JON BONSO

AWS CERTIFIED CLOUD PRACTITIONER EXAM

Tutorials Dojo Study Guide

TABLE OF CONTENTS

INTRODUCTION

We are in an age of rapid technological innovation and information exchange. New technologies are being produced every day by different industries, governments, and researchers to make life more enjoyable. Hence, people are also beginning to shift their infrastructures onto the cloud, especially onto Amazon Web Services (AWS). The cloud is the perfect platform for innovation. It allows you to obtain compute and storage capacity simply through a click of a button. There is no need to meticulously allocate capital anymore for physical infrastructure and setting them up yourself.

For several years, AWS has been recognized as the leading cloud provider in the market[1]. They have been continuously upgrading their services to deliver customer satisfaction and drive customer success. Every year, you can expect AWS to deliver something new to the table. And since the AWS cloud is already so vast, industries will need trained people who understand how the AWS cloud operates and how to maximize solutions that will produce the best results. AWS formalizes this process of training and recognition through their highly valued **AWS Certifications**.

The path for learning cloud is like a long and exciting journey. Becoming an AWS Cloud Practitioner is a great way to start it off. It opens up a lot of career opportunities for you, and you can choose the path that you want to take. You can become a cloud solutions architect, a cloud developer, a system operations administrator, data analyst or a specialist of your choosing. The AWS Cloud Practitioner course is the first step in helping you understand the value of moving to the cloud, as well as the basic AWS services which are fundamental and crucial for building success in AWS.

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 that you take our AWS Certified Cloud Practitioner video course with included hands-on labs and our high-quality practice exams to further expand your knowledge and improve your test-taking skills.

1 https://aws.amazon.com/blogs/aws/aws-named-as-a-leader-in-gartners-infrastructure-as-a-service-iaas-magic-quadrant-for-the-9th-consecutiveyear/

AWS CERTIFIED CLOUD PRACTITIONER EXAM OVERVIEW

Amazon Web Services began its Global Certification Program in 2013 with the primary purpose of validating the technical skills and knowledge of IT Professionals in building secure and reliable cloud-based applications using the AWS Cloud. On April 2013, AWS launched its first-ever AWS Certification test called the AWS Certified Solutions Architect Associate exam. This was followed by the AWS Certified SysOps Administrator and AWS Certified Developer Associate exams.

Amazon has been continuously expanding and updating its certification program year after year. They launched a series of Professional and Specialty-level certifications that cover various topics like DevOps, machine learning, data analytics, advanced networking, and many others. As the number of AWS services increases, a new and updated version of the AWS certification exam is released regularly to reflect the recent service changes and include the new knowledge areas.

On December 2017, AWS launched its entry-level certification test called the AWS Certified Cloud Practitioner exam. This exam is recommended for professionals with a non-technical background and individuals who are quite new to the IT industry, including college students and fresh graduates. The Cloud Practitioner exam checks your understanding of the different cloud concepts, cloud terminologies, AWS services, and other basic topics in AWS. It has an exam code of CLF-C02 and has no prerequisites – meaning you can take it directly without having to earn any prior certification, degree, or training.

The exam contains a mixture of scenario-based and easy WH questions that can either be in multiple-choice or multiple-response formats. The first question type has one correct answer and three incorrect responses, while the multiple-response format has two or more correct responses out of five or more options. The exam costs a hundred US dollars and can be taken either from a local testing center or online from the comfort of your home.

The Cloud Practitioner certification exam has a total of 65 questions that you should complete within 90 minutes or one hour and a half. The score range for this test is from 100 to 1,000, with a minimum passing score of 700. AWS is using a scaled scoring model to equate scores across multiple exam types that may have different difficulty levels. An email of your result will be sent to you after a few days, and the complete score report will be available to your AWS Certification account afterward.

The AWS Certified Cloud Practitioner exam is the easiest one among all of the AWS certification tests. It's easier than the others because most of the items being asked are just WH questions, so you'll see a one-liner question that starts with What, When, Where, Who, Why, Which, and How. Some items have one statement describing the scenario and another line asking the actual question. However, these questions do not exceed two lines, meaning that the items are fairly concise. The options are short too, which can be the name of an AWS Service, a phrase, or a brief statement.

This is in stark contrast with the Associate, Professional, and Specialty-level AWS exams, where you'll see long-winded scenarios and options. That's why the Cloud Practitioner exam is considered entry-level and very manageable to get a pass. However, this does not mean you don't have to study for it. The exam contains a handful of difficult questions on AWS Billing, the AWS Shared Responsibility Model, and other advanced cloud concepts; thus, you still have to study for this test to ensure a passing score.

Individuals who unfortunately did not pass the AWS exam must wait for 14 days before they are allowed to retake the exam. There is no hard limit on the number of exam attempts, so you can try again and again until you pass the exam. Take note that on each attempt, the full registration price of the exam must be paid.

Your AWS Certification Account will have a record of your complete exam results within 5 business days of completing your exam. The score report contains a table of your performance for each exam domain, which indicates whether you met the competency level required for these domains or not. AWS uses a compensatory scoring model, which means that you do not necessarily need to pass each and every individual section.

You will pass this exam as long as you get an overall score of 700 across 4 domains. Each section has a specific score weighting that translates to the number of questions; hence, some sections have more questions than others. Your Score Performance table highlights your strengths and weaknesses that you need to improve on.

Exam Details

The AWS Certified Cloud Practitioner (CLF-C02) examination is intended for individuals who have the knowledge and skills necessary to effectively demonstrate an overall understanding of the AWS Cloud, independent of specific technical roles addressed by other AWS certifications (for example, Solutions Architect - Associate, Developer - Associate, or SysOps Administrator - Associate). It is composed of identification and enumeration questions that are formatted as either multiple-choice or multiple-response.

For multiple-choice types of questions, you will have to choose one correct response out of four options. For multiple-response types of questions, you will have to choose two or more correct responses out of five or more options. You can take the exam via online proctoring or from a testing center close to you.

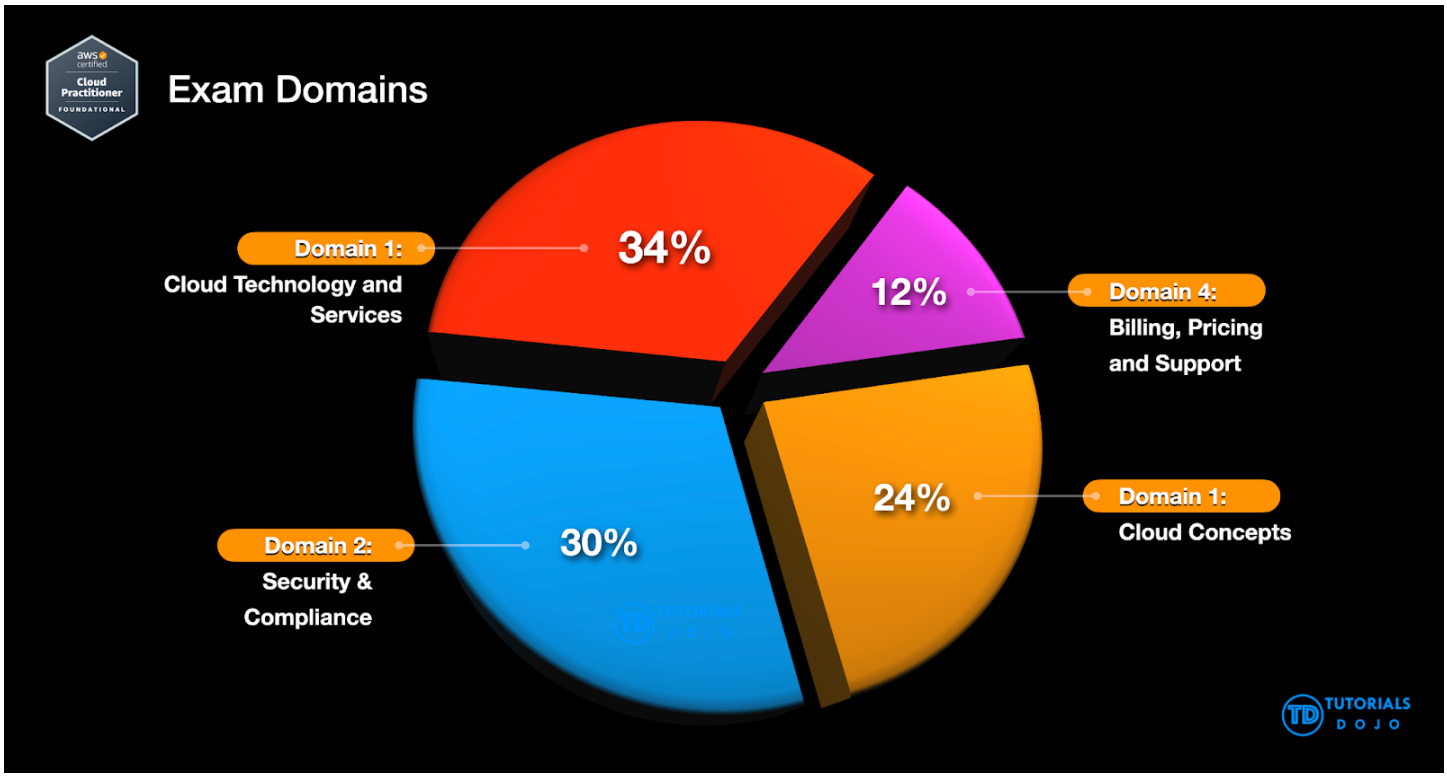
Exam Code:	CLF-C02
Prerequisites:	None
No. of Questions:	65
Score Range:	100-1000
Cost:	100 USD
Passing Score:	700
Time Limit:	90 minutes

Exam Domains

The AWS Certified Cloud Practitioner (CLF-C02) exam has four different domains, each with a corresponding weight and topic coverage. The domains are as follows:

- Domain 1: Cloud Concepts (24%)
- Domain 2: Security & Compliance (30%)
- Domain 3: Cloud Technology and Services (34%)
- Domain 4: Billing, Pricing and Support (12%)

One exam domain is comprised of several task statements. A task statement is a sub-category of the exam domain that contains the required cloud concepts, knowledge, and skills for you to accomplish a particular task or activity in AWS. In the AWS Certified Cloud Practitioner (CLF-C02) test, the **Domain 3: Cloud Technology and Services** has the biggest weighting in the exam at 34% so expect to see a lot of technology-related scenarios in the exam. Conversely, the exam domain with the least exam weighting is **Domain 4: Billing, Pricing & Support** so you have to limit the time you spend studying under this knowledge area.



Domain 1: Cloud Concepts

1.1. Define the benefits of the AWS Cloud.
1.2. Identify design principles of the AWS Cloud.
1.3. Understand the benefits of and strategies for migration to the AWS Cloud.
1.4. Understand concepts of cloud economics.

Domain 2: Security and Compliance

2.1. Understand the AWS shared responsibility model.
2.2. Understand AWS Cloud security, governance, and compliance concepts.
2.3. Identify AWS access management capabilities.
2.4. Identify components and resources for security.

Domain 3: Cloud Technology and Services

3.1. Define methods of deploying and operating in the AWS Cloud.
3.2. Define the AWS global infrastructure.
3.3. Identify AWS compute services.
3.4. Identify AWS database services.
3.5. Identify AWS network services.
3.6. Identify AWS storage services.
3.7. Identify AWS artificial intelligence and machine learning (AI/ML) services and analytics services.
3.8. Identify services from other in-scope AWS service categories.

Domain 4: Billing, Pricing and Support

4.1. Compare AWS pricing models.
4.2. Understand resources for billing, budget, and cost management.
4.3. Identify AWS technical resources and AWS Support options.

Exam-Related AWS Topics and Services

The official exam guide contains a list of key tools, technologies, and concepts that may show up on the Cloud Practitioner test. Keep in mind that this is just a non-exhaustive list of the tools and technologies that may or may not appear on the exam. This list can change at any time and is primarily given to test-takers to help them understand the general scope of services, features, or technologies for this certification. In addition, the general tools and technologies in this list appear in no particular order.

Here are the topics, AWS services, and concepts that you should focus on for your upcoming exam. You have to review your knowledge on:

• APIs • Benefits of migrating to the AWS Cloud • AWS Cloud Adoption Framework (AWS CAF) • AWS Compliance • Compute • Cost management • Databases • Amazon EC2 instance types (for example, Reserved, On-Demand, Spot) • AWS global infrastructure (for example, AWS Regions, Availability Zones) • Infrastructure as code (IaC) • AWS Knowledge Center • Machine learning • Management and governance • Migration and data transfer • Network services	• AWS Partner Network • AWS Prescriptive Guidance • AWS Pricing Calculator • AWS Professional Services • AWS re:Post • AWS SDKs • Security • AWS Security Blog • AWS Security Center • AWS shared responsibility model • AWS Solutions Architects • Storage • AWS Support Center • AWS Support plans • AWS Well-Architected Framework

Remember that out of the 4 exam domains, the Cloud Technology and Services domain has the biggest coverage in the exam, at 34 percent. This means that a third of the questions in the entire AWS Certified Cloud Practitioner exam covers the many cloud services and features in AWS. Most of these AWS services can be grouped according to their primary functions or use cases, such as Analytics, Application Integration, Compute, Database, Networking, et cetera.

The Appendix section of the exam guide also includes a list of relevant AWS services that you should focus on, so in your exam, make sure that you review the following AWS services.

For Analytics, we have Amazon Athena, AWS Data Exchange, Amazon EMR, AWS Glue, Amazon Kinesis, Amazon Managed Streaming for Apache Kafka (Amazon MSK), Amazon OpenSearch Service, Amazon QuickSight and Amazon Redshift.

For Application Integration, learn about Amazon EventBridge, Amazon Simple Notification Service (Amazon SNS) , Amazon Simple Queue Service (Amazon SQS) and AWS Step Functions. Additionally, you should also know how Business Applications in AWS works such as Amazon Connect and Amazon Simple Email Service (Amazon SES).

For Cloud Financial Management, research on how you can utilize the AWS Billing Conductor, AWS Budgets, AWS Cost and Usage Report, AWS Cost Explorer and AWS Marketplace in your organization.

For Computing services, study AWS Batch, Amazon EC2, AWS Elastic Beanstalk, Amazon Lightsail, AWS Local Zones, AWS Outposts and AWS Wavelength.Containers services are also in scope so make sure you know Amazon Elastic Container Service, Amazon Elastic Kubernetes Service and Amazon Elastic Container Registry (Amazon ECR). Serverless compute options like AWS Fargate and AWS Lambda are also covered.

For Customer Engagement, you have AWS Activate for Startups, AWS IQ, AWS Managed Services (AMS) and the various AWS Support Plan types that you can avail of. End User Computing services are also covered such as Amazon AppStream 2.0, Amazon WorkSpaces and Amazon WorkSpaces Web.

For Databases, you have Amazon Aurora, Amazon DynamoDB, Amazon MemoryDB for Redis, Amazon Neptune, Amazon RDS and other cloud databases in AWS.

For Developer Tools, familiarize yourself with the CI/CD services in AWS, namely AWS AppConfig, AWS CLI, AWS Cloud9, AWS CloudShell, AWS CodeArtifact, AWS CodeBuild, AWS CodeCommit, AWS CodeDeploy, AWS CodePipeline, AWS CodeStar and AWS X-Ray.

AWS Amplify, AWS AppSync and AWS Device Farm are the relevant services for Frontend Web & Mobile section while AWS IoT Core and AWS IoT Greengrass are for the Internet of Things (IoT) services in AWS.

There are also several Machine Learning services covered in the exam such as Amazon Comprehend, Amazon Kendra, Amazon Lex, Amazon Polly, Amazon Rekognition, Amazon SageMaker, Amazon Textract, Amazon Transcribe and Amazon Translate. Don’t worry too much as the AI-related questions for this category should be on the easy level and not in-depth.

The Cloud Practitioner exam covers a handful of services that relates to Management and Governance. These are AWS Auto Scaling, AWS CloudFormation, AWS CloudTrail, Amazon CloudWatch, AWS Compute Optimizer, AWS Config, AWS Control Tower, AWS Health Dashboard, AWS Launch Wizard, AWS License Manager, AWS Management Console, AWS Organizations, AWS Resource Groups and Tag Editor, AWS Service Catalog, AWS Systems Manager, AWS Trusted Advisor and AWS Well-Architected Tool.

For Migration and Transfer, it covers AWS Application Discovery Service, AWS Application Migration Service, AWS Database Migration Service (AWS DMS), AWS Migration Hub, AWS Schema Conversion Tool (AWS SCT), AWS Snow Family, AWS Transfer Family. For Networking and Content Delivery category, Amazon API Gateway, Amazon CloudFront, AWS Direct Connect, AWS Global Accelerator, Amazon Route 53, Amazon VPC and AWS VPN.

For Security, Identity, and Compliance category, prepare to see a range of AWS services that you can use to secure your enterprise applications and AWS resources. Check out the AWS Artifact, AWS Audit Manager, AWS

Certificate Manager (ACM), AWS CloudHSM, Amazon Cognito, Amazon Detective, AWS Directory Service, AWS Firewall Manager, Amazon GuardDuty, AWS Identity and Access Management (IAM), AWS IAM Identity Center (AWS Single Sign-On), Amazon Inspector, AWS Key Management Service (AWS KMS), Amazon Macie, AWS Network Firewall, AWS Resource Access Manager (AWS RAM), AWS Secrets Manager, AWS Security Hub, AWS Shield and AWS WAF. You have to pay attention to how these services work together and know the appropriate AWS service to use for a particular business case or situation.

Lastly, don't forget to study the plethora of cloud storage services at your disposal, such as AWS Backup, Amazon Elastic Block Store (Amazon EBS), Amazon Elastic File System (Amazon EFS), AWS Elastic Disaster Recovery, Amazon FSx, Amazon S3, Amazon S3 Glacier and the various types of AWS Storage Gateway.

Exam Scoring System

You can get a score from 100 to 1,000 with a minimum passing score of **700** when you take the AWS Certified Cloud Practitioner exam. AWS uses a scaled scoring model to associate scores across multiple exam types that may have different levels of difficulty. Your complete score report will be sent to you by email 1 - 5 business days after your exam. However, as soon as you finish your exam, you'll immediately see a pass or fail notification on the testing screen.

For individuals who unfortunately do not pass their exams, you must wait 14 days before you are allowed to retake the exam. There is no hard limit on the number of attempts you can retake an exam. Once you pass, you'll receive various benefits such as a discount coupon which you can use for your next AWS exam.

Once you receive your score report via email, the result should also be saved in your AWS Certification account already. The score report contains a table of your performance on each domain and it will indicate whether you have met the level of competency required for these domains. Take note that you do not need to achieve competency in all domains for you to pass the exam. At the end of the report, there will be a score performance table that highlights your strengths and weaknesses which will help you determine the areas you need to improve on.

Section	Score Performance		
	% of Scored Items	Needs Improvement	Meets Competencies
Domain 1.0: Cloud Concepts	34%		
Domain 2.0: Security and Compliance	30%		
Domain 3.0: Cloud Technology and Services	24%		
Domain 4.0: Billing, Pricing and Support	12%		

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 CLOUD PRACTITIONER EXAM STUDY GUIDE

The AWS Certified Cloud Practitioner exam or AWS CCP is the easiest to achieve among all the AWS certification exams. This certification covers most, if not all, fundamental knowledge that one should know when venturing into the Cloud. The AWS CCP course intends to provide practitioners a fundamental understanding of the AWS Cloud without having to dive deep into the technicalities. This includes the AWS Global Infrastructure, best practices in using AWS Cloud, pricing models, technical support options, and many more. You can view the complete details and guidelines for the certification exam here.

What to review

1. The AWS Cloud Services

Currently, AWS offers more than 160+ services and products to their customers. And every year, the list grows longer. You don't have to memorize every single service and function to pass the exam (although that would be amazing if you did!). What's important is that you familiarize yourself with the more commonly used services such as those under **compute**, **storage**, **databases**, **security**, **networking and content delivery, management and governance,** and a few others**.** To quickly view over the different categories, you may visit this link.

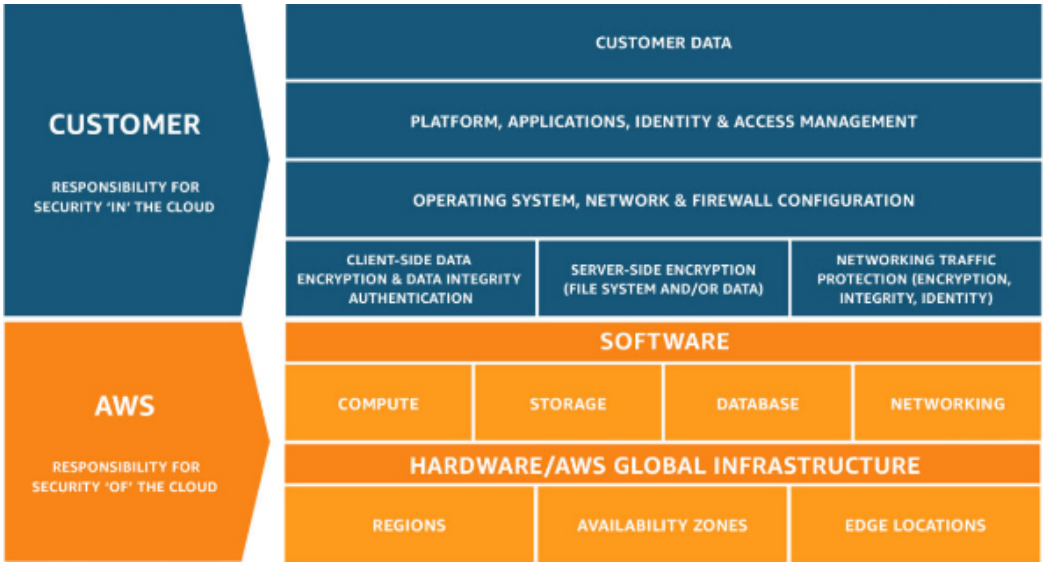
To help you get started with the familiarization, this AWS whitepaper contains an overview of the different AWS services along with their definitions and use cases. It is also important to know what cloud computing introduces into the industry, and how the AWS Global Infrastructure is set up to help you maximize the capabilities of cloud computing. Aside from questions on the different services, questions about Regions and Availability Zones commonly pop up in the exam as well.

2. Best Practices when Architecting for the Cloud

This section is highly important and might comprise the bulk of your CCP exam. Focus on reading the contents of this AWS Well-Architected Framework whitepaper. The best practices are essentially the ways you can take advantage of AWS Cloud's strengths. This paper elaborates on the different pillars that make up a well-architected system. Reading through the design principles and core services of each pillar will help you connect the dots between the best practices and AWS services. Lastly, you can visit this site to gather more information and view additional content for your review of this section.

3. Security in the Cloud

Security in the AWS Cloud is another major part of your CCP Exam. AWS has defined the security controls that they manage and the security controls that you manage through the **Shared Responsibility Model** below.



The primary resource that you should be studying for this section is this whitepaper. The AWS Security Best Practices whitepaper discusses the many ways you can secure your applications and services. I suggest you thoroughly review the following:

1) Data encryption at rest and in transit (EBS, S3, EC2, RDS, etc)

2) Identity and Access Management (IAM)

3) VPC and Application Network Security (security groups, ACLs, etc)

4) Monitoring and Logging of your Infrastructure (Cloudwatch, cloudtrail, etc)

5) AWS Compliance Programs

4. AWS Pricing Model

One of the advantages of using Cloud is having on-demand capacity provisioning. Therefore, it is also crucial for you to understand the provider's pricing model. AWS charges you in multiple ways. There is no exact model that applies to all, since different AWS services have their own cost plans. However, AWS has three fundamental drivers of cost that usually apply to any kind of service. They are:

i. Compute cost
ii. Storage cost
iii. Outbound data transfer cost

Aside from on-demand capacity provisioning, AWS also offers you multiple ways to lower your total cost, such as the option to reserve capacity or create a savings plan.

Detailed information about each of these costs can be seen in this whitepaper, which also serves as your main study material for this section. The purpose of studying cost and pricing models is to help you optimize your costs in AWS. AWS provides a great tool to calculate expected monthly costs, known as the **AWS Pricing Calculator**. Note that the CCP exam frequently asks scenarios where you'd have to optimize your costs.

5. AWS Support Plans

AWS offers different types of support plans namely: Basic, Developer, Business, Enterprise and Enterprise On-Ramp. It is important to know how each support plan differs from one another. With that said, this webpage will serve as your primary study material. You might miss the subtle details if you don't read each support plan properly, so be sure to take note of these details.

In tandem with learning the AWS Support Plans is studying AWS Trusted Advisor. AWS Trusted Advisor is a tool that offers best practice checks and recommendations across various categories such as Cost Optimization, Security, Fault Tolerance, Performance, Operational Excellence and Service Limits. You do not need to memorize each check in AWS Trusted Advisor, though browsing through them is an advantage.

How to review

As with any exam, the very first step is always the same - **KNOWING WHAT TO STUDY**. Although we have already enumerated them in the previous section, I highly suggest you go over the AWS Certified Cloud Practitioner Exam Guide again and see the exam contents.

AWS already has a vast number of free resources available for you to prepare for the exam. I suggest you first read Overview of Amazon Web Services whitepaper, and gain a good understanding of the different AWS concepts and services. Again, you don't need to memorize every single AWS service and function there. Rather, focus on the services that are more commonly used by the industry. You can check out the amazing Tutorials Dojo cheat sheets to supplement your review for this section.

After reviewing the services whitepaper, I recommend reading the whitepaper How Pricing Works next. The AWS CCP exam frequently throws out tricky questions about pricing, TCO and cost optimization. Be extra careful in answering questions that ask for the most cost effective solution. Always prioritize utility over pricing, since there might be a choice in the question where it is the cheapest solution, but is not appropriate for the scenario's needs. You can compare the pricing of the different services here on this website.

The AWS Security Best Practices whitepaper discusses what you'll need to know for AWS Security. Also, familiarize yourself with the **Shared Responsibility Model**. This frequently comes up in the AWS CCP exam.

With security, you should know the following:

- Protect your data in AWS and going out of AWS. Different services have different encryption methods and protocols.
- Network level security and subnet level security. There are many ways you can secure your VPC and the services inside it, such as NACLs and security groups.
- Be comfortable with IAM. Focus on concepts of IAM users, groups, policies and roles.
- Understand AWS monitoring and logging features such as Cloudwatch, CloudWatch Logs, VPC Logs and CloudTrail.

The last whitepaper you need to review is the AWS Well-Architected Framework whitepaper. The material nicely wraps up all the AWS services, products, features, and pricing that you've learned. It is very important to understand what the best practices are, since scenario questions in the exam always revolve around these topics. You can open up an AWS Management Console to help you visualize what is being discussed in this paper.

After reading through all the whitepapers, the last section of your review is the AWS Support Plans. This is a quick browse of a webpage, and shouldn't take you long to study. Take note of what support plans are available, and how they differ from each other. There might be questions in the exam that ask which support plan offers some specific service.

AWS also provides a free, online virtual course called AWS Cloud Practitioner Essentials which you can take to better prepare yourself for the AWS CCP exam. This course contains a set of video lectures that summarize everything you've read so far in your review, and discuss topics you might have missed.

Also check out this article: **Top 5 FREE AWS Review Materials**.

Common Exam Scenarios

Scenario	Solution
Domain 1: Cloud Concepts	
A key financial benefit of migrating systems hosted on your on-premises data center to AWS.	- Replaces upfront capital expenses (CAPEX) with low variable operational expenses (OPEX). - Reduce the Total Cost of Ownership (TCO)
4 cloud architectures design principle in AWS	1. Design for failure. Decouple your components Implement elasticity 2. Think parallel
A cloud architecture for mission-critical workloads in AWS which must be highly-available.	Use multiple Availability Zones
A change or a failure in one component should not cascade to other components.	Loose coupling
You need to enable your Amazon EC2 instances in the public subnet to connect to the public Internet.	Internet Gateway
You need to enable your EC2 instances in the private subnet to connect to the public Internet.	NAT Gateway
Domain 2: Security and Compliance	
A security management tool to configure your AWS WAF rules across your accounts.	AWS Firewall Manager
A company needs to download the compliance-related documents in AWS such as Service Organization Controls (SOC) reports	AWS Artifact
Improve the security of IAM users.	- Enable Multi-Factor Authentication (MFA) - Configure a strong password policy
An IAM identity that uses access keys to manage cloud resources via AWS CLI.	IAM User

Grant temporary access to your AWS resources.	IAM Role
Apply and easily manage the common access permissions to a large number of IAM users in AWS.	IAM Group
Grant the required permissions to access your Amazon S3 resources.	Bucket Policy User Policy
You must provide temporary AWS credentials for users who have authenticated via their social media logins as well as for guest users who do not require any authentication.	Amazon Cognito Identity Pool
A startup needs to evaluate the newly created IAM policies.	IAM Policy Simulator
A service that discovers, classifies, and protects sensitive data such as personally identifiable information (PII) or intellectual property.	Amazon Macie
A threat detection service that continuously monitors for malicious activity to protect your AWS account.	Amazon GuardDuty
Prevent unauthorized deletion of Amazon S3 objects.	Enable Multi-Factor Authentication (MFA)
A company needs to control the traffic going in and out of their VPC subnets.	Network Access Control List (NACL)
What acts as a virtual firewall in AWS that controls the traffic at the EC2 instance level?	Security Group
Set up an automated security assessment service to improve the security and compliance of your applications.	Amazon Inspector
Domain 3: Cloud Technology and Services	
A company needs to use the AWS global network to improve availability of deployed applications on AWS using an anycast static IP address.	AWS Global Accelerator
You need to securely transfer hundreds of petabytes of data into and out of the AWS Cloud.	AWS Snowball Edge

A type of an EC2 instance that allows you to use your existing server-bound software licenses.	Dedicated Host
A service that allows you to continuously monitor and log account activities such as the user actions made from the AWS Management Console and AWS SDKs.	AWS CloudTrail
A highly available and scalable cloud DNS web service in AWS.	Amazon Route 53
Store the results of I/O-intensive SQL database queries to improve the application performance.	Amazon ElastiCache
A combination of AWS services that allows you to serve the static files with lowest possible latency.	Amazon S3 Amazon CloudFront
Automatically scale the capacity of an AWS cloud resource based on the incoming traffic to improve availability and reduce failures	AWS Auto Scaling
A company needs to migrate an on-premises MySQL database to Amazon RDS.	AWS Database Migration Service (AWS DMS)
Automatically transfer your infrequently accessed data in your S3 bucket to a more cost-effective storage class.	S3 Lifecycle Policy
You need to upload a single object as a set of parts to improve throughput and have a quicker recovery from any network issues.	Use Multipart Upload API
A company needs to establish a dedicated connection between their on-premises network and their AWS VPC.	AWS Direct Connect
A Machine Learning service that allows you to add a visual analysis feature to your applications.	Amazon Rekognition
A source control service that allows you to host Git-based repositories.	AWS CodeCommit
A service that can trace user requests in your application.	AWS X-Ray

A company needs to retrieve the instance ID, public keys, and public IP address of their EC2 instance.	Instance metadata
You need to speed up the content delivery of static assets to your customers around the globe	Amazon CloudFront
Create and deploy infrastructure-as-code templates	AWS CloudFormation
You have to encrypt the log data that is stored and managed by AWS CloudTrail.	AWS Key Management Service (AWS KMS)
A database service that can be used to store JSON documents.	Amazon DynamoDB
Domain 4: Billing, Pricing and Support	
A designated technical point of contact that will maintain an operationally healthy AWS environment.	Technical Account Manager (TAM)
A tool that inspects your AWS environment and makes recommendations that follows AWS best practices.	AWS Trusted Advisor
A startup needs to estimate the costs of moving their application to AWS.	AWS Pricing Calculator
Set coverage targets and receive alerts when your utilization drops.	AWS Budgets
A type of Reserved Instance that allows you to change its instance family, instance type, platform, scope, or tenancy.	Convertible RI
Take advantage of unused EC2 capacity in the AWS Cloud and provides up to 90% discount.	Spot Instance
You need to centrally manage policies and consolidate billing across multiple AWS accounts.	AWS Organizations
The most cost-efficient storage option for retaining database backups that allows occasional data retrieval in minutes.	Amazon Glacier

Forecast future costs and usage of your AWS resources based on your past consumption.	AWS Cost Explorer
Categorize and track AWS costs on a detailed level.	Cost allocation tags
A company launched a new VPC which is way beyond the default service limit.	Request a service limit increase in AWS Support Center
The most cost-effective option when you purchase a Reserved Instance for a 1-year term.	All Upfront
You have to combine usage volume discounts of your multiple AWS accounts.	Consolidated Billing
Sell your catalog of custom AMIs in AWS	AWS Marketplace

Validate Your Knowledge

When you are feeling confident with your review, it is best to validate your knowledge through sample exams. **Tutorials Dojo** offers a very useful and well-reviewed set of practice tests for the Cloud Practitioner exam takers **here** as well as a **video course with included hands-on labs** to help you prepare well. 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. You can pair ou video course and practice exams with this study guide eBook.

If you have scored well on the **Tutorials Dojo AWS Certified Cloud Practitioner 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.

Sample Practice Test Questions:

Question 1

Which of the following is true on how AWS lessens the time to provision your IT resources?

1. It provides an AI-powered IT ticketing platform for fulfilling resource requests.
2. It provides various ways to programmatically provision IT resources.
3. It provides an automated system of requesting and fulfilling IT resources from third-party vendors.
4. It provides express service to deliver your servers to your data centers fast.

Correct Answer: 2

Cloud computing is the on-demand delivery of compute power, database, storage, applications, and other IT resources via the internet with pay-as-you-go pricing.

Whether you are using it to run applications that share photos to millions of mobile users or to support business critical operations, a cloud services platform provides rapid access to flexible and low cost IT resources. With cloud computing, you don't need to make large upfront investments in hardware and spend a lot of time on the heavy lifting of managing that hardware. Instead, you can provision exactly the right type and size of computing resources you need to power your newest idea or operate your IT department. You can access as many resources as you need, almost instantly, and only pay for what you use.

With Cloud Computing, you can stop spending money running and maintaining data centers. You can then focus on projects that differentiate your business, not the infrastructure. Cloud computing lets you focus on your own customers, rather than on the heavy lifting of racking, stacking, and powering servers.

With the cloud, businesses no longer need to plan for and procure servers and other IT infrastructure weeks or months in advance. Instead, they can instantly spin up hundreds or thousands of servers in minutes and deliver results faster. AWS provides you various ways and tools to programmatically provision IT resources such as AWS CLI, AWS API and the web-based AWS Management Console.

