like to launch child processes that return a result to a parent, then you should consider Amazon SWF.	Use Step Functions when you need to coordinate service components in the development of highly scalable and auditable <b>applications</b> . Use SQS when you need a reliable, highly scalable, hosted <b>queue</b> for sending, storing, and receiving <b>messages</b> between services.	SWF API actions are <b>task-oriented</b> . SQS API actions are <b>message-oriented</b> .  The SWF Console and visibility APIs provide an <b>application-centric view</b> that lets you search for executions, drill down into an execution's details, and administer executions. SQS requires <b>implementing</b> such additional functionality.



<p>With Step Functions, you write state machines in <b>declarative JSON</b>. With Amazon SWF, you write a <b>decider program</b> to separate activity steps from decision steps. This provides you complete control over your orchestration logic, but increases the complexity of developing applications. You may write decider programs in the programming language of your choice, or you may use the Flow framework, which is a library for building SWF applications, to use programming constructs that structure asynchronous interactions for you.</p>	<p>Step Functions <b>keeps track</b> of all tasks and events in an application. Amazon SQS requires you to <b>implement your own</b> application-level tracking, especially if your application uses multiple queues. The Step Functions Console and visibility APIs provide an <b>application-centric view</b> that lets you search for executions, drill down into an execution's details, and administer executions. Amazon SQS requires <b>implementing</b> such additional functionality.</p>	<p>SWF <b>keeps track</b> of all tasks and events in an application. SQS requires you to <b>implement your own</b> application-level tracking, especially if your application uses multiple queues.</p>
	<p>Step Functions offers several <b>features that facilitate application development</b>, such as passing data between tasks and flexibility in distributing tasks. Amazon SQS requires you to <b>implement</b> some application-level functionality.</p>	<p>SWF offers several <b>features that facilitate application development</b>, such as passing data between tasks, signaling, and flexibility in distributing tasks. SQS requires you to <b>implement</b> some application-level functionality.</p>
	<p>You can use Amazon SQS to build basic workflows to coordinate your distributed application, but you get this facility out-of-the-box with Step Functions, alongside other application-level capabilities.</p>	<p>In addition to a core SDK that calls service APIs, SWF provides the <b>Flow Framework</b> with which you can write distributed applications using programming constructs that structure asynchronous interactions.</p>



## Application Load Balancer vs Network Load Balancer vs Classic Load Balancer

	Application Load Balancer	Network Load Balancer	Classic Load Balancer
Quick Info	Application Load Balancer is one of the newer types of load balancers. AWS recommends that you use this instead of the classic load balancer. This ELB operates at OSI Layer 7 (HTTP)	Network Load Balancer is one of the newer types of load balancers. AWS recommends that you use this instead of the classic load balancer. This ELB operates at OSI Layer 4 (TCP)	The classic load balancer is the older type of load balancer in AWS. AWS recommends that you use ALB or NLB instead for your load balancing requirements. Operates at either Layer 4 or Layer 7
Use Case	If you need flexible application management, use an Application Load Balancer.	If extreme performance and static IP is needed for your application, use a Network Load Balancer.	If you have an existing application that was built within the EC2-Classic network, then use a Classic Load Balancer.
Supported Protocols	HTTP and HTTPS	TCP, UDP and TLS	TCP, SSL/TLS, HTTP, HTTPS
Supported Platform	VPC	VPC	VPC and EC2-Classic (deprecated)
Can load balance to multiple ports on the same instance	Yes (Useful for containerized applications)	Yes	No
Supports Websockets	Yes	Yes	No
IP addresses as load balancing targets	Yes	Yes	No
Deletion Protection	Yes	Yes	No
Configurable idle connection timeout	Yes	No	Yes
Supports sticky sessions	Yes	No	Yes

## UNIQUE FEATURES

### Application Load Balancer

- You can set Lambda functions as load balancing targets
- Only ALB supports the following content-based routing method:
  - Path based routing
  - Host-based routing
  - HTTP header based-routing
  - HTTP method based-routing
  - Query string parameter based-routing
  - Source IP address CIDR based-routing
- Natively supports HTTP/2 IPv6
- Support for multiple SSL certificates on the ALB using
- Server Name Indication (SNI)
- Allows tag based IAM permission policies



