JON BONSO

AWS CERTIFIED SOLUTIONS ARCHITECT ASSOCIATE

SAA-C03

Tutorials Dojo
Study Guide and Cheat Sheets

TABLE OF CONTENTS

INTRODUCTION

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

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

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

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

Note: We took extra care to come up with these study guides and cheat sheets, however, this is meant to be just a supplementary resource when preparing for the exam. We highly recommend working on **hands-on sessions**, **SAA-C03 video course** and **practice exams** to further expand your knowledge and improve your test taking skills.

AWS CERTIFIED SOLUTIONS ARCHITECT ASSOCIATE EXAM - STUDY GUIDE AND TIPS

The AWS Certified Solutions Architect Associate SAA-C03 exam, or SAA for short, is one of the most sought after certifications in the Cloud industry. This certification attests to your knowledge of the AWS Cloud and building a well-architected infrastructure in AWS.

As a Solutions Architect, it is your responsibility to be familiar with the services that meet your customer requirements. Aside from that, you should also have the knowledge to create an efficient, secure, reliable, fault tolerant, and cost-effective infrastructure out of these services. Your AWS SA Associate exam will be based upon these topics.

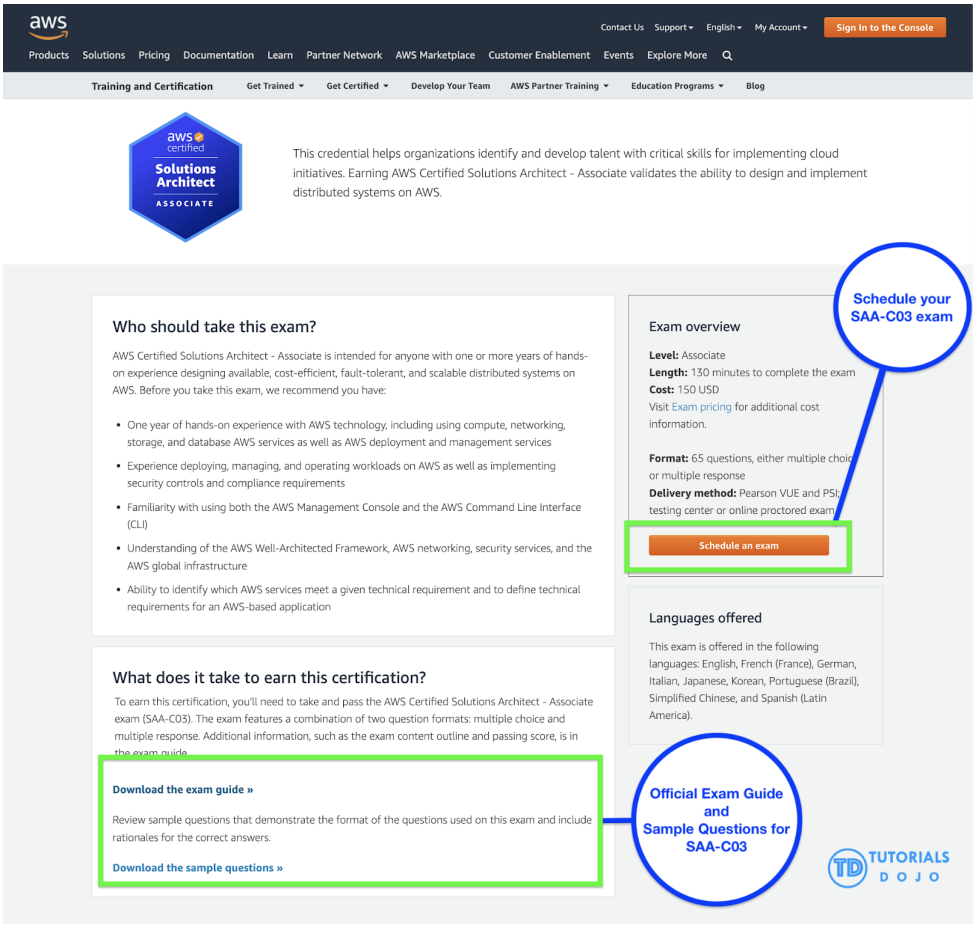
Whitepapers, FAQs, and the AWS Documentation will be your primary study materials for this exam. Experience in building systems will also be helpful, since the exam consists of multiple scenario type questions. You can learn more details on your exam through the official SAA-C03 Exam Guide here. Do a quick read on it to be aware of how to prepare and what to expect on the exam itself.

SAA-C03 Study Materials

As a starting point for your AWS Certified Solutions Architect Associate exam studies, we recommend taking the FREE AWS Certified Cloud Practitioner Essential digital course. This free and highly interactive course aims to improve AWS Cloud knowledge by covering different AWS Cloud concepts, AWS services, security, architecture, pricing, and support plans. If you are quite new to AWS, taking and completing this digital course should be your first step for your SAA-C03 exam prep.

There are a lot of posts on the Internet claiming the “best” course for the AWS Certified Solutions Architect Associate SAA-C03 Exam. However, some of these resources are already obsolete and don't cover the latest topics that were recently introduced in the SAA-C03 test. How can I ensure that you are using the right study materials for your upcoming AWS Certified Solutions Architect Associate test?