Hence, the correct answer is: **It provides various ways to programmatically provision IT resources.**

The option that says: **It provides an AI-powered IT ticketing platform for fulfilling resource requests** is incorrect because AWS doesn't have this kind of ticketing platform. What AWS actually does is it allows you to programmatically provision IT resources using AWS CLI, AWS API, and the web-based AWS Management Console.

The option that says: **It provides an automated system of requesting and fulfilling IT resources from third-party vendors** is incorrect because AWS primarily is the cloud vendor and it doesn't rely on third-party vendors to provision your resources.

The option that says: **It provides express service to deliver your servers to your data centers fast** is incorrect because AWS actually handles the underlying servers needed to run the cloud resources you requested. Remember that Cloud Computing is the on-demand delivery of compute power, database, storage, applications, and other IT resources via the Internet and not from your on-premises data centers.

References:
https://docs.aws.amazon.com/whitepapers/latest/aws-overview/six-advantages-of-cloud-computing.html
https://d1.awsstatic.com/whitepapers/aws-overview.pdf

Question 2

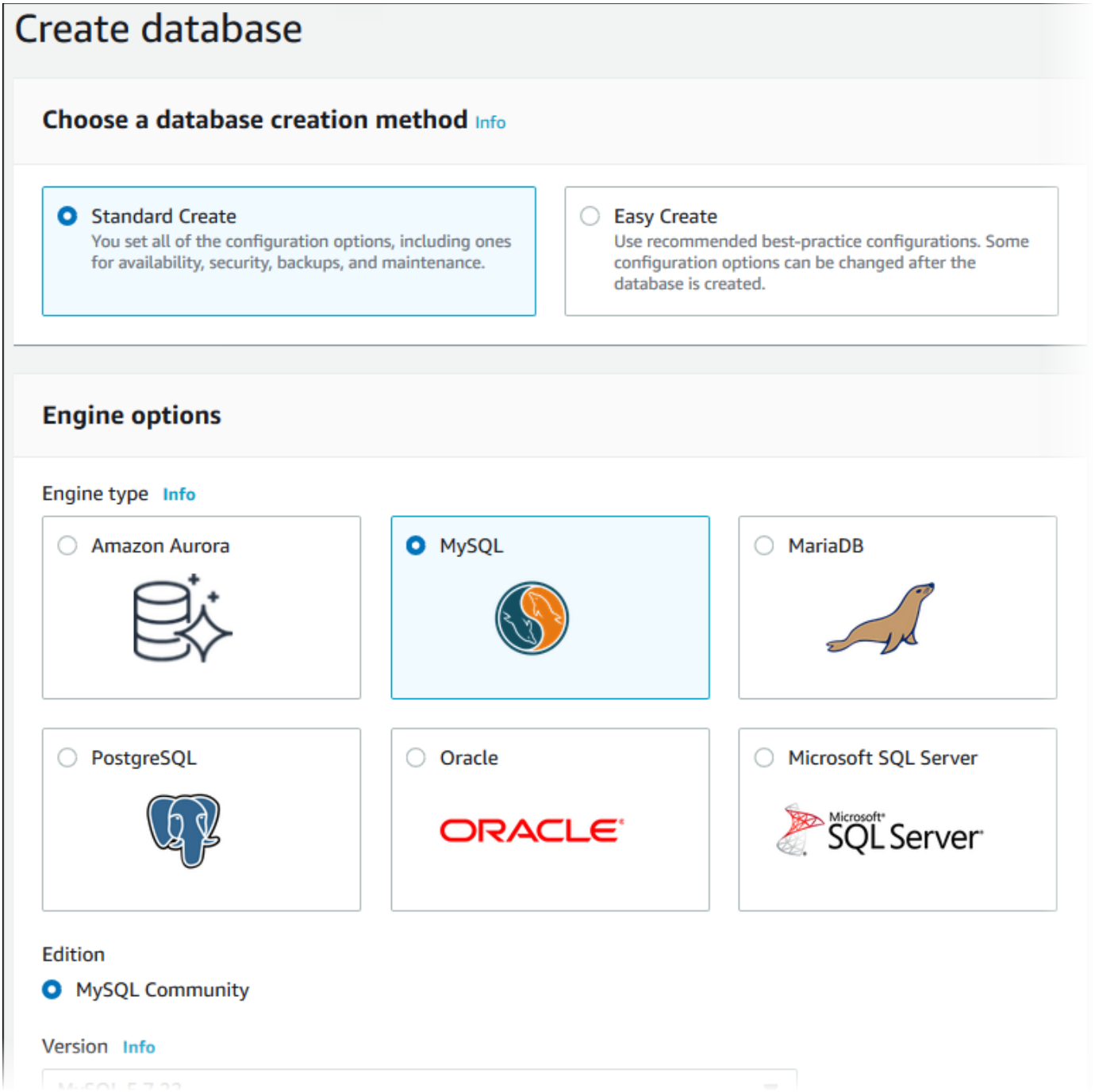
Which among the options below can you use to launch a new Amazon RDS database cluster to your VPC in a quick and easy manner? (Select TWO)

1. AWS Management Console
2. AWS Concierge
3. AWS CodePipeline
4. AWS CloudFormation
5. AWS Systems Manager

Correct Answers: 1,4

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

You can launch a new RDS database cluster using the AWS Management Console, AWS CLI, and AWS CloudFormation. The AWS Management Console provides a web-based way to administer AWS services. You can sign in to the console and create, list, and perform other tasks with AWS services for your account. These tasks might include starting and stopping Amazon EC2 instances and Amazon RDS databases, creating Amazon DynamoDB tables, creating IAM users, and so on. The AWS Command Line Interface (CLI), on the other hand, is a unified tool to manage your AWS services.



AWS CloudFormation provides a common language for you to describe and provision all the infrastructure resources in your cloud environment. CloudFormation allows you to use programming languages or a simple text file to model and provision, in an automated and secure manner, all the resources needed for your applications across all regions and accounts.

Hence, the correct answers are: **AWS Management Console** and **AWS CloudFormation**.

AWS Concierge is incorrect because this is actually a senior customer service agent who is assigned to your account when you subscribe to an Enterprise or qualified Reseller Support plan. This customer service agent is not authorized to launch an RDS cluster on your behalf.

AWS CodePipeline is incorrect because this is just a fully managed continuous delivery service that helps you automate your release pipelines for fast and reliable application and infrastructure updates.

AWS Systems Manager is incorrect because this is just a unified user interface so you can view operational data from multiple AWS services, and allows you to automate operational tasks across your AWS resources.

References:
https://docs.aws.amazon.com/IAM/latest/UserGuide/console.html
https://aws.amazon.com/cli/
https://aws.amazon.com/cloudformation/

Check out this AWS CloudFormation Cheat Sheet:
https://turon.tutorialsdojo.com/aws-cheat-sheet-aws-cloudformation/

Click **here** for more **AWS Certified Cloud Practitioner practice exam questions**.

What to expect from the exam

There are two types of questions on the examination:

- Multiple-choice: Has one correct response and three incorrect responses (distractors).
- Multiple-response: Has two or more correct responses out of five or more options.

Distractors, or incorrect answers, are response options that an examinee with incomplete knowledge or skill would likely choose. However, they are generally plausible responses that fit in the content area defined by the test objective.

Unanswered questions are scored as incorrect; there is no penalty for guessing.

Majority of questions are usually scenario based. Some will ask you to identify a specific service or concept. While others will ask you to select multiple responses that fit the given requirements. No matter the style of the question, as long as you understand what is being asked, then you will do fine.

Your examination may include unscored items that are placed on the test by AWS to gather statistical information. These items are not identified on the form and do not affect your score.

The AWS Certified Cloud Practitioner (CLF-C02) examination is a pass or fail exam. Your results for the examination are reported as a scaled score from 100 through 1000, with a minimum passing score of 700. Right after the exam, you will immediately know whether you passed or you failed. And in the succeeding business days, you should receive your complete results with the score breakdown (and hopefully the certificate too).

A few more tips:

1. Be sure to get proper sleep the night before, and don't be lazy in preparing for the exam. If you feel that you aren't ready enough, you can just reschedule your exam.
2. Come early to the exam venue so that you have time to handle mishaps if there are any.
3. Read the exam questions properly, but don't spend too much time on a question you don't know the answer to. You can always go back to it after you answer the rest.
4. Keep your reviewer if you plan on taking other AWS certifications in the future. It will be handy for sure.
5. And be sure to visit the Tutorials Dojo website to see our latest AWS reviewers, cheat sheets and other guides.

AWS BASICS

AWS Global infrastructure

Amazon Web Services provides the most extensive global footprint compared to any other cloud providers in the market, and it opens up new regions faster than others/. AWS maintains numerous geographic regions around the globe, from North America, South America, Europe, Asia Pacific, and the Middle East. AWS serves over a million active customers in more than 190 countries.

AWS is able to support this massive workload, thanks to its Global Cloud Infrastructure which consists of Availability Zones, Regions, and Edge Networks.

The AWS Global Cloud Infrastructure is the most secure, extensive, and reliable cloud platform in the industry today, which offers a wide range of cloud service offerings. AWS is the top choice of small and medium enterprises for deploying their application workloads across the globe and for distributing content closer to their end-users with low-latency. It provides you a highly available and fault-tolerant cloud infrastructure where and when you need it.

AWS owns and operates thousands of servers and networking devices that are running in various data centers, scattered around the globe. A data center is a physical facility that houses hundreds of computer systems, network devices, and storage appliances. You can run your applications in two or more data centers to achieve high availability; so if there is an outage in one of the data centers, you still have other servers running in another data center. A data center can also deliver cached content to your global end-users to improve response times.

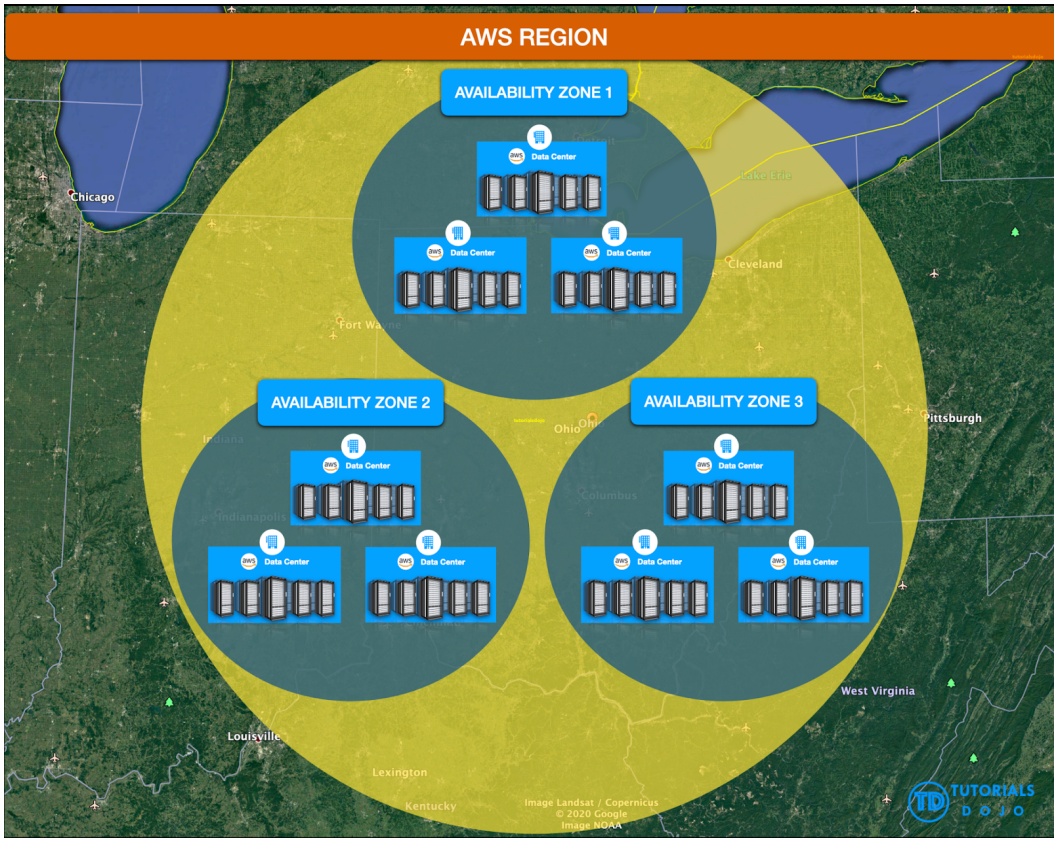
At its core, the AWS Global Infrastructure utilizes multiple data centers and group them into Availability Zones, Regions, and Edge Locations. Let's discuss these components one by one.

Availability Zone

An Availability Zone consists of one or more data centers, each with redundant power, networking, and connectivity. The data centers of a single Availability Zone, or "AZ" for short, are typically within 100 kilometers or 60 miles of each other. Think of it as a cluster of interconnected data centers in a specific geographic zone, that can help your applications become highly available – hence the name, Availability Zone.

AWS Region

An AWS Region consists of multiple Availability Zones. AWS has various regions available in North America, South America, Europe, Asia, and other parts of the globe. Since a single AZ consists of multiple data centers, your system can achieve a higher level of fault-tolerance by running it in two or more AZs. This enables companies to build highly available, fault-tolerant, and scalable cloud architecture instead of running their applications on a single datacenter.



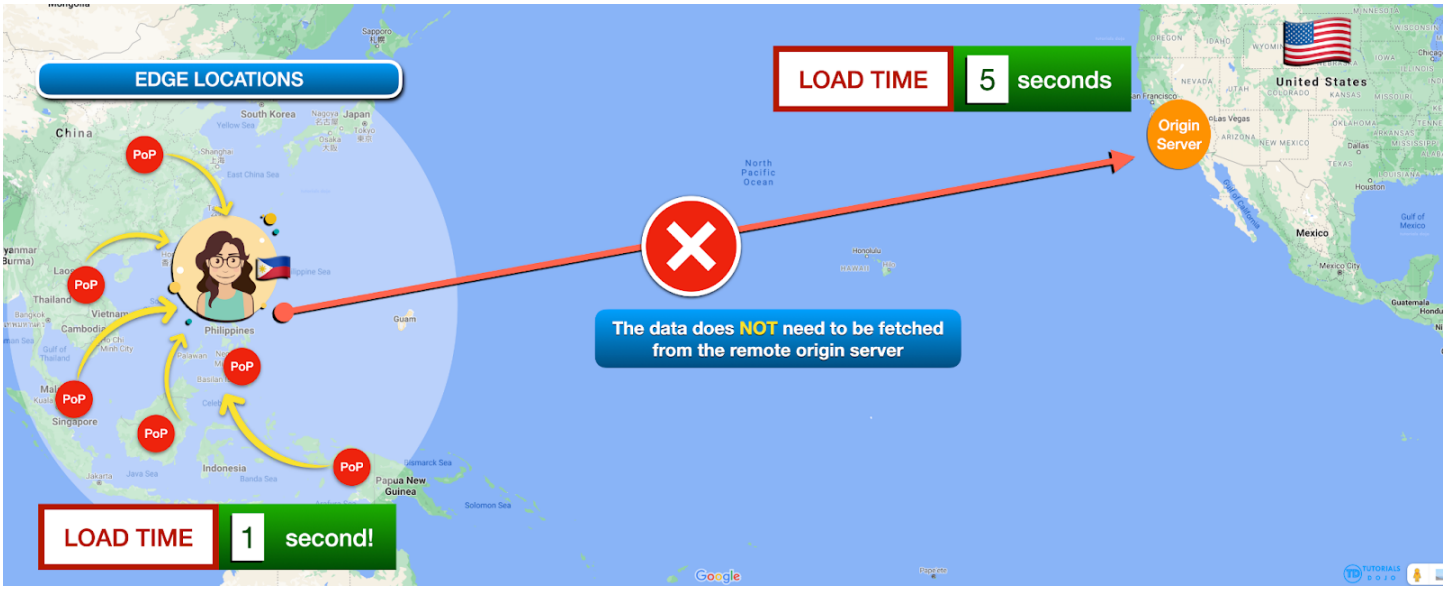
Remember that a single AZ consists of one or more data centers. Since you can deploy your application or your database to multiple AZs in a single region, your systems will still be running even if three or more data centers experienced an outage simultaneously.

To improve the durability of your data, you can also replicate it to two or more regions. This is helpful for disaster recovery and backups. The Availability Zones of a single AWS Region are typically within hundreds of kilometers or miles of each other. However, these AZs are still within a specific country to comply with the data sovereignty requirement. This is particularly useful if you have sensitive data that must only be stored in a certain location or country for data privacy compliance.

There is also a type of region called an AWS Local Region which is just a single data center designed to complement an existing AWS Region. An AWS Local Zone has less redundancy than a regular AWS Region since it is only composed of a single data center. The main purpose of having this localized region is to make the compute, storage, database, and other selected AWS services closer to a certain country or geographical location where there is no existing AWS Region.

Edge Locations

The other component of the AWS Global Cloud Infrastructure is the edge networks of Point-of-Presence or PoP. It consists of Edge Locations and Regional Edge Caches, which enables you to distribute your content with low-latency to your global users. Basically, a PoP serves as an access point that allows two different networks to communicate with each other. By using these global edge networks, a user request doesn't need to travel far back to your origin just to fetch data. The cached contents can quickly be retrieved from regional edge caches that are closer to your end-users. This is also referred to as a Content Delivery Network or CDN.



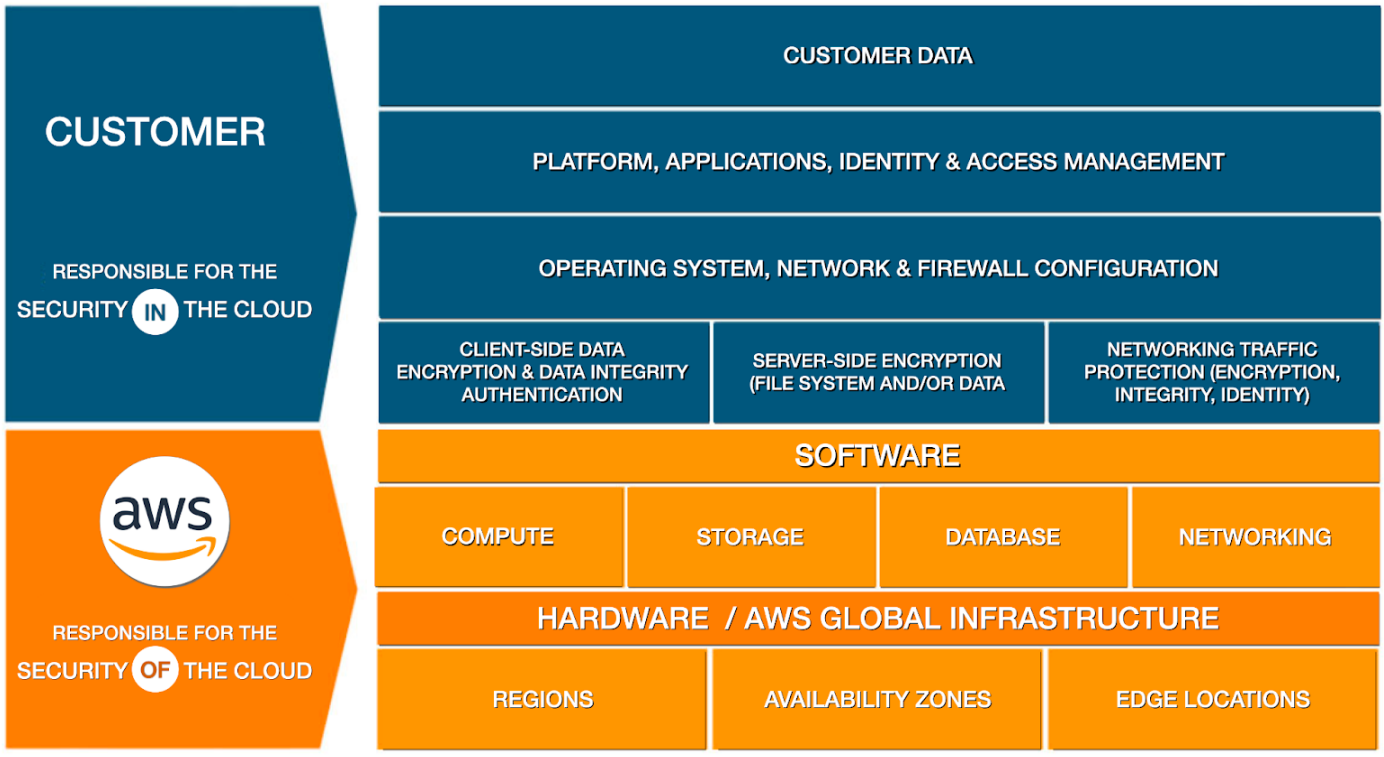
So for example, you have high-resolution images stored in a server in California. You can cache these media files to an edge location in the Philippines, India, or Singapore to allow your customers in Asia to retrieve these photos faster. The images will be loaded quickly because it is fetched to an edge server near your users, instead of retrieving it from its origin server in California.

AWS Shared Responsibility Model

Cloud computing is defined as a model for enabling ubiquitous, convenient, and on-demand network access to a shared pool of configurable computing resources. This shared pool of configurable computing resources is actually composed of thousands of high-powered physical rack servers that are scattered in multiple data centers all across the globe. Each of these servers has enterprise-grade processors that can generate hundreds or even thousands of virtual machines that can be used by thousands of customers.

A physical server is also called a host computer which runs a host operating system and a hypervisor. This hypervisor is responsible for generating multiple virtual machines; each with its own guest operating system that is chosen by the customer. So if you have a virtual machine right now in the US East Northern Virginia region, your server instance is actually running in one of the physical rack servers that are located in one of the many data centers within the state of Virginia.

Aside from virtual machines, customers can also use another type of computing resource called an abstracted service, which can be a ready-to-use database, storage, or messaging service in the cloud. These abstracted services are called as such because the cloud service provider abstracts, or removes away the responsibility of server maintenance, patching, and troubleshooting from the end customer.



If you are running your applications in the cloud, you should be aware of who is responsible for each and every component of your cloud solutions. There is a concept called Shared Responsibility Model in Amazon Web Services that defines the specific things that AWS is responsible for and the items for which the customer has full responsibility. The Shared Responsibility Model also covers a set of IT controls that are managed exclusively by AWS, by the customer, or by both parties. This ensures security and compliance in every building block of your cloud infrastructure.

Let's first step back and ask ourselves these questions:

- Who is responsible for patching the operating system of your Amazon EC2 instance?
- Who is responsible for applying the security patches of the guest operating system that your EC2 instance is using?
- Who is responsible for running the host operating system and the virtualization layer that powers your Amazon EC2 instances?
- Who is responsible for managing all your IAM user access and secret keys?
- Who is responsible for maintaining the underlying server of your AWS Lambda functions?
- Who is responsible for the Service and Communications Protection or Zone Security of your data?
- Who is responsible for the physical security of the servers and the entire network of data centers of the AWS Global Infrastructure?
- Who is responsible for designing encryption-at-rest strategies and other security features in your Amazon RDS database?
- Who is responsible for the security "of" the cloud and the security "in" the cloud?

The Shared Responsibility Model depicts how AWS and the customer share the responsibility of securing the physical infrastructure that powers the AWS cloud as well as the configuration management that protects the end-user data. AWS is responsible for the security "of" the cloud while the customer is responsible for the security "in" the cloud. The keywords here are "of" and "in" the cloud which looks the same at first glance but has a significant difference if you analyze it further. These prepositions describe the scope of responsibility of AWS and the customer.

Security "OF" the Cloud – The Responsibility of AWS

The phrase "Security of the Cloud" means that AWS is responsible for protecting the entire physical infrastructure that runs all of the available services offered in the AWS Cloud. We have discussed earlier that the "cloud" is actually a shared pool of configurable computing resources composed of thousands of high-powered physical rack servers and networking devices scattered in multiple data centers around the world. The AWS Cloud simply won't exist without these servers and data centers. As a cloud service provider, AWS is mainly responsible for the security "of" its global cloud infrastructure including all the hardware, software, networking, and physical facilities that run its various cloud services.

AWS owns hundreds of data centers across multiple countries and each data center hosts thousands of bare-metal servers that are linked together to form the AWS Cloud. A data center is a physical facility where a network of computers, storage systems, and computing infrastructure are hosted. Just like any other building, a data center needs physical security to protect the countless IT assets that are residing on its premises. Remember that the customer data actually exists in one of the storage volumes within a single data center or is distributed across multiple Availability Zones or AWS Regions. Millions of companies are hosting their mission-critical applications on these servers so a failure in one of the rack servers or data centers might cause production issues or data loss for the customer.

AWS has the obligation of maintaining the host operating system and the virtualization layer of its physical servers. This includes the task of applying the OS patches to both the host operating system and the hypervisor that instantiates and runs the Amazon EC2 virtual machines. Installing firmware updates on its computing and networking hardware is also in scope. Aside from implementing physical and environmental controls for its data centers, AWS also covers the activities that ensure the availability, reliability, and scalability of its cloud service.

Security "IN" the Cloud – The Responsibility of the Customer

The customer is responsible for the "Security in the Cloud" which basically means protecting its custom data that is processed and stored within the AWS Global Infrastructure. There are certain configuration and management tasks that the customer can do in order to encrypt and secure its data in the cloud. Customers can add security groups to their EC2 instances, set up network ACLs for their Amazon VPCs, enable data encryption, and use other readily-available security features that are provided by AWS to secure their sensitive data.

The level of customer responsibility is determined by the type of cloud services that they are actually using which can be an Infrastructure as a Service (IaaS) or an abstracted service. An example of this is Amazon EC2 which is an Infrastructure as a Service that provides a pay-as-you-go model for your computing and virtualization needs. Amazon EC2 provides a range of options that you can choose from to fully customize, configure, and secure your own computing resources.

Customers are responsible for updating and applying the security patches of the guest operating system that's being used by their Linux or Windows instances. Take note that AWS takes care of the host operating system of the physical host server that generates the virtual machines that come with a guest OS defined by the customer. The customer is also expected to set up the virtual firewall, security group, network ACL, and other security features on every EC2 instance it owns.

Client-side and server-side data encryption are also managed by the customer which can be achieved by enabling the encryption options available in Amazon EC2, Amazon EBS, Amazon FSx, and other services. Customers are also expected to handle the identity and access management of their EC2 fleet as well as the Service and Communications Protection, which is also known as Zone Security.

An Amazon EC2 instance typically resides within a single Available Zone only but certain route table configurations can be implemented in its Amazon VPC that may inadvertently allow unauthorized access to the customer data. This can be prevented by routing or zoning the data within the specific environments that you define using an AWS Network Firewall, Transit Gateway, and other networking services.

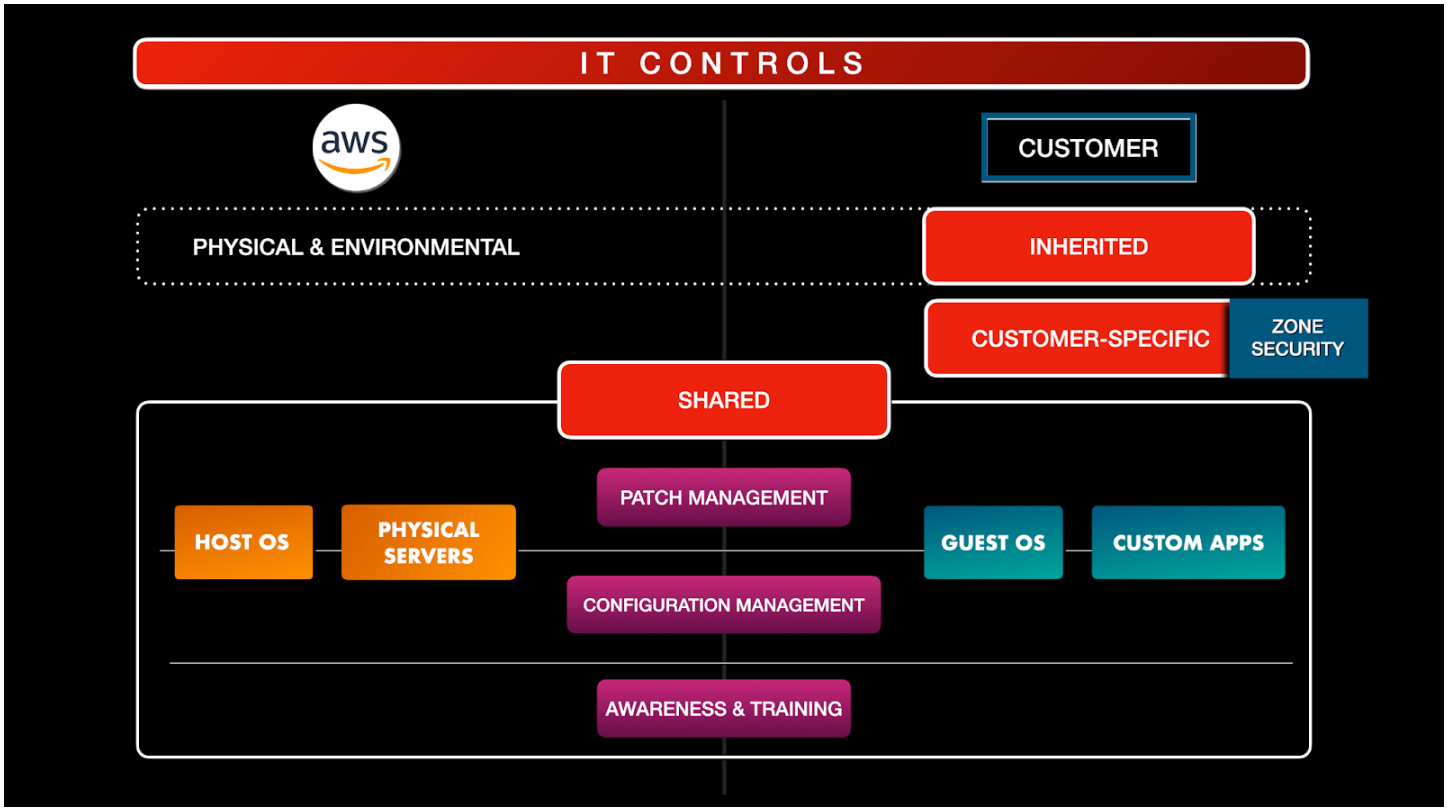
For abstracted services, AWS handles almost everything from the infrastructure layer, the operating system, the software, and the platforms including the external dependencies used by the service. Consider Amazon S3 and Amazon DynamoDB. Amazon S3 is an extremely scalable storage service in AWS that can be used immediately by the customer without launching its own virtual storage network. AWS is the one that provides all the required physical storage devices that allow the customer to just upload large amounts of data without worrying about server or storage limits. Most or all of the underlying layers that make up the service are abstracted from the customer's perspective.

The same is true for Amazon DynamoDB. This NoSQL database can store millions of records without burdening the customer to provision more storage capacity. Abstracted services are also called Platform-as-a-Service or Software-as-a-Service depending on their type. Infrastructure security is already provided by these services but customers can further secure their data by using encryption features, configuring endpoints, and crafting IAM policies to allow fine-grained permissions to meet their compliance requirements.

IT Controls

This shared responsibility model of AWS and its customers also extends to IT controls. The management, operation, and verification of IT controls may be shared by both parties. There are inherited controls, shared controls, and customer-specific controls that must be taken into account too. For inherited IT controls, the customer fully inherits certain items from AWS such as the physical and environmental controls of the data centers and their related assets.

Shared controls apply to both the infrastructure and customer layers where AWS and the customer work together to properly manage all the facets of the cloud infrastructure. AWS provides the core infrastructure while the customer can implement their own set of controls within their use of AWS services. Examples of shared controls are patch management, configuration management, as well Awareness & Training.



For patch management, AWS handles the patching of the host OS and troubleshooting various issues within the AWS infrastructure, while customers are in charge of patching their guest OS and securing their enterprise applications.

For configuration management, AWS maintains the configuration of its infrastructure devices and physical servers but the customer is still responsible for configuring their own guest operating systems, databases, and custom applications. For Awareness & Training, both parties must train their respective employees and ensure that their staff is aware of the shared responsibility on both sides.

Customer-specific controls are the tasks that are solely the responsibility of the customer based on the applications and systems they are running within the AWS cloud. This covers the aspect of Zone Security where customers can manually modify certain routes to their resources or filter the traffic to better control access to its cloud resources and data.

AWS vs Customer Responsibility Examples

Now that we have discussed the concept of the AWS Shared Responsibility Model, you might find it easier now to answer the set of questions that were asked to you at the beginning.

Who is responsible for patching the operating system of your Amazon EC2 instance?

- This question needs more clarification since an operating system in AWS can be the host OS or a guest OS. If the question is about the host operating system, then the entity responsible for patching is AWS and if it is the latter, then it is the customer's obligation.

Who is responsible for applying the security patches of the guest operating system that your EC2 instance is using?

- This task clearly falls on the shoulders of the customer since we are talking about guest operating systems.

Who is responsible for running the host operating system and the virtualization layer that powers your Amazon EC2 instances?

- In this case, the one responsible is AWS because it mentions the host OS and the virtualization layer, which is also known as the hypervisor or the virtual machine monitor.

Who is responsible for managing all your IAM user access and secret keys?

- For this question, IAM refers to Identity and Access Management which is part of the security in the cloud. Managing the identity and access of the cloud resources is within the scope of what the customer manages.

Who is responsible for maintaining the underlying server of your AWS Lambda functions?

- The answer here is AWS since it is the cloud service provider that provides the underlying physical servers that power the cloud such as the AWS Lambda serverless service. AWS Lambda is a type of an abstracted service that is considered to be a Function as a Service that provides computing capabilities for a short amount of time. The term "serverless" simply means that the customer has "less" server management responsibilities as these tasks are managed by AWS.

Who is responsible for the Service and Communications Protection or Zone Security of your data?

- We have discussed the concept of Zone Security in this eBook and by now, you should know that this is one of the IT controls that is handled exclusively by the customer. Again, Zone Security is the concept of zoning your data within a particular network space that you define to prevent unauthorized access to sensitive data.

Who is responsible for the physical security of the servers and the entire network of data centers of the AWS Global Infrastructure?

- AWS is responsible for all the hardware and data centers that power its global cloud infrastructure.

Who is responsible for designing encryption-at-rest strategies and other security features in your Amazon RDS database?

- You have a key phrase "designing encryption at rest strategies" for Amazon RDS. The customer is responsible for client-side and server-side data encryption but this is usually done by enabling the security options available in AWS. Configuration management is primarily implemented by the customer through the readily-available features in AWS and not by manually designing security strategies yourself. Therefore, the answer here should be AWS since they have a team of developers and engineers that design and add data encryption strategies for Amazon RDS and for other services.

Who is responsible for the security "of" the cloud? Who takes care of the security "in" the cloud?