- Can be configured for slow start (linearly increase the number of requests sent to target)
- Supports round-robin load balancing
- You can offload the authentication functionality from your apps into ALB
- Can redirect an incoming request from one URL to another URL INCLUDING HTTP to HTTPS
- You can set HTTP or custom responses for incoming requests to the ALB, offloading this task from your application

### Network Load Balancer

- High throughput/low latency ELB
- Can be assigned a static IP address
- Can be assigned an elastic IP address
- Preserves source IP address of non-HTTP applications on EC2 instances
- Offer multi-protocol listeners, allowing you to run applications such as DNS that rely on both TCP and UDP protocols on the same port behind a Network Load Balancer.
- TLS Termination

### Classic Load Balancer

- You can create custom security policies detailing which ciphers and protocols are supported by the ELB
- Supports both IPv4 and IPv6 for EC2-classic network

Common features between the three load balancers

- Has instance health check features
- Has built-in CloudWatch monitoring
- Logging features
- Support zonal failover
- Support connection draining when deregistering targets/instances
- Support cross-zone load balancing (evenly distributes traffic across registered instances in enabled AZs)
- Support SSL offloading/termination
- Backend server encryption
- Resource-based IAM permission policies



## S3 Transfer Acceleration vs Direct Connect vs VPN vs Snowball vs Snowmobile

### S3 Transfer Acceleration (TA)

- Amazon S3 Transfer Acceleration makes public Internet transfers to S3 faster, as it leverages Amazon CloudFront's globally distributed AWS Edge Locations.
- There is no guarantee that you will experience increased transfer speeds. If S3 Transfer Acceleration is not likely to be faster than a regular S3 transfer of the same object to the same destination AWS Region, AWS will not charge for the use of S3 TA for that transfer.
- This is not the best transfer service to use if transfer disruption is not tolerable.
- S3 TA provides the same security benefits as regular transfers to Amazon S3. This service also supports multi-part upload.
- **S3 TA vs AWS Snow\***
  - The AWS Snow\* Migration Services are ideal for moving large batches of data at once. In general, if it will take more than a week to transfer over the Internet, or there are recurring transfer jobs and there is more than 25Mbps of available bandwidth, S3 Transfer Acceleration is a good option.
  - Another option is to use AWS Snowball or Snowmobile to perform initial heavy lift moves and then transfer incremental ongoing changes with S3 Transfer Acceleration.
- **S3 TA vs Direct Connect**
  - AWS Direct Connect is a good choice for customers who have a private networking requirement or who have access to AWS Direct Connect exchanges. S3 Transfer Acceleration is best for submitting data from distributed client locations over the public Internet, or where variable network conditions make throughput poor.
- **S3 TA vs VPN**
  - You typically use (IPsec) VPN if you want your resources contained in a private network. VPN tools such as OpenVPN allow you to setup stricter access controls if you have a private S3 bucket. You can complement this further with the increased speeds from S3 TA.

### AWS Direct Connect

- Using AWS Direct Connect, data that would have previously been transported over the Internet can now be delivered through a **private physical network connection** between AWS and your datacenter or corporate network. Customers' traffic will remain in AWS global network backbone, after it enters AWS global network backbone.
- Benefits of Direct Connect vs internet-based connections
  - reduced costs
  - increased bandwidth
  - a more consistent network experience
- Each AWS Direct Connect connection can be configured with one or more **virtual interfaces**. Virtual interfaces may be configured to access AWS services such as Amazon EC2 and Amazon S3 using public IP space, or resources in a VPC using private IP space.
- You can run IPv4 and IPv6 on the same virtual interface.
- Direct Connect does not support multicast.



- A Direct Connect connection is **not redundant**. Therefore, a second line needs to be established if redundancy is required. Enable *Bidirectional Forwarding Detection* (BFD) when configuring your connections to ensure fast detection and failover.
- AWS Direct Connect offers SLA.
- Direct Connect vs IPsec VPN
  - A VPC VPN Connection utilizes IPSec to establish **encrypted network connectivity** between your intranet and Amazon VPC **over the Internet**. VPN Connections can be configured in minutes and are a good solution if you have an immediate need, have low to modest bandwidth requirements, and can tolerate the inherent variability in Internet-based connectivity. AWS Direct Connect **does not involve the Internet**; instead, it uses **dedicated, private network connections** between your intranet and Amazon VPC.
- You can combine one or more Direct Connect dedicated network connections with the Amazon VPC VPN. This combination provides an IPsec-encrypted private connection that also includes the benefits of Direct Connect.