The best thing to do is to check the official AWS Certification website for the most up-to-date information. You can also head on to the official AWS Certification page for the **AWS Certified Solutions Architect Associate SAA-C03** exam. This page is where you can find the actual link to schedule your SAA-C03 exam as well as get the official SAA-C03 **Exam Guide** and **Sample Questions** as shown below:



For the exam version (SAA-C03), you should also know the following services:

- AWS Global Accelerator
- Elastic Fabric Adapter (EFA)
- Elastic Network Adapter (ENA)
- AWS ParallelCluster
- Amazon FSx
- AWS DataSync
- AWS Directory Service
- High Performance Computing
- Aurora Serverless

There are more SAA-C03 topics that we have recently added to our:

- **AWS Certified Solutions Architect Associate Practice Exams**
- **AWS Certified Solutions Architect Associate Video Course**

Core AWS Services to Focus On for the SAA-C03 Exam

1. EC2 - As the most fundamental compute service offered by AWS, you should know about EC2 inside out.
2. Lambda - Lambda is the common service used for serverless applications. Study how it is integrated with other AWS services to build a full stack serverless app.
3. Elastic Load Balancer - Load balancing is very important for a highly available system. Study about the different types of ELBs, and the features each of them supports.
4. Auto Scaling - Study what services in AWS can be auto scaled, what triggers scaling, and how auto scaling increases/decreases the number of instances.
5. Elastic Block Store - As the primary storage solution of EC2, study on the types of EBS volumes available. Also study how to secure, backup and restore EBS volumes.
6. S3 / Glacier - AWS offers many types of S3 storage depending on your needs. Study what these types are and what differs between them. Also review on the capabilities of S3 such as hosting a static website, securing access to objects using policies, lifecycle policies, etc. Learn as much about S3 as you can.
7. Storage Gateway - There are occasional questions about Storage Gateway in the exam. You should understand when and which type of Storage Gateway should be used compared to using services like S3 or EBS. You should also know the use cases and differences between DataSync and Storage Gateway.
8. EFS - EFS is a service highly associated with EC2, much like EBS. Understand when to use EFS, compared to using S3, EBS or instance store. Exam questions involving EFS usually ask the trade off between cost and efficiency of the service compared to other storage services.
9. RDS / Aurora - Know how each RDS database differs from one another, and how they are different from Aurora. Determine what makes Aurora unique, and when it should be preferred from other databases (in terms of function, speed, cost, etc). Learn about parameter groups, option groups, and subnet groups.
10. DynamoDB - The exam includes lots of DynamoDB questions, so read as much about this service as you can. Consider how DynamoDB compares to RDS, Elasticache and Redshift. This service is also commonly used for serverless applications along with Lambda.
11. Elasticache - Familiarize yourself with Elasticache redis and its functions. Determine the areas/services where you can place a caching mechanism to improve data throughput, such as managing session state of an ELB, optimizing RDS instances, etc.
12. VPC/NACL/Security Groups - Study every service that is used to create a VPC (subnets, route tables, internet gateways, nat gateways, VPN gateways, etc). Also, review on the differences of network access control lists and security groups, and during which situations they are applied.
13. Route 53 - Study the different types of records in Route 53. Study also the different routing policies. Know what hosted zones and domains are.
14. IAM - Services such as IAM Users, Groups, Policies and Roles are the most important to learn. Study how IAM integrates with other services and how it secures your application through different policies. Also read on the best practices when using IAM.

15. CloudWatch - Study how monitoring is done in AWS and what types of metrics are sent to CloudWatch. Also read upon Cloudwatch Logs, CloudWatch Alarms, and the custom metrics made available with CloudWatch Agent.
16. CloudTrail - Familiarize yourself with how CloudTrail works, and what kinds of logs it stores as compared to CloudWatch Logs.
17. Kinesis - Read about Kinesis sharding and Kinesis Data Streams. Have a high level understanding of how each type of Kinesis Stream works.
18. CloudFront - Study how CloudFront helps speed up websites. Know what content sources CloudFront can serve from. Also check the kinds of certificates CloudFront accepts.
19. SQS - Gather info on why SQS is helpful in decoupling systems. Study how messages in the queues are being managed (standard queues, FIFO queues, dead letter queues). Know the differences between SQS, SNS, SES, and Amazon MQ.
20. SNS - Study the function of SNS and what services can be integrated with it. Also be familiar with the supported recipients of SNS notifications.
21. SWF / CloudFormation / OpsWorks - Study how these services function. Differentiate the capabilities and use cases of each of them. Have a high level understanding of the kinds of scenarios they are usually used in.

Based on our exam experience, you should also know when to use the following:

- AWS DataSync vs Storage Gateway
- FSx (Cold and Hot Storage)
- Cross-Region Read Replicas vs. Multi-Az RDS - which database provides high-availability
- Amazon Object key vs Object Metadata
- Direct Connect vs. Site-to-Site VPN
- AWS Config vs AWS CloudTrail
- Security Group vs NACL
- NAT Gateway vs NAT Instance
- Geolocation routing policy vs. Geoproximity routing policy on Route 53

The AWS Documentation and FAQs will be your primary source of information. You can also visit **Tutorials Dojo's AWS Cheat Sheets** to gain access to a repository of thorough content on the different AWS services mentioned above. Complete our **AWS Certified Solutions Architect Associate Video Course** and aim to get a consistent 90% score on all sets of our **AWS Certified Solutions Architect Associate Practice Exams**.

Lastly, try out these services yourself by signing up in AWS and performing some lab exercises. Experiencing them on your own will help you greatly in remembering what each service is capable of.