- AWS is responsible for the security of the cloud while the customer is the one responsible for the security in the cloud. AWS is in charge of protecting the entire physical infrastructure that runs all of the available services offered in the AWS Cloud. On the other hand, the customer has the duty of protecting its custom data that is processed and stored within the AWS Global Infrastructure.

The Advantages of Cloud Computing

The advent of cloud computing ushered in a new era for companies to design, manage, and operate their IT resources in a more scalable, easier, and cost-effective way. Companies, both large and small, can easily launch their online solutions and computing resources in a matter of minutes instead of days, weeks, or months. There's no need to buy their own physical servers or manage their traditional on-premises data center in order to run their applications.

With just a click of a button, customers can have on-demand access to a wide range of virtual machines, storage services, databases, and other IT resources that allow enterprises to run their production workloads and even offer their services globally. Both the upfront capital expenditures and monthly operating expenses of the company can be significantly lowered due to its revolutionary cloud economics that gives its customers an affordable pay-as-you-go pricing option.

The cloud also provides unparalleled flexibility to your server provisioning process, resource management, serverless computing, automation, software development lifecycle, and many other facets of your enterprise IT infrastructure. It allows you to quickly launch new resources automatically to better serve your customers when the demand is high and, conversely, decommission unnecessary IT assets that are not in use anymore to save on costs. Cloud Service Providers offer a plethora of computing options that provide a better price-to-performance ratio than manually running and maintaining bare-metal servers on-premises.

Companies right now can launch their entire enterprise systems and other dependencies to the cloud in a few minutes, which usually took days or weeks in the past – way back when cloud computing was not available. The cloud provides a lower total cost of ownership compared with running an infrastructure environment on-premises or in a co-location facility. These levels of flexibility, scalability, and cost-benefit were practically non-existent before, which is why millions of companies from all over the globe are moving their local on-site systems to the cloud to avail themselves of all its many benefits.

There are a lot of advantages to using cloud computing over a traditional on-premises data center. With Cloud Computing, you can:

- Trade fixed expense for variable expense
- Benefit from massive economies of scale
- Stop guessing capacity
- Increase speed and agility
- Stop spending money running and maintaining data centers
- Go global in minutes

These are the major advantages that companies can enjoy if they leverage on the power of cloud computing to run their IT solutions.

Trading 'fixed expense' for 'variable expense' simply means that you will gain greater flexibility in paying for your IT resources. The traditional on-premises environment has fixed expenses with high upfront costs, while the cloud has variable monthly expenses that require lower upfront fees.

The benefit from massive economies of scale is actually based on the microeconomic concept, where goods are produced on a larger scale with fewer input costs. This concept is known as the Economies of Scale and depends on how massive the entity is. Companies with highly efficient large-scale production can sell their goods and services way cheaper than businesses with small-scale operations. The same is true in cloud computing, where the cost of a virtual server becomes affordable since the operating cost can be distributed to a larger customer base.

The ability to stop the guessing capacity for your applications is a unique trait of the cloud. There is a selection of helpful features in cloud computing that will help you stop guessing capacity entirely. You can accurately match the real usage patterns of your systems to your computing capacity and not just provide a mere estimate.

The cloud's global infrastructure can provide an increase in speed and agility for developing new solutions. Launching your servers, storage services, network devices, and other resources is a hundred times faster in the cloud as opposed to doing them on your own on-premises data center. This allows you to scale your business at an accelerated pace!

Next, you can stop spending money running and maintaining data centers because the cloud service providers will handle this for you. Running a data center entails a large sum of capital, herculean effort, and technical expertise in order for your systems to run as smoothly as possible. By offloading these tasks to a cloud service provider, you can focus more on building enterprise applications that generate cash flow for your organization. You don't need to spend money to operate a fleet of servers or several on-site facilities to run your systems globally.

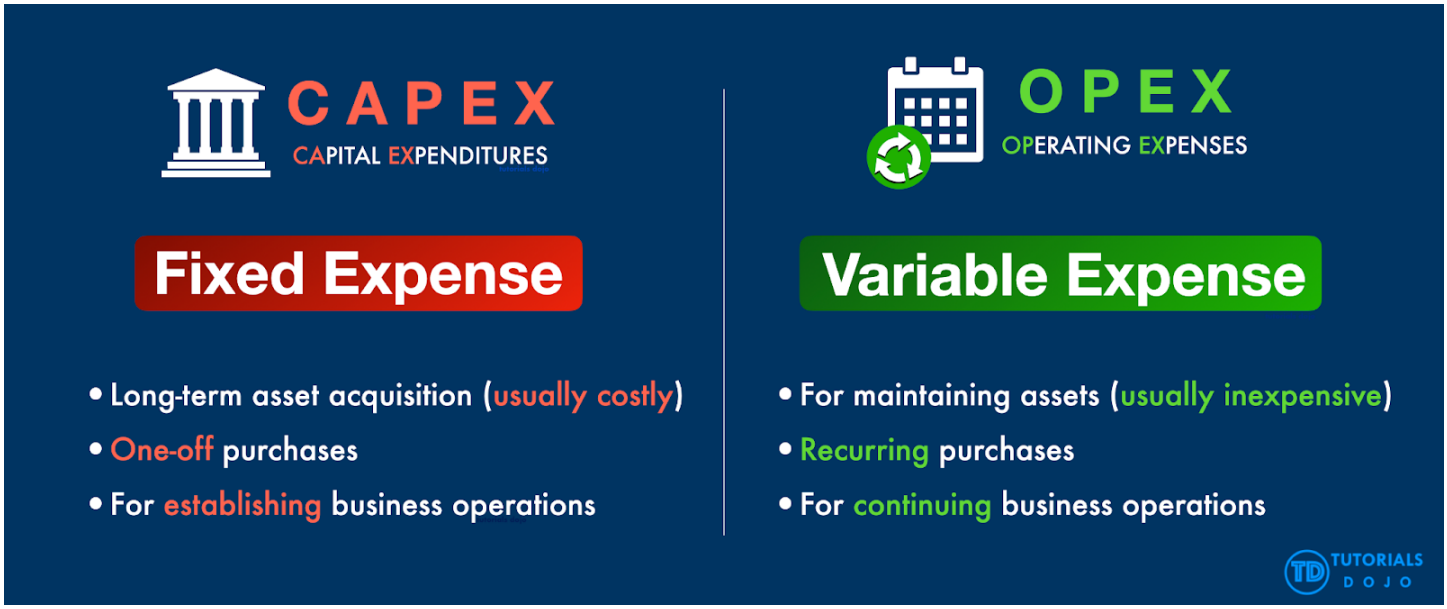
Lastly, going global in minutes is basically the capability to serve more customers by launching your services in more geographic regions and not only on your base country. All Cloud Service Providers have a global infrastructure spanning hundreds of countries and regions. It only takes several clicks to provision and control thousands of resources worldwide which allows you to serve countless number of people and not just in your own locality.

These are the major advantages of cloud computing. The meaning of each of these items is somehow straightforward, but there's still a wealth of related information that you must know to completely understand the positive tailwinds of using cloud computing over the traditional model.

Let's discuss these advantages one by one in this section.

Trade Fixed Expense for Variable Expense

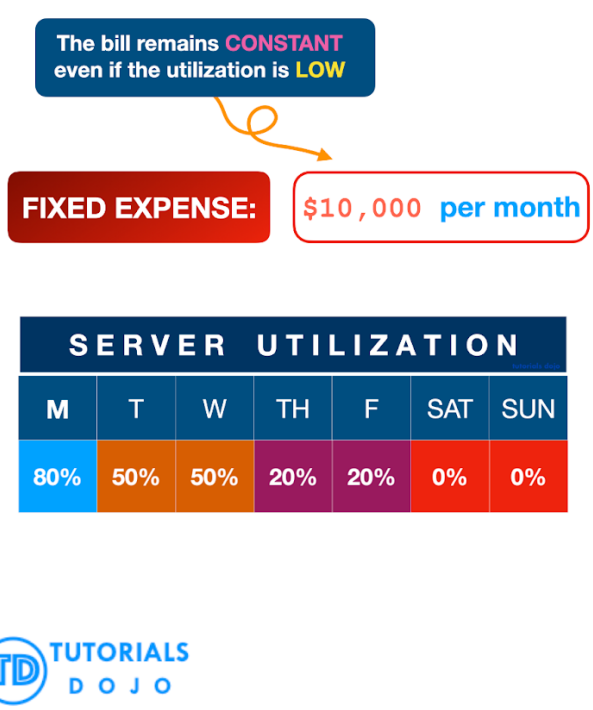
Trading "fixed expense" for "variable expense" simply means that you will gain more flexibility when it comes to paying your IT resources. There are two types of expenses that you have to consider here – you have the fixed expense and the variable expense. These two relate to both the CAPEX and OPEX of the company. Capital expenditures, or CAPEX for short, refers to the long-term asset acquisition or one-off purchases incurred by the company to establish its business operations. Operating expenses, or OPEX, on the other hand, are the recurring payables for running the resources necessary to operate the business.

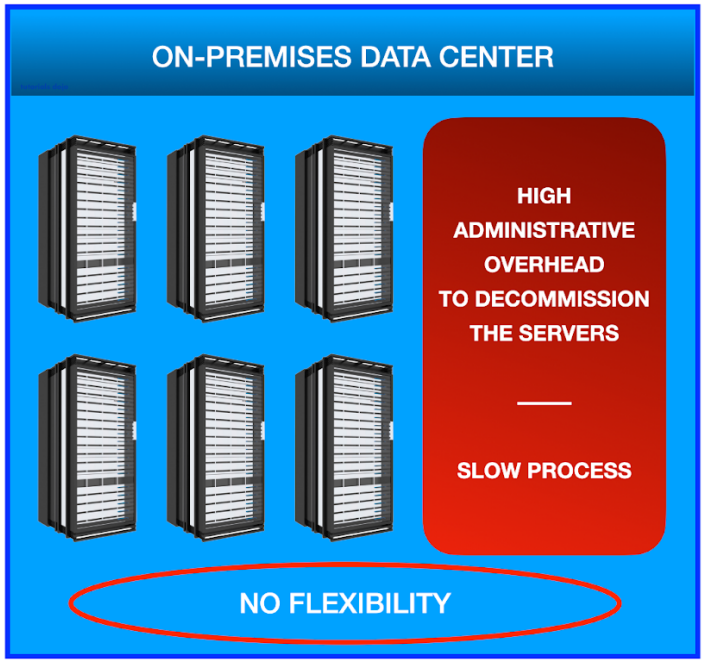


Running a physical data center has a set of fixed expenses that are constant or don't vary over time. Most companies spend large upfront investments to buy the necessary hardware for their data centers, such as rack servers for computing, routers for networking, hardware security modules for encryption, HDD and SSD drives for storage, as well as the physical facilities where all their IT assets are located. If you don't need your bare-metal servers anymore, you cannot just discard them right away since you have already paid a substantial amount of money for them in the first place.

There is also a risk of not fully utilizing all your local computing resources. For example, you have 10 servers that run your enterprise web application round-the-clock. All these 10 servers fully use 100% of their CPU power during peak hours of the work week. However, these servers are not used at all on weekends, which indicates that the 10 servers are just idle throughout the non-working days and on holidays. This implies that companies are paying fixed fees for computing hours that wildly vary depending on the demand and incoming traffic. Organizations pay the exact same amount of bills even if they only utilize 80%, 50%, or 20% of their total computing capacity. Worse, they still have to pay for the electricity, staff, rent, supplies, and everything else in their data center, even if they are not using the servers at all!

Many customers are also constrained by the limit of the computing capabilities of their existing servers and data centers. If your application receives a surge of incoming traffic, you only have a maximum of 10 servers to do all of the processing, based on our earlier example. Adding more servers to your data center entails a lot of money, effort, and management overhead to implement. Similarly, the process of decommissioning unused servers and resources entails the same amount of extensive labor. The bottom line is that running a traditional data center is rigid and lacks the agility that is necessary to cope quickly with the ever-changing business trends.





In cloud computing, you will have a wide selection of services to choose from with variable expenses rather than fixed expenses that you have to pay upfront. The cloud service provider is the one that runs all the physical data centers and other assets, so the customers won't have to. This effectively lowers the CAPEX, or the funds necessary to start the business operations.

Your monthly expenses will also vary greatly based on how much CPU capacity or resources you actually consume. You are not stuck using a fixed number of servers to run your applications. If the demand goes up, the cloud can automatically provision more servers without any manual effort on your part; and if the demand dwindles down, the cloud will reduce the number of servers through auto-scaling. You pay a higher bill for running more servers to scale out your computing capacity; however, that can be offset with a lower bill during non-peak hours since you will be running servers in less quantity. This is what your company's OPEX will look like in the cloud, as the variations in your application usage patterns affect your monthly bill.

You can even go serverless to lessen or totally free yourself from the cumbersome tasks of managing your application servers. The cloud allows you to pay your computing resources by the hour, by the second, or even by the number of milliseconds that your function or application actually spent for processing. So if your serverless app runs for just 10 seconds for the entire month, you will only have to pay for that 10 seconds! Wow! This variable expense truly provides customers with a game-changing experience and greater flexibility in paying for their IT resources which were not available before in the traditional model.

That is how the cloud allows you to trade "fixed expense" for "variable expense." It removes the barrier for small, medium, and even large companies in launching their solutions since the computing capabilities are available to them at their fingertips. No facility, data center, server, router, or any physical asset is needed!

Benefit from Massive Economies of Scale

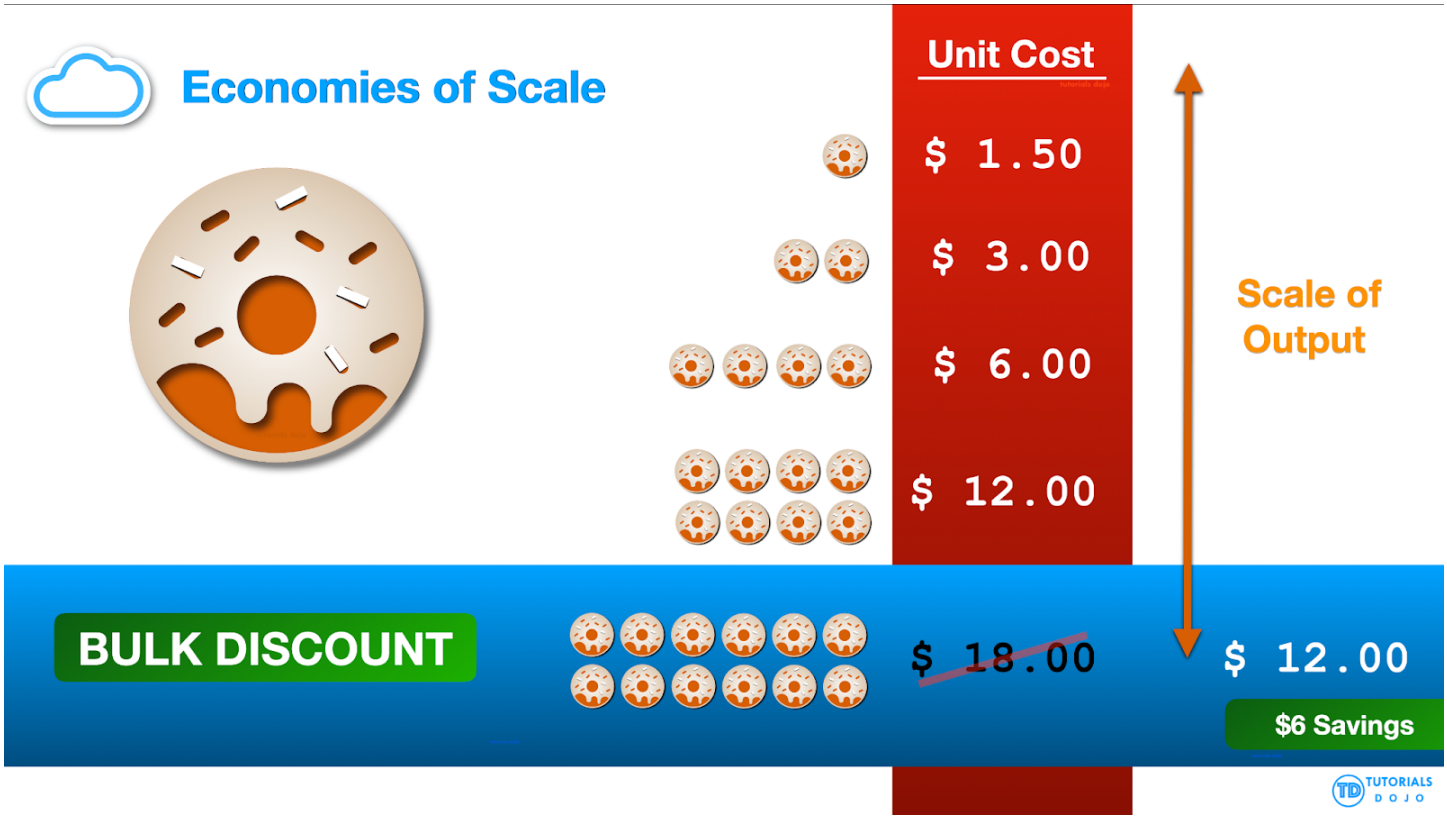
The public cloud is quite expansive, and it is exponentially growing each and every day! It has a global Infrastructure that spans various countries around the world and consists of millions of servers and IT resources. It is a network of physical data centers, colocation facilities, point-of-presence or PoP locations, and other components which collectively powers the cloud.

Each data center may have hundreds of thousands of rack servers, storage appliances, networking devices, and other digital assets. These data centers are grouped together to form an Availability Zone, and in turn, these multiple Availability Zones are clustered together within 100 kilometers or 60 miles of each other to make up a cloud Region. Each Region is a separate geographic area that covers one or more cities or provinces within a specific country only, which has its own unique data sovereignty requirements. Cloud Service Providers have multiple regions in many countries interconnected through an extensive and ever-growing global network infrastructure that's used by millions of customers.

There's no denying that the cloud is absolutely massive! This is where the benefit of massive economies of scale comes in. The term "economies of scale" is a microeconomic concept that describes a state where the unit cost decrease with the increase in the scale of the output being produced by a company. In other words, the products are sold cheaper since the company's production capacity is bigger.

More units of goods or services are produced on a larger scale with fewer input costs. Companies that have highly efficient large-scale production can sell their products cheaper to more customers than the companies with small-scale operations.

Think of it this way. Suppose you're buying a piece of a soft yummy donut with a creamy custard filling inside. It costs a dollar and fifty cents for each piece, so if you buy 2, you have to pay 3 dollars; buy 4 donuts, pay 6 dollars; buy 8 donuts, pay 12 dollars; buy a dozen donuts, pay 18 dollars, and so on and so forth.



As you increase the number of donuts you purchase, there's a good chance that the store can give you a bulk discount for getting more. The store can reduce the price of each donut to just one dollar if you buy a dozen; thus, you only pay 12 dollars instead of 18 dollars. That's a discount of 6 dollars since you are buying in large quantities.

This is the same logic in wholesale stores like Costco, Walmart, Target, and others. You usually get low warehouse prices at Costco for buying your groceries in bulk, and also because these products are directly from the manufacturer. A cost-efficient supply chain allows these wholesale companies to sell their products way cheaper than selling them individually via retail.

Similarly, a cloud service provider can sell or lend its computing resources to thousands of customers for less. The cloud's massive surplus of IT assets and affordable pricing options drive the costs lower as the expenses are spread over a larger number of goods.

This is the unique, strong point of cloud computing over a traditional on-premises data center. It has a significant cost-benefit brought by its massive economies of scale that is impossible to achieve on a small localized facility with far fewer resources than the cloud.

Stop Guessing Capacity

Another advantage of the cloud is having more control over your computing capacity. You can stop guessing how much capacity you really need if you are using the cloud. This is because cloud computing has a pay-as-you-go pricing option where you only pay for the computing time that you actually consume. The virtual servers are billed by the hour or by the second, and you even stop incurring server costs by terminating your computing resources.

Historically, IT firms have to guess the capacity or come up with the best estimate on how big or small their computing, storage, and network resources will be. Organizations are often constrained by the limited capability of their physical infrastructure.

Suppose a company plans to launch a new web application that will accept requests from its 100,000 customers worldwide. Most companies will conduct load testing first to get an idea of how much CPU, RAM, and network bandwidth they'll provision for their brand-new servers. However, this is still a rough estimate, and the actual computing power they need might be way lower or much higher than their initial baseline. So going back to our example, the number of active customers can exceed a hundred thousand users over time which will cause the demand to spike up. Conversely, this particular number may slump, causing underutilization of your servers.

Conducting a capacity plan prior to deploying an application often ends up either having underutilized resources or overutilized servers. It's too difficult to right-size your resources since your physical IT inventory is limited, and the needed parts to upgrade your capacity must be purchased and installed separately. These problems are a thing of the past if you leverage the competitive edge of the cloud. Customers can access as much or as little capacity as they need to better serve their customers.

Using the cloud enables companies to stop guessing the capacity for their production workloads. You'll be able to scale up or down at a moment's notice to match the real usage patterns of your systems.

Cloud computing offers different ways to match the computing needs of your system in near real-time. You can either scale horizontally or vertically, allowing you to scale up, scale down, scale-out, and scale in your computing capacity. The process of vertical scaling involves scaling up or scaling down the size of one existing machine, and in contrast, horizontal scaling is the process of scaling out or scaling in the number of machines for your application. Doing horizontal or vertical scaling in the cloud is easy and can be done automatically, as opposed to doing it in a physical data center which entails a lot of manual effort, resources, and time to do so.

For instance, one virtual machine with 2 CPU cores and 2 gigabytes of RAM can be vertically scaled up to 4 CPU cores and 4 gigabytes of RAM. This will allow your application to serve more requests during peak hours. If the demand for your application decreases, you can scale down the server by trimming the CPU to 1 core with 1 gigabyte of RAM only to avoid underutilization. These scaling activities can be done in a few minutes in the cloud rather than in days or weeks in a local data center. The process of scaling up and scaling down your cloud server does not require you to manually add or remove CPU, RAM, or storage drives in your machine, which usually takes a long while to do so on-premises.

The same is true of horizontal scaling, which allows the number of your servers to scale out or scale in automatically. Say you are using a total of 10 distributed servers to run a single application. You can scale out the total number of computing resources from 10 servers to 20 servers or more via Auto Scaling. This is beneficial during peak hours of the work week when most of your customers are using your application. You can reduce or 'scale in' the number of servers on weekends or at night when fewer people are accessing your systems. So from a total of 20 servers during the day, your application will run on just 2 servers at night time to save on costs.

Furthermore, you can also adopt a serverless architecture in the cloud to free you from the task of scaling and maintaining your own servers. A serverless application can be in the form of a simple function or a container that only runs when invoked. If your application has no request at all, then your application will not consume any computing resources. It will only run when there's an active request, and this is the only time you'll get billed. Serverless services can also handle millions of requests worldwide by automatically scaling their internal resources without any manual intervention on your part.

These are the helpful features of cloud computing that will help you stop guessing capacity entirely. Your computing capacity accurately matches the real usage patterns of your systems and is not just a mere estimate, which is prone to error.

Increase Speed and Agility

The next advantage of cloud computing is the increase in speed and agility of your overall IT capability. This allows your company to do your regular tasks faster and be more flexible for any change that might arise during your product development and business operations.

Basically, the speed of launching your servers, storage services, network devices, and other resources is a hundred times faster in the cloud as opposed to doing them on your own on-premises network. Automatic scaling can be immediately achieved, which further hastens the momentum of your product delivery, system upgrades, and expansion. The task of manually purchasing, installing, and configuring the required hardware and equipment is also eliminated, allowing you to scale your business at an accelerated pace.

You can deploy an entire solution with just a click of a button and expedite your software development process by reducing the time needed to make those test environments available to your developers. This fast rate of deployment is also applicable if it's time to unveil your applications in production. You don't have to wait for your manufacturer or retail partner to launch a new set of servers, storage drives, and network connections, among others, since you can do it all by yourself in the cloud.

It also increases the agility of your organization since you have a myriad of services that you can use at your disposal. Your development team will be more agile as you can easily shift to a totally new implementation if you stumbled upon a blocker in your current workflow. The associated cost to experiment, develop and test different solutions is significantly lower, which drives innovation and digital transformation within your organization.

The cloud provides a variety of ways to accomplish tasks and workloads in your enterprise. You are not constrained by the limited physical assets of your organization. With hundreds of services to choose from, you can easily launch your systems on a virtual machine, a container, a serverless application, or in any platform that's right for you.

In addition, you can leverage the automation tools and features that are available in the cloud. Automating a time-consuming manual task will certainly speed up your operations. You will gain more free time to explore various solutions since the cloud service provider handles all the boring maintenance tasks for you. This makes your team or department truly agile, as you have more control over your activities than working in an on-site data center.

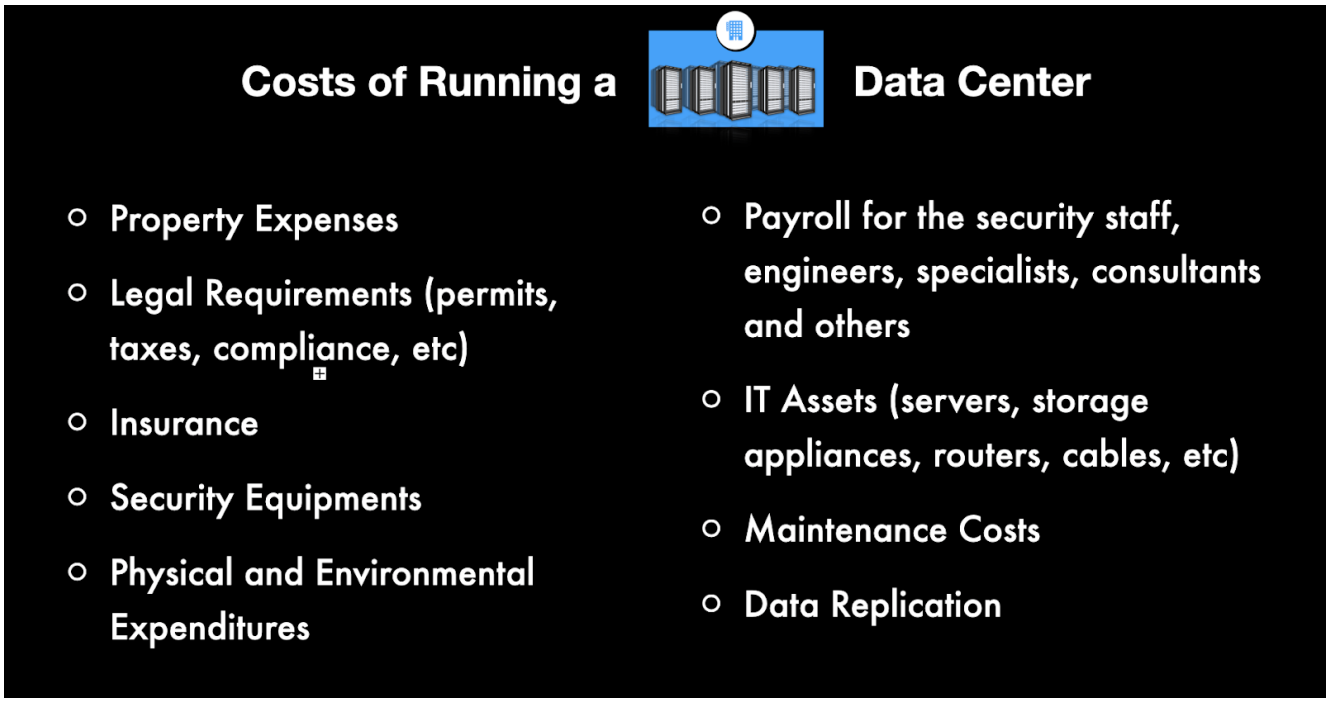
Cloud computing indeed increases the speed and agility of your entire IT infrastructure.

Stop Spending money running and maintaining Data Centers

Another advantage of cloud computing is the option to stop spending money running and maintaining your own data centers. As we have discussed earlier, running a data center entails a lot of capital, effort, and expertise for your systems to run as smoothly as possible. Most people nowadays haven't seen, heard nor even visited an actual data center at all so it's understandable that a majority of people don't understand how expensive it is to launch, run, and maintain an on-site location.

A data center is a physical facility or a building where all your IT resources are located. This is a real place that you can actually see and visit. In the state of Virginia, for example, you will spot numerous data centers scattered in Ashburn, Sterling, Manassas, and other cities or counties operated by various companies. These data centers may operate as an Internet Exchange point, a Colocation facility, a disaster recovery and business continuity site, or as one of the cloud nodes for a global cloud network. Again, these are actual locations with physical addresses that companies pay for.

Just like your house or an apartment, you have to pay rent on a monthly basis or take out a property loan before you can move in all your stuff on-premises. You have to secure all the legal requirements before starting your operation, such as business permits, property taxes, compliance requirements, environmental conformance, amortization, workplace health and safety standards, et cetera. Companies also have to consider natural disasters and damages to their property, compelling them to procure various types of insurance to safeguard their investment.



Take note that this is just one of the layers of a data center that covers the actual perimeters of the physical location. Additional costs are still needed to place the required CCTV cameras, security guards, intrusion detection technology, and other security measures. Power outages, flood mitigation, and a slew of external factors must be considered as well.

The large upfront investment required to build a dedicated data center is too costly and exorbitant for small companies, which is why bigger corporations offer shared facilities so that one location can be shared by more customers. In cloud computing, this is referred to as a colocation facility that helps companies to lower costs.

Another component of a data center is called the data layer, which usually contains sensitive customer information. You have to hire a staff of security personnel, computer engineers, hardware specialists, tech professionals, systems managers, and other consultants just to get started. Hundreds of physical host servers, storage appliances, routers, cables, and certain pieces of IT equipment must be purchased and configured to run your own virtualization layer. Maintenance tasks such as firmware updates, installing OS patches, troubleshooting, and other responsibilities also fall on your shoulder. If one piece of equipment fails, your team must troubleshoot and fix the problem, even if it happens during the weekends or in the wee hours of the morning.

Ensuring the availability and reliability of your systems is one critical factor too. You must replicate your dataset to at least two other data centers to improve the availability of your data in the event of outages or natural disasters. Replicating data to multiple locations can double or triple your operating costs since you will have to maintain, secure, and hire more staff to fulfill this need.

We have discussed the various expenses associated with operating a data center – both the upfront investment required and the monthly operating costs to keep it running. This is the reason why cloud computing is gaining traction around the globe, as it removes the large upfront investment required in running a data center. This allows startups and companies to start their business operations faster than ever before. You can stop spending money running and maintaining data centers if you choose the cloud as an integral part of your infrastructure.

Go Global in Minutes

The benefit of going global in minutes can be realized by leveraging the power of the cloud. This advantage is too difficult to achieve if you're just maintaining your own set of data centers in one or multiple locations. The geographic area that a small company can cover is dwarfed by the cloud's expansive global network.

All major Cloud Service Providers have a global infrastructure footprint that spans hundreds of countries and regions. It only takes several clicks to provision and control thousands of resources worldwide which allows you to serve countless number of people and not just in your own locality.

One excellent example is by using a Content Delivery Network service or CDN for your application. A CDN consists of a global network of point-of-presence locations or PoPs scattered in various countries. This will reduce the latency or the loading time of your website, images, videos, and static assets since the data are all delivered from the PoPs and not from your local point of origin. So if you have a high-resolution photo hosted in a server in New York, that data won't need to travel across the Atlantic ocean just for your European customers to see.

That wide geographic distance between your origin server in the US and other countries causes significant latency and slow site performance. You cannot just build thousands of physical edge locations in hundreds of countries overnight, but you can use certain cloud services to meet your needs without doing the heavy lifting.

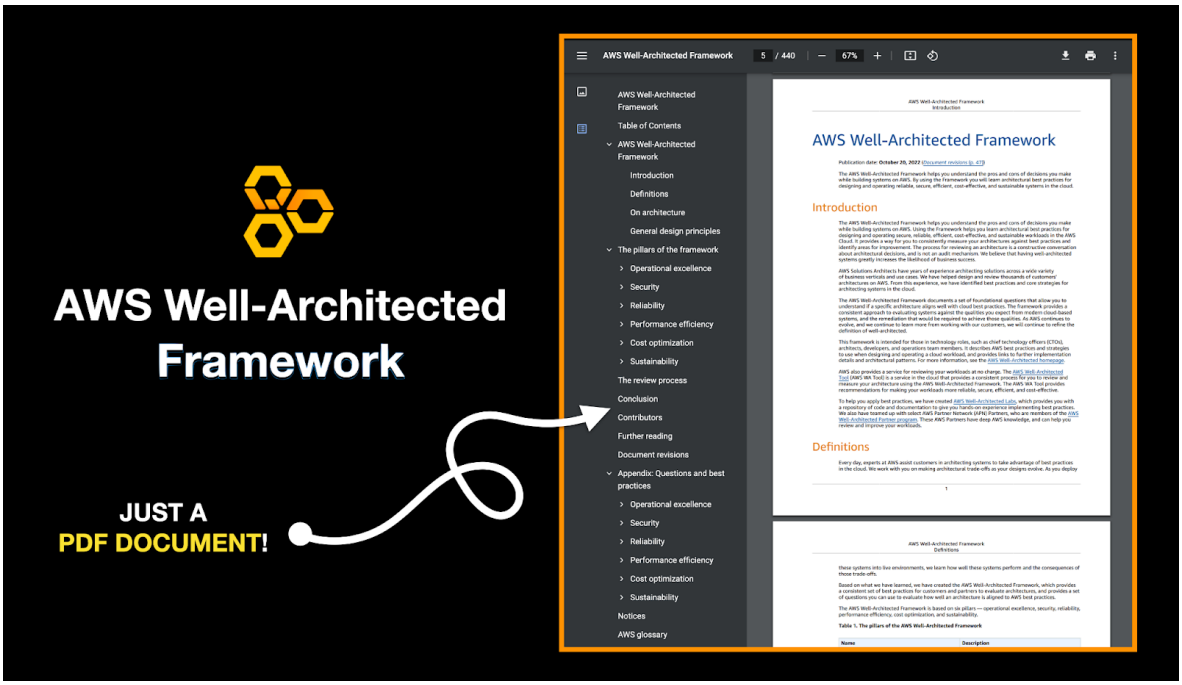
There are also foreign laws and security requirements that companies have to comply with. Data sovereignty requirements vary from one industry to another, and there are certain regional rules that you have to strictly follow or be faced with hefty fines. Cloud computing provides an option to quickly establish a digital presence in other countries while being compliant with its data protection and privacy laws.

Suppose you're running an international online banking application for European Union countries, namely France, Spain, Italy, Netherlands, Germany, and others. By law, the customer data must reside in a network of data centers that are physically located within mainland Europe. The data you collect from the EU countries is subject to the laws and regulations of the European Union, where that particular data is collected and processed. This is called GDPR or the General Data Protection Regulation.