## AWS VPN

- AWS VPN is comprised of two services:
  - AWS Site-to-Site VPN enables you to securely connect your on-premises network or branch office site to your Amazon VPC.
  - AWS Client VPN enables you to securely connect users to AWS or on-premises networks.
- Data transferred between your VPC and datacenter routes over an encrypted VPN connection to help maintain the confidentiality and integrity of data in transit.
- If data that passes through Direct Connect moves in a dedicated private network line, AWS VPN instead encrypts the data before passing it through the Internet.
- VPN connection throughput can depend on multiple factors, such as the capability of your customer gateway, the capacity of your connection, average packet size, the protocol being used, TCP vs. UDP, and the network latency between your customer gateway and the virtual private gateway.
- All the VPN sessions are **full-tunnel VPN**. (cannot split tunnel)
- AWS Site-to-Site VPN enable you to create **failover** and CloudHub solutions **with AWS Direct Connect**.
- AWS Client VPN is designed to connect devices to your applications. It allows you to choose from **OpenVPN-based client**.

## Snowball

- Snowball is a **petabyte-scale data transport** solution that uses secure appliances to transfer large amounts of data into and out of AWS.
- Benefits of Snowball include:
  - lower network costs,
  - Shorter transfer times,
  - and security using 256-bit encryption keys you manage through AWS Key Management Service (KMS)..
- Similar to Direct Connect, AWS Snowball is **physical hardware**. It includes a 10GBaseT network connection. You can order a device with either **50TB** or an **80TB** storage capacity.
- Data transported via Snowball are stored in Amazon S3 once the device arrives at AWS centers.
- AWS Snowball is not only for shipping data into AWS, but also out of AWS.



- AWS Snowball can be used as a quick order for additional temporary petabyte storage.
- For security purposes, data transfers must be completed **within 90 days of a Snowball's preparation**.
- When the transfer is complete and the device is ready to be returned, the E Ink shipping label will automatically update to indicate the correct AWS facility to ship to, and you can track the job status by using Amazon Simple Notification Service (SNS), text messages, or directly in the console.
- Snowball is the best choice if you need to more securely and quickly transfer terabytes to many petabytes of data to AWS. Snowball can also be the right choice if you don't want to make expensive upgrades to your network infrastructure, if you frequently experience large backlogs of data, if you're located in a physically isolated environment, or if you're in an area where high-bandwidth Internet connections are not available or cost-prohibitive.
- If you will be transferring data to AWS on an ongoing basis, it is better to use AWS Direct Connect.
- If multiple users located in different locations are interacting with S3 continuously, it is better to use S3 TA.
- You **cannot** export data directly from S3 Glacier. It should be first restored to S3.

## Snowmobile

- Snowmobile is Snowball with larger storage capacity. Snowmobile is literally a mobile truck.
- Snowmobile is an **Exabyte-scale data transfer** service.
- You can transfer up to **100PB** per Snowmobile.
- Snowmobile uses multiple layers of security to help protect your data including dedicated security personnel, GPS tracking, alarm monitoring, 24/7 video surveillance, and an optional escort security vehicle while in transit. All data is encrypted with 256-bit encryption keys you manage through the AWS Key Management Service (KMS).
- After the data transfer is complete, the Snowmobile will be returned to your designated AWS region where your data will be uploaded into the AWS storage services such as S3 or Glacier.
- Snowball vs Snowmobile
  - To migrate large datasets of 10PB or more in a single location, you should use Snowmobile. For datasets less than 10PB or distributed in multiple locations, you should use Snowball.
  - If you have a high speed backbone with hundreds of Gb/s of spare throughput, then you can use Snowmobile to migrate the large datasets all at once. If you have limited bandwidth on your backbone, you should consider using multiple Snowballs to migrate the data incrementally.
  - Snowmobile **does not** support data export. Use Snowball/Snowball Edge for this cause.
- When the data import has been processed and verified, AWS performs a software erasure based on NIST guidelines.