Common Exam Scenarios

Scenario	Solution
Domain 1: Design Secure Architectures	
Encrypt EBS volumes restored from the unencrypted EBS snapshots	Copy the snapshot and enable encryption with a new symmetric CMK while creating an EBS volume using the snapshot.
Limit the maximum number of requests from a single IP address.	Create a rate-based rule in AWS WAF and set the rate limit.
Grant the bucket owner full access to all uploaded objects in the S3 bucket.	Create a bucket policy that requires users to set the object's ACL to bucket-owner-full-control.
Protect objects in the S3 bucket from accidental deletion or overwrite.	Enable versioning and MFA delete.
Access resources on both on-premises and AWS using on-premises credentials that are stored in Active Directory.	Set up SAML 2.0-Based Federation by using a Microsoft Active Directory Federation Service.
Secure the sensitive data stored in EBS volumes	Enable EBS Encryption
Ensure that the data-in-transit and data-at-rest of the Amazon S3 bucket is always encrypted	Enable Amazon S3 Server-Side or use Client-Side Encryption
Secure the web application by allowing multiple domains to serve SSL traffic over the same IP address.	Use AWS Certificate Manager to generate an SSL certificate. Associate the certificate to the CloudFront distribution and enable Server Name Indication (SNI).
Control the access for several S3 buckets by using a gateway endpoint to allow access to trusted buckets.	Create an endpoint policy for trusted S3 buckets.
Enforce strict compliance by tracking all the configuration changes made to any AWS services.	Set up a rule in AWS Config to identify compliant and non-compliant services.
Domain 2: Design Resilient Architectures	
Set up asynchronous data replication to another RDS DB instance hosted in another AWS Region	Create a Read Replica
A parallel file system for “hot” (frequently accessed) data	Amazon FSx For Lustre

Implement synchronous data replication across Availability Zones with automatic failover in Amazon RDS.	Enable Multi-AZ deployment in Amazon RDS.
Needs a storage service to host “cold” (infrequently accessed) data	Amazon S3 Glacier
Set up a relational database and a disaster recovery plan with an RPO of 1 second and RTO of less than 1 minute.	Use Amazon Aurora Global Database.
Monitor database metrics and send email notifications if a specific threshold has been breached.	Create an SNS topic and add the topic in the CloudWatch alarm.
Set up a DNS failover to a static website.	Use Route 53 with the failover option to a static S3 website bucket or CloudFront distribution.
Implement an automated backup for all the EBS Volumes.	Use Amazon Data Lifecycle Manager to automate the creation of EBS snapshots.
Monitor the available swap space of your EC2 instances	Install the CloudWatch agent and monitor the SwapUtilizationmetric.
Domain 3: High-Performing Architectures	
Implement a fanout messaging.	Create an SNS topic with a message filtering policy and configure multiple SQS queues to subscribe to the topic.
A database that has a read replication latency of less than 1 second.	Use Amazon Aurora with cross-region replicas.
A specific type of Elastic Load Balancer that uses UDP as the protocol for communication between clients and thousands of game servers around the world.	Use Network Load Balancer for TCP/UDP protocols.
Monitor the memory and disk space utilization of an EC2 instance.	Install Amazon CloudWatch agent on the instance.
Retrieve a subset of data from a large CSV file stored in the S3 bucket.	Perform an S3 Select operation based on the bucket's name and object's key.
Upload 1 TB file to an S3 bucket.	Use Amazon S3 multipart upload API to upload large objects in parts.

Improve the performance of the application by reducing the response times from milliseconds to microseconds.	Use Amazon DynamoDB Accelerator (DAX)
Retrieve the instance ID, public keys, and public IP address of an EC2 instance.	Access the url: http://169.254.169.254/latest/meta-data/ using the EC2 instance.
Route the internet traffic to the resources based on the location of the user.	Use Route 53 Geolocation Routing policy.
Domain 4: Design Cost-Optimized Architectures	
A cost-effective solution for over-provisioning of resources.	Configure a target tracking scaling in ASG.
The application data is stored in a tape backup solution. The backup data must be preserved for up to 10 years.	Use AWS Storage Gateway to backup the data directly to Amazon S3 Glacier Deep Archive.
Accelerate the transfer of historical records from on-premises to AWS over the Internet in a cost-effective manner.	Use AWS DataSync and select Amazon S3 Glacier Deep Archive as the destination.
Globally deliver the static contents and media files to customers around the world with low latency.	Store the files in Amazon S3 and create a CloudFront distribution. Select the S3 bucket as the origin.
An application must be hosted to two EC2 instances and should continuously run for three years. The CPU utilization of the EC2 instances is expected to be stable and predictable.	Deploy the application to a Reserved instance.
Implement a cost-effective solution for S3 objects that are accessed less frequently.	Create an Amazon S3 lifecycle policy to move the objects to Amazon S3 Standard-IA.
Minimize the data transfer costs between two EC2 instances.	Deploy the EC2 instances in the same Region.
Import the SSL/TLS certificate of the application.	Import the certificate into AWS Certificate Manager or upload it to AWS IAM.

Validate Your Knowledge

When you are feeling confident with your review, it is best to validate your knowledge through sample exams. You can take this practice exam from AWS for free as additional material, but do not expect your real exam to be on the same level of difficulty as this practice exam on the AWS website. **Tutorials Dojo** offers a very useful and well-reviewed set of practice tests for AWS Solutions Architect Associate SAA-C03 takers **here**. Each test contains unique questions that will surely help verify if you have missed out on anything important that might appear on your exam. You can pair our practice exams with this study guide eBook to further help in your exam preparations.