Each country has its own data privacy law that dictates the data residency and data sovereignty that companies should abide by. This includes the government, financial industry, personal health information for public health, and other sectors that handle sensitive data.

AWS Well-Architected Framework

A building is only as strong as its foundation. This fundamental principle is not just applicable to masonry but also to cloud architecture as well. There's a particular framework that can guide you in designing a secure, high-performing, resilient, and efficient architecture for your various applications and workloads in the AWS Cloud. This framework is called the AWS Well-Architected Framework.



What is the AWS Well-Architected Framework?

Basically, the AWS Well-Architected Framework is a body of knowledge that describes the key concepts, design principles, and architectural best practices to help you design and run efficient workloads in AWS. You can ensure that your cloud architecture aligns with the AWS best practices by answering several foundational questions provided by this framework. It also comes with related services and tools that you can use to measure the overall efficiency of your design. This will empower you to improve your existing IT infrastructure in terms of operations, security, reliability, efficiency, cost optimization, and sustainability.

The AWS Well-Architected Framework is categorized into several pillars, which reflect the various components of your architecture. Each pillar has its own roster of key topics that revolve around a specific subject matter. For every key topic, there's a list of common design patterns and anti-patterns for your architecture. A design pattern is a common blueprint of what's actually being used by thousands of companies to fulfill a particular use case more effectively, while an anti-pattern is its complete opposite which often produces entirely ineffective or subpar results. These design patterns and anti-patterns reflect the associated technical concepts, design principles, and strategies for that knowledge area.

The key topics in each pillar have a corresponding implementation guide that you can follow along. The risk level is also described if this recommendation is not established in your architecture. There's also a list of benefits included if you apply the recommended best practice on your own cloud infrastructure.

You'll discover a whole range of industry best practices in the AWS Well-Architected Framework. Essentially, a best practice is a proven method or technique that has been generally accepted as the best or the superior way to other known alternatives, as it often produces better results compared with other means. This distinct practice is considered the standard way of doing things in the status quo.

How does the AWS Well-Architected Framework work?

So how does the AWS Well-Architected Framework improve the design of your cloud solutions?

Suppose you are developing an online solution that handles sensitive financial information. Your application has passed all the integration tests and is now ready for production deployment. However, you still want to verify that your cloud infrastructure is indeed secure as part of your corporate compliance.

You can check the security pillar of the AWS Well-Architected Framework that focuses on protecting your data, files, and overall systems. This includes key topics on data integrity, managing user permissions, and establishing controls to detect security incidents.

In essence, you can improve your cloud designs by simply answering the evaluation questions and following the best practices provided by this framework. These questions will shed light on your existing or new architecture in the AWS Cloud. It has questions like:

- "How do you protect your data at rest?"
- "How do you protect your data in transit?"
- "How do you manage identities for people and machines?"
- …and so on and so forth.

Your answer to these questions can show if your cloud architecture is secure or not. If you responded *"I don't know"* in the *"How do you protect your data at rest?"* question, then that means your architecture is not secure and has a high number of security vulnerabilities. This signifies that you don't employ encryption and tokenization schemes in your system.

The same goes for the *"How do you protect your data in transit?"* query. If you answer that you do not protect your data in transit, then that indicates your architecture has no firewall rules, network authentication, secure key management, and other mechanisms to keep your sensitive data safe as it traverses through different

systems and networks. With this realization, you can now resolve the deficiencies in your system by following the prescriptive guidance provided by the AWS Well-Architected Framework.

Your team can start incorporating the various security best practices in your AWS workloads. So for 'protecting your data at rest', you will need to implement secure key management, enforce encryption at rest, automate data at rest protection, enforce access control, and use mechanisms to keep unauthorized people away from your data.

Conversely, you can protect your data in transit by authenticating network communications, automating the detection of unintended data access, implementing secure key and certificate management, enforcing encryption in transit via SSL, plus other recommendations that are mentioned in the security pillar of AWS Well-Architected Framework.

And there you have it! We have successfully exposed the fatal flaws of your cloud architecture, like opening a can of worms by merely asking the right architectural questions. Your designs will truly be well-architected because you adhere to the industry-standard best practices of the AWS Well-Architected Framework, which is quite time-consuming to do if you just blindly audit your own systems yourself.

Considerations in using the AWS Well-Architected Framework

You might be thinking: do I really need to follow all of the rules, recommendations, and guidelines of this framework? Is it a strict requirement to use the AWS Well-Architected Framework as a baseline for all of my cloud architectures? And is it really beneficial to spend all the time, effort, money, and resources on this in the long run?

Every company has its own set of requirements, strategies, and budget constraints in building its unique systems. It's not a hard rule to abide by each and every recommendation of this framework, especially if you are just building a prototype or simply testing out stuff in the cloud.

Things are quite different if you are actually designing a cloud solution for production use, where you have to ensure high availability, reliability, security, performance, and cost-effectiveness, amongst others. Your cloud infrastructure must meet all strict regulatory compliance, data protection requirements, and privacy laws. Failure to do so can incur a substantial fine and loss of revenue to the business. Worse, the company can face irreparable damage to its reputation that leads to the demise of its enterprise. This is where having a guide like the AWS Well-Architected Framework becomes absolutely indispensable.

A cloud solution that complies with all the relevant guidelines of the AWS Well-Architected Framework is way better than one with no proper structure at all. It is as if you are building your house on a proverbial solid rock rather than on sinking sand. Your architecture will have a more robust form that can withstand unexpected outages and security vulnerabilities. This is because you are reinforcing your IT infrastructure with the proven cloud best practices that were brought by the AWS team's years of experience and expertise.

It can assist you in determining the vital tradeoffs that you can do, which enables you to select an optimal approach for your particular use case. You'll gain awareness of which component of your solution can be traded off for the other part of your architecture. One aspect of your architecture can either be traded off or be set as a top priority to fulfill a certain need. This eliminates the time-consuming activity of figuring it out on your own and the unreliable guesswork in your design process. Companies can further improve their application performance by trading consistency, durability, and space for time and speed by choosing the optimal architectural trade-offs.

For example, you can trade reliability for lower costs for your prototype applications running in your test environments. This is completely acceptable since non-production environments are usually not bound by a service level agreement or SLA that requires system reliability and high availability. In fact, you are actually wasting a lot of money if you implement too many redundant resources for a test application to optimize its reliability, even though it is not used in production at all. Thus, trading reliability for lower costs is okay in this case.

It's a different story for mission-critical applications in your production environment with a more stringent recovery time and recovery point objective. In this situation, the earlier trade-off is not applicable since greater reliability and availability are expected. You must accept the high operating costs associated with achieving reliability which entails running redundant databases, application servers, and other resources to comply with your RTO and RPO requirements. A lower cost of running your production workloads is not usually traded off at the expense of reliability for obvious reasons.

You can also prioritize performance over costs since customer satisfaction can drive increased revenue for the company, and almost 100% of the time, companies won't ever trade off security over anything else due to the harmful consequences therewith.

These are the helpful insights and benefits you can obtain by using the AWS Well-Architected Framework. Knowing the correct facet of your architecture that you should trade off over the other can fully optimize your workloads while remaining compliant with your corporate obligations.

AWS has millions of customers around the world, and every day, its experts assist customers from all around the globe in designing enterprise-grade cloud solutions. The extensive years of cloud research, development, and experience were collected and condensed to form this powerful knowledge base. You can leverage this wealth of real-world expertise by simply adopting the AWS Well-Architected Framework on your architectural design workflow.

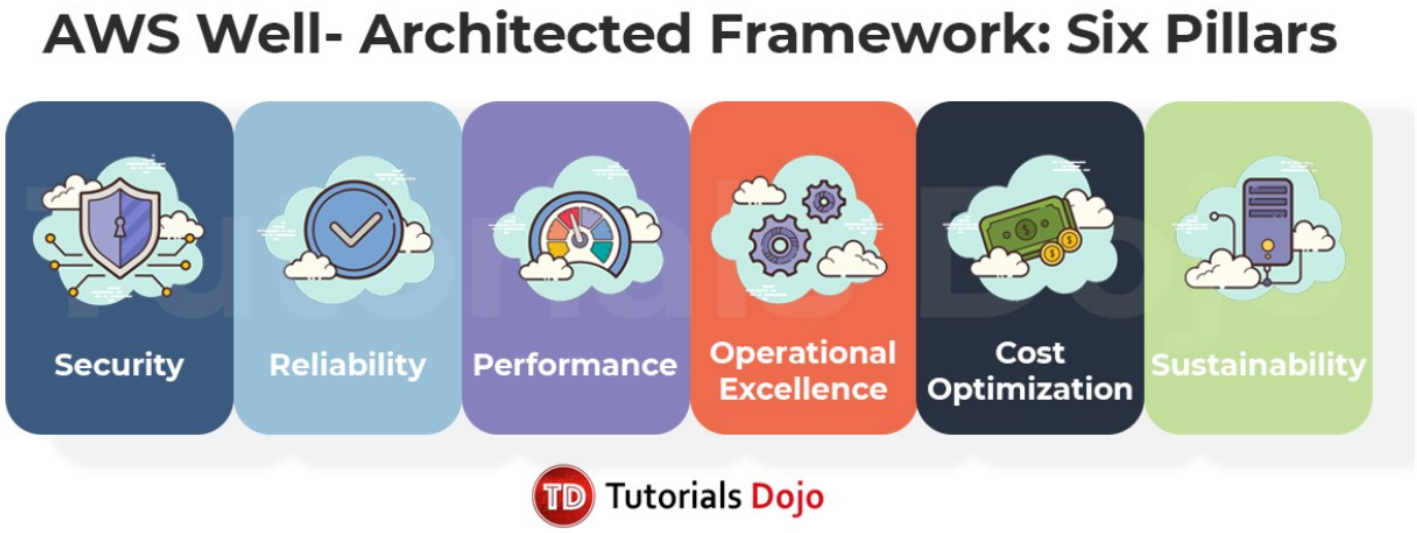
Your company can avoid costly mistakes in terms of money and effort by embracing this fantastic framework in your cloud design. Additionally, you can now measure your architecture against AWS best practices and easily identify your problem areas for improvement.

The Pillars of the AWS Well-Architected Framework

Designing a cloud architecture is like constructing your own home. You have the option of choosing what type of dwelling place you would like to construct, whether it be a mansion, modern farmhouse, midcentury modern ranch, colonial, victorian, bungalow, or any other style you prefer. It all depends on your specific needs, which affect how you design your shelter.

You need to have a blueprint first of how you envision your home would look like. A temporary scaffolding will initially be put together on-premises so the builders can start constructing the floors, frames, walls, roof, and pillars of your residence. These pillars are those upright columns that support the overall structure of your building. If these pillars are poorly constructed, the structural integrity of your house would be compromised, which might result in fatal accidents. This means your property won't withstand natural calamities like earthquakes, winter storms, or hurricanes when it arrives. That's the importance of having strong pillars to ensure that your abode has a firm foundation.

The same goes for designing your cloud architecture in AWS. The AWS Well-Architected Framework is also composed of several pillars where each has its own set of design principles, best practices, strategies, key AWS services, and other useful resources. Conforming with the architectural recommendations of these pillars will guarantee the reliability, security, efficiency, and sustainability of your cloud design.



The pillars in the AWS Well-Architected Framework are Operational Excellence, Security, Reliability, Performance Efficiency, Cost Optimization, and Sustainability. These are primarily used as a guide since every company has its own list of technical requirements. A customer may or may not apply the architectural best practices that are recommended by these pillars depending on their current needs, service level agreement, or preference.

Let's discuss the core concept, design principles, and best practice areas of each of these pillars one by one.

Operational Excellence

The Operational Excellence pillar revolves around how you run your operations in the AWS Cloud to deliver business value. There are design principles and concepts in this pillar that will reveal if your production workloads are operating excellently or poorly.

This includes your ability to effectively run workloads in AWS, gain helpful insight into your cloud operations, and continuously improve your supporting processes and procedures. For example, you can design your AWS workloads to have loosely-coupled components which can be updated on a regular basis and where the changes can be made in small, reversible increments. You have to set protocols in place to continuously improve the supporting processes of your cloud operations. These supporting processes can be in the form of continuous improvement, knowledge management, post-incident analysis, feedback loops, and other protocols that support your primary processes in AWS. Think of these supporting processes as the scaffolding of your actual house. These are not actual parts of your building per se, but they work as structural reinforcement that allows you to build your home more effectively.

This pillar also tackles risk mitigation, wherein you have to anticipate failures by identifying potential sources of failure so that they can be removed or mitigated. Disaster recovery exercises can verify the effectiveness of your response procedures in anticipation of a possible system outage. This is why setting up regular game days or team drills to test your disaster recovery action plan is immensely helpful.

Operational Excellence also depends on how much support the organization provides to achieve its business objectives. It has the following design principles:

- Perform operations as code
- Make frequent, small, reversible changes
- Refine operations procedures frequently
- Anticipate failure
- Learn from all operational failures

There are four best practice areas for operational excellence in the cloud. These are:

- Organization
- Prepare
- Operate
- Evolve

Security

The second pillar is all about the overall security of your cloud solutions. This is actually self-explanatory, but this particular concept encompasses a critical part of your infrastructure. Security is always the top priority for organizations, and it should not be traded off with any other aspect of your cloud design.

The security pillar verifies your ability to protect the data, systems, and assets by taking advantage of various cloud services available at your disposal. Traceability is a big part of this pillar, where you have to monitor and track the changes to your environment and resources in real time. Collecting API call logs and metric data is essential to automatically investigate production incidents and to be able to take the appropriate course of action quickly. The aim here is to improve your overall security posture.

For example, you can enable traceability by using AWS Config to record, audit, and evaluate changes to AWS resources in your production environment. Since you know the specific component that was compromised, the process of resolving the security issues can be expedited. You can implement data encryption, tokenization, SSL, and firewalls to protect your data in transit as well as at rest. Granting the least privilege to your staff with the minimum permissions required to perform a task is highly recommended too. These are the signs that your cloud architecture is secure.

There are many design principles that you can adopt to reinforce your workload security. They are:

- Implement a strong identity foundation
- Enable traceability
- Apply security at all layers
- Automate security best practices
- Protect data in transit and at rest
- Keep people away from data
- Prepare for security events

The security in the AWS Cloud is composed of six knowledge areas. We have

- Foundations for Security
- Identity and access management
- Detection
- Infrastructure protection
- Data protection
- Incident response

Reliability

The third pillar is about reliability which is focused on the ability of your systems to work correctly and consistently. This will give you confidence that your applications remain reliable even if there are traffic surges, unexpected system changes, or natural disasters battering your infrastructure. The reliability pillar also includes the ability to operate and test your production workloads throughout its entire lifecycle.

The keyword for this reliability pillar is recovery. You must design your systems to have the ability to recover easily from service disruptions, natural disasters, application failures, and other outages. Your architecture should also dynamically acquire computing resources to meet the growing demand as needed.

Suppose your system was able to recover from infrastructure or service disruptions. That means you successfully implemented the best practices mentioned in the reliability pillar. If a company implements Amazon EC2 Auto Scaling on multiple Availability Zones with an Application Load Balancer, then that will allow the application to automatically recover unhealthy applications that are running on their Amazon EC2 instances. This is because Auto Scaling can dynamically acquire computing resources to avoid system degradations which can lead to outages.

Another example would be to use Cross-Region Replication for your databases, S3 buckets, and other resources to increase the ability of your systems to recover. Most of the time, production outage occurs when national disasters or power interruptions happen in the proximity of your primary AWS Region. Remember that your cloud resources are actually hosted in physical data centers, and your resources might be affected if the electricity and network connection are cut-off from these data facilities. You'll be able to recover from failures faster if you have both a separate computing capacity and replicated data available in another AWS Region. Again, the keyword for this particular pillar is recovery.

The Reliability pillar has several design principles that can help improve the reliability of your cloud designs. These principles are:

- Automatically recover from failure
- Test recovery procedures
- Scale horizontally to increase aggregate workload availability
- Stop guessing capacity
- Manage change through automation

This pillar also describes the best practice for these four areas:

- Foundations for Reliability
- Workload Architecture
- Change Management
- Failure Management

Performance Efficiency

This fourth pillar is called Performance Efficiency, which is the ability to use resources to meet your system requirements efficiently. This focuses on achieving and maintaining a high level of efficiency even as your customer demand changes. The performance of your cloud architecture can be improved by adopting new technologies, such as going serverless or changing the design of your system for a more efficient approach.

For instance, a company is planning to replace all of its physical servers on-premises with AWS serverless compute services. The lower cost, high scalability, and other benefits of serverless computing can be quickly realized after the migration. This shows that the company is aiming for performance efficiency as it wants to take advantage of the advanced technologies available in AWS.

The Performance Efficiency pillar has the following design principles to help you achieve and maintain efficient workloads in your cloud architecture:

- Democratize advanced technologies
- Go global in minutes
- Use serverless architectures
- Experiment more often
- Consider mechanical sympathy

The best practice areas that are covered in this pillar are as follows:

- Selection
- Review
- Monitoring
- Trade-offs

Cost Optimization

The fifth one is the Cost Optimization pillar which is essentially the ability to run your systems and deliver business value at the lowest price point possible. Cost optimization is a continual process of improving your workload to achieve the business outcomes expected of you while minimizing costs. This will allow your organization to increase revenue and maximize its return on investment.

Cost Optimization can also be achieved by adopting a consumption model where companies will only pay for the resources they actually consume. Opting for pay-as-you-go pricing can help you remove the reliance on elaborate forecasting to determine what would be the expected usage of your compute resources. This type of forecasting could be extremely inaccurate and entails lots of guesswork. Traditional pricing schemes are not mandatory in AWS, so companies can pay only for the resources that it actually uses. It also provides them the ability to increase or decrease their resource usage to meet their ever-changing business requirements.

There are design principles that you have to consider for cost optimization. They are:

- Implement cloud financial management
- Adopt a consumption model
- Measure overall efficiency
- Stop spending money on undifferentiated heavy lifting
- Analyze and attribute expenditure

This pillar has five focus areas that you can do to fully optimize the costs of running your cloud workloads. These are:

- Practice Cloud Financial Management
- Expenditure and usage awareness
- Cost-effective resources
- Manage demand and supplying resources
- Optimize over time

Sustainability

The sustainability pillar is all about sustainable development, which addresses the long-term environmental, economic, and societal impact of your business operations. Basically, sustainable development is a type of "development that meets the needs of the present without compromising the ability of future generations to meet their own needs" as defined by the United Nations World Commission on Environment and Development.

Your business operations may have negative environmental impacts such as carbon emissions, unrecyclable waste, and damage to shared natural resources. This pillar has a spotlight on environmental sustainability which is a shared responsibility between you and AWS. It works like the Shared Security Responsibility Model in AWS, but instead of Security, this one is focused on Sustainability. AWS is responsible for optimizing the sustainability "of" the cloud, while the customers are responsible for the sustainability "in" the cloud. AWS delivers efficient shared infrastructure, water stewardship, and sources renewable power, whereas the customers optimize their workloads and resource utilization by minimizing the total resources required to run their production workloads.

You can use these design principles to maximize sustainability and minimize the environmental impact of your cloud workloads:

- Understand your impact
- Establish sustainability goals
- Maximize utilization
- Anticipate and adopt new, more efficient hardware and software offerings,
- Use managed services
- Reduce the downstream impact of your cloud workloads

There are knowledge areas in the sustainability pillar that represents opportunities to employ the best practices to reduce the sustainability impact of your cloud workloads. You can increase the overall energy efficiency of your cloud systems by optimizing your workload placement and switching to a more efficient deployment pattern. This can also be done by maximizing the utilization of your resources and minimizing the total resources needed to support your workload.

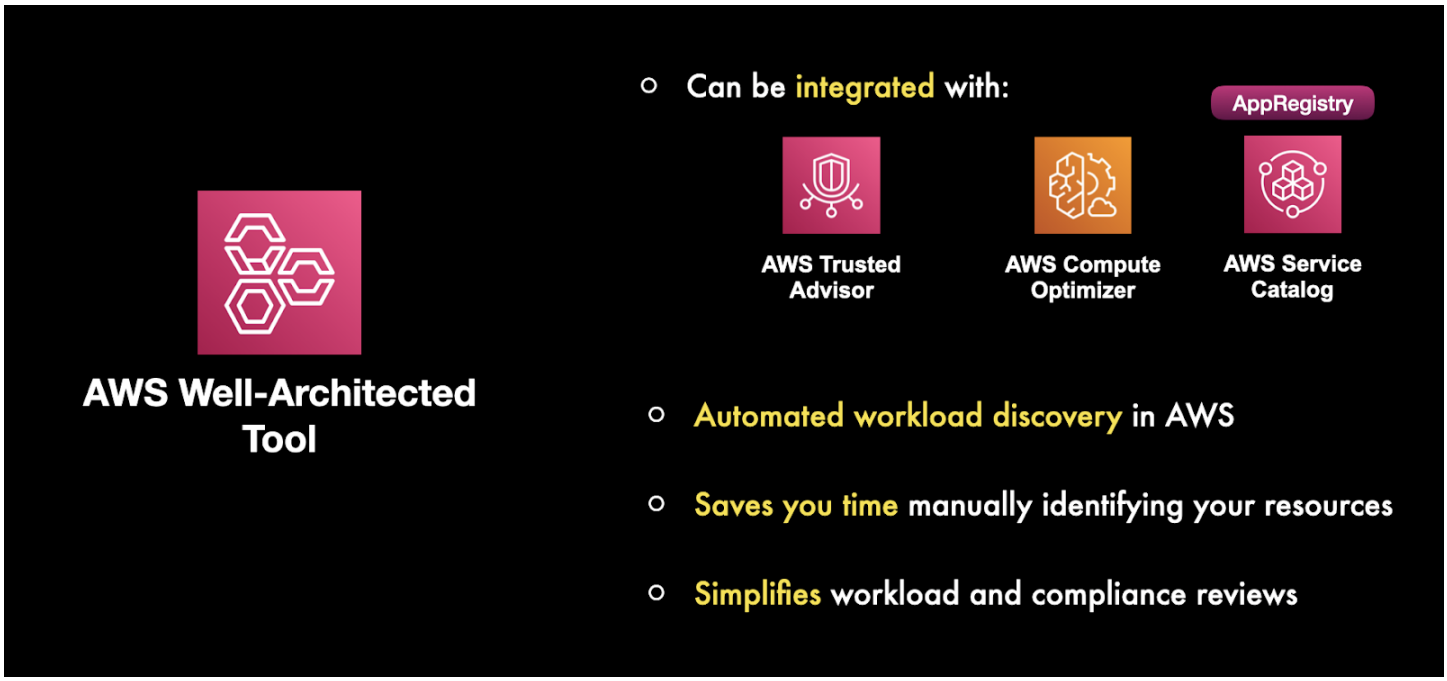
The best practice areas for the Sustainability pillar are:

- Region selection
- User behavior patterns
- Software and architecture patterns
- Data patterns
- Hardware patterns
- Development and deployment process

AWS Well-Architected Tool

The AWS Well-Architected Framework is a body of knowledge that describes the key concepts, design principles, and architectural best practices for designing and running efficient workloads in AWS. This framework is actually just a document in its raw form which means you have to manually do the architectural checks on your cloud architecture yourself to determine if you are using the recommended AWS best practices.

Suppose you want to measure the security compliance of your AWS infrastructure. You have no choice but to read the entire Security Pillar section of the AWS Well-Architected Framework and manually inspect your cloud stack to see if the design principles are incorporated or not. It's very tedious to track all your components one by one if they conform to the best practices recommended by AWS. This task is quite difficult to accomplish and consumes much of your time. The combination of a steep learning curve and management overhead makes it unsustainable in the long run, which is why most companies give up on adopting this framework.



Good thing is that AWS provides a way to easily incorporate the design principles and best practices at your fingertips. This is possible through the AWS Well-Architected Tool or WA Tool for short. The AWS Well-Architected Tool is nothing but a self-service console that you can use to check your cloud architectures against the AWS Well-Architected Framework. It also is available at absolutely no charge in the AWS Management Console, so you can use this as often as you want.

This tool is designed to help you review the state of your applications and cloud workloads against architectural best practices in the AWS Cloud. It also helps you to easily identify opportunities for improvement and track your progress over time. The AWS WA Tool can be integrated with other AWS services, such as the AWS Trusted Advisor, AWS Compute Optimizer, and AWS Service Catalog, among others.

For example, you can use the AppRegistry module of the AWS Service Catalog to register your custom application and its associated resource collection. This application can be connected to the Well-Architected Tool via the AppRegistry for easier workload discovery. Since the task of discovering the different components of your workload is automated, you will be able to save a lot of time since you won't have to manually search your resources as part of assessing your architecture. You can even enable the AWS Trusted Advisor to simplify your workload reviews by providing automated context for supported questions.

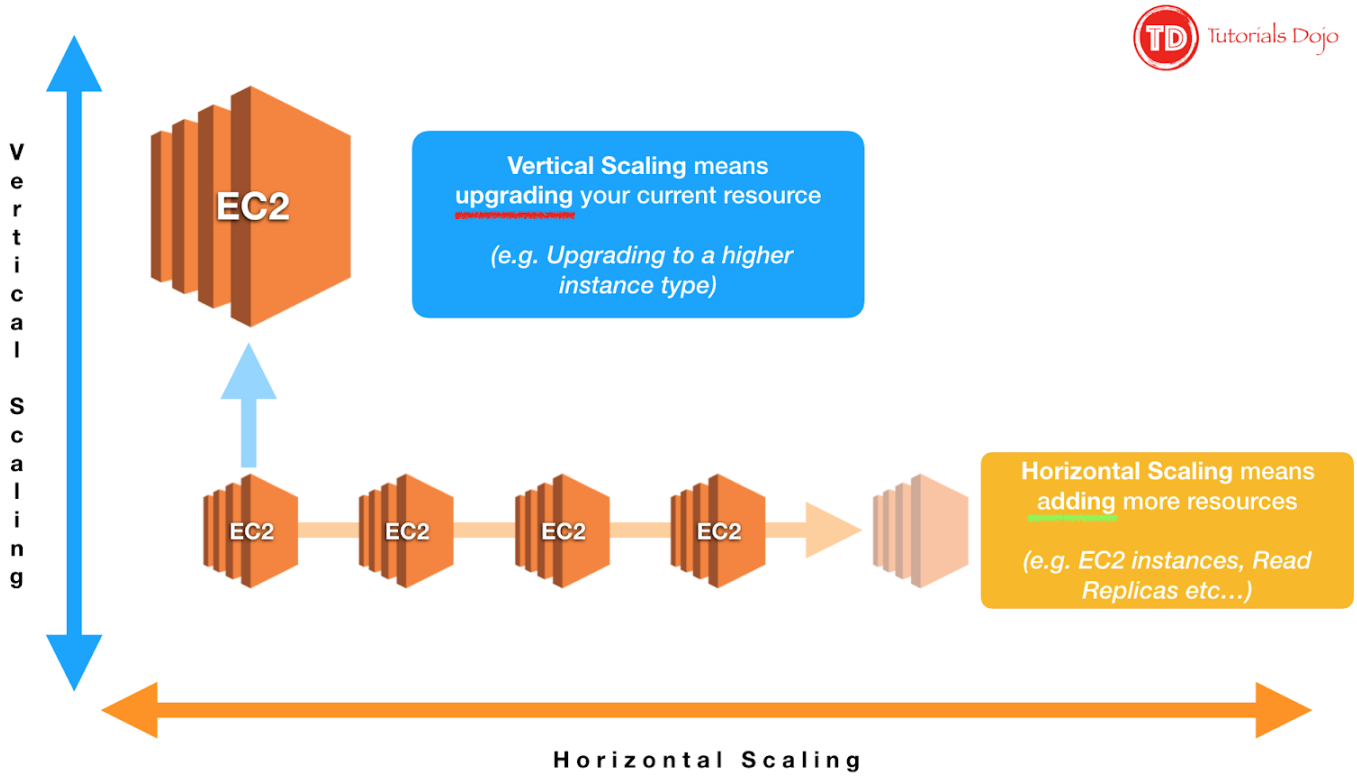
Furthermore, the AWS Well-Architected Tool provides APIs that you can use to extend the AWS Well-Architected functionality into your existing applications, architecture governance processes, and workflows. These APIs provide programmatic access to the AWS Well-Architected Tool without using the AWS Management Console. You can create custom programs using these APIs to fetch the workloads, best practices, and measurements programmatically. The AWS Command Line Interface, or the AWS CLI, also has available commands that you use to invoke the Well-Architected Tool APIs.

There are three primary steps in using the AWS Well-Architected Tool. The first step is to define your workload, and second, conduct an architectural review of your cloud systems. The last step is to apply the best practices and improvement plans which are identified in the review.

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.

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.

12. Cloud Architecture Best Practices

There are various best practices that you can follow which can help you build an application in the AWS cloud. The notable ones are:

1. **Decouple your components** - the key concept is to build components that do not have tight dependencies on each other so that if one component were to fail for some reason, the other components in the system will continue to work. This is also known as loose coupling. This reinforces the Service-Oriented Architecture (SOA) design principle that the more loosely coupled the components of the system are, the better and more stable it scales.
2. **Think parallel** - This internalizes the concept of parallelization when designing architectures in the cloud. It encourages you to implement parallelization whenever possible and to also automate the processes of your cloud architecture.
3. **Implement elasticity** - This principle is implemented by automating your deployment process and streamlining the configuration and build process of your architecture. This ensures that the system can scale in and scale out to meet the demand without any human intervention.
4. **Design for failure** - This concept encourages you to be a pessimist when designing architectures in the cloud and assume that the components of your architecture will fail. This reinforces you to always design your cloud architecture to be highly available and fault-tolerant.

Disaster Recovery Concepts

- **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 Strategies With AWS

- **Backup and Restore** - storing backup data on S3 and recovering 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 multiple availability zones in AWS

Related AWS Services

- **Amazon S3** as a destination for backup data that might be needed quickly to perform a restore.
- **AWS DataSync** for transferring very large data sets by shipping storage devices directly to AWS.
- **Server Migration Service** for performing agentless server migrations from on-premises to AWS.
- **Database Migration Service and Schema Conversion Tool** for moving databases from on-premises to AWS and automatically converting SQL schema from one engine to another.
- **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.
- **AWS 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 provisioning 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.

AWS Support Plans

Amazon Web Services provides a range of tools, technology, people, and programs to help you launch, run, and troubleshoot your AWS workloads. You can choose from the different Support Plans in AWS that suit your needs and budget. Each support plan has a unique set of features, capabilities, and actual resource teams that you can tap on for getting technical assistance, ensuring site reliability, and maintaining the high performance of your AWS cloud infrastructure. It also provides automated checks for reducing costs, improving performance, and strengthening the security of your cloud systems so you can focus better on more important aspects of your business.

There are 5 different support plans available on AWS; we have Basic, Developer, Business, Enterprise On-Ramp, and Enterprise. The entry-level option is called the Basic plan, which is free of charge, while others have their own respective monthly subscription fees. The Developer plan is the cheapest among all the paid support plans. This is followed by the Business plan, Enterprise On-Ramp plan, and the Enterprise plan, which is the most expensive one among all the five support plans. You can use more features if you opt for a higher tier, such as the Business or Enterprise plan. However, it is up to you to decide which support plan better suits your company's needs based on the features that you require, such as the support response time, architectural guidance, programmatic case management via the AWS Support API, third-party software support, Proactive Self Service Programs in AWS Systems Manager, technical account management, account assistance and many more.

These plans are subject to a 30-day minimum term with a unique set of features that you and your company can utilize. AWS can allocate actual people to assist you and your company in setting up, troubleshooting, securing, and improving the overall performance of your cloud infrastructure. For example, you can raise a case ticket on the AWS Support Center Console if you encounter a problem in one of your AWS resources. The support personnel that provides you with technical assistance on the AWS Support Center can either be a Cloud Support Associate or a Cloud Support Engineer. Cloud Support Associates can only give limited support, while Cloud Support Engineers provide more extensive assistance. You can also have priority access to the Concierge Support Team, the AWS Managed Services team, and a dedicated Technical Account Manager for your AWS account, which is known as TAM. Your access to these diverse AWS teams depends on the support tier that you signed up for.

AWS has 5 different Support Plans:

1. Basic
2. Developer
3. Business
4. Enterprise
5. Enterprise On-Ramp

Let's check out each support plan one by one.

Basic Support Plan

The Basic Support plan is included for all AWS customers by default. This means that once you create your AWS account, certain support features are already available for you to use. You'll have 24-by-7 access to the AWS customer service, documentation, whitepapers, and the community-based AWS re:Post site. Since this is just a basic plan, expect some delayed response times when you raise a ticket on its online AWS Support Center. You can also access the AWS Personal Health Dashboard, which provides a personalized view of the health of the AWS services that you are using.

Access to the core AWS Trusted Advisor checks are available as well, which can help you provision your AWS resources based on a set of cloud computing best practices that increase performance and improve overall security. These core Trusted Advisor checks are only a subset of the entire AWS Trusted Advisor features. You can only access the full set of best practice checks in the AWS Trusted Advisor on higher support plans, namely the Business, Enterprise On-Ramp, and Enterprise.

Developer Support Plan

The Developer support plan is recommended if you are just experimenting or testing out your prototype in AWS. However, this is not a hard rule, and you can still use this even if you have production workloads running in the AWS Cloud. Just be mindful of the limitations of this support plan which you might need down the line. This tier includes a limited number of best practice checks in AWS Trusted Advisor, like basic security checks and service quota, but not the full set.

In terms of technical support, you will be given access to Amazon's Cloud Support Associates, who can provide tech support to your AWS issues during business hours only, as well as prioritized responses on the AWS re:Post website. You can also raise unlimited support cases using 1 primary contact, which is the root user of your AWS account. Do take note that you will only be able to reach the Support Associates via the web, excluding any phone or chat interactions. The business hours in AWS are usually from 8:00 AM in the morning to 6:00 PM at night in your country's time zone, which is set in your My Account console. This schedule excludes public holidays and weekends.

If your issue only requires general guidance, then the response time should be within 24 hours, but if it is a system-impaired case involving one or more AWS services, you can expect an answer within 12 hours. So if you have a Developer support plan and you raised a case online on the AWS Support Center on a Saturday, the AWS team will only be able to work on your request on the following Monday. Again, this support plan only provides support during weekdays from 8 AM to 6 PM. The AWS Support team won't be able to prioritize your request, and there's no phone or chat assistance too for this particular support tier.