## Service Control Policies vs IAM Policies

Service Control Policies (SCP)	IAM Policies
<ul style="list-style-type: none"><li>• SCPs are mainly used along with AWS Organizations organizational units (OUs).</li><li>• SCPs do not replace IAM Policies such that they do not provide actual permissions. To perform an action, you would still need to grant appropriate IAM Policy permissions.</li><li>• Even if a Principal is allowed to perform a certain action (granted through IAM Policies), an attached SCP will override that capability if it enforces a Deny on that action.</li><li>• SCP takes precedence over IAM Policies.</li><li>• SCPs can be applied to the root of an organization or to individual accounts in an OU.</li><li>• When you apply an SCP to an OU or an individual AWS account, you choose to either enable (<a href="#">whitelist</a>), or disable (<a href="#">blacklist</a>) the specified AWS service. Access to any service that isn't explicitly allowed by the SCPs associated with an account, its parent OUs, or the master account is denied to the AWS accounts or OUs associated with the SCP.</li><li>• Any account has only those permissions permitted by every parent above it. If a permission is blocked at any level above the account, either implicitly (by not being included in an Allow policy statement) or explicitly (by being included in a Deny policy statement), a user or role in the affected account can't use that permission, even if there is an attached IAM policy granting Administrator permissions to the user.</li><li>• SCPs affect only principals that are managed by accounts that are part of the organization.</li></ul>	<ul style="list-style-type: none"><li>• IAM Policies operate at the Principal level. There are two types of IAM policies<ul style="list-style-type: none"><li>- Identity-based policies - attached to an IAM user, group, or role.</li><li>- Resource-based policies - attached to an AWS resource such as an S3 bucket.</li></ul></li><li>• IAM Policies can grant/deny a Principal permissions to perform certain actions to certain resources. This can be used together with SCP to ensure stricter controls in AWS Organizations. An IAM policy can be applied only to IAM users, groups, or roles, and it can never restrict the root identity of the AWS account.</li><li>• IAM Policies cannot be attached to OUs.</li><li>• An IAM Policy can allow or deny actions. An explicit allow overrides an implicit deny. An explicit deny overrides an explicit allow.</li></ul>



## EC2 Container Services ECS vs Lambda

### Amazon EC2 Container Service (ECS)

- Amazon ECS is a highly scalable, high performance container management service that supports Docker containers and allows you to easily run applications on a managed cluster of Amazon EC2 instances. ECS eliminates the need for you to install, operate, and scale your own cluster management infrastructure.
- With ECS, deploying containerized applications is easily accomplished. This service fits well in running batch jobs or in a microservice architecture. You have a central repository where you can upload your Docker Images from ECS container for safekeeping called Amazon ECR.
- Applications in ECS can be written in a stateful or stateless manner.
- The Amazon ECS CLI supports Docker Compose, which allows you to simplify your local development experience as well as easily set up and run your containers on Amazon ECS.
- Since your applications still run on EC2 instances, server management is your responsibility. This gives you more granular control over your system.
- It is up to you to manage scaling and load balancing of your EC2 instances as well, unlike in AWS Lambda where functions scale automatically.
- You are charged for the costs incurred by your EC2 instances in your clusters. Most of the time, Amazon ECS costs more than using AWS Lambda since your active EC2 instances will be charged by the hour.
- One version of Amazon ECS, known as AWS Fargate, will fully manage your infrastructure so you can just focus on deploying containers. AWS Fargate has a different pricing model from the standard EC2 cluster.
- ECS will automatically recover unhealthy containers to ensure that you have the desired number of containers supporting your application.