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

Sample SAA-C03 Practice Test Questions

Question 1

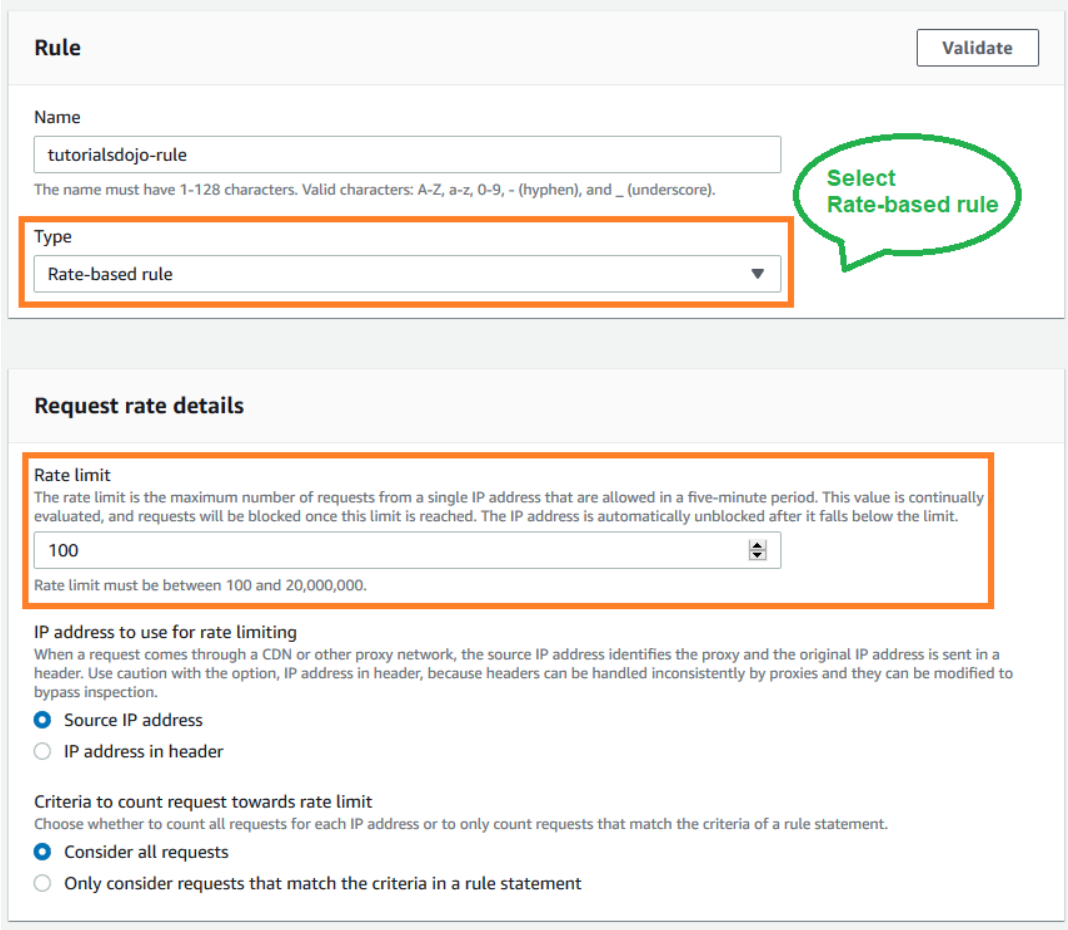
A company hosted an e-commerce website on an Auto Scaling group of EC2 instances behind an Application Load Balancer. The Solutions Architect noticed that the website is receiving a large number of illegitimate external requests from multiple systems with IP addresses that constantly change. To resolve the performance issues, the Solutions Architect must implement a solution that would block the illegitimate requests with minimal impact on legitimate traffic.

Which of the following options fulfills this requirement?

1. Create a regular rule in AWS WAF and associate the web ACL to an Application Load Balancer.
2. Create a custom network ACL and associate it with the subnet of the Application Load Balancer to block the offending requests.
3. Create a rate-based rule in AWS WAF and associate the web ACL to an Application Load Balancer.
4. Create a custom rule in the security group of the Application Load Balancer to block the offending requests.

Correct Answer: 3

AWS WAF is tightly integrated with Amazon CloudFront, the Application Load Balancer (ALB), Amazon API Gateway, and AWS AppSync – services that AWS customers commonly use to deliver content for their websites and applications. When you use AWS WAF on Amazon CloudFront, your rules run in all AWS Edge Locations, located around the world close to your end-users. This means security doesn't come at the expense of performance. Blocked requests are stopped before they reach your web servers. When you use AWS WAF on regional services, such as Application Load Balancer, Amazon API Gateway, and AWS AppSync, your rules run in the region and can be used to protect Internet-facing resources as well as internal resources.



A rate-based rule tracks the rate of requests for each originating IP address and triggers the rule action on IPs with rates that go over a limit. You set the limit as the number of requests per 5-minute time span. You can use this type of rule to put a temporary block on requests from an IP address that's sending excessive requests. Based on the given scenario, the requirement is to limit the number of requests from the illegitimate requests without affecting the genuine requests. To accomplish this requirement, you can use AWS WAF web ACL. There are two types of rules in creating your own web ACL rule: regular and rate-based rules. You need to select the latter to add a rate limit to your web ACL. After creating the web ACL, you can associate it with ALB. When the rule action triggers, AWS WAF applies the action to additional requests from the IP address until the request rate falls below the limit.