The Developer Support plan also provides access to the basic Support Automation Workflows that are available in the AWS Systems Manager service. Support Automation Workflows or SAWs are automation runbooks that are written and maintained by the AWS Support team. These SAW runbooks could help you troubleshoot common issues with your AWS resources automatically, including a range of custom tasks like running the EC2Rescue tool on your unreachable EC2 servers, resetting passwords or SSH keys on your Linux instances, fixing Remote Desktop Protocol connections on your Windows servers, and so much more. All of these custom runbooks are available in the Automation section of the AWS Systems Manager console.

Support Automation Workflows have two types:

- Basic runbooks
- Premium runbooks.

The first type has a prefix of AWSSupport and only covers basic automation tasks. On the other hand, the premium runbooks have a prefix of AWSPremiumSupport, which provides more advanced automation workflows. The basic runbooks are only covered in the Developer Support plan while the Business, Enterprise On-Ramp, and Enterprise Support customers have access to the premium runbooks with the AWSPremiumSupport prefix.

So if you have a Developer plan and you want to move your Amazon EC2 instance to another subnet, Availability Zone, or VPC, then you can take advantage of the AWSSupport-CopyEC2Instance runbook. This support automation workflow can automatically copy your instance to another subnet, another AZ, or another Amazon VPC without any manual intervention. Apart from that, you can also use the `AWSSupport-ResetAccess`, `AWSSupport-ExecuteEC2Rescue`, `AWSSupport-ListEC2Resources`, and any other runbooks with an `AWSupport` prefix.

Business Support Plan

This support tier is recommended if you have one or more production workloads in AWS. The Business Support plan is one step higher than the Developer Support plan, so a handful of new and upgraded features are included in this tier.

This includes the full set of best practice checks in AWS Trusted Advisor and has 24-by-7 access to Cloud Support Engineers via phone, web, and chat. Keep in mind that the people who will assist you are Cloud Support Engineers, who are more capable than the first-level Cloud Support Associates. Additionally, you have 24-by-7 tech support for your AWS issues and not just during business hours or weekdays. You can submit unlimited support cases on the AWS Support Center using one or more IAM Users in your AWS account on top of your root user access.

Your company can enjoy prioritized responses on the AWS re:Post site, including the AWS Support App in Slack for quicker access. This AWS Support App enables you to easily manage your AWS support cases on Slack and allows you to invite your team members to chat channels, respond to case updates, and chat directly with AWS support agents. For this support tier, AWS Support will provide contextual architectural guidance to your use cases and not just basic general support.

The response times for the general guidance and system-impaired issues are similar to the Developer Support plan, which are 24 hours and 12 hours, respectively. Moreover, you'll also get enhanced support for production systems. Degraded or impaired AWS resource issues have an SLA of 4 hours, while outages to your production systems have a one hour response time.

You can access the AWS Support API if you choose to have a Business Support plan. The AWS Support API is basically an application programming interface, or a web service, that provides programmatic access to some of the features in the AWS Support Center. This API provides two different operation types: support case management operations and AWS Trusted Advisor operations. The first operation type can help you manage the entire lifecycle of your AWS support cases, like creating, updating, or completing a support case, while Trusted Advisor operations allow you to access the AWS Trusted Advisor checks programmatically via AWS SDK without manually checking them on your AWS Management Console.

The Business Support plan is also suitable for companies that use a lot of third-party software. This tier provides support for interoperability, configuration guidance, and troubleshooting for your third-party tools and software that require interaction with your AWS resources. It supports both the basic and premium Support Automation Workflows in AWS thus, you can use all runbooks with the prefix of AWSSupport and AWSPremiumSupport in AWS Systems Manager Automation.

Access to Infrastructure Event Management is available for an additional fee. Essentially, the AWS Infrastructure Event Management, or IEM, offers architecture guidance and operational support during the preparation and execution of the planned events in your company. This is helpful if you have a scheduled shopping holiday, product launches, system migrations, or any event launches in the coming days or weeks ahead. These activities may cause unnecessary system degradation or even site outages if your cloud architecture is not properly optimized. With IEM, you can easily assess operational readiness, mitigate risks, and execute your planned activity confidently with assistance from AWS experts.

Furthermore, access to the AWS Managed Services, or AMS, is also available for an additional fee if you want to. AMS can help you operate your AWS infrastructure on your behalf, augmenting your existing internal teams with advanced cloud operation skills and capacity which conforms with the baseline operations you have set. AMS provides you with AWS experts to run your workloads in AWS effectively, such as a designated Cloud Service Delivery Manager, a Cloud Architect, an AMS security team, or all three.

Enterprise On-Ramp Support Plan

The next one that we will discuss is the Enterprise On-Ramp support plan. This is highly recommended if you are running business-critical production workloads in AWS Cloud with strict Service Level Agreements or SLA – meaning, your applications have a shorter recovery time objective or RTO.

The Enterprise On-Ramp support plan has the same set of features that the Business support plan has but with more capabilities. Its response times for general guidance and system-impaired issues are similar to the Business Support plan. Impaired AWS resource issues also have an SLA of 4 hours and a one-hour response time baseline for production outages. In addition to this, the Enterprise On-Ramp support tier also supports business-critical system outages, with less than 30 minutes of target response time.

Priority access to the AWS Concierge Support Team is provided too. The Concierge Support Team is your primary contact for your AWS Billing and AWS Support concerns. Infrastructure Event Management is also included, which you can use for one event launch per year. Consultative review and architectural guidance are provided based on your custom applications, systems, or architectures.

Enterprise Support Plan

Finally, we have the Enterprise Support plan, which is usually recommended if you have mission-critical workloads in AWS. All of the features in the On-Ramp support plan are practically the same, with just a slight variation.

The Enterprise Support plan is the most expensive support plan among all others. It comes with the AWS Trusted Advisor Priority feature, which prioritizes recommendations that the AWS account team curated. It has the quickest response times for mission-critical workloads of 15 minutes, which is faster compared with the 30-minute RTO of its Enterprise-On Ramp counterpart. You can use the Infrastructure Event Management capability more than once on this support plan, and not just once per year.

This option grants you access to online self-paced labs and to the AWS Incident Detection and Response program for an additional fee. Basically, the AWS Incident Detection and Response program is a 24-by-7 proactive monitoring and incident management for your selected production workloads that are regularly conducted by AWS experts. You can also access the AWS Support Proactive Services, including operational reviews of your resources, operational workshops, and deep dive sessions.

You'll also be given a designated Technical Account Manager (TAM) that you count on and not just a random pool of Technical Account Managers that may change over time.

Comparison of AWS Support Plans

	DEVELOPER	BUSINESS	ENTERPRISE
Use Case	Recommended if you are experimenting or testing in AWS.	Recommended if you have production workloads in AWS.	Recommended if you have business and/or mission critical workloads in AWS.
AWS Trusted Advisor Best Practice Check	7 Core checks	Full set of checks	Full set of checks
Architectural Guidance	General	Contextual to your use-cases	Consultative review and guidance based on your applications
Technical Account Manager	✗	✗	Designated Technical Account Manager (TAM) to proactively monitor your environment and assist with optimization.
Training	✗	✗	Access to online self-paced labs
Account Assistance	✗	✗	Concierge Support Team
Enhanced Technical Support	Business hours** email access to Cloud Support Associates. Unlimited cases / 1 primary contact	24x7 phone, email, and chat access to Cloud Support Engineers Unlimited cases / Unlimited contacts (IAM supported)	24x7 phone, email, and chat access to Clou Support Engineers Unlimited cases / Unlimited contacts (IAM supported)
Programmatic Case Management	✗	AWS Support API	AWS Support API
Third-Party Software Support	✗	Interoperability & configuration guidance and troubleshooting	Interoperability & configuration guidance and troubleshooting
Proactive Programs	Access to Support Automation Workflows with prefixes AWSSupport	Access to Support Automation Workflows with prefixes AWSSupport and AWSPremiumSupport Access to Infrastructure Event Management for additional fee	• Access to Support Automation Workflows with prefixes AWSSupport and AWSPremiumSupport • Infrastructure Event Management • Well-Architected Reviews • Operations Reviews • Technical Account Manager (TAM) coordinates access to programs and other AWS experts as needed

Customers with an Enterprise support plan are eligible for additional services that are not available in the Developer or Business plans. Aside from having a designated Technical Account Manager, you will also have the following benefits if you opt for an Enterprise-level support in AWS:

- Infrastructure Event Management
- Architecture Support
- White-glove case routing
- Management business reviews
- Concierge Support Team

Technical Support Response Times

	DEVELOPER	BUSINESS	ENTERPRISE
Case Severity / Response Times*	General guidance: < 24 business hours**	General guidance: < 24 hours	General guidance: < 24 hours
	System impaired: < 12 business hours**	System impaired: < 12 hours	System impaired: < 12 hours
		Production system impaired: < 4 hours	Production system impaired: < 4 hours
		Production system down: < 1 hour	Production system down: < 1 hour
			Business-critical system down: < 15 minutes

You can also choose a type of AWS Support Plan based on your production workload. If you are only experimenting, testing or doing a Proof of Concept (POC) in AWS, it is recommended that you choose the Developer plan. If you have production workloads running in AWS, it is suitable to opt for the Business plan. Lastly, if you have mission-critical workloads, it is better to stick with an Enterprise plan because it provides the most efficient response times to support your systems.

With its Enhanced Technical Support, the Enterprise Support plan provides you with 24x7 access to the AWS Cloud Support Engineers via phone, chat, and email. You can also have an unlimited number of contacts that can open an unlimited amount of cases. AWS also provides you with a response time of less than 15 minutes in the event that your business-critical systems go down.

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, Amazon EMR,** 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**.
  - Estimate the cost of migrating your architecture to the cloud.
  - Generate the lowest cost estimate for your workload.

Sources:
https://d1.awsstatic.com/whitepapers/aws_pricing_overview.pdf
https://aws.amazon.com/pricing/
https://aws.amazon.com/ec2/pricing/reserved-instances/pricing/

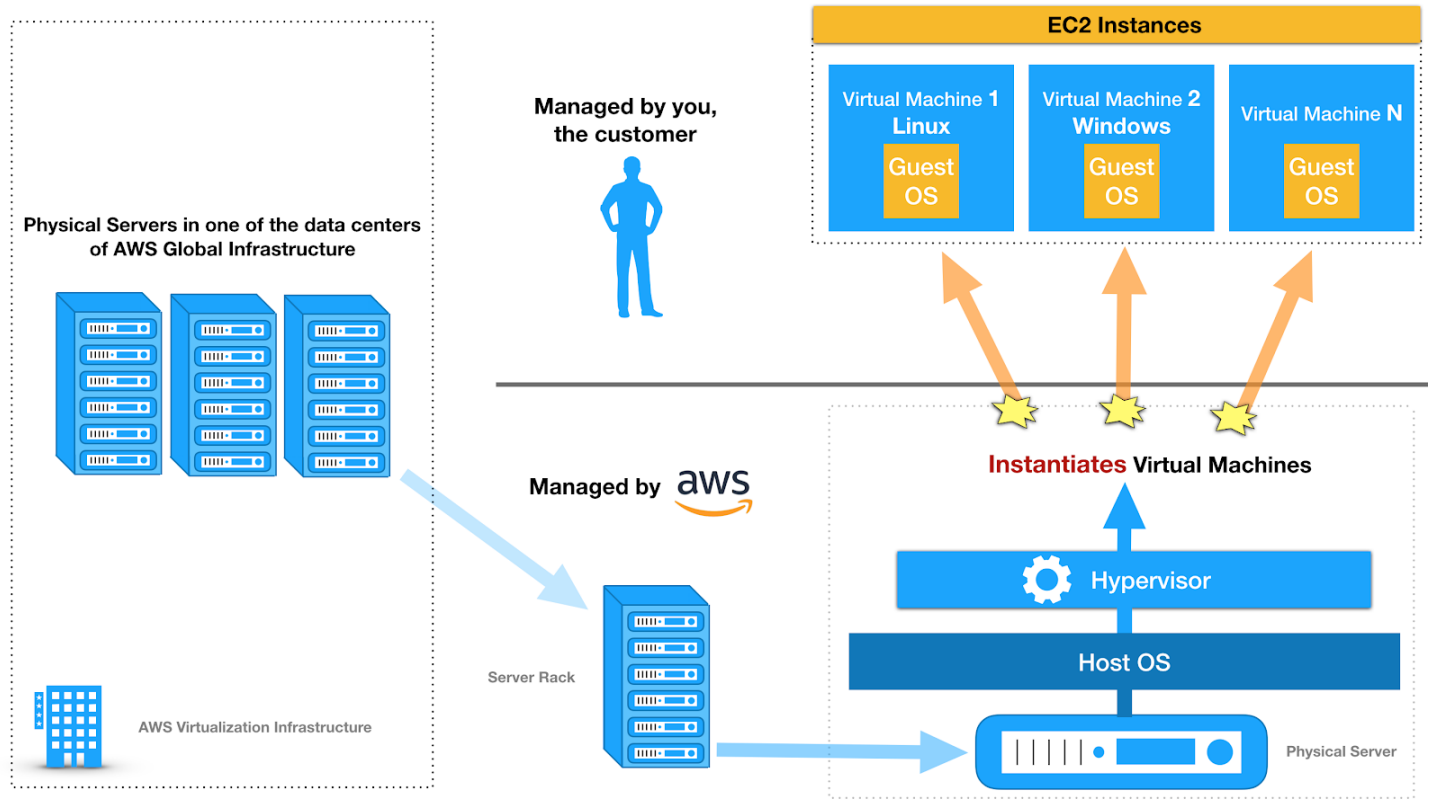
AWS CHEAT SHEETS

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

Amazon EC2 is a computing service that runs virtual servers in the cloud. It allows you to launch Linux or Windows virtual machines to host your applications and manage them remotely – wherever you are in the globe.

You and AWS have a shared responsibility in managing your Amazon EC2 virtual machines. AWS manages the data centers, physical facilities, the hardware components, the host operating system, and the virtualization layer that powers the entire Amazon EC2 service. On the other hand, you are responsible for your guest operating system, applying OS patches, setting up security access controls, and managing your data.

Amazon EC2 can be integrated into other AWS services to accomplish a certain task or to meet your specifications. It can be used to do a variety of functions – from running applications, hosting a self-managed database, processing batch jobs and so much more.

An Amazon EC2 virtual machine is somewhat similar to your desktop or laptop that you may be using right now. It also has a CPU, a Random Access Memory, a Network Interface, an IP address, and even a system image backup. You can also attach a Solid State Drive, or a Hard Disk Drive (HDD) to your EC2 instance for more storage; You can even connect it to a shared network file system, to allow multiple computers to access the same files.

Just like your computer, you can also integrate a lot of other AWS services with Amazon EC2. You can attach various storage, networking, and security services to an Amazon EC2 instance. There are many options available to purchase your EC2 instance, that can help you lower down your operating costs. Some AWS Services are even using Amazon EC2 as its underlying compute component. These services orchestrate or control a group of EC2 instances to perform a specific function, such as scaling or batch processing. It is also used on AWS-managed databases, containers, serverless computing engines, microservices, and many more! This is why Amazon EC2 is considered as the basic building block in AWS – it is used in almost every service!

For storage, you can use different AWS Storage services with your Amazon EC2 instance to store and process data. You can attach an Instance store for your temporary data or an Amazon EBS volume for persistent storage.

You can also mount a file system to your EC2 instances. You can connect to it to Amazon EFS or Amazon FSx. For your static media files or object data, you can store them in Amazon S3 then retrieve them back to your EC2 instance via an API or through an HTTP and FTP client.

For networking, you launch your EC2 instance on either a public or a private subnet in a Virtual Private Cloud or VPC. You can associate an Elastic IP address to your instance for it to have a static IPv4 address. An elastic network interface can also be used as a virtual network card for your EC2 instance. If you have a group of interdependent instances, you can organize them on a placement group. This placement group can be a

cluster, a spread, or a partition type that enables you to minimize correlated failures, lower network latency, and achieve high throughput.

AWS also offers enhanced networking features to provide high-performance networking capabilities by using an Elastic Network Adapter or an Intel 82599 Virtual Function (VF) interface. If you have a High-Performance Computing workload or machine learning applications, you can attach an Elastic Fabric Adapter to your instance to provide a higher network throughput than your regular TCP transport.

For scaling, you can use Amazon EC2 Auto Scaling to automatically add more EC2 instances to process the increasing number of traffic in your application. Auto Scaling can also terminate the underutilized instances if the demand decrease – this can cut down your server expenses in half, or even more!

For system image back up, you can take a snapshot of your EC2 instance by creating an Amazon Machine Image, or AMI.

The AMI is just like a disk image of your Mac, Linux or Windows computer that contains custom data and system configurations that you have set. It enables you to launch a pre-configured Amazon EC2 instance that can be used for auto-scaling, migration and backups. If your EC2 instance crashed, you can easily restore your data using an AMI. It is also helpful if you want to move your server to another Available Zone, another Region or even another AWS account. You can also launch one or more EC2 instances using a single AMI.

There are more AWS services and features that you can integrate with Amazon EC2. We will cover these services in the succeeding chapters of this eBook.

Components of an EC2 Instance

You must know the components of an EC2 instance, since this is one of the core AWS services that you'll be encountering the most in the exam.

1) When creating an EC2 instance, you always start off by choosing a **base AMI or Amazon Machine Image**. An AMI contains the OS, settings, and other applications that you will use in your server. AWS has many pre-built AMIs for you to choose from, and there are also custom AMIs created by other users which are sold on the AWS Marketplace for you to use. If you have created your own AMI before, it will also be available for you to select. AMIs cannot be modified after launch.
2) After you have chosen your AMI, you select the **instance type and size** of your EC2 instance. The type and size will determine the physical properties of your instance, such as CPU, RAM, network speed, and more. There are many instance types and sizes to choose from and the selection will depend on your workload for the instance. You can freely modify your instance type even after you've launched your instance, which is commonly known as “right sizing”.
3) Once you have chosen your AMI and your hardware, you can now configure your instance settings.
    a) If you are working on the console, the first thing you'll indicate is the **number of instances** you'd like to launch with these specifications you made.
    b) You specify whether you'd like to launch **spot instances** or use another instance billing type (on-demand or reserved).
    c) You configure which **VPC and subnet** the instance should be launched in, and whether it should receive a **public IP address** or not.
    d) You choose whether to include the instance in a **placement group** or not.
    e) You indicate if the instance will be joined to one of your **domains/directories**.
    f) Next is the **IAM role** that you'd like to provide to your EC2 instance. The IAM role will provide the instance with permissions to interact with other AWS resources indicated in its permission policy.
    g) **Shutdown behavior** lets you specify if the instance should only be stopped or should be terminated once the instance goes into a stopped state. If the instance supports **hibernation**, you can also enable the hibernation feature.
    h) You can enable the **termination protection** feature to protect your instance from accidental termination.
    i) If you have **EFS file systems** that you'd like to immediately mount to your EC2 instance, you can specify them during launch.
    j) Lastly, you can specify if you have commands you'd like your EC2 instance to execute once it has launched. These commands are written in the **user data** section and submitted to the system.
4) After you have configured your instance settings, you now need to add **storage** to your EC2 instance. A volume is automatically created for you since this volume will contain the OS and other applications of your AMI. You can add more storage as needed and specify the type and size of EBS storage you'd like

to allocate. Other settings include specifying which EBS volumes are to be included for termination when the EC2 instance is terminated, and encryption.

5) When you have allocated the necessary storage for your instances, next is adding **tags** for easier identification and classification.
6) After adding in the tags, you now create or add **security groups** to your EC2 instance, which will serve as firewalls to your servers. Security groups will moderate the inbound and outbound traffic permissions of your EC2 instance. You can also add, remove, and modify your security group settings later on.
7) Lastly, the access to the EC2 instance will need to be secured using one of your **key pairs**. Make sure that you have a copy of this key pair so that you'll be able to connect to your instance when it is launched. There is no way to reassociate another key pair once you've launched the instance. You can also proceed without selecting a key pair, but then you would have no way of directly accessing your instance unless you have enabled some other login method in the AMI or via Systems Manager.
8) Once you are happy with your instance, proceed with the launch. Wait for your EC2 instance to finish preparing itself, and you should be able to connect to it if there aren't any issues.

References:
https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EC2_GetStarted.html
https://tutorialsdojo.com/amazon-elastic-compute-cloud-amazon-ec2/

Types of EC2 Instances

1. **General Purpose** — Provides a balance of compute, memory, and networking resources, and can be used for a variety of diverse workloads. Instances under the T-family have burstable performance capabilities to provide higher CPU performance when CPU is under high load, in exchange for CPU credits. Once the credits run out, your instance will not be able to burst anymore. More credits can be earned at a certain rate per hour depending on the instance size.
2. **Compute Optimized** — Ideal for compute bound applications that benefit from high performance processors. Instances belonging to this family are well suited for batch processing workloads, media transcoding, high performance web servers, high performance computing, scientific modeling, dedicated gaming servers and ad server engines, machine learning inference and other compute intensive applications.
3. **Memory Optimized** — Designed to deliver fast performance for workloads that process large data sets in memory.
4. **Accelerated Computing** — Uses hardware accelerators or co-processors to perform functions such as floating point number calculations, graphics processing, or data pattern matching more efficiently than on CPUs.
5. **Storage Optimized** — Designed for workloads that require high, sequential read and write access to very large data sets on local storage. They are optimized to deliver tens of thousands of low-latency, random I/O operations per second (IOPS) to applications.

6. **Nitro-based** — The Nitro System provides bare metal capabilities that eliminate virtualization overhead and support workloads that require full access to host hardware. When you mount EBS Provisioned IOPS volumes on Nitro-based instances, you can provision from 100 IOPS up to 64,000 IOPS per volume compared to just up to 32,000 on other instances.

References:
https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/instance-types.html
https://tutorialsdojo.com/amazon-elastic-compute-cloud-amazon-ec2/

Instance Purchasing Options

AWS offers multiple options for you to purchase compute capacity that will best suit your needs. Aside from pricing on different instance types and instance sizes, you can also specify how you'd like to pay for the compute capacity. With EC2 instances, you have the following purchase options:

1) **On-Demand Instances** – You pay by the hour or the second depending on which instances you run for each running instance. If your instances are in a stopped state, then you do not incur instance charges. No long term commitments.
2) **Savings Plans** – Receive discounts on your EC2 costs by committing to a consistent amount of usage, in USD per hour, for a term of 1 or 3 years. You can achieve higher discount rates by paying a portion of the total bill upfront, or paying full upfront. There are two types of Savings Plans available:
    a) **Compute Savings Plans** provide the most flexibility since it automatically applies your discount regardless of instance family, size, AZ, region, OS or tenancy, and also applies to Fargate and Lambda usage.
    b) **EC2 Instance Savings Plans** provide the lowest prices but you are committed to usage of individual instance families in a region only. The plan reduces your cost on the selected instance family in that region regardless of AZ, size, OS, or tenancy. You can freely modify your instance sizes within the instance family in that region without losing your discount.
3) **Reserved Instances (RI)** – Similar to Saving Plans but less flexible since you are making a commitment to a consistent instance configuration, including instance type and Region, for a term of 1 or 3 years. You can also pay partial upfront or full upfront for higher discount rates. A Reserved Instance has four instance attributes that determine its price:
    a) Instance type
    b) Region
    c) Tenancy - shared (default) or single-tenant (dedicated) hardware.
    d) Platform or OS

    Reserved Instances are automatically applied to running On-Demand Instances provided that the specifications match. A benefit of Reserved Instances is that you can sell unused Standard Reserved Instances in the AWS Marketplace.

There are also different types of RIs for you to choose from:

a) **Standard RIs** - Provide the most significant discount rates and are best suited for steady-state usage.
b) **Convertible RIs** - Provide a discount and the capability to change the attributes of the RI as long as the resulting RI is of equal or greater value.
c) **Scheduled RIs -** These are available to launch within the time windows you reserve. This option allows you to match your capacity reservation to a predictable recurring schedule that only requires a fraction of a day, a week, or a month.

	Standard RI	**Convertible RI**
Applies to usage across all Availability Zones in an AWS region	Yes	Yes
Can be shared between multiple accounts within a consolidated billing family.	Yes	Yes
Change Availability Zone, instance size (for Linux OS), networking type	Yes	Yes
Change instance families, operating system, tenancy, and payment option	No	Yes
Benefit from Price Reductions	No	Yes
Can be bought/sold in Marketplace	Yes	No

4) **Spot Instances** – Unused EC2 instances that are available for a cheap price, which can reduce your costs significantly. The hourly price for a Spot Instance is called a Spot price. The Spot price of each instance type in each Availability Zone is set by Amazon EC2, and is adjusted gradually based on the long-term supply of and demand for Spot Instances. Your Spot Instance runs whenever capacity is available and the maximum price per hour that you've placed for your request exceeds the Spot price. When the Spot price goes higher than your specified price, your Spot Instance will be stopped or terminated after a two minute warning. Use Spot Instances only when your workloads can be interrupted

5) **Dedicated Hosts** – You 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. Support for multiple instance sizes on the same Dedicated Host is available for the following instance families: c5, m5, r5, c5n, r5n, and m5n. Dedicated Hosts also offers options for upfront payment for higher discounts.
6) **Dedicated Instances** – Pay by the hour for instances that run on single-tenant hardware. Dedicated Instances that belong to different AWS accounts are physically isolated at a hardware level. Only your compute nodes run in single-tenant hardware; EBS volumes do not.

	Dedicated Hosts	Dedicated Instances
Billing	Per-host billing	Per-instance billing
Visibility of sockets, cores, and host ID	Provides visibility on the number of sockets and physical cores	No visibility
Host and instance affinity	Allows you to consistently deploy your instances to the same physical server over time	Not supported
Targeted instance placement	Provides additional visibility and control over how instances are placed on a physical server	Not supported
Automatic instance recovery	Supported	Supported
Bring Your Own License (BYOL)	Supported	Not supported
Instances must run within a VPC	Yes	Yes
Can be combined with other billing options	On-demand Dedicated Hosts, Reserved Dedicated Hosts, Savings Plans	On-demand Instances, Reserved Dedicated Instances, Dedicated Spot Instances

7) **Capacity Reservations** – Allows you to reserve capacity for your EC2 instances in a specific Availability Zone for any duration. No commitment required.

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.

Security Groups And Network Access Control Lists

Security groups and network ACLs are your main lines of defense in protecting your VPC network. These services act as firewalls for your VPCs and control inbound and outbound traffic based on the rules you set. Although both of them are used for VPC network security, they serve two different functions and operate in a different manner.

Security groups operate on the instance layer. They serve as virtual firewalls that control inbound and outbound traffic to your VPC resources. Not all AWS services support security groups, but the general idea is that if the service involves servers or EC2 instances then it should also support security groups. Examples of these services are:

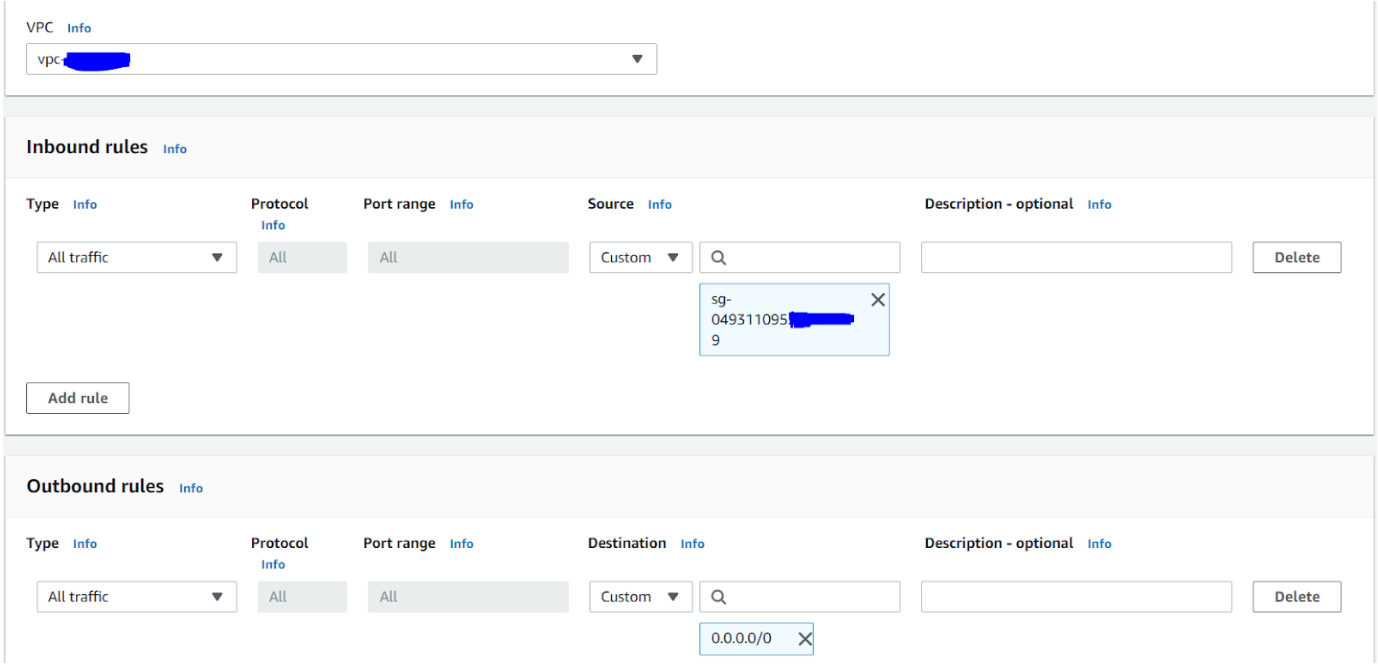
1. Amazon EC2
2. AWS Elastic Beanstalk
3. Amazon Elastic Load Balancing
4. Amazon RDS
5. Amazon EFS
6. Amazon EMR
7. Amazon Redshift
8. Amazon Elasticache

To control the flow of traffic to your VPC resources, you define rules in your security group which specify the types of traffic that are allowed. A security group rule is composed of traffic type (SSH, RDP, etc), internet protocol (tcp or udp), port range, origin of the traffic for inbound rules or destination of the traffic for outbound rules, and an optional description for the rule. Origins and destinations can be defined as definite IP addresses, IP address ranges, or a security group ID. If you reference a security group ID in your rule then all resources that are associated with the security group ID are counted in the rule. This saves you the trouble of entering their IP addresses one by one.

You can only create rules that allow traffic to pass through. Traffic parameters that do not match any of your security group rules are automatically denied. By default, newly created security groups do not allow any inbound traffic while allowing all types of outbound traffic to pass through. Security groups are also stateful, meaning if you send a request from your instance, the response traffic for that request is allowed to flow in regardless of inbound rules. Responses to allowed inbound traffic are allowed to flow out, regardless of outbound rules. One thing to remember is, when you are adding rules to allow communication between two VPC instances, you should enter the private IP address of those instances and not their public IP or Elastic IP address.

Security groups are associated with network interfaces, and not the instances themselves. When you change the security groups of an instance, you are changing the security groups associated with its network interface. By default, when you create a network interface, it's associated with the default security group for the VPC, unless you specify a different security group. Network interfaces and security groups are bound to the VPC

they are launched in, so you cannot use them for other VPCs. However, security groups belonging to a different VPC can be referenced as the origin and destination of a security group rule of peered VPCs.

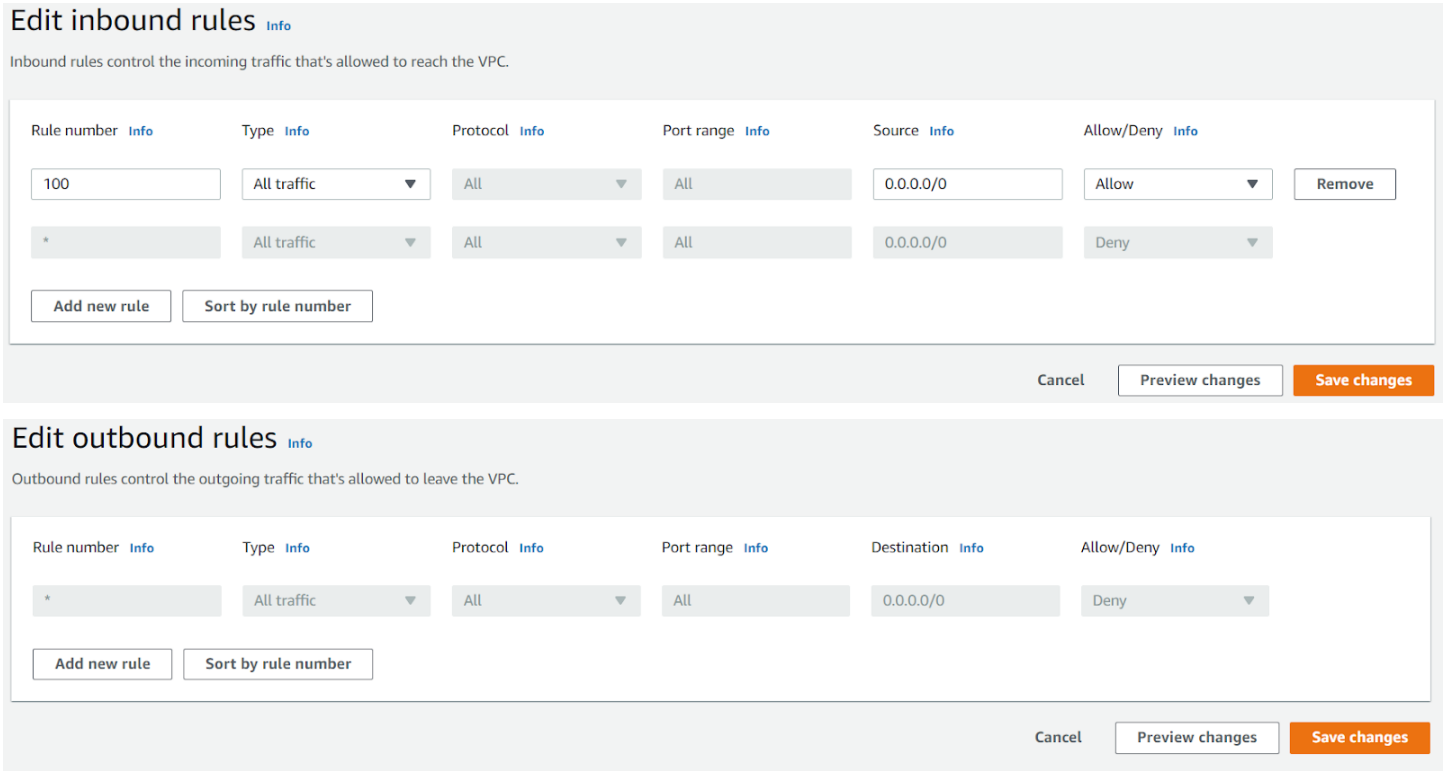


Network ACLs operate on the subnet layer, which means they protect your whole subnet rather than individual instances. Similar to security groups, traffic is managed through the use of rules. A network ACL rule consists of a rule number, traffic type, protocol, port range, source of the traffic for inbound rules or destination of the traffic for outbound rules, and an allow or deny setting.