### AWS Lambda

- AWS Lambda is a function-as-a-service offering that runs your code in response to events and automatically manages the compute resources for you, since Lambda is a serverless compute service. With Lambda, you do not have to worry about managing servers, and directly focus on your application code.
- Lambda automatically scales your function to meet demands. It is noteworthy, however, that Lambda has a maximum execution duration per request of 900 seconds or 15 minutes.
- To allow your Lambda function to access other services such as Cloudwatch Logs, you would need to create an execution role that has the necessary permissions to do so.
- You can easily integrate your function with different services such as API Gateway, DynamoDB, CloudFront, etc. using the Lambda console.
- You can test your function code locally in the Lambda console before launching it into production. Currently, Lambda supports only a number of programming languages such as Java, Go, PowerShell, Node.js, C#, Python, and Ruby. ECS is not limited by programming languages since it mainly caters to Docker.
- Lambda functions must be stateless since you do not have volumes for data storage.
- You are charged based on the number of requests for your functions and the duration, the time it takes for your code to execute. To minimize costs, you can throttle the number of concurrent executions running at a time, and the execution time limit of the function.
- With Lambda@Edge, AWS Lambda can run your code across AWS locations globally in response to Amazon CloudFront events, such as requests for content to or from origin servers and viewers. This makes it easier to deliver content to end users with lower latency.



## SNI Custom SSL vs Dedicated IP Custom SSL

Server Name Indication (SNI) Custom SSL	Dedicated IP Custom SSL
<ul style="list-style-type: none"><li>◆ Relies on the SNI extension of the TLS protocol, which allows multiple domains to serve SSL traffic over the same IP address.</li><li>◆ Offers the same level of security when using Dedicated IP Custom SSL.</li><li>◆ If you configure CloudFront to serve HTTPS requests using SNI, CloudFront associates your alternate domain name with an IP address for each edge location. The IP address to your domain name is determined during the SSL/TLS handshake negotiation, and isn't dedicated to your distribution.</li><li>◆ Some older browsers do not support SNI and will not be able to establish a connection with CloudFront to load the HTTPS version of your content.</li><li>◆ You can use SNI Custom SSL with no upfront or monthly fees for certificate management.</li></ul>	<ul style="list-style-type: none"><li>◆ Mainly useful for browsers that do not support SNI.</li><li>◆ For this feature, the Amazon content delivery network allocates dedicated IP addresses to serve your SSL content at each Edge location.</li><li>◆ You will need to upload a SSL certificate and associate it with your CloudFront distributions.</li><li>◆ You can associate more than two custom SSL certificate with your AWS Account by submitting a CloudFront Limit Increase Form.</li><li>◆ This method works for every HTTPS request, regardless of the browser or other viewer that the user is using.</li><li>◆ Because of the added cost associated with dedicating IP addresses per SSL certificate, AWS charges a fixed monthly fee of \$600 for each custom SSL certificate you associate with your content delivery network distributions, pro-rated by the hour.</li><li>◆ You can switch to using a custom SSL/TLS certificate with SNI instead and eliminate the charge that is associated with dedicated IP addresses.</li></ul>





## EC2 Instance Health Check vs ELB Health Check vs Auto Scaling and Custom Health Check

### EC2 instance health check

- Amazon EC2 performs automated checks on every running EC2 instance to identify hardware and software issues.
- Status checks are performed every minute and each returns a pass or a fail status.
  - If all checks pass, the overall status of the instance is OK.
  - If one or more checks fail, the overall status is impaired.
- Status checks are built into EC2, so they cannot be disabled or deleted.
- You can create or delete alarms that are triggered based on the result of the status checks.
- There are two types of status checks

#### System Status Checks

- These checks detect underlying problems with your instance that require AWS involvement to repair. When a system status check fails, you can choose to wait for AWS to fix the issue, or you can resolve it yourself.

#### Instance Status Checks

- Monitor the software and network configuration of your individual instance. Amazon EC2 checks the health of an instance by sending an address resolution protocol (ARP) request to the ENI. These checks detect problems that require your involvement to repair.