Hence, the correct answer is: **Create a rate-based rule in AWS WAF and associate the web ACL to an Application Load Balancer.**

The option that says: **Create a regular rule in AWS WAF and associate the web ACL to an Application Load Balancer** is incorrect because a regular rule only matches the statement defined in the rule. If you need to add a rate limit to your rule, you should create a rate-based rule.

The option that says: **Create a custom network ACL and associate it with the subnet of the Application Load Balancer to block the offending requests** is incorrect. Although NACLs can help you block incoming traffic, this option wouldn't be able to limit the number of requests from a single IP address that is dynamically changing.

The option that says: **Create a custom rule in the security group of the Application Load Balancer to block the offending requests** is incorrect because the security group can only allow incoming traffic. Remember that you can't deny traffic using security groups. In addition, it is not capable of limiting the rate of traffic to your application unlike AWS WAF.

References:
https://docs.aws.amazon.com/waf/latest/developerguide/waf-rule-statement-type-rate-based.html
https://aws.amazon.com/waf/faqs/

Check out this AWS WAF Cheat Sheet:
https://tutorialsdojo.com/aws-waf/

Question 2

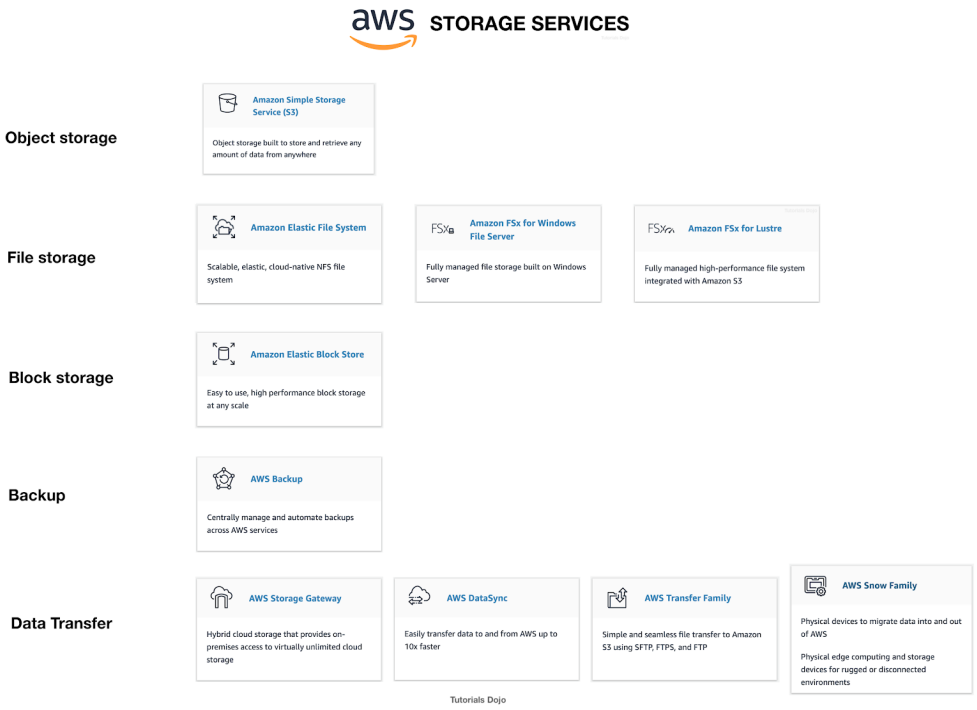
An AI-powered Forex trading application consumes thousands of data sets to train its machine learning model. The application's workload requires a high-performance, parallel hot storage to process the training datasets concurrently. It also needs cost-effective cold storage to archive those datasets that yield low profit.

Which of the following Amazon storage services should the developer use?

1. Use Amazon FSx For Lustre and Amazon EBS Provisioned IOPS SSD (io1) volumes for hot and cold storage respectively.
2. Use Amazon FSx For Lustre and Amazon S3 for hot and cold storage respectively.
3. Use Amazon Elastic File System and Amazon S3 for hot and cold storage respectively.
4. Use Amazon FSx For Windows File Server and Amazon S3 for hot and cold storage respectively.

Correct Answer: 2

Hot storage refers to the storage that keeps frequently accessed data (hot data). **Warm storage** refers to the storage that keeps less frequently accessed data (warm data). **Cold storage** refers to the storage that keeps rarely accessed data (cold data). In terms of pricing, the colder the data, the cheaper it is to store, and the costlier it is to access when needed.



Amazon FSx For Lustre is a high-performance file system for fast processing of workloads. Lustre is a popular open-source **parallel file system** which stores data across multiple network file servers to maximize performance and reduce bottlenecks.

Amazon FSx for Windows File Server is a fully managed Microsoft Windows file system with full support for the SMB protocol, Windows NTFS, Microsoft Active Directory (AD) Integration.

Amazon Elastic File System is a fully-managed file storage service that makes it easy to set up and scale file storage in the Amazon Cloud.

Amazon S3 is an object storage service that offers industry-leading scalability, data availability, security, and performance. S3 offers different storage tiers for different use cases (frequently accessed data, infrequently accessed data, and rarely accessed data).