In network ACL, rules are evaluated starting with the lowest numbered rule. As soon as a rule matches traffic, it's applied regardless of any higher-numbered rule that might contradict it. And unlike security groups, you can create allow rules and deny permissions in NACL for both inbound and outbound rules. Perhaps you want to allow public users to have HTTP access to your subnet, except for a few IP addresses that you found to be malicious. You can create an inbound HTTP allow rule that allows 0.0.0.0/0 and create another inbound HTTP deny rule that blocks these specific IPs. If no rule matches a traffic request or response then it is automatically denied. Network ACLs are also stateless, so sources and destinations need to be allowed on both inbound and outbound for them to freely communicate with the resources in your subnet.

Every VPC comes with a default network ACL, which allows all inbound and outbound traffic. You can create your own custom network ACL and associate it with a subnet. By default, each custom network ACL denies all inbound and outbound traffic until you add rules. Note that every subnet must be associated with a network ACL. If you don't explicitly associate a subnet with a network ACL, the subnet is automatically associated with the default network ACL. A network ACL can be associated with multiple subnets. However, a subnet can be associated with only one network ACL at a time.

One last thing to note is, for subnets that handle public network connections, you might encounter some issues if you do not add an allow rule for your ephemeral ports. The range varies depending on the client's operating system. A NAT gateway uses ports 1024-65535 for example.



References:
https://docs.aws.amazon.com/vpc/latest/userguide/VPC_SecurityGroups.html
https://docs.aws.amazon.com/vpc/latest/userguide/vpc-network-acls.html
https://tutorialsdojo.com/security-group-vs-nacl/

EC2 Placement Groups

Launching EC2 instances in a placement group influences how they are placed in underlying AWS hardware. Depending on your type of workload, you can create a placement group using one of the following placement strategies:

- **Cluster** – your instances are placed close together inside an Availability Zone. A cluster placement group can span peered VPCs that belong in the same AWS Region. This strategy enables workloads to achieve low-latency, high network throughput network performance.
- **Partition** – spreads your instances across logical partitions, called partitions, such that groups of instances in one partition do not share the underlying hardware with groups of instances in different partitions. A partition placement group can have partitions in multiple Availability Zones in the same Region, with a maximum of seven partitions per AZ. This strategy reduces the likelihood of correlated hardware failures for your application.
- **Spread** – strictly places each of your instances across distinct underlying hardware racks to reduce correlated failures. Each rack has its own network and power source. A spread placement group can have partitions in multiple Availability Zones in the same Region, with a maximum of seven running EC2 instances per AZ per group.

If you try to add more instances to your placement group after you create it, or if you try to launch more than one instance type in the placement group, you might get an insufficient capacity error. If you stop an instance in a placement group and then start it again, it still runs in the placement group. However, the start fails if there isn't enough capacity for the instance. To remedy the capacity issue, simply retry the launch until you succeed.

Some limitations you need to remember:

- You can't merge placement groups.
- An instance cannot span multiple placement groups.
- You cannot launch Dedicated Hosts in placement groups.
- A cluster placement group can't span multiple Availability Zones.

References:
https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/placement-groups.html
https://tutorialsdojo.com/amazon-elastic-compute-cloud-amazon-ec2/

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

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.
- 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.
- Natively supports the following languages:
    - Node.js
    - Java
    - C#
    - Go
    - Python
    - Ruby
    - PowerShell
- You can also provide your own custom runtime.

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.
- **Event source** – an AWS service or a custom service that triggers your function and executes its logic.
- **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.

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.

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.

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.

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 Batch

- Enables you to run batch computing workloads on the AWS Cloud.
- It is a regional service that simplifies running batch jobs across multiple AZs within a region.

Features

- Batch manages compute environments and job queues, allowing you to easily run thousands of jobs of any scale using EC2 and EC2 Spot.
- Batch chooses where to run the jobs, launching additional AWS capacity if needed.
- Batch carefully monitors the progress of your jobs. When capacity is no longer needed, it will be removed.
- Batch provides the ability to submit jobs that are part of a pipeline or workflow, enabling you to express any interdependencies that exist between them as you submit jobs.

Security

- Take advantage of IAM policies, roles, and permissions.

Monitoring

- You can use the **AWS Batch event stream for CloudWatch Events** to receive near real-time notifications regarding the current state of jobs that have been submitted to your job queues.
- Events from the AWS Batch event stream are ensured to be delivered at least one time.
- CloudTrail captures all API calls for AWS Batch as events.

Pricing

- There is no additional charge for AWS Batch. You pay for resources you create to store and run your application.

Sources:
https://docs.aws.amazon.com/batch/latest/userguide/
https://aws.amazon.com/batch/features/
https://aws.amazon.com/batch/pricing/
https://aws.amazon.com/batch/faqs/

Amazon Elastic Container Registry (ECR)

- A managed AWS Docker registry service.
- Amazon ECR is a regional service.

Features

- ECR supports Docker Registry HTTP API V2 allowing you to use Docker CLI commands or your preferred Docker tools in maintaining your existing development workflow.
- ECR stores both the containers you create and any container software you buy through AWS Marketplace.
- ECR stores your container images in Amazon S3.
- ECR supports the ability to define and organize repositories in your registry using namespaces.
- You can transfer your container images to and from Amazon ECR via HTTPS.

Pricing

- You pay only for the amount of data you store in your repositories and data transferred to the Internet.

Sources:
https://docs.aws.amazon.com/AmazonECR/latest/userguide/
https://aws.amazon.com/ecr/features/
https://aws.amazon.com/ecr/pricing/
https://aws.amazon.com/ecr/faqs/

AWS Savings Plan

- Savings Plan is a flexible pricing model that helps you save up cost on Amazon EC2, AWS Fargate, and AWS Lambda usage.
- You can purchase Savings Plans from any account, payer or linked.
- By default, the benefit provided by Savings Plans is applicable to usage across all accounts within an AWS Organization/consolidated billing family. You can also choose to restrict the benefit of Savings Plans to only the account that purchased them.
- Similar to Reserved Instances, you have All Upfront, Partial upfront, or No upfront payment options.

Plan Types

- **Compute Savings Plans** - provide the most flexibility and prices that are up to 66 percent off of On-Demand rates. These plans automatically apply to your EC2 instance usage, regardless of instance family (example, M5, C5, etc.), instance sizes (example, c5.large, c5.xlarge, etc.), Region (for example, us-east-1, us-east-2, etc.), operating system (for example, Windows, Linux, etc.), or tenancy (Dedicated, default, dedicated host). They also apply to your Fargate and Lambda usage.
    - You can move a workload between different instance families, shift your usage between different regions, or migrate your application from Amazon EC2 to Amazon ECS using Fargate at any time and continue to receive the discounted rate provided by your Savings Plan.
- **EC2 Instance Savings Plans** - provide savings up to 72 percent off On-Demand, in exchange for a commitment to a specific instance family in a chosen AWS Region (for example, M5 in N. Virginia US-East-1). These plans automatically apply to usage regardless of instance size, OS, and tenancy within the specified family in a region.
    - You can change your instance size within the instance family (example, from c5.xlarge to c5.2xlarge) or the operating system (example, from Windows to Linux), or move from Dedicated tenancy to Default and continue to receive the discounted rate provided by your Savings Plan.

Savings Plan vs RIs

	Compute Savings Plans	EC2 Instance Savings Plans	Convertible RIs	Standard RIs
Savings over On-Demand	Up to 66 percent	Up to 72 percent	Up to 66 percent	Up to 72 percent
Automatically applies pricing to any instance family	✓	–	–	–
Automatically applies pricing to any instance size	✓	✓	Regional only	Regional only
Automatically applies pricing to any tenancy or OS	✓	✓	–	–
Automatically applies to Amazon ECS using Fargate and Lambda	✓	–	–	–
Automatically applies pricing across AWS Regions	✓	–	–	–
Term length options of 1 or 3 years	✓	✓	✓	✓

Monitoring

- The **Savings Plans Inventory** page shows a detailed overview of the Savings Plans you own.
- If you're a user in a linked account of AWS Organizations, you can view the Savings Plans owned by your specific linked account.
- If you're a user in the payer account in AWS Organizations, you can view Savings Plans owned only by the payer account, or you can view Savings Plans owned by all accounts in AWS Organizations.
- You can use AWS Budgets to set budgets for your Savings Plan utilization, coverage, and costs.

Sources:
https://aws.amazon.com/savingsplans/
https://docs.aws.amazon.com/savingsplans/latest/userguide/what-is-savings-plans.html
https://aws.amazon.com/savingsplans/faq/

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.
- You can't delete an S3 bucket using the Amazon S3 console if the bucket contains 100,000 or more objects. You can't delete an S3 bucket using the AWS CLI if versioning is enabled.

Storage Classes

- Storage Classes for Frequently Accessed Objects
  - S3 **STANDARD** for **general-purpose** storage of frequently accessed data.
- 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.
    - There are no retrieval fees in S3 Intelligent-Tiering.
- S3 GLACIER
    - For long-term **archive**
    - S3 Glacier provides the following storage classes: S3 Glacier Instant Retrieval, S3 Glacier Flexible Retrieval, and S3 Glacier Deep 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.

Bucket Configurations

Subresource	Description
location	Specify the AWS Region where you want S3 to create the bucket.
policy and *ACL* (access control list)	All your resources are private by default. Use bucket policy and ACL options to grant and manage bucket-level permissions.
website	You can configure your bucket for static website hosting.
logging	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.
tagging	S3 provides the *tagging* subresource to store and manage tags on a bucket. AWS generates a cost allocation report with usage and costs aggregated by your tags.

Objects

- Each S3 object has **data**, a **key**, and **metadata**.
- Tagging
    - You can associate up to 10 tags with an object. Tags associated with an object must have unique tag keys.

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	You pay for all bandwidth into and out of S3, except for the following: ● Data transferred in from the internet ● Data transferred out to an Amazon EC2 instance, when the instance is in the same AWS Region as the S3 bucket ● Data transferred out to Amazon CloudFront You also pay a fee for any data transferred using Amazon S3 Transfer Acceleration.

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.
  - 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.

Sources:
https://docs.aws.amazon.com/AmazonS3/latest/dev/Welcome.html
https://aws.amazon.com/s3/faqs/

Amazon S3 Glacier

- **Long-term archival** solution optimized for infrequently used data, or "cold data."
- 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 an average annual durability of 99.999999999% 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*

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.
- 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

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.

Features

- Different types of storage options: **General Purpose SSD (gp2,gp3)**, **Provisioned IOPS SSD (io1,io2)**, **Throughput Optimized HDD (st1)**, and **Cold HDD (sc1)** volumes up to **16 TiB** in size **or 64TiB** for io2 Block Express.
- You can mount multiple volumes on the same instance, and you can mount a Provisioned IOPS volume to multiple instances at a time using Amazon EBS Multi-Attach.
- Enable Multi-Attach on EBS Provisioned IOPS io1 volumes to allow a single volume to be concurrently attached to up to sixteen AWS Nitro System-based Amazon EC2 instances within the same AZ.
- 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.
- EBS fast snapshot restore allows you to create a volume from a snapshot that is fully initialized. This removes the latency of I/O operations on the block when accessed for the first time.

Types of EBS Volumes

Volume Name	General Purpose SSD		Provisioned IOPS SSD	
Volume type	gp3	gp2	io2	io1
Description	General Purpose SSD volume that balances price performance for a wide variety of transactional workloads	General Purpose SSD volume that balances price performance for a wide variety of transactional workloads	High performance SSD volume designed for **business-critical** latency-sensitive applications	High performance SSD volume designed for latency-sensitive transactional workloads
Use Cases	virtual desktops, medium sized single instance databases such as MSFT SQL Server and Oracle DB, low-latency interactive apps, dev & test, boot volumes	Boot volumes, low-latency interactive apps, dev & test	Workloads that require sub-millisecond latency, and sustained IOPS performance or more than 64,000 IOPS or 1,000 MiB/s of throughput	Workloads that require sustained IOPS performance or more than 16,000 IOPS and I/O-intensive database workloads
Volume Size	1 GB – 16 TB	1 GB – 16 TB	4 GB – 16 TB / 64 TB for io2 block express	4 GB – 16 TB
Durability	99.8% - 99.9% durability	99.8% - 99.9% durability	99.999%	99.8% - 99.9%
Max IOPS / Volume	16,000	16,000	64,000 / 256,000 for io2 block express	64,000
Max Throughput / Volume	1000 MB/s	250 MB/s	1,000 MB/s / 4,000 MiB/s for io2 block express	1,000 MB/s
Max IOPS / Instance	260,000	260,000	160,000 / 260,000 MiB/s for io2 block express	260,000
Max IOPS / GB	N/A	N/A	500 IOPS/GB / 1,000 IOPS/GB for io2 block express	50 IOPS/GB
Max Throughput / Instance	7,500 MB/s	7,500 MB/s	4,750 MB/s / 7,500 MB/s for io2 block express	7,500 MB/s

Latency	single digit millisecond	single digit millisecond	single digit millisecond	single digit millisecond
Multi-Attach	No	No	Yes	Yes

Volume Name	Throughput Optimized HDD	Cold HDD
Volume type	st1	sc1
Description	Low cost HDD volume designed for frequently accessed, throughput-intensive workloads	Throughput-oriented storage for data that is infrequently accessed Scenarios where the lowest storage cost is important
Use Cases	Big data, data warehouses, log processing	Colder data requiring fewer scans per day
Volume Size	125 GB – 16 TB	125 GB – 16 TB
Durability	99.8% - 99.9% durability	99.8% - 99.9% durability
Max IOPS / Volume	500	250
Max Throughput / Volume	500 MB/s	250 MB/s
Max IOPS / Instance	260,000	260,000
Max IOPS / GB	N/A	N/A
Max Throughput / Instance	7,500 MB/s	7,500 MB/s
Multi-Attach	No	No

FEATURES	SSD Solid State Drive	HDD Hard Disk Drive
Best for workloads with:	***small, random*** I/O operations	***large, sequential*** I/O operations
Can be used as a bootable volume?	Yes	No
Suitable Use Cases	- Best for **transactional workloads** - Critical business applications that require sustained IOPS performance - Large database workloads such as MongoDB, Oracle, Microsoft SQL Server and many others…	- Best for ***large streaming workloads*** requiring consistent, fast throughput at a low price - Big data, Data warehouses, Log processing - Throughput-oriented storage for large volumes of data that is ***infrequently*** accessed
Cost	moderate / high 	low 
Dominant Performance Attribute	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 host 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

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.
- 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.
- 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.
- 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.

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 500 Mbps and 60,000 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.

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.
- 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.

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.
- 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 vs S3

- Performance Comparison

	Amazon EFS	Amazon EBS Provisioned IOPS
Per-operation latency	Low, consistent latency.	Lowest, consistent latency.
Throughput scale	Multiple GBs per second	Single GB per second

	Amazon EFS	Amazon S3
Per-operation latency	Low, consistent latency.	Low, for mixed request types, and integration with CloudFront.
Throughput scale	Multiple GBs per second	Multiple GBs per second

- Storage Comparison

	Amazon EFS	Amazon EBS Provisioned IOPS
Availability and durability	Data are stored redundantly across multiple AZs.	Data are 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.

	Amazon EFS	Amazon S3
Availability and durability	Data are stored redundantly across multiple AZs.	Stored redundantly across multiple AZs.

Access	Up to thousands of EC2 instances from multiple AZs can connect concurrently to a file system.	One to millions of connections over the web.
Use cases	Big data and analytics, media processing workflows, content management, web serving, and home directories.	Web serving and content management, media and entertainment, backups, big data analytics, data lake.

- We have more comparisons for EFS, S3, and EBS in our **Comparison of AWS Services** section.

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/
https://aws.amazon.com/efs/when-to-choose-efs/

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.
    - **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.
        - **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.
    - **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%.

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.
- Authentication and access control with IAM.

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/

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

Amazon Aurora

- A fully managed relational database engine that's compatible with **MySQL** and **PostgreSQL**.
- Aurora includes a high-performance storage subsystem. The underlying storage grows automatically as needed, up to 128 terabytes.
- Aurora will keep your database up-to-date with the latest patches.
- Aurora is fault-tolerant and self-healing.
- 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 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.
- High Availability and Fault Tolerance
    - When you create Aurora Replicas across Availability Zones, RDS automatically provisions and maintains them synchronously.
    - 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.
    - Aurora has a Backtrack feature that rewinds or restores the DB cluster to the time you specify. However, take note that the Amazon Aurora Backtrack feature is not a total replacement for fully backing up your DB cluster since the limit for a backtrack window is only 72 hours.
- 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
    - 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.

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

- 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.

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**.
- Basic building block of RDS is the **DB instance**, which is an isolated database environment in the cloud.
- Each DB instance runs a **DB engine**.
- 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 Instance:
    - 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.

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.
- 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 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:

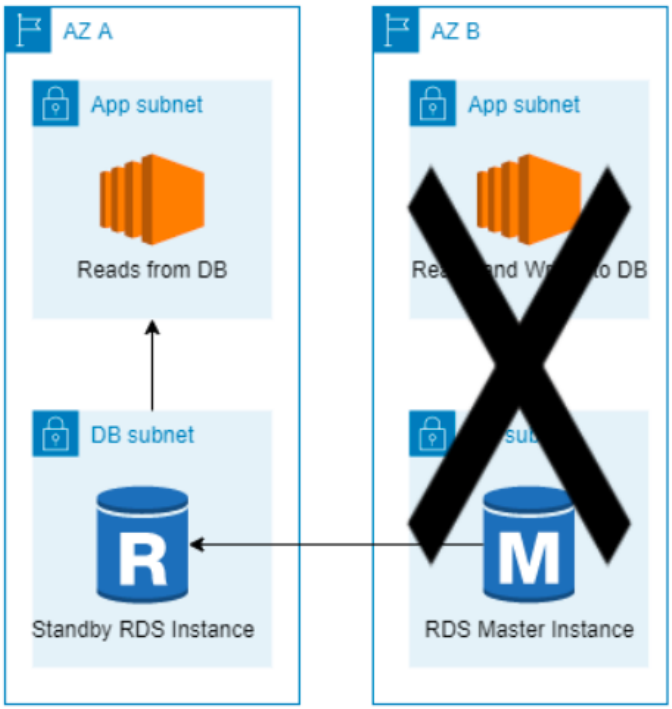
DB Instance	Accessed By
In a VPC	An EC2 Instance in the Same VPC
	An EC2 Instance in a Different VPC
	An EC2 Instance Not in a VPC
	A Client Application Through the Internet
Not in a VPC	An EC2 Instance in a VPC
	An EC2 Instance Not in a VPC
	A Client Application Through the Internet

Tagging

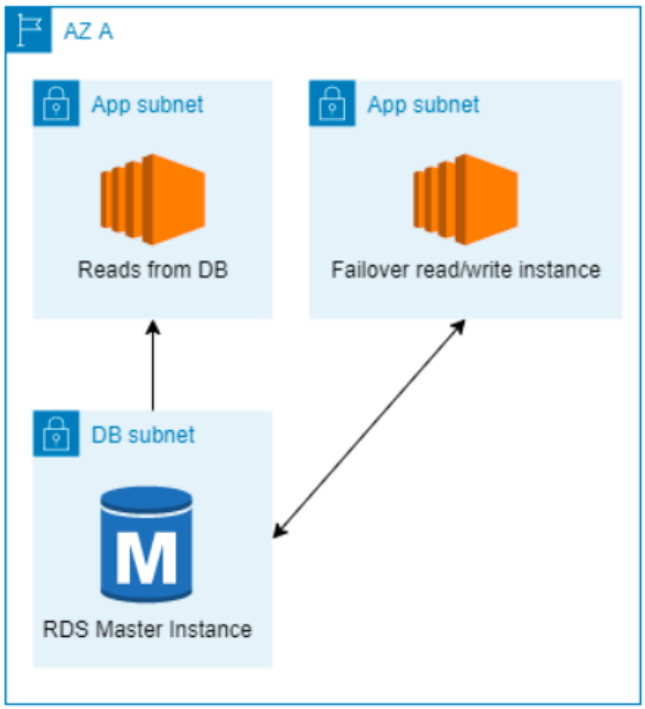
- An RDS tag is a **name-value pair** that you define and associate with an RDS resource. The name is referred to as the key. Supplying a value for the key is optional.
- All Amazon RDS resources can be tagged.
- Use tags to organize your AWS bill to reflect your own cost structure.
- A *tag set* can contain as many as 50 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. This is available for both Standard and Enterprise editions.
- 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**



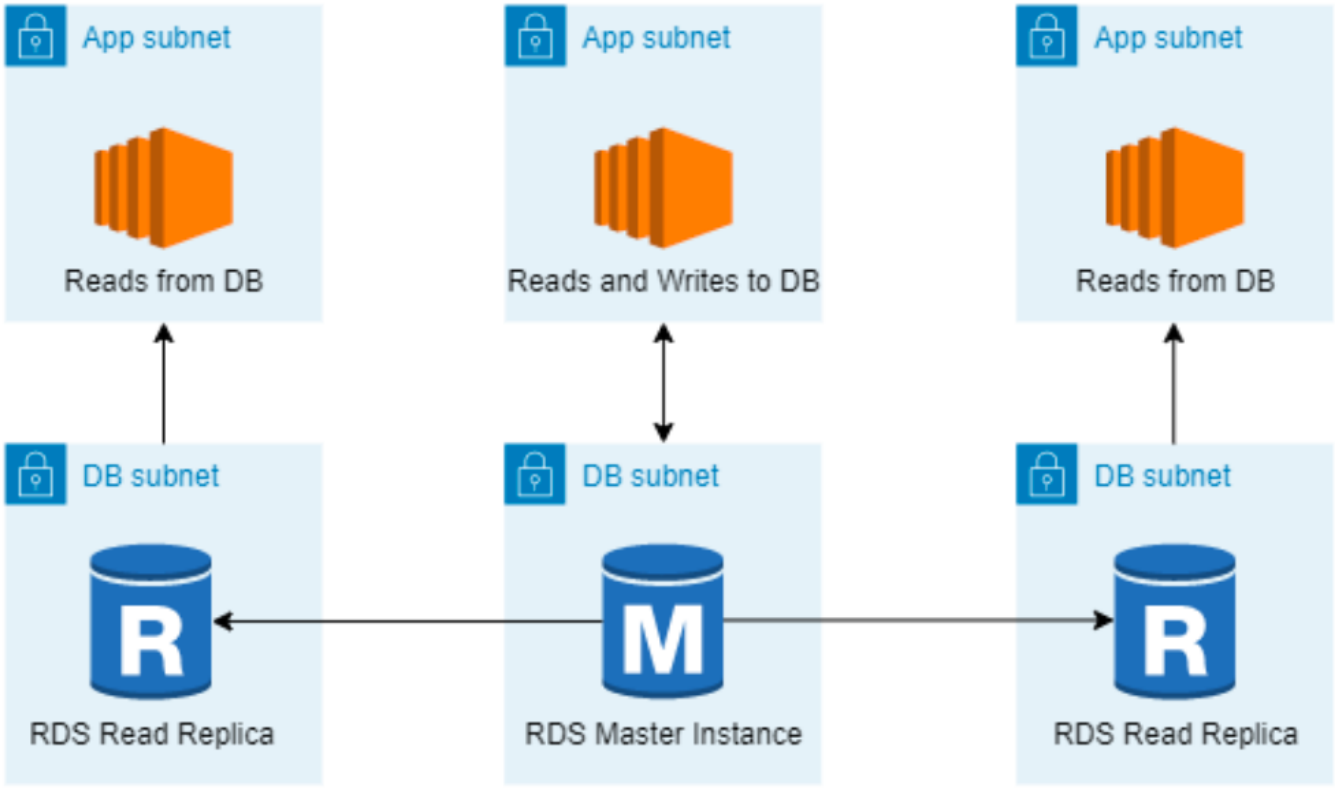
Standby DB instance in another subnet. AZ B experiences outage.



Read replica promoted to Master instance. Spawn a new read/write app in AZ A. Reroute traffic using failover policies.

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.



Multi-AZ Deployments vs Read Replicas

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.
- The first snapshot of a DB instance contains the data for the full DB instance. Subsequent snapshots of the same DB instance are incremental.

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
- 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.
- 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.
- 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.

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 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.
- 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.
- Transactions provide atomicity, consistency, isolation, and durability (ACID) in DynamoDB, helping you to maintain data correctness in your applications.

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.
- Maximum number of tags per resource: 50

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.
    - You can restore backups as new DynamoDB tables in other regions.
- 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.

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.
  - 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**

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.

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 **only one** 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.

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** environment in the AWS Cloud.
- ElastiCache works with both the **Redis** and **Memcached** engines.
- Elasticache can be used for storing session state.

	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

- 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.

	Redis (cluster mode enabled)	Redis (cluster mode disabled)	Memcached
Data Types	string, sets, sorted sets, lists, hashes, bitmaps, hyperloglog, geospatial indexes	string, sets, sorted sets, lists, hashes, bitmaps, hyperloglog, geospatial indexes	string, objects (like databases)
Data Partitioning (distribute your data among multiple nodes)	Supported	Unsupported	Supported
Modifiable cluster	Only versions 3.2.10 and later	Yes	Yes
Online resharding	Only versions 3.2.10 and later	No	No
Encryption	3.2.6, 4.0.10 and later	3.2.6, 4.0.10 and later	Unsupported
Sub-millisecond latency	Yes	Yes	Yes
FedRAMP, PCI DSS and HIPAA compliant	3.2.6, 4.0.10 and later	3.2.6, 4.0.10 and later	No
Multi-threaded (make use of multiple processing cores	No	No	Yes
Node type upgrading	No	Yes	No
Engine upgrading	Yes		
Cluster replication (create multiple copies of a primary cluster)	Supported	Supported	Unsupported
Multi-AZ for automatic failover	Required	Optional	Unsupported
Transactions (execute a group of commands as an isolated and atomic operation)	Supported	Supported	Unsupported
Pub/Sub capability	Yes	Yes	No
Backup and restore (keep your data on disk with a point in time snapshot)	Supported	Supported	Unsupported
Lua Scripting (execute transactional Lua scripts)	Supported	Supported	Unsupported
Use Case Tutorials Dojo	• You need to partition your data across two to 90 node groups (clustered mode only). • You need geospatial indexing (clustered mode or non-clustered mode). • You don't need to support multiple databases. • Plus features of non-clustered mode	• You need complex data types, such as strings, hashes, lists, sets, sorted sets, and bitmaps. • You need to sort or rank in-memory datasets. • You need persistence of your key store. • You need to replicate your data from the primary to one or more read replicas for read intensive applications. • You need automatic failover if your primary node fails. • You need pub/sub capabilities. • You need backup and restore capabilities. • You need to support multiple databases.	• You need the simplest model possible. • You need to run large nodes with multiple cores or threads. • You need the ability to scale out and in, adding and removing nodes as demand on your system increases and decreases. • You need to cache objects, such as a database. • Needs Auto Discovery to simplify the way an application connects to a cluster.

- 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 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.
- 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 a per-second billing 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.
- Allows creating, deploying, and managing a RESTful API to expose backend HTTP endpoints, Lambda functions, or other AWS services.
- Together with Lambda, API Gateway forms the app-facing part of the AWS serverless infrastructure.
- 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.
    - Amazon API Gateway offers general availability of HTTP APIs, which gives you the ability to route requests to private ELBs AWS AppConfig, Amazon EventBridge, Amazon Kinesis Data Streams, Amazon SQS, AWS Step Functions and IP-based services registered in AWS CloudMap such as ECS tasks. Previously, HTTP APIs enabled customers to only build APIs for their serverless applications or to proxy requests to HTTP endpoints.
    - You can create data mapping definitions from an HTTP API's method request data (e.g. path parameters, query string, and headers) to the corresponding integration request parameters and from the integration response data (e.g. headers) to the HTTP API method response parameters.
    - Use wildcard custom domain names (*.example.com) to create multiple URLs that route to one API Gateway HTTP API.
    - You can configure your custom domain name to route requests to different APIs. Using multi-level base path mappings, you can implement path-based API versioning and migrate API traffic between APIs according to request paths with many segments.
- 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.
- 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
- 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.
- 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 a mediastore container or a media package 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.
- 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.

- 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.
- Monitoring
    - 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.
- 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.

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 Elastic Load Balancing

- Distributes incoming application or network traffic across multiple targets, such as **EC2 instances**, **containers (ECS)**, **Lambda functions,** and **IP addresses**, in multiple Availability Zones.

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.

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.
    - 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.
        - 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.
    - 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.
- **Gateway Load Balancer**
    - Enables you to deploy, scale, and manage virtual appliances, such as firewalls, intrusion detection and prevention systems, and deep packet inspection systems.
    - Operates at the third layer of the Open Systems Interconnection (OSI) model, the network layer. It listens for all IP packets across all ports and forwards traffic to the target group that's specified in the listener rule.
    - Gateway Load Balancers use Gateway Load Balancer endpoints to securely exchange traffic across VPC boundaries. A Gateway Load Balancer endpoint is a VPC endpoint that provides private connectivity between virtual appliances in the service provider VPC and application servers in the service consumer VPC.
    - Traffic to and from a Gateway Load Balancer endpoint is configured using route tables.
- **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.
    - 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 store 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.

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	*listener*
Outbound	
Destination	**Port Range**
instance security group	*instance listener*
instance security group	*health check*

Summary of Features

Feature	Application Load Balancer	Network Load Balancer	Gateway Load Balancer
Protocols	HTTP, HTTPS, gRPC	TCP, UDP, TLS	IP
Platforms	VPC	VPC	VPC
Health checks	HTTP, HTTPS, gRPC	TCP, HTTP, HTTPS	TCP, HTTP, HTTPS
Cloudwatch Metrics	✓	✓	✓
Logging	✓	✓	✓
Zonal Failover	✓	✓	✓
Connection Draining (deregistration delay)	✓	✓	✓
Load Balancing to multiple ports on the same instance	✓	✓	✓
IP addresses as targets	✓	✓ (TCP, TLS)	✓
Load balancer deletion protection	✓	✓	✓
Configuration idle connection timeout	✓		
Cross-zone load balancing	✓	✓	✓
Sticky sessions	✓	✓	✓
Static IP		✓	
Elastic IP address		✓	
Preserve Source IP address	✓	✓	✓
Resource-based IAM permissions	✓	✓	✓
Tag-based IAM permissions	✓	✓	✓
Slow start	✓		
Web sockets	✓	✓	✓
PrivateLink Support		✓ (TCP, TLS)	✓ (GWLBE)
Source IP address CIDR-based routing	✓		

Feature	Application Load Balancer	Network Load Balancer	Gateway Load Balancer
Layer 7			
Path-based routing	✓		
Host-based routing	✓		
Native HTTP/2	✓		
Redirects	✓		
Fixed response	✓		
Lambda functions as targets	✓		
HTTP header-based routing	✓		
HTTP method-based routing	✓		
Query string parameter-based routing	✓		
Security			
SSL offloading	✓	✓	
Server Name Indication (SNI)	✓	✓	
Back-end server encryption	✓	✓	
User authentication	✓		
Session Resumption	✓	✓	
Terminates flow/proxy behavior	✓	✓	✓

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 Gateway Load Balancer is running and the number of Gateway Load Balancer Capacity Units (GLCU) used by Gateway 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

Know the following Concepts

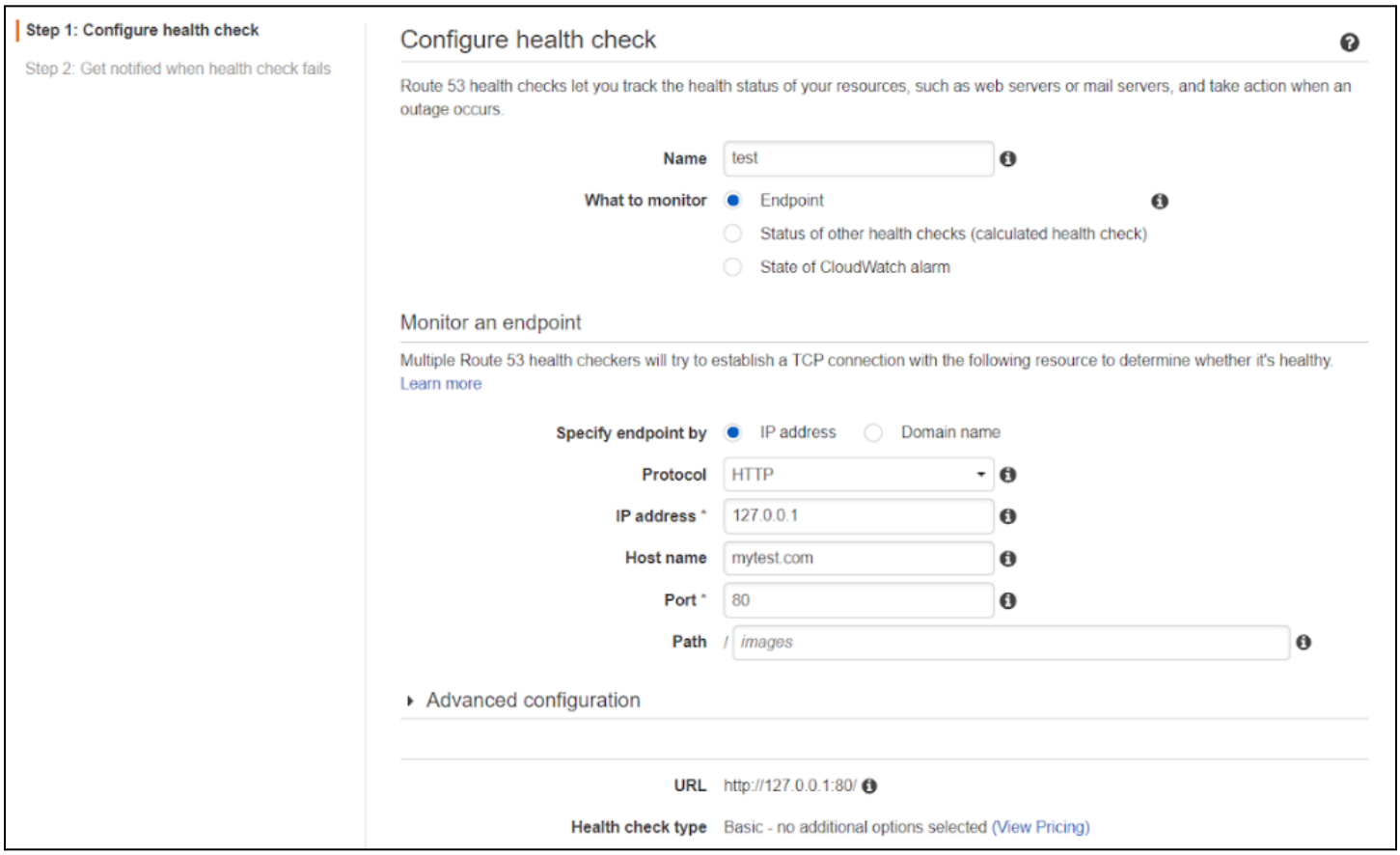
- Domain Registration Concepts - domain name, domain registrar, domain registry, domain reseller, top-level domain
- DNS Concepts
  - **Alias record** - a type of record that you can create to route traffic to AWS resources.
  - DNS query
  - DNS resolver
  - Domain Name System (DNS)
  - Private DNS
  - **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)

Records

- 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.