### Elastic Load Balancer (ELB) health check

- To discover the availability of your registered EC2 instances, a load balancer periodically sends pings, attempts connections, or sends requests to test the EC2 instances.
- The status of the instances that are healthy at the time of the health check is InService. The status of any instances that are unhealthy at the time of the health check is OutOfService.
- When configuring a health check, you would need to provide the following:
  - a specific port
  - protocol to use
    - HTTP/HTTPS health check succeeds if the instance returns a 200 response code within the health check interval.
    - A TCP health check succeeds if the TCP connection succeeds.
    - An SSL health check succeeds if the SSL handshake succeeds.
  - ping path
- ELB health checks do not support WebSockets.
- The load balancer routes requests only to the healthy instances. When an instance becomes impaired, the load balancer resumes routing requests to the instance only when it has been restored to a healthy state.
- The load balancer checks the health of the registered instances using either
  - the default health check configuration provided by Elastic Load Balancing or
  - a health check configuration that you configure (auto scaling or custom health checks for example).
- Network Load Balancers use active and passive health checks to determine whether a target is available to handle requests.
  - With active health checks, the load balancer periodically sends a request to each registered target to check its status. After each health check is completed, the load balancer node closes the connection that was established.
  - With passive health checks, the load balancer observes how targets respond to connections, which enables it to detect an unhealthy target before it is reported as unhealthy by active health checks. You cannot disable, configure, or monitor passive health checks.

### Auto Scaling and Custom health checks

- All instances in your Auto Scaling group start in the healthy state. Instances are assumed to be healthy unless EC2 Auto Scaling receives notification that they are unhealthy. This notification can come from one or more of the following sources:
  - Amazon EC2 (default)
  - Elastic Load Balancing
  - A custom health check.
- After Amazon EC2 Auto Scaling marks an instance as unhealthy, it is scheduled for replacement. If you do not want instances to be replaced, you can suspend the health check process for any individual Auto Scaling group.
- If an instance is in any state other than running or if the system status is impaired, Amazon EC2 Auto Scaling considers the instance to be unhealthy and launches a replacement instance.
- If you attached a load balancer or target group to your Auto Scaling group, Amazon EC2 Auto Scaling determines the health status of the instances by checking both the EC2 status checks and the Elastic Load Balancing health checks.
- Amazon EC2 Auto Scaling waits until the health check grace period ends before checking the health status of the instance. Ensure that the health check grace period covers the expected startup time for your application.
- Health check grace period does not start until lifecycle hook actions are completed and the instance enters the InService state.
- With custom health checks, you can send an instance's health information directly from your system to Amazon EC2 Auto Scaling.



## Backup and Restore vs Pilot Light vs Warm Standby vs Multi - site

You should select the most appropriate DR plan to meet your company RTO and RPO. Consider also your budget and which system elements are most critical for your business.

Backup Restore	Pilot Light
<ul style="list-style-type: none"><li>• This DR plan provides the slowest system restoration after a DR event.</li><li>• You take frequent snapshots of your data such as those in Amazon EBS Volumes and Amazon RDS databases, and you store them in a durable and secure storage location such as Amazon S3.</li><li>• There are many ways for you to move data in and out of S3<ul style="list-style-type: none"><li>- Transfer over the network via S3 Transfer Acceleration</li><li>- Transfer over a dedicated network line using AWS Direct Connect</li><li>- Transfer using transport hardware such as AWS Snowball and Snowmobile</li></ul></li><li>• With S3 Glacier, you get to reduce a large portion of your costs compared to using S3 Standard, since Glacier is meant for long term archival storage which is perfect for backups.</li><li>• AWS Storage Gateway enables snapshots of your on-premises data volumes to be transparently copied into S3 for backup.<ul style="list-style-type: none"><li>- Storage-cached volumes allow you to store your primary data in S3, but keep your frequently accessed data local for low-latency access.</li><li>- Gateway-VTL of AWS Storage Gateway serves as a replacement for traditional magnetic tape backup.</li></ul></li><li>• You can quickly create local volumes or Amazon EBS volumes from snapshots in S3.</li><li>• You can create AMIs out of your EC2 instances which preserve the following:<ul style="list-style-type: none"><li>- A template for the root volume for the instance (for example, an operating system, an application server, and applications)</li><li>- Launch permissions that control which AWS accounts can use the AMI to launch instances</li><li>- A block device mapping that specifies the volumes to attach to the instance when it's launched</li></ul></li><li>• Backup and restore is used in combination with other DR plans since it is crucial to always have a working backup of your system.</li></ul>	<ul style="list-style-type: none"><li>• The pilot light method gives you a quicker recovery time than the backup-and-restore method because the core pieces of the system are already running and are continually kept up to date, but is not as fast as Warm Standby.</li><li>• You can maintain a pilot light by configuring and running the <b>most critical core elements</b> of your system in AWS.</li><li>• When the time comes for recovery, you can rapidly provision a full-scale production environment around the critical core.</li><li>• Pilot light is an example of <b>active/passive failover configuration</b>. Infrastructure elements for the pilot light itself typically include your database servers, which would be configured for data mirroring replication.</li><li>• Restoring the rest of the system includes utilizing EBS snapshots and EC2 AMIs that you should be regularly generating.</li><li>• Pilot light tends to be more costly than backup and restore since you leave a few core AWS resources running all the time.</li><li>• From a networking point of view, you have two main options for provisioning web servers:<ul style="list-style-type: none"><li>- Use Elastic IP addresses, which can be pre-allocated and pre-identified, and associate them with your instances.</li><li>- Use Elastic Load Balancing (ELB) to distribute traffic to multiple instances.</li></ul></li><li>• You would then update your DNS records to point at your EC2 instance or point to your load balancer using a CNAME.</li><li>• Consider redundancy especially at your data layer (enable multi-AZ, cluster sharding, etc).</li><li>• If your data is constantly changing and failover occurs, you would have to reverse replicate your data in the DR site back to the primary site, so that any data updates received while the primary site was down can be replicated back, without the loss of data.</li></ul>