The question has two requirements:

1. High-performance, parallel hot storage to process the training datasets concurrently.
2. Cost-effective cold storage to keep the archived datasets that are accessed infrequently

In this case, we can use **Amazon FSx For Lustre** for the first requirement, as it provides a high-performance, parallel file system for hot data. On the second requirement, we can use Amazon S3 for storing the cold data. Amazon S3 supports a cold storage system via Amazon S3 Glacier / Glacier Deep Archive.

Hence, the correct answer is: **Use Amazon FSx For Lustre and Amazon S3 for hot and cold storage respectively**.

Using Amazon FSx For Lustre and Amazon EBS Provisioned IOPS SSD (io1) volumes for hot and cold storage respectively is incorrect because the Provisioned IOPS SSD (io1) volumes are designed as a hot storage to meet the needs of I/O-intensive workloads. EBS has a storage option called Cold HDD but it is not used for storing cold data. In addition, EBS Cold HDD is a lot more expensive than using Amazon S3 Glacier / Glacier Deep Archive.

Using Amazon Elastic File System and Amazon S3 for hot and cold storage respectively is incorrect because although EFS supports concurrent access to data, it does not have the high-performance ability that is required for machine learning workloads.

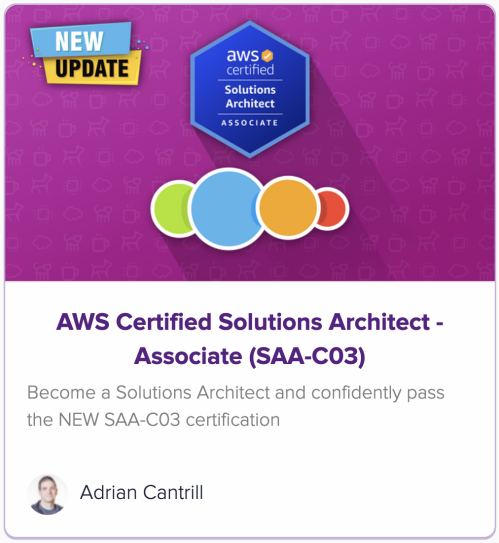
Using Amazon FSx For Windows File Server and Amazon S3 for hot and cold storage respectively is incorrect because Amazon FSx For Windows File Server does not have a parallel file system, unlike Lustre.

References:
https://aws.amazon.com/fsx/
https://docs.aws.amazon.com/whitepapers/latest/cost-optimization-storage-optimization/aws-storage-services.html
https://aws.amazon.com/blogs/startups/picking-the-right-data-store-for-your-workload/

Check out this Amazon FSx Cheat Sheet:
https://tutorialsdojo.com/amazon-fsx/

Additional SAA-C03 Training Materials

There are a few top-rated AWS Certified Solutions Architect Associate SAA-C03 video courses that you can check out as well, which can complement your exam preparations especially if you are the type of person who can learn better through visual courses instead of reading long whitepapers. We recommend the AWS Certified Solutions Architect - Associate video course by Adrian Cantrill which covers both the fundamentals and advanced concepts of the SAA-C03 exam in detail:



We also have a concise **AWS Certified Solutions Architect Associate video training course** that will equip you with the exam-specific knowledge that you need to understand in order to pass the SAA-C03 exam:

Based on the feedback of thousands of our students in our practice test course, the combination of any of these video courses plus our practice tests and this study guide eBook were enough to pass the exam and even get a good score.

Deep Dive on AWS Services

The Solutions Architect Associate exam will test your knowledge on choosing the right service for the right situation. There are many cases wherein two services may seem applicable to a situation, but one of them fulfills the requirement better or the other options have incorrect statements. In this deep dive section, we'll be going through different scenarios that you might encounter in the SAA exam. These scenarios can be related to the behavior of a service feature, integration of different services, or how you should use a certain service. We will go as detailed as we can in this section so that you will not only know the service, but also understand what it is capable of. We will also be adding official AWS references and/or diagrams to supplement the scenarios we'll discuss. Without further ado, let's get right into it.

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/

Storage with Highest IOPS for EC2 Instance

When talking about storage and IOPS in EC2 instances, the first thing that pops into the minds of people is Amazon EBS Provisioned IOPS. Amazon EBS Provisioned IOPS volumes are the highest performing EBS volumes designed for your critical, I/O intensive applications. These volumes are ideal for both IOPS-intensive and throughput-intensive workloads that require extremely low latency. And since they are EBS volumes, your data will also persist even after shutdowns or reboots. You can create snapshots of these volumes and copy them over to your other instances, and much more.

But what if you require really high IOPS, low latency performance, and the data doesn't necessarily have to persist on the volume? If you have this requirement then the instance store volumes on specific instance types might be more preferable than EBS Provisioned IOPS volumes. EBS volumes are attached to EC2 instances virtually, so there is still some latency in there. Instance store volumes are physically attached to the EC2 instances themselves, so your instances are able to access the data much faster. Instance store volumes can come in HDD, SSD or NVME SSD, depending on the instance type you choose. Available storage space will depend on the instance type as well.