Route 53 Health Checks and DNS Failover



- 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
- 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.

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://docs.aws.amazon.com/Route53/latest/DeveloperGuide/Welcome.html
https://aws.amazon.com/route53/features/
https://aws.amazon.com/route53/pricing/

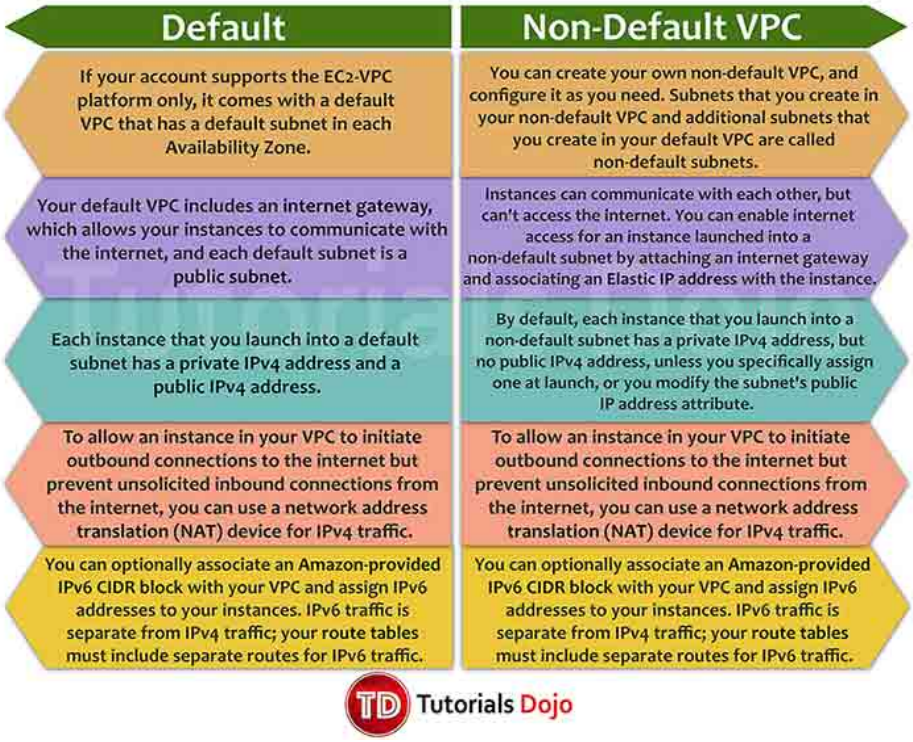
Amazon VPC

- 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.

Default vs Non-Default VPC



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

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
- 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.

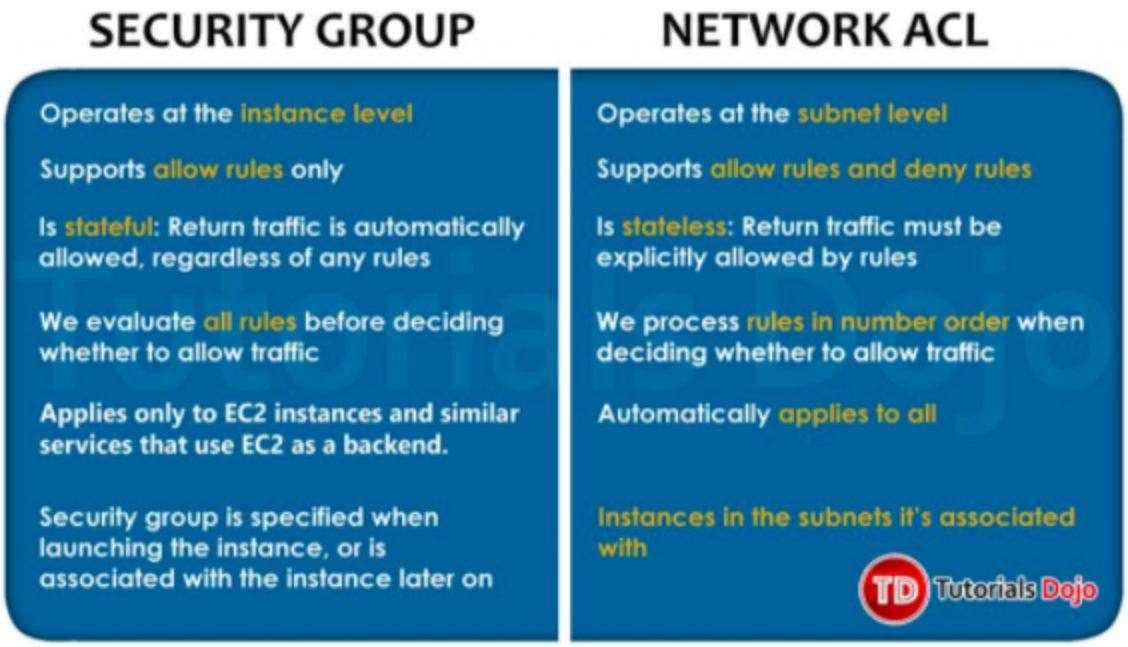
Subnet Routing

- 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.
    - For custom ACLs, you need to add a rule for ephemeral ports, usually with the range of 32768-65535. If you have a NAT Gateway, ELB or a Lambda function in a VPC, you need to enable 1024-65535 port range.
- Flow logs — capture information about the IP traffic going to and from network interfaces in your VPC that is published to CloudWatch Logs.

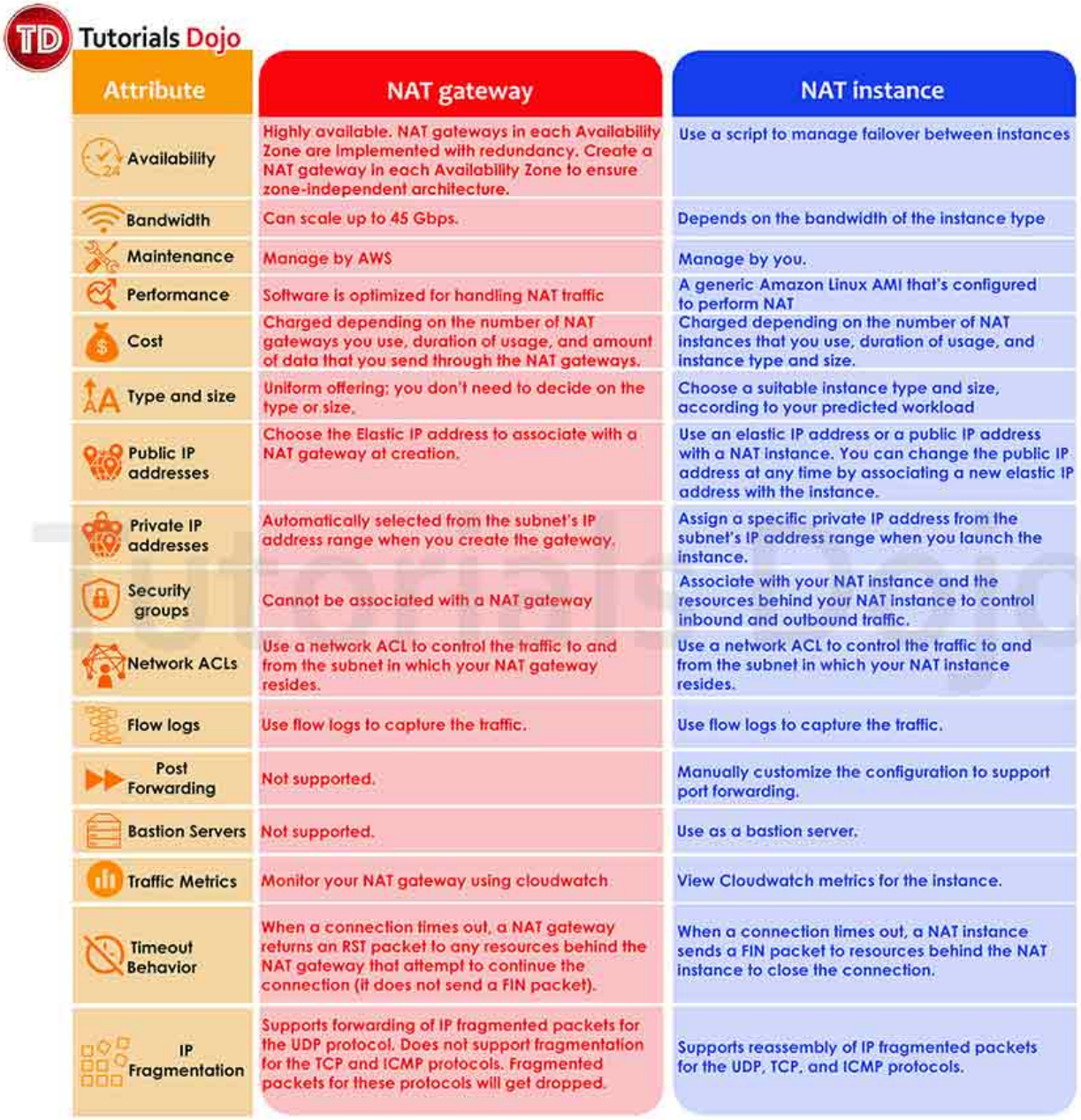


- Diagram of security groups and NACLs in a VPC

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.
- Internet Gateways
    - Allows communication between instances in your VPC and the internet.
    - Imposes no availability risks or bandwidth constraints on your network 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 Instance vs NAT Gateways



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.
Post 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	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.

- 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.
- 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.

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/

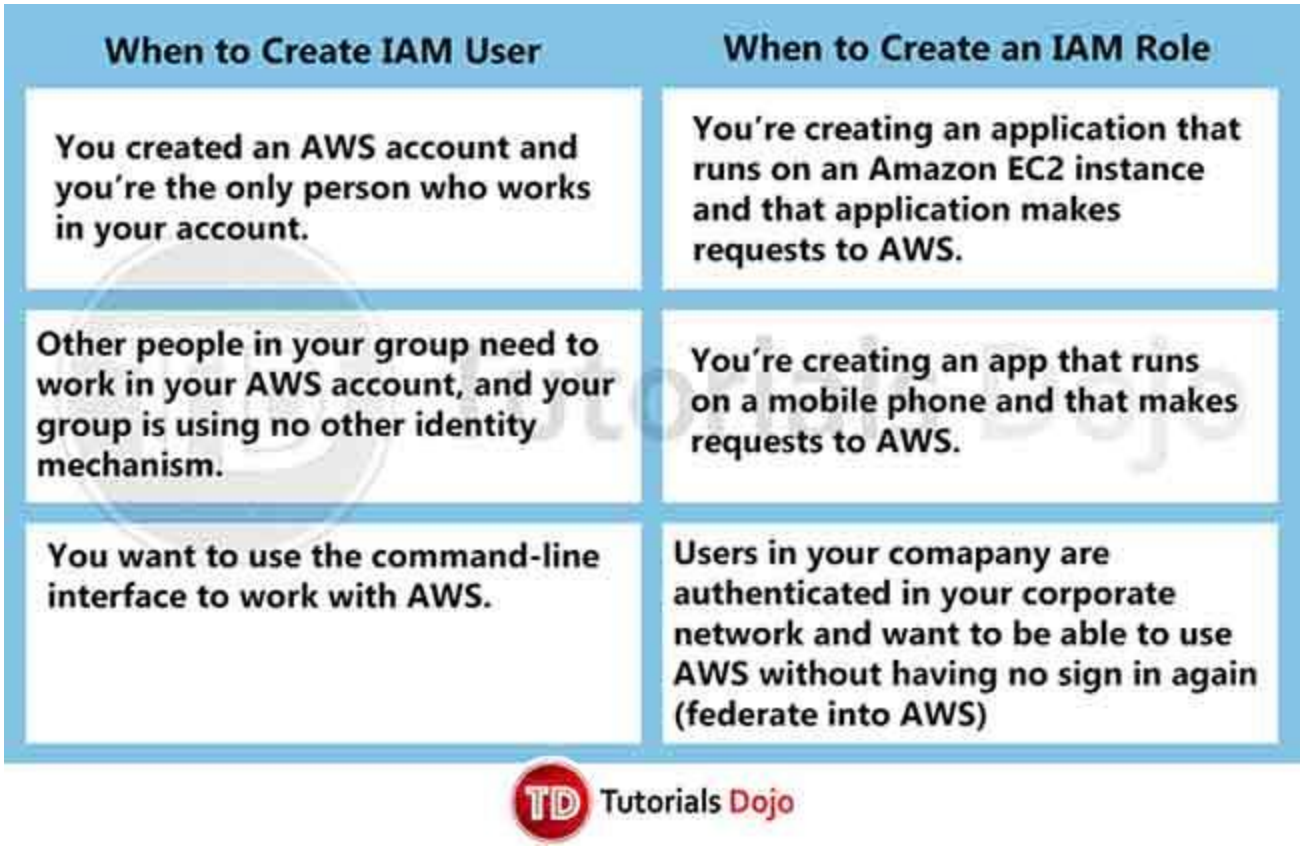
SECURITY AND IDENTITY

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 add two-factor authentication to your account and to individual users for extra security.
    - 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.
    - Your unique account sign-in page URL: 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.
    - You can generate and download a credential report that lists all users on your AWS account. The report also shows the status of passwords, access keys, and MFA devices.
- **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 username and password.
        - To authenticate from the API or AWS CLI, you must provide your access key and secret key.
    - **Authorization**
        - 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 “allow”.
    - **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.
    - Users or groups can have multiple policies attached to them that grant different permissions.



- **Policies**
    - Most permission policies are JSON policy documents.
    - 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.

  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.

- **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.
- Assume Role Options
  - AssumeRole - Returns a set of temporary security credentials that you can use to access AWS resources that you might not normally have access to. These temporary credentials consist of an access key ID, a secret access key, and a security token. Typically, you use *AssumeRole* within your account or for cross-account access.
    - You can include multi-factor authentication (MFA) information when you call *AssumeRole*. This is useful for cross-account scenarios to ensure that the user that assumes the role has been authenticated with an AWS MFA device.
  - AssumeRoleWithSAML - Returns a set of temporary security credentials for users who have been authenticated via a SAML authentication response. This allows you to link your enterprise identity store or directory to role-based AWS access without user-specific credentials or configuration.
  - AssumeRoleWithWebIdentity - Returns a set of temporary security credentials for users who have been authenticated in a mobile or web application with a web identity provider. Example providers include Amazon Cognito, Login with Amazon, Facebook, Google, or any OpenID Connect-compatible identity provider.
- STS Get Tokens
  - GetFederationToken - Returns a set of temporary security credentials (consisting of an access key ID, a secret access key, and a security token) for a federated user. You must call the GetFederationToken operation using the long-term security credentials of an IAM user. A typical use is in a proxy application that gets temporary security credentials on behalf of distributed applications inside a corporate network.
  - GetSessionToken - Returns a set of temporary credentials for an AWS account or IAM user. The credentials consist of an access key ID, a secret access key, and a security token. You must call

the GetSessionToken operation using the long-term security credentials of an IAM user. Typically, you use GetSessionToken if you want to use MFA to protect programmatic calls to specific AWS API operations.

- **IAM Access Analyzer**
  - Provides policy checks that help you proactively validate policies when creating them. These checks analyze your policy and report errors, warnings, and suggestions with actionable recommendations that help you set secure and functional permissions.
  - IAM Access Analyzer continuously monitors for new or updated resource policies and permissions granted for S3 buckets, KMS keys, SQS queues, IAM roles, Lambda functions, and Secrets Manager secrets.
- **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 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.

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.

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 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

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 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.

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 handles 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.

Other Additional Features

- You can scan Amazon S3 buckets across multiple AWS accounts, and perform scoping of scans by object prefix.
- An estimation of the costs of these job runs is sent to you for review before you run them.
- Once a job is submitted, findings are generated in the Amazon Macie console and sent out through Amazon EventBridge where sensitive data location information is included in the findings. This allows for identification of sensitive data within objects using detail such as line numbers, page numbers, record index, or column and row numbers.

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.

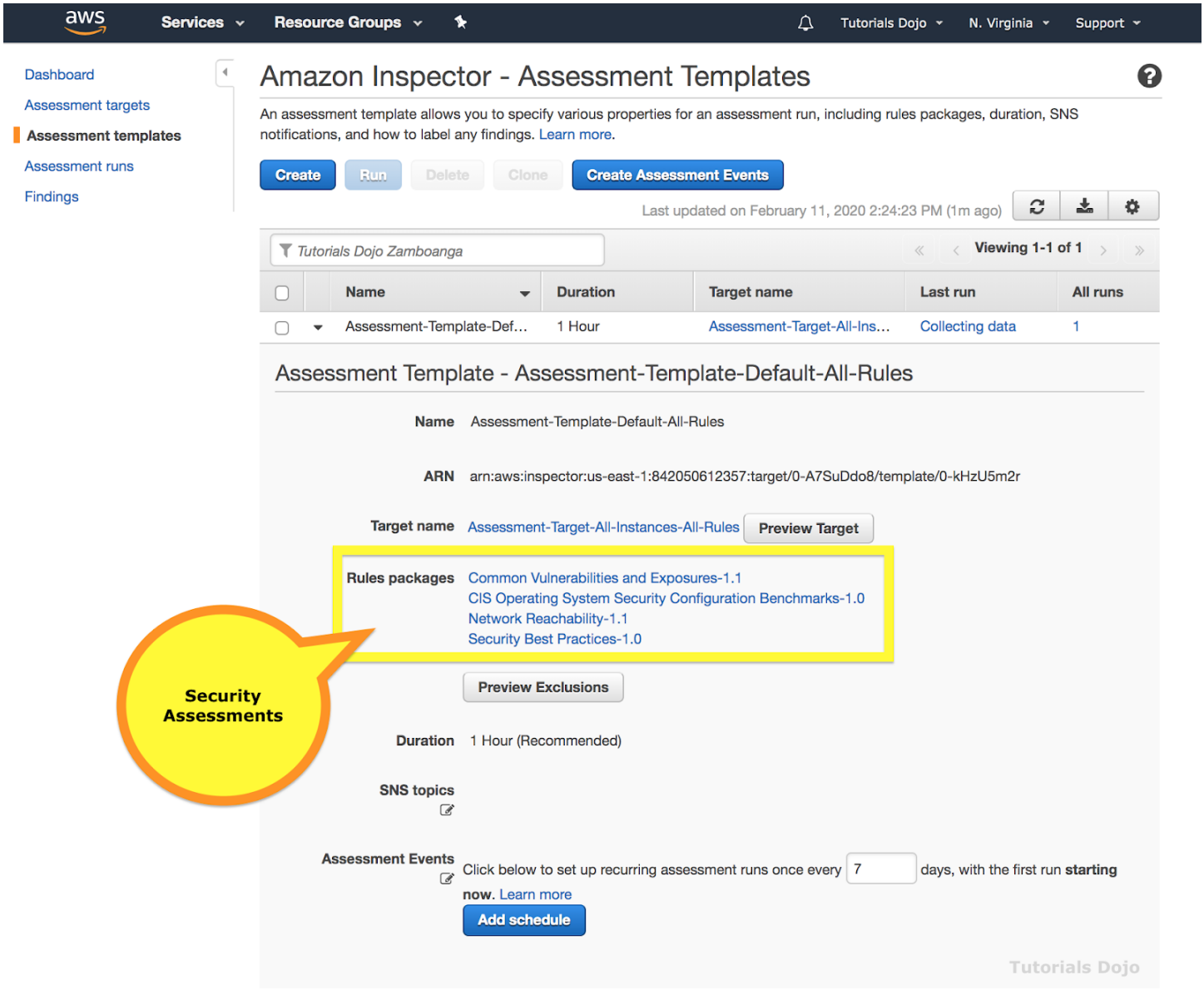
Sources:
https://aws.amazon.com/shield/features/
https://aws.amazon.com/shield/pricing/
https://aws.amazon.com/shield/faqs/

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.



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 **management account** is the AWS account you use to create your organization. You cannot change which account in your organization is the management account.
    - From the management 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 management 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 management account, that is part of an organization. A member account can belong to only one organization at a time. The management 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.

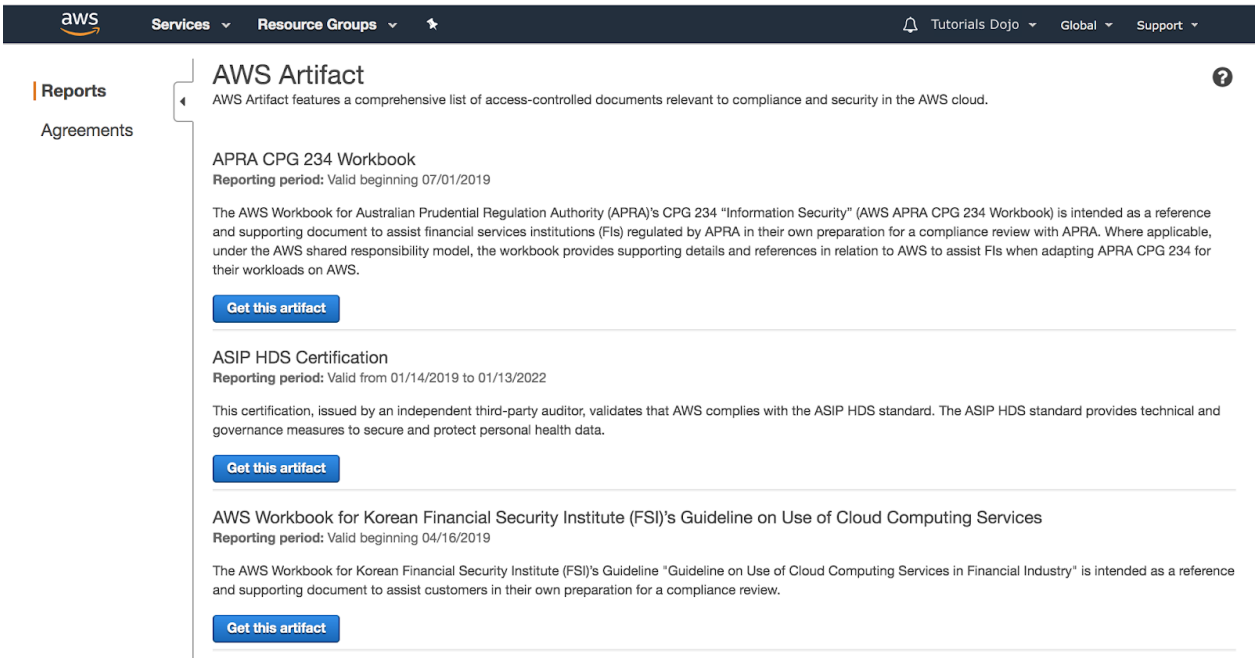
Pricing

- 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 management account and all member accounts. Only the management account in an organization can accept agreements in AWS Artifact Organization Agreements.
    - Management 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 management 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 management 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 management 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 management 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.
- You start by requesting one or more Snowball Edge Compute Optimized or Snowball Edge Storage Optimized devices in the AWS Management Console based on how much data you need to transfer and the compute power needed for local processing.
- Once a device arrives, you connect it to your local network and set the IP address either manually or automatically with DHCP. Then use the Snowball Edge client software, job manifest, and unlock code to verify the integrity of the Snowball Edge device or cluster, and unlock it for use.
- All logistics and shipping is done by Amazon, so when copying is complete and the device is ready to be returned, the E Ink shipping label will automatically update the return address. Once the device ships, you can receive tracking status via messages sent by Amazon SNS, generated texts and emails, or directly from the console.
- Snowball Edge devices are designed to be requested and used within a single AWS Region. The device may not be requested from one Region and returned to another.
- Snowball Edge encrypts all data with 256-bit encryption.

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/

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
- 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.
- 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 *groups* 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 *launch configuration* 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.
- 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.

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.

- 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.

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://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.
- CloudFormation automates the provisioning and updating of your infrastructure in a safe and controlled manner.

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.

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.
    - 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.
        - Insights events
            - Not logged by default
            - Insights events capture unusual activity in your AWS account. If you have Insights events enabled, CloudTrail detects unusual activity and logs this to S3.
            - Insights events provide relevant information, such as the associated API, incident time, and statistics, that help you understand and act on unusual activity.
            - Insights events are logged only when CloudTrail detects changes in your account's API usage that differ significantly from the account's typical usage patterns.

- **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, DynamoDB tables, 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/

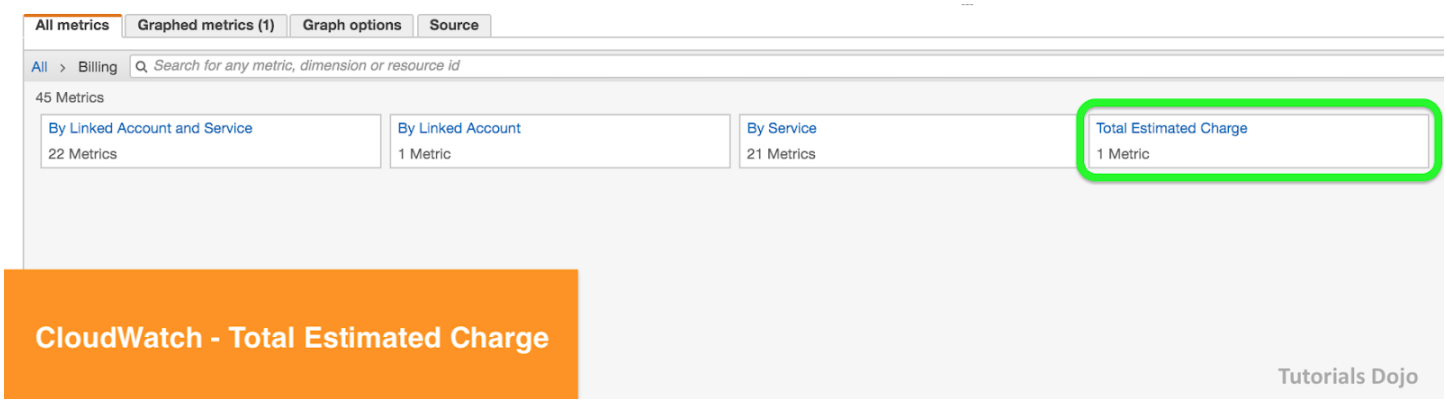
Amazon CloudWatch

- 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 is basically a metrics repository. An AWS service, such as Amazon EC2, puts metrics into the repository and you retrieve statistics based on those metrics. If you put your own custom metrics into the repository, you can retrieve statistics on these metrics as well.
- 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.
    - **Dimensions** - a name/value pair that uniquely identifies a metric.
        - You can assign up to 10 dimensions to a 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.

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
    - When an alarm is on a dashboard, it turns red when it is in the *ALARM* state.
    - 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.
    - You can also monitor your estimated AWS charges by using Amazon CloudWatch Alarms. However, take note that you can only track the estimated AWS charges in CloudWatch and not the actual utilization of your resources. Remember that you can only set coverage targets for your reserved EC2 instances in AWS Budgets or Cost Explorer, but not in CloudWatch.



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.

CloudWatch Events / Amazon EventBridge

- 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 Agent

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

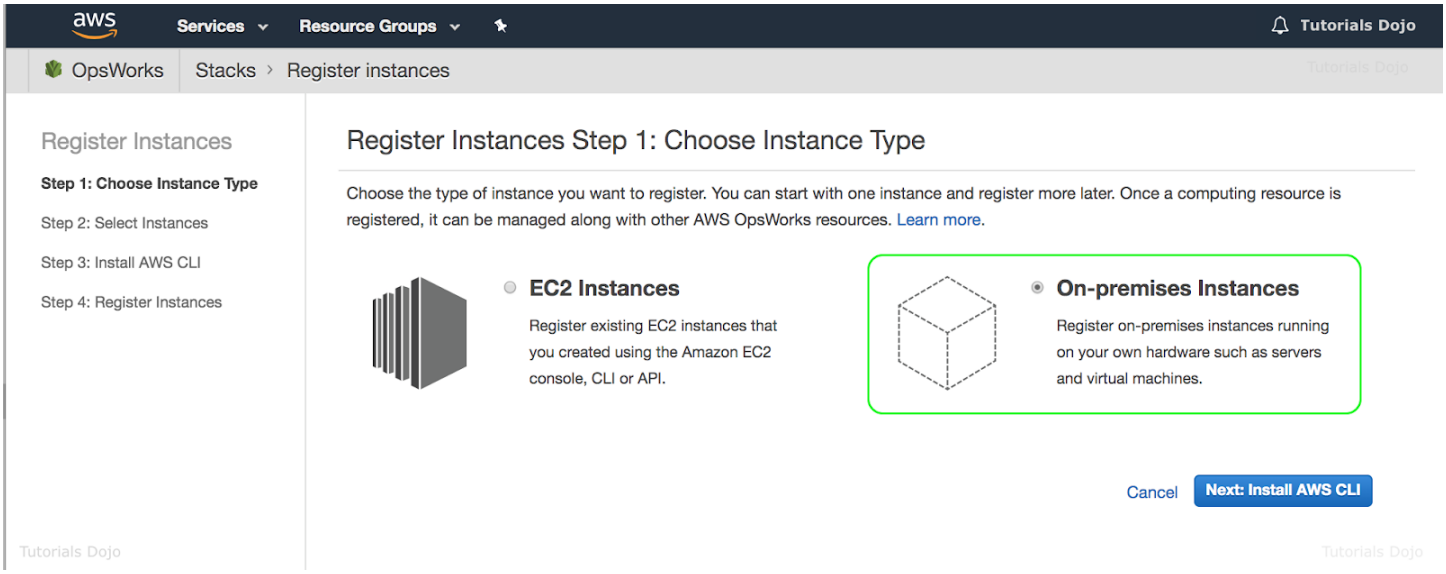
Pricing

- 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

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.
- With AWS OpsWorks, you can automate how nodes are configured, deployed, and managed, whether they are Amazon EC2 instances or on-premises devices:



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.
- **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.
- **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.

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 Management Console

- **Resource Groups**
  - A collection of AWS resources that are all in the same AWS region, and that match criteria provided in a query.
  - Resource groups make it easier to manage and automate tasks on large numbers of resources at one time.
  - Two types of queries on which you can build a group:
    - Tag-based
    - AWS CloudFormation stack-based
- **Tag Editor**
  - Tags are words or phrases that act as metadata for identifying and organizing your AWS resources. The tag limit varies with the resource, but most can have up to 50 tags.
  - You can sort and filter the results of your tag search to find the tags and resources that you need to work with.

Sources:
https://docs.aws.amazon.com/awsconsolehelpdocs/latest/gsg
https://docs.aws.amazon.com/ARG/latest/userguide/

AWS Trusted Advisor

- Trusted Advisor analyzes your AWS environment and provides best practice recommendations in five categories:
    - Cost Optimization
    - Performance
    - Security
    - Fault Tolerance
    - Service Limits
- Access to the seven core Trusted Advisor checks are available to all AWS users.
- Access to the full set of Trusted Advisor checks are available to Business and Enterprise Support plans.

Sources:
https://aws.amazon.com/premiumsupport/trustedadvisor/
https://aws.amazon.com/premiumsupport/ta-faqs/
https://www.amazonaws.cn/en/support/trustedadvisor/best-practices/

ANALYTICS

Amazon Kinesis

- 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.
- **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.
- **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.
- **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**

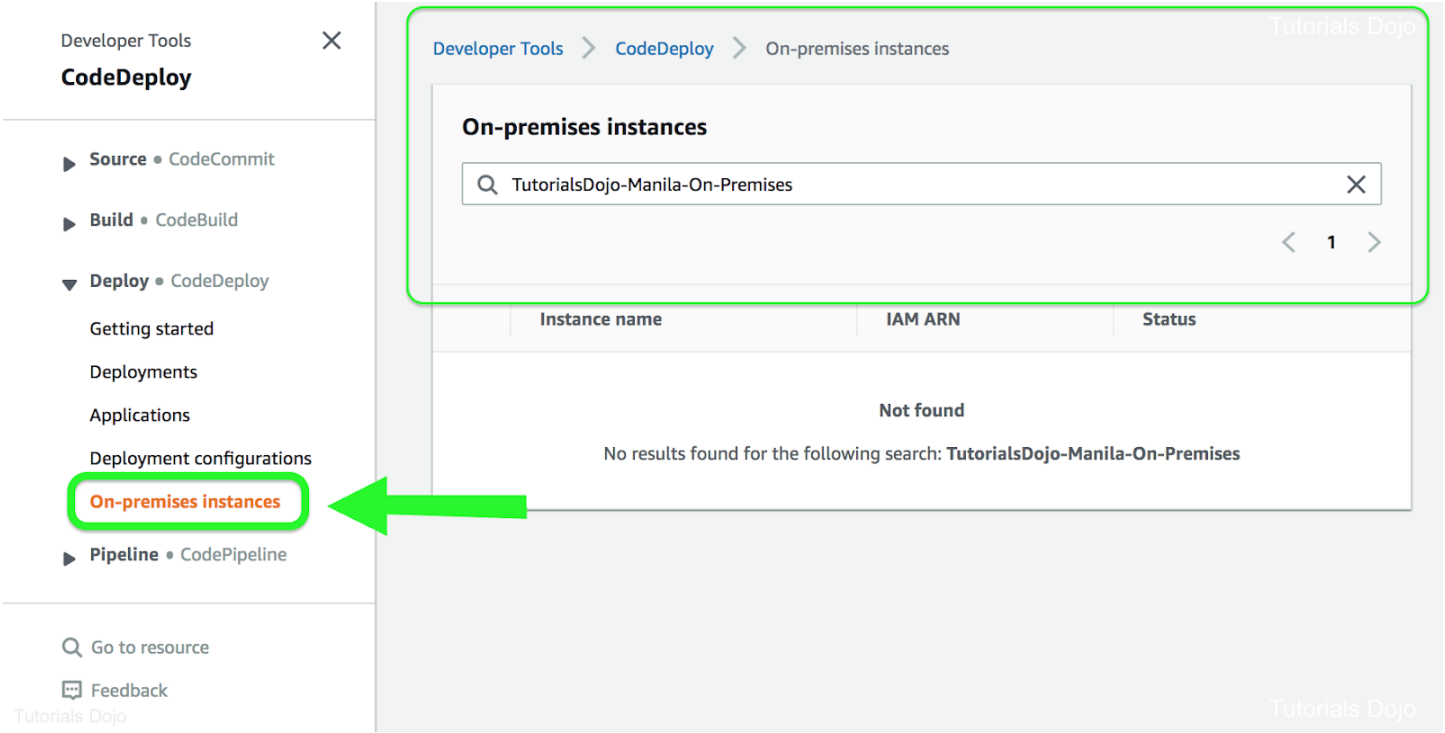
    - Kinesis Data Analytics is **serverless** and takes care of everything required to continuously run your application.
    - Kinesis Data Analytics elastically scales applications to keep up with any volume of data in the incoming data stream.
    - Kinesis Data Analytics delivers sub-second processing latencies so you can generate real-time alerts, dashboards, and actionable insights.
- **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.