## Warm Standby

- This DR plan is faster in system restoration than performing Pilot Light after a DR event, but is not as fast as having a Multi-site System.
- Warm standby describes a DR scenario in which a scaled-down version of a fully functional environment is always running in the cloud.
- Since it is not only your core elements that are running all the time, warm standby is usually more costly than pilot light.
- Warm standby is another example of **active/passive failover configuration**.
- Servers can be left running in a minimum number of EC2 instances on the smallest sizes possible. Once failover occurs, quickly resize them and add scaling capabilities. It is best to place these instances behind a load balancer as well.
- For the data layer, the practice is similar to pilot light where a standby resource is present and changing data is constantly being replicated to the other.
- In the case of failure of the production system, the standby environment will be scaled up for production load , and DNS records will be changed to route all traffic to AWS.
- If your data is constantly changing and failover occurs, you would have to reverse replicate your data in the DR site back to the primary site, so that any data updates received while the primary site was down can be replicated back, without the loss of data.

## Multi-site

- This DR plan is the fastest in system restoration during a DR event.
- Multi-site is a one-to-one copy of your infrastructure that is located and running in another region or AZ, known as an **active-active configuration**.
- Because of this, multi-site is the most expensive among all DR plans.
- Multi-site gives you the best RTO and RPO as no downtime is expected and little to no data loss should be experienced. In addition to recovery point options, there are various replication methods, such as synchronous and asynchronous methods.
- You can use a DNS service that supports weighted routing, such as Amazon Route 53, to route production traffic to different sites that deliver the same application or service.
- During failover, you can quickly increase compute capacity by using AWS Auto Scaling or by resizing your instances to a larger size.
- Multiple services in AWS such as RDS offer a multi-AZ feature which allows you to provision resources in a different location for a more fault-tolerant setup.
- If your data is constantly changing and failover occurs, you would have to reverse replicate your data in the DR site back to the primary site, so that any data updates received while the primary site was down can be replicated back, without the loss of data.



## Latency Routing vs Geoproximity Routing vs Geolocation Routing

### Latency Routing

#### Definition

Lets Route 53 serve user requests from the AWS Region that provides the lowest latency. It does not, however, guarantee that users in the same geographic region will be served from the same location.

Latency-based routing is based on latency measurements performed over a period of time, and the measurements reflect changes in network connectivity and routing.

#### How it works

To use latency-based routing, you create **latency records** for your resources in multiple AWS Regions. When Route 53 receives a DNS query for your domain or subdomain, it determines which AWS Regions you've created latency records for, determines which region gives the user the lowest latency, and then selects a latency record for that region. Route 53 responds with the value from the selected record, such as the IP address for a web server.

Record sets can be created using any record type supported by Route 53, except NS or SOA records.