Reference:
https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/InstanceStorage.html

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

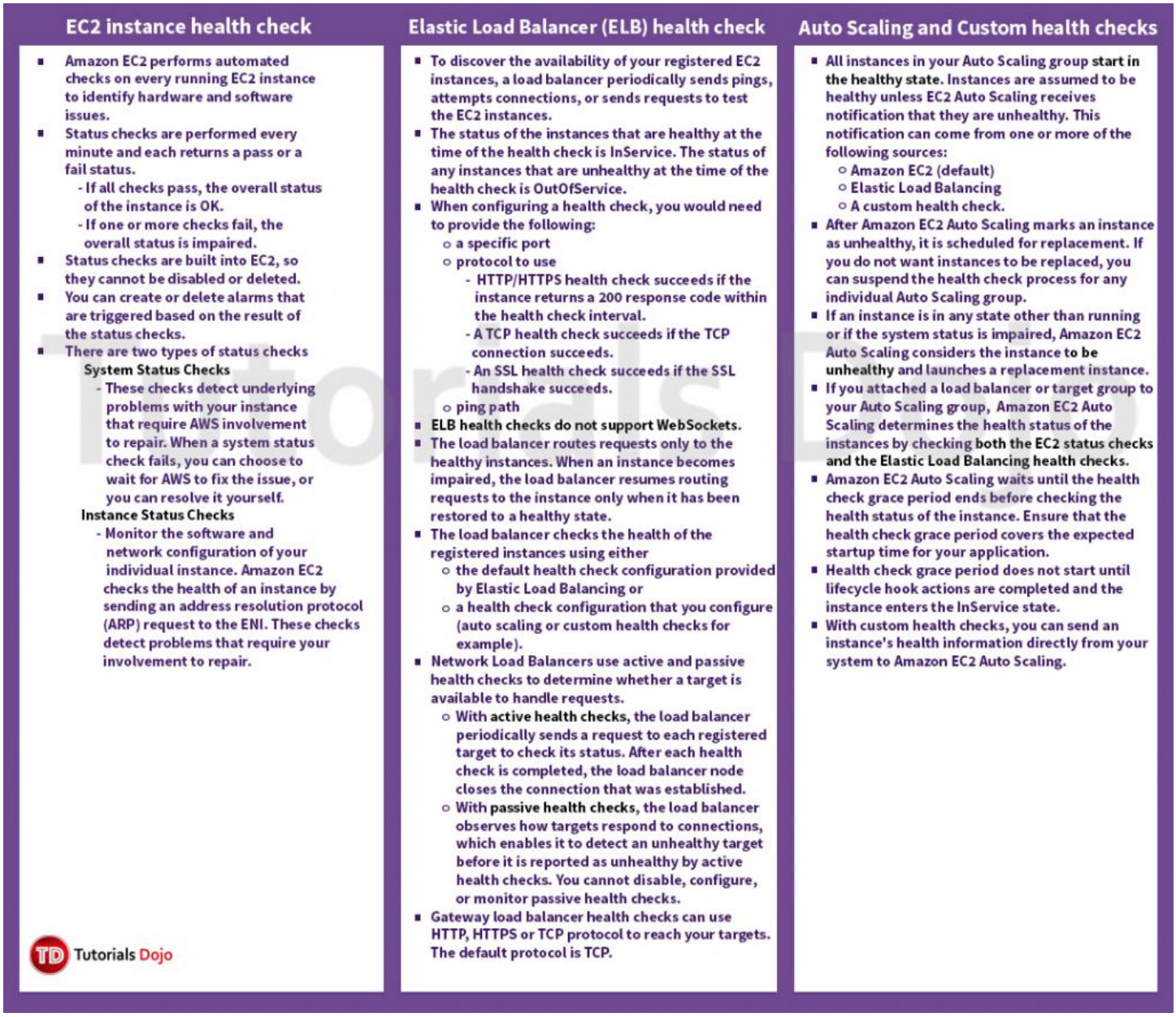
4) **Spot Instances** – Unused EC2 instances that are available for a cheap price, which can reduce your costs significantly. The hourly price for a Spot Instance is called a Spot price. The Spot price of each instance type in each Availability Zone is set by Amazon EC2, and is adjusted gradually based 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.

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

Comparison of Different Types of EC2 Health Checks



EC2 instance health check

- Amazon EC2 performs automated checks on every running EC2 instance to identify hardware and software issues.
- Status checks are performed every minute and each returns a pass or a fail status.
  - If all checks pass, the overall status of the instance is OK.
  - If one or more checks fail, the overall status is impaired.
- Status checks are built into EC2, so they cannot be disabled or deleted.
- You can create or delete alarms that are triggered based on the result of the status checks.
- There are two types of status checks
  - **System Status Checks**
    - These checks detect underlying problems with your instance that require AWS involvement to repair. When a system status check fails, you can choose to wait for AWS to fix the issue, or you can resolve it yourself.
  - **Instance Status Checks**
    - Monitor the software and network configuration of your individual instance. Amazon EC2 checks the health of an instance by sending an address resolution protocol (ARP) request to the ENI. These checks detect problems that require your involvement to repair.