Sources:
https://aws.amazon.com/kinesis/

DEVELOPMENT

AWS CodeDeploy

- A **fully managed deployment service** that automates software deployments to a variety of compute services such as Amazon EC2, AWS Fargate, AWS Lambda, and your on-premises servers.
    - Advantages of using Blue/Green Deployments vs In-Place Deployments
        - An application can be installed and tested in the new replacement environment and deployed to production simply by rerouting traffic.
        - If you're using the EC2/On-Premises compute platform, switching back to the most recent version of an application is faster and more reliable. Traffic can just be routed back to the original instances as long as they have not been terminated. With an in-place deployment, versions must be rolled back by redeploying the previous version of the application.
        - If you're using the EC2/On-Premises compute platform, new instances are provisioned and contain the most up-to-date server configurations.
        - If you're using the AWS Lambda compute platform, you control how traffic is shifted from your original AWS Lambda function version to your new AWS Lambda function version.
- With AWS CodeDeploy, you can also deploy your applications to your on-premises data centers.



- Pricing
    - There is no additional charge for code deployments to Amazon EC2 or AWS Lambda.
    - You are charged per on-premises instance update using AWS CodeDeploy.

Sources:
https://aws.amazon.com/codedeploy/features/?nc=sn&loc=2
https://docs.aws.amazon.com/codedeploy/latest/userguide/welcome.html
https://aws.amazon.com/codedeploy/faqs/?nc=sn&loc=6

AWS CodePipeline

- A fully managed **continuous delivery service** that helps you automate your release pipelines for application and infrastructure updates.
- You can easily integrate AWS CodePipeline with third-party services such as GitHub or with your own custom plugin.
- Concepts
    - A **pipeline** defines your release process workflow, and describes how a new code change progresses through your release process.
    - A pipeline comprises a series of **stages** (e.g., build, test, and deploy), which act as logical divisions in your workflow. Each stage is made up of a sequence of actions, which are tasks such as building code or deploying to test environments.
- Features
    - AWS CodePipeline can pull source code for your pipeline directly from AWS CodeCommit, GitHub, Amazon ECR, or Amazon S3.
    - It can run builds and unit tests in AWS CodeBuild.
    - It can deploy your changes using AWS CodeDeploy, AWS Elastic Beanstalk, Amazon ECS, AWS Fargate, Amazon S3, AWS Service Catalog, AWS CloudFormation, and/or AWS OpsWorks Stacks.
- Limits
    - Maximum number of total pipelines per Region in an AWS account is 300
    - Number of stages in a pipeline is minimum of 2, maximum of 10
- Pricing
    - You are charged per active pipeline each month. Newly created pipelines are free to use during the first 30 days after creation.

Sources:
https://aws.amazon.com/codepipeline/features/?nc=sn&loc=2
https://aws.amazon.com/codepipeline/pricing/?nc=sn&loc=3
https://docs.aws.amazon.com/codepipeline/latest/userguide/welcome.html
https://aws.amazon.com/codepipeline/faqs/?nc=sn&loc=5

AWS CodeBuild

- A fully managed **continuous integration service** that compiles source code, runs tests, and produces software packages that are ready to deploy.
- Features
  - AWS CodeBuild runs your builds in preconfigured build environments that contain the operating system, programming language runtime, and build tools (such as Apache Maven, Gradle, npm) required to complete the task. You just specify your source code's location and select settings for your build, such as the build environment to use and the build commands to run during a build.
  - AWS CodeBuild builds your code and stores the artifacts into an Amazon S3 bucket, or you can use a build command to upload them to an artifact repository.
  - AWS CodeBuild provides build environments for
    - Java
    - Python
    - Node.js
    - Ruby
    - Go
    - Android
    - .NET Core for Linux
    - Docker
  - You can define the specific commands that you want AWS CodeBuild to perform, such as installing build tool packages, running unit tests, and packaging your code.
  - You can integrate CodeBuild into existing CI/CD workflows using its source integrations, build commands, or Jenkins integration.
  - CodeBuild can connect to AWS CodeCommit, S3, GitHub, and GitHub Enterprise and Bitbucket to pull source code for builds.
  - CodeBuild allows you to use Docker images stored in another AWS account as your build environment, by granting resource level permissions.
  - It now allows you to access Docker images from any private registry as the build environment. Previously, you could only use Docker images from public DockerHub or Amazon ECR in CodeBuild.
- Pricing
  - You are charged for compute resources based on the duration it takes for your build to execute. The per-minute rate depends on the compute type that you use.

Sources:
https://aws.amazon.com/codebuild/features/?nc=sn&loc=2
https://aws.amazon.com/codebuild/pricing/?nc=sn&loc=3
https://aws.amazon.com/codebuild/faqs/?nc=sn&loc=5
https://docs.aws.amazon.com/codebuild/latest/userguide/getting-started.html

AWS CodeCommit

- A **fully-managed source control** service that hosts secure Git-based repositories, similar to Github.
- You can create your own code repository and use Git commands to interact with your own repository and other repositories.
- You can store and version any kind of file, including application assets such as images and libraries alongside your code.
- The AWS CodeCommit Console lets you visualize your code, pull requests, commits, branches, tags and other settings.
- High Availability
    - CodeCommit stores your repositories in Amazon S3 and Amazon DynamoDB.
- Monitoring
    - CodeCommit uses AWS IAM to control and monitor who can access your data as well as how, when, and where they can access it.
    - CodeCommit helps you monitor your repositories via AWS CloudTrail and AWS CloudWatch.
    - You can use Amazon SNS to receive notifications for events impacting your repositories. Each notification will include a status message as well as a link to the resources whose event generated that notification.
- Pricing
    - The first 5 active users per month are free of charge. You also get to have unlimited repositories, with 50 GB-month total worth of storage, and 10,000 Git requests/month at no cost.
    - You are billed for each active user beyond the first 5 per month. You also get an additional 10GB-month of storage per active user, and an additional 2,000 Git requests per active user.

Sources:
https://aws.amazon.com/codecommit/
https://docs.aws.amazon.com/codecommit/latest/userguide/welcome.html
https://aws.amazon.com/codecommit/faqs/

AWS X-Ray

- AWS X-Ray analyzes and debugs production, distributed applications, such as those built using a microservices architecture. With X-Ray, you can identify performance bottlenecks, edge case errors, and other hard to detect issues.
- AWS X-Ray provides an end-to-end, cross-service, application-centric view of requests flowing through your application by aggregating the data gathered from individual services in your application into a single unit called a *trace*.
- You pay based on the number of traces recorded, retrieved, and scanned. A trace represents a request to your application and may include multiple data points, such as for calls to other services and database access.

Sources:
https://aws.amazon.com/xray/features/
https://aws.amazon.com/xray/pricing/
https://docs.aws.amazon.com/xray/latest/devguide/aws-xray.html
https://aws.amazon.com/xray/faqs/

AWS BILLING AND COST MANAGEMENT

- **Cost Explorer** tracks and analyzes your AWS usage. It is free for all accounts.
- Use **Budgets** to manage budgets for your account.
- Use **Bills** to see details about your current charges.
- Use **Payment History** to see your past payment transactions.
- AWS Billing and Cost Management closes the billing period at midnight on the last day of each month and then calculates your bill.
- At the end of a billing cycle or at the time you choose to incur a one-time fee, AWS charges the credit card you have on file and issues your invoice as a downloadable PDF file.
- With CloudWatch, you can create billing alerts that notify you when your usage of your services exceeds thresholds that you define.
- Use **cost allocation tags** to track your AWS costs on a detailed level. AWS provides two types of cost allocation tags, an *AWS generated tags* and *user-defined tags*.

AWS Free Tier

- When you create an AWS account, you're automatically signed up for the free tier for **12 months**.
- You can use a number of AWS services for free, as long as you haven't surpassed the allocated usage limit.
- To help you stay within the limits, you can track your free tier usage and set a **billing alarm with AWS Budgets** to notify you if you start incurring charges.

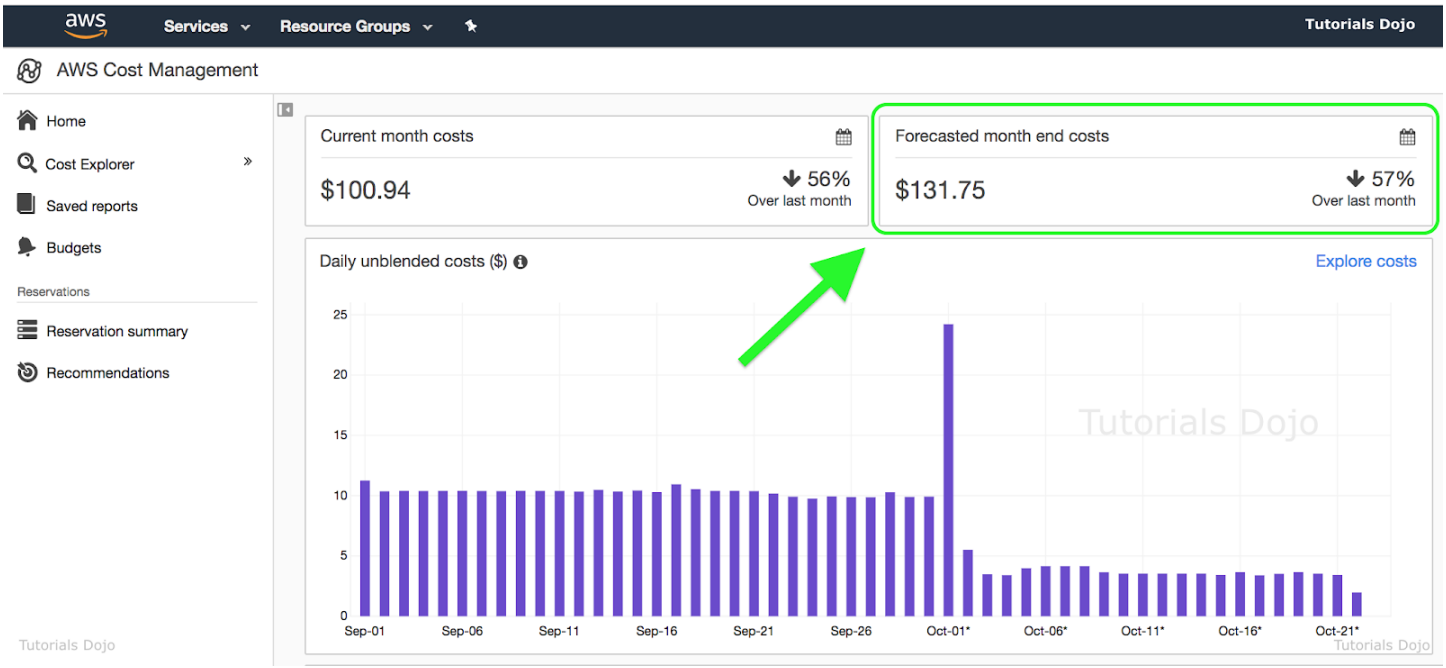
AWS Cost and Usage Reports

- The AWS Cost and Usage report provides information about your use of AWS resources and estimated costs for that usage.
- The AWS Cost and Usage report is a .csv file or a collection of .csv files that is stored in an S3 bucket. Anyone who has permissions to access the specified S3 bucket can see your billing report files.
- You can use the Cost and Usage report to track your Reserved Instance Utilization, charges, and allocations.
- Reports can be automatically uploaded into AWS Redshift and/or AWS QuickSight for analysis.

AWS Cost Explorer

- Cost Explorer includes a default report that helps you visualize the costs and usage associated with your TOP FIVE cost-accruing AWS services, and gives you a detailed breakdown on all services in the table view.
- You can view data for up to the last 12 months, forecast how much you're likely to spend for the next three months, and get recommendations for what Reserved Instances to purchase.

- Cost Explorer must be enabled before it can be used. You can enable it only if you're the owner of the AWS account and you signed in to the account with your root credentials.

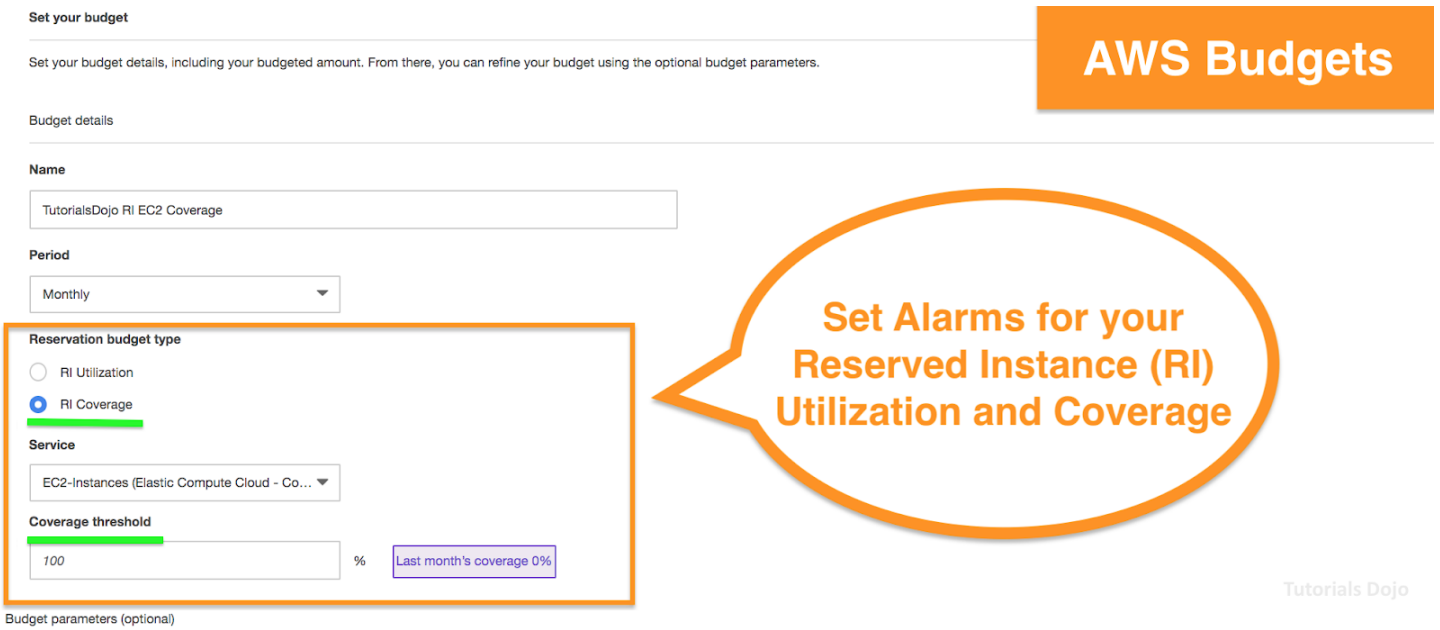


- If you're the owner of a management account in an organization, enabling Cost Explorer enables Cost Explorer for all of the organization accounts. You can't grant or deny access individually.
- You can create forecasts that predict your AWS usage and define a time range for the forecast.
- Other default reports available are:
    - The **EC2 Monthly Cost and Usage report** lets you view all of your AWS costs over the past two months, as well as your current month-to-date costs.
    - The **Monthly Costs by Linked Account report** lets you view the distribution of costs across your organization.
    - The **Monthly Running Costs report** gives you an overview of all of your running costs over the past three months, and provides forecasted numbers for the coming month with a corresponding confidence interval.

AWS Budgets

- Set custom budgets that alert you when your costs or usage exceed or are forecasted to exceed your budgeted amount.
- With Budgets, you can view the following information:
    - How close your plan is to your budgeted amount or to the free tier limits
    - Your usage to date, including how much you have used of your Reserved Instances
    - Your current estimated charges from AWS and how much your predicted usage will incur in charges by the end of the month

- How much of your budget has been used



- Budget information is updated up to three times a day.
- Types of Budgets:
  - **Cost budgets** – Plan how much you want to spend on a service.
  - **Usage budgets** – Plan how much you want to use one or more services.
  - **RI utilization budgets** – Define a utilization threshold and receive alerts when your RI usage falls below that threshold.
  - **RI coverage budgets** – Define a coverage threshold and receive alerts when the number of your instance hours that are covered by RIs fall below that threshold.
- Budgets can be tracked at the monthly, quarterly, or yearly level, and you can customize the start and end dates.
- Budget alerts can be sent via email and/or Amazon SNS topic.
- First two budgets created are free of charge.

Sources:
https://aws.amazon.com/aws-cost-management/aws-budgets/
https://aws.amazon.com/aws-cost-management/aws-cost-explorer/
https://aws.amazon.com/aws-cost-management/aws-cost-and-usage-reporting/
https://aws.amazon.com/aws-cost-management/faqs/
https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2

APPLICATION INTEGRATION

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
- **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 **Unlimited Throughput** - Standard queues support a nearly unlimited number of transactions per second (TPS) per action. **At-Least-Once Delivery** - A message is delivered at least once, but occasionally more than one copy of a message is delivered. **Best-Effort Ordering** - Occasionally, messages might be delivered in an order different from which they were sent.	Available in the US East (N. Virginia), US East (Ohio) US West (Oregon), EU (Ireland), Asia Pacific (Sydney), and Asia Pacific (Tokyo)regions. **High Throughput** - 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. **Exactly-Once Processing** - A message is delivered once and remains available until a consumer processes and deletes it. Duplicates aren't introduced into the queue. **First-in-First-Out Delivery** - The order in which messages are sent and received is strictly preserved.
Send data between applications when the throughput is important. Tutorials Dojo	Send data between applications when the order of events is important.

- **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 FIFO topics support the forwarding of messages to SQS FIFO queues. You can also 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.

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.
- 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.

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/

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.
  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.
    - 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.
- 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/

COMPARISON OF AWS SERVICES

S3 vs EBS vs EFS

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 burstring throughput.
Durability	Stored redundantly across multiple AZs; has 99.999999999% 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. S3 Intelligent Tiering - 99.9%	Has 99.999% availability	99.9% SLA. Runs in multi-AZ

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 Amazon EBS Multi-Attach enables you to attach a single Provisioned IOPS SSD (io1 or io2) volume to up to 16 Nitro-based instances that are in the same Availability Zone.	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 *Block Public Access* 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 *mount target*, 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.
Backup and Restoration	Use versioning or cross-region replication	All EBS volume types offer durable snapshot capabilities.	EFS to EFS replication through third party tools or AWS DataSync
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 99.999999999% of objects across multiple Availability Zones.
- S3 and Glacier are designed for availability of 99.99%.
- 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 can only be transitioned to the Glacier Deep Archive storage class.
- S3 (standard, intelligent-tiering, standard-IA, and one zone-IA) and Glacier are 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)	S3 Intelligent Tiering
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	For long-lived data that have unpredictable access patterns
Durability	99.999999999% (11 9's)	99.999999999% (11 9's)	99.999999999% (11 9's)	99.999999999% (11 9's)
Availability	99.99%	99.9%	99.5%	99.9%
Availability SLA	99.9%	99%	99%	99%
Number of Availability Zones	At least 3	At least 3	Only 1	At least 3
Minimum capacity charge per object	N/A	128KB	128KB	N/A
Minimum storage duration charge	N/A	30 days	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.	Directly PUT into S3 Intelligent-Tiering or set Lifecycle policies to transition objects from the S3 Standard to the S3 Intelligent-Tiering storage class.
Retrieval fee	N/A	per GB retrieved	per GB retrieved	N/A
First byte latency	milliseconds	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	S3 Intelligent to S3 One Zone-IA or 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.	Data with unknown or changing access patterns, optimize storage costs automatically, and unpredictable workloads

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.
- It is recommended to use multipart upload for objects larger than 100MB.

RDS vs DynamoDB

Tutorials Dojo	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	- 128 TB for Aurora engine. - 64 TB for MySQL, MariaDB, Oracle, and PostgreSQL engines. - 16 TB for SQL Server engine.	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. Magnetic – Amazon RDS also supports magnetic storage for backward compatibility.	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	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. RDS also supports storage auto scaling Read replicas are available in Amazon RDS for MySQL, MariaDB, and PostgreSQL as well as Amazon Aurora.	Support tables of virtually any size with horizontal scaling. For tables using on-demand capacity mode, DynamoDB instantly accommodates your workloads as they ramp up or down to any previously reached traffic level. For tables using provisioned capacity, DynamoDB delivers automatic scaling of throughput and storage based on your previously set capacity.
Security	Isolate your database in your own virtual network. Connect to your on-premises IT infrastructure using industry-standard encrypted IPsec VPNs. You can configure firewall settings and control network access to your database instances. Integrates with IAM.	Integrates with IAM.
Encryption	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. Supports Transparent Data Encryption in SQL Server and Oracle. Supports the use of SSL to secure data in transit.	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	A monthly charge for each database instance that you launch. Option to reserve a DB instance for a one or three year term and receive discounts in pricing, compared to On-Demand instance pricing.	Charges for reading, writing, and storing data in your DynamoDB tables, along with any optional features you choose to enable. There are specific billing options for each of DynamoDB's capacity modes.
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. Tutorials Dojo

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.

RDS vs Aurora

	Aurora	RDS
Type of database	Relational database	
Features	• MySQL and PostgreSQL compatible. • 5x faster than standard MySQL databases and 3x faster than standard PostgreSQL databases. • Use Parallel Query to run transactional and analytical workloads in the same Aurora database, while maintaining high performance. • You can distribute and load balance your unique workloads across different sets of Aurora DB instances using custom endpoints. • Aurora Serverless allows for on-demand, autoscaling of your Aurora DB instance capacity.	• Has several database instance types for different kinds of workloads and supports five database engines - MySQL, PostgreSQL, MariaDB, Oracle, and SQL Server. • Can use either General Purpose Storage and Provisioned IOPS storage to deliver a consistent IOPS performance
Maximum storage capacity	• 128 TB	• 64 TB for MySQL, MariaDB, Oracle, and PostgreSQL engines • 16 TB for SQL Server engine
DB instance classes	• Memory Optimized classes - for workloads that need to process large data sets in memory. • Burstable classes - provides the instance the ability to burst to a higher level of CPU performance when required by the workload.	• Standard classes - for a wide range of workloads, you can use general purpose instance. It offers a balance of compute, memory, and networking resources. • Memory Optimized classes - for workloads that need to process large data sets in memory. • Burstable classes - provides the instance the ability to burst to a

		higher level of CPU performance when required by the workload.
Availability and durability	• Amazon Aurora uses RDS Multi-AZ technology to automate failover to one of up to 15 Amazon Aurora Replicas across three Availability Zones • Amazon Aurora Global Database uses storage-based replication to replicate a database across multiple AWS Regions, with typical latency of less than 1 second. • Self-healing: data blocks and disks are continuously scanned for errors and replaced automatically.	• 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.
Backups	• Point-in-time recovery to restore your database to any second during your retention period, up to the last five minutes. • Automatic backup retention period up to thirty-five days. • Backtrack to the original database state without needing to restore data from a backup.	• 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.

Scalability	• Aurora automatically increases the size of your volumes as your database grows larger (increments of 10 GB). • Aurora also supports replica auto-scaling, where it automatically adds and removes DB replicas in response to changes in performance metrics. • Cross-region replicas provide fast local reads to your users, and each region can have an additional 15 Aurora replicas to further scale local reads.	• The MySQL, MariaDB, SQL Server, Oracle, and PostgreSQL engines scale your storage automatically as your database workload grows with zero downtime. • Read replicas are available for Amazon RDS for MySQL, MariaDB, PostgreSQL, Oracle, and SQL Server. Amazon RDS creates a second DB instance using a snapshot of the source DB instance and uses the engines' native asynchronous replication to update the read replica whenever there is a change to the source. • Can scale compute and memory resources (vertically) of up to a maximum of 32 vCPUs and 244 GiB of RAM.
Security	• Isolate the database in your own virtual network via VPC. • Connect to your on-premises IT infrastructure using encrypted IPsec VPNs or Direct Connect and VPC Endpoints. • Configure security group firewall and network access rules to your database instances. • Integrates with IAM.	
Encryption	• Encrypt your databases using keys you manage through AWS KMS. With Amazon Aurora encryption, data stored at rest is encrypted, as are its automated backups, snapshots, and replicas in the same cluster. • Supports the use of SSL (AES-256) to secure data in transit.	• Encrypt your databases using keys you manage through AWS KMS. With Amazon RDS encryption, data stored at rest is encrypted, as are its automated backups, read replicas, and snapshots. • Supports Transparent Data Encryption in SQL Server and Oracle. • Supports the use of SSL to secure data in transit

DB Authentication	• Password authentication • Password and IAM database authentication	• Password authentication • Password and IAM database authentication • Password and Kerberos authentication
Maintenance	• Amazon Aurora automatically updates the database with the latest patches. • Amazon Aurora Serverless enables you to run your database in the cloud without managing/maintaining any database infrastructure.	• Amazon RDS will update databases with the latest major and minor patches on scheduled maintenance windows. You can exert optional control over when and if your database instance is patched.
Monitoring	• Use Enhanced Monitoring to collect metrics from the operating system instance. • Use Performance Insights to detect database performance problems and take corrective action. • Uses Amazon SNS to receive a notification on database events.	
Pricing	• A monthly charge for each database instance that you launch if you use on-demand. This includes both the instance compute capacity and the amount of storage being used. • Option to reserve a DB instance for a one or three-year term (reserve instances) and receive discounts in pricing.	

Use Cases	• Enterprise applications - a great option for any enterprise application that uses relational database since it handles provisioning, patching, backup, recovery, failure detection, and repair. • SaaS applications - without worrying about the underlying database that powers the application, you can concentrate on building high-quality applications. • Web and mobile gaming - since games need a database with high throughput, storage scalability, and must be highly available. Aurora suits the variable use pattern of these apps perfectly.	• Web and mobile applications - since the application needs a database with high throughput, storage scalability, and must be highly available. RDS also fulfills the needs of such highly demanding apps. • E-commerce applications - a managed database service that offers PCI compliance. You can just focus on building high-quality customer experiences without thinking of the underlying database. • Mobile and online games - game developers don't need to worry about provisioning, scaling, and monitoring of database servers since RDS manages the database infrastructure.

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
Up to five security groups per instance	1 nACL per subnet
Up to 50 rules per security group	Up to 20 rules per nACL
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, io2), 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:	*small, random* I/O operations	*large, sequential* I/O operations
Can be used as a bootable volume?	Yes	No
Suitable Use Cases	- Best for **transactional workloads** - Critical business applications that require sustained IOPS performance - Large database workloads such as MongoDB, Oracle, Microsoft SQL Server and many others…	- Best for *large streaming workloads* requiring consistent, fast throughput at a low price - Big data, Data warehouses, Log processing - Throughput-oriented storage for large volumes of data that is *infrequently* accessed
Cost	moderate / high 	low 
Dominant Performance Attribute	IOPS	Throughput (`MiB/s`)

Provisioned IOPS SSD (io1, io2) 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. Provisioned IOPS SSD io2 is an upgrade of Provisioned IOPS SSD io1. It offers higher 99.999% durability and higher IOPS per GiB ratio with 500 IOPS per GiB, all at the same cost as io1 volumes.

Volume Name	General Purpose SSD		Provisioned IOPS SSD	
Volume type	gp3	gp2	io2	io1
Description	General Purpose SSD volume that balances price performance for a wide variety of transactional workloads	General Purpose SSD volume that balances price performance for a wide variety of transactional workloads	High performance SSD volume designed for **business-critical** latency-sensitive applications	High performance SSD volume designed for latency-sensitive transactional workloads
Use Cases	virtual desktops, medium sized single instance databases such as MSFT SQL Server and Oracle DB, low-latency interactive apps, dev & test, boot volumes	Boot volumes, low-latency interactive apps, dev & test	Workloads that require sub-millisecond latency, and sustained IOPS performance or more than 64,000 IOPS or 1,000 MiB/s of throughput	Workloads that require sustained IOPS performance or more than 16,000 IOPS and I/O-intensive database workloads
Volume Size	1 GB – 16 TB	1 GB – 16 TB	4 GB – 16 TB	4 GB – 16 TB
Durability	99.8% - 99.9% durability	99.8% - 99.9% durability	99.999%	99.8% - 99.9%
Max IOPS / Volume	16,000	16,000	64,000	64,000
Max Throughput / Volume	1000 MB/s	250 MB/s	1,000 MB/s	1,000 MB/s
Max IOPS / Instance	260,000	260,000	160,000	260,000
Max IOPS / GB	N/A	N/A	500 IOPS/GB	50 IOPS/GB
Max Throughput / Instance	7,500 MB/s	7,500 MB/s	4,750 MB/s	7,500 MB/s
Latency	single digit millisecond	single digit millisecond	single digit millisecond	single digit millisecond

Multi-Attach	No	No	Yes	Yes

Volume Name	Throughput Optimized HDD	Cold HDD
Volume type	st1	sc1
Description	Low cost HDD volume designed for frequently accessed, throughput-intensive workloads	Throughput-oriented storage for data that is infrequently accessed Scenarios where the lowest storage cost is important
Use Cases	Big data, data warehouses, log processing	Colder data requiring fewer scans per day
Volume Size	125 GB – 16 TB	125 GB – 16 TB
Durability	99.8% - 99.9% durability	99.8% - 99.9% durability
Max IOPS / Volume	500	250
Max Throughput / Volume	500 MB/s	250 MB/s
Max IOPS / Instance	260,000	260,000
Max IOPS / GB	N/A	N/A
Max Throughput / Instance	7,500 MB/s	7,500 MB/s
Multi-Attach	No	No

Application Load Balancer vs Network Load Balancer vs Gateway Load Balancer

Feature	Application Load Balancer	Network Load Balancer	Gateway Load Balancer
Protocols	HTTP, HTTPS, gRPC	TCP, UDP, TLS	IP
Platforms	VPC	VPC	VPC
Health checks	HTTP, HTTPS, gRPC	TCP, HTTP, HTTPS	TCP, HTTP, HTTPS
Cloudwatch Metrics	✓	✓	✓
Logging	✓	✓	✓
Zonal Failover	✓	✓	✓
Connection Draining (deregistration delay)	✓	✓	✓
Load Balancing to multiple ports on the same instance	✓	✓	✓
IP addresses as targets	✓	✓ (TCP, TLS)	✓
Load balancer deletion protection	✓	✓	✓
Configuration idle connection timeout	✓		
Cross-zone load balancing	✓	✓	✓
Sticky sessions	✓	✓	✓
Static IP		✓	
Elastic IP address		✓	
Preserve Source IP address	✓	✓	✓
Resource-based IAM permissions	✓	✓	✓
Tag-based IAM permissions	✓	✓	✓
Slow start	✓		
Web sockets	✓	✓	✓
PrivateLink Support		✓ (TCP, TLS)	✓ (GWLBE)
Source IP address CIDR-based routing	✓		

Feature	Application Load Balancer	Network Load Balancer	Gateway Load Balancer
Layer 7			
Path-based routing	✓		
Host-based routing	✓		
Native HTTP/2	✓		
Redirects	✓		
Fixed response	✓		
Lambda functions as targets	✓		
HTTP header-based routing	✓		
HTTP method-based routing	✓		
Query string parameter-based routing	✓		
Security			
SSL offloading	✓	✓	
Server Name Indication (SNI)	✓	✓	
Back-end server encryption	✓	✓	
User authentication	✓		
Session Resumption	✓	✓	
Terminates flow/proxy behavior	✓	✓	✓

Common features between the load balancers:

- Has instance health check features
- Has built-in CloudWatch monitoring
- Logging features
- Support zonal failover
- Support cross-zone load balancing (evenly distributes traffic across registered instances in enabled AZs)
- Resource-based IAM permission policies
- Tag-based IAM permissions
- Flow stickiness - all packets are sent to one target and return the traffic that comes from the same target.

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 matter.
- 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, know 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 unhealth containers to ensure that you have the desired number of containers supporting your application.

Tutorials Dojo

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.

FINAL REMARKS

Whether you are a student wanting to learn more about the cloud, or a fresh graduate trying to enter the industry, or even an experienced professional exploring a new field, the cloud is absolutely a fun and exciting space to be in. There are so many things you can do today that were not feasible before with a local infrastructure setup. All you need is a browser and Internet connectivity and you'll have your whole environment right at your fingertips. And as the days go by, more and more people aspire to be AWS Certified. More and more people want to learn cloud computing and bring their careers to newer heights. And with these certifications, they're like investments on yourself and on your skills. These achievements are acknowledged by everyone in the community.

We at Tutorials Dojo are dedicated to help you achieve these results. We do our best to constantly produce practical and valuable content for everyone who is preparing for his/her AWS certification exams. We have written blogs, guides, cheat sheets, and practice exams that are also constantly being updated based on our experiences and on the feedback of our students. We listen and we deliver.

So if you are currently reading our final remarks, we want to say thank you for choosing Tutorials Dojo and we hope you'll continue supporting us. We also wish you the very best on your future AWS certification exams! Our forums are always open for feedback and we would love to hear from you. It is you, our students, who are the front-runners that help improve the content that we produce.

Once you feel that you have learned the basics, we recommend testing your knowledge through our AWS Certified Cloud Practitioner Video course and AWS Certified Cloud Practitioner Practice Exams. You can also try the free sampler version of our full practice test course here. And if you have any issues, concerns, or constructive feedback on our eBook, feel free to contact us at support@tutorialsdojo.com.

ABOUT THE AUTHOR

Jon Bonso (10x AWS Certified)

Born and raised in the Philippines, Jon is the Co-Founder of **Tutorials Dojo**. Now based in Sydney, Australia, he has over a decade of diversified experience in Banking, Financial Services, and Telecommunications. He's 10x AWS Certified and has worked with various cloud services such as Google Cloud, and Microsoft Azure. Jon is passionate about what he does and dedicates a lot of time creating educational courses. He has given IT seminars to different universities in the Philippines for free and has launched educational websites using his own money and without any external funding.