#### Use Case

Use when you have **resources in multiple AWS Regions** and you want to route traffic to the **region that provides the best latency**

### Geoproximity Routing

#### Definition

Lets Amazon Route 53 route traffic to your resources based on the geographic location of your users and your resources.

You can also optionally choose to route more traffic or less to a given resource by specifying a value, known as a bias. A bias expands or shrinks the size of the geographic region from which traffic is routed to a resource.

#### How it works

To use geoproximity routing, you must use **Route 53 traffic flow**.

You create traffic flow policies for your resources and specify one of the following values for each policy:

- If you're using AWS resources, you can set the AWS Region where your resource is created
- If you're using non-AWS resources, you can enter the latitude and longitude of the resource

#### Use Case

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.



# Geolocation Routing

## Definition

Resources serve traffic based on the geographic location of your users, meaning the location that DNS queries originate from.

## How it works

Geolocation works by **mapping IP addresses to locations**. Some IP addresses aren't mapped to geographic locations, so Amazon Route 53 will receive some DNS queries from locations that it can't identify.

You can create a default record that handles both queries from IP addresses that aren't mapped to any location and queries that come from locations that you haven't created geolocation records for. If you don't create a **default record**, Route 53 returns a "no answer" response for queries from those locations.

No two records should specify the same geographic location.

## Use Case

Use when you want to route traffic **based on the location of your users**.

- You can localize your content and present some or all of your website in the language of your users.
- You can restrict distribution of content to only the locations in which you have distribution rights.
- Useful for balancing load across endpoints in a predictable, easy-to-manage way, so that each user location is consistently routed to the same endpoint.





## Multi-AZ deployments vs. Multi-Region deployments vs. Read Replicas

Multi-AZ deployments	Multi-Region deployments	Read Replicas
Main purpose is high availability	Main purpose is disaster recovery and local performance	Main purpose is scalability
Non-Aurora: synchronous replication; Aurora: asynchronous replication	Asynchronous replication	Asynchronous replication
Non-Aurora: only the primary instance is active; Aurora: all instances are active	All regions are accessible and can be used for reads	All read replicas are accessible and can be used for readscaling
Non-Aurora: automated backups are taken from standby; Aurora: automated backups are taken from shared storage layer	Automated backups can be taken in each region	No backups configured by default
Always span at least two Availability Zones within a single region	Each region can have a Multi-AZ deployment	Can be within an Availability Zone, Cross-AZ, or Cross-Region
Non-Aurora: database engine version upgrades happen on primary; Aurora: all instances are updated together	Non-Aurora: database engine version upgrade is independent in each region; Aurora: all instances are updated together	Non-Aurora: database engine version upgrade is independent from source instance; Aurora: all instances are updated together
Automatic failover to standby (non-Aurora) or read replica (Aurora) when a problem is detected	Aurora allows promotion of a secondary region to be the master	Can be manually promoted to a standalone database instance (non-Aurora) or to be the primary instance (Aurora)



## FINAL REMARKS AND TIPS

That's a wrap! Thank you once again for choosing our Study Guide and Cheat Sheets for the AWS Certified Solutions Architect Associate (SAA-C02) exam. The Tutorials Dojo team spent considerable time and effort to produce this content to help you pass the AWS exam.

We also recommend that before you take the actual SAA-C02 exam, allocate some time to check your readiness first by taking our AWS practice test course in the Tutorials Dojo Portal. This will help you identify the topics that you need to improve on and help reinforce the concepts that you need to fully understand in order to pass the SAA-C02 exam. It also has different training modes that you can choose from such as Timed mode, Review mode, Section-Based tests, Topic-based tests, and Final test plus bonus flashcards. In addition, you can read the technical discussions in our forums or post your queries if you have one. If you have any issues, concerns or constructive feedback on our eBook, feel free to contact us at [support@tutorialsdojo.com](mailto:support@tutorialsdojo.com).

On behalf of the Tutorials Dojo team, I wish you all the best in your upcoming AWS Certified Solutions Architect - Associate exam. May it help advance your career, as well as increase your earning potential.

With the right strategy, hard work, and unrelenting persistence, you can definitely make your dreams a reality!  
You can make it!

Sincerely,  
Jon Bonso, Adrian Formaran and the Tutorials Dojo Team