Tutorials Dojo

Elastic Load Balancer (ELB) health check

- To discover the availability of your registered EC2 instances, a load balancer periodically sends pings, attempts connections, or sends requests to test the EC2 instances.
- The status of the instances that are healthy at the time of the health check is InService. The status of any instances that are unhealthy at the time of the health check is OutOfService.
- When configuring a health check, you would need to provide the following:
  - a specific port
  - protocol to use
    - HTTP/HTTPS health check succeeds if the instance returns a 200 response code within the health check interval.
    - A TCP health check succeeds if the TCP connection succeeds.
    - An SSL health check succeeds if the SSL handshake succeeds.
  - ping path
- ELB health checks do not support WebSockets.
- The load balancer routes requests only to the healthy instances. When an instance becomes impaired, the load balancer resumes routing requests to the instance only when it has been restored to a healthy state.
- The load balancer checks the health of the registered instances using either
  - the default health check configuration provided by Elastic Load Balancing or
  - a health check configuration that you configure (auto scaling or custom health checks for example).
- Network Load Balancers use active and passive health checks to determine whether a target is available to handle requests.
  - With **active health checks**, the load balancer periodically sends a request to each registered target to check its status. After each health check is completed, the load balancer node closes the connection that was established.
  - With **passive health checks**, the load balancer observes how targets respond to connections, which enables it to detect an unhealthy target before it is reported as unhealthy by active health checks. You cannot disable, configure, or monitor passive health checks.
- Gateway load balancer health checks can use HTTP, HTTPS or TCP protocol to reach your targets. The default protocol is TCP.

Auto Scaling and Custom health checks

- All instances in your Auto Scaling group start in the healthy state. Instances are assumed to be healthy unless EC2 Auto Scaling receives notification that they are unhealthy. This notification can come from one or more of the following sources:
  - Amazon EC2 (default)
  - Elastic Load Balancing
  - A custom health check.
- After Amazon EC2 Auto Scaling marks an instance as unhealthy, it is scheduled for replacement. If you do not want instances to be replaced, you can suspend the health check process for any individual Auto Scaling group.
- If an instance is in any state other than running or if the system status is impaired, Amazon EC2 Auto Scaling considers the instance to be unhealthy and launches a replacement instance.
- If you attached a load balancer or target group to your Auto Scaling group, Amazon EC2 Auto Scaling determines the health status of the instances by checking both the EC2 status checks and the Elastic Load Balancing health checks.
- Amazon EC2 Auto Scaling waits until the health check grace period ends before checking the health status of the instance. Ensure that the health check grace period covers the expected startup time for your application.
- Health check grace period does not start until lifecycle hook actions are completed and the instance enters the InService state.
- With custom health checks, you can send an instance's health information directly from your system to Amazon EC2 Auto Scaling.

Reference:
https://tutorialsdojo.com/ec2-instance-health-check-vs-elb-health-check-vs-auto-scaling-and-custom-health-check/

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/

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.

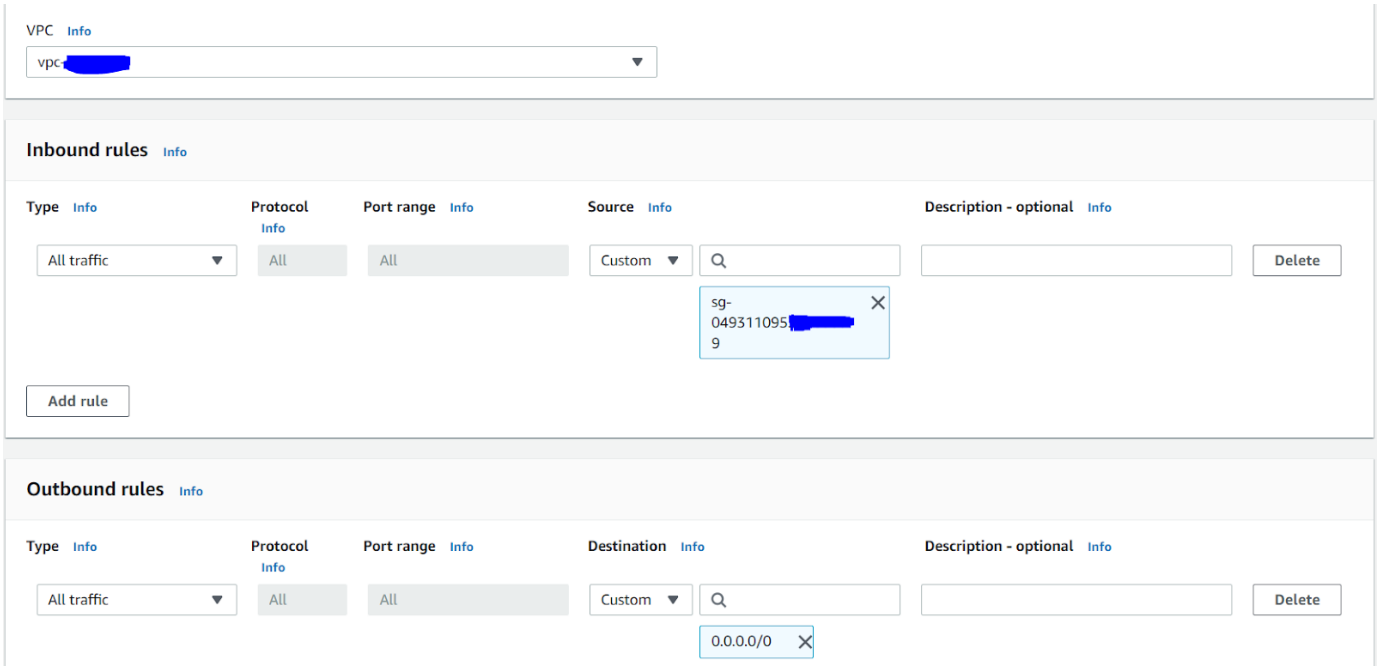
Security groups operate on the instance layer. They serve as virtual firewalls that control inbound and outbound traffic to your VPC resources. Not all AWS services support security groups, but the general idea is that if the service involves servers or EC2 instances then it should also support security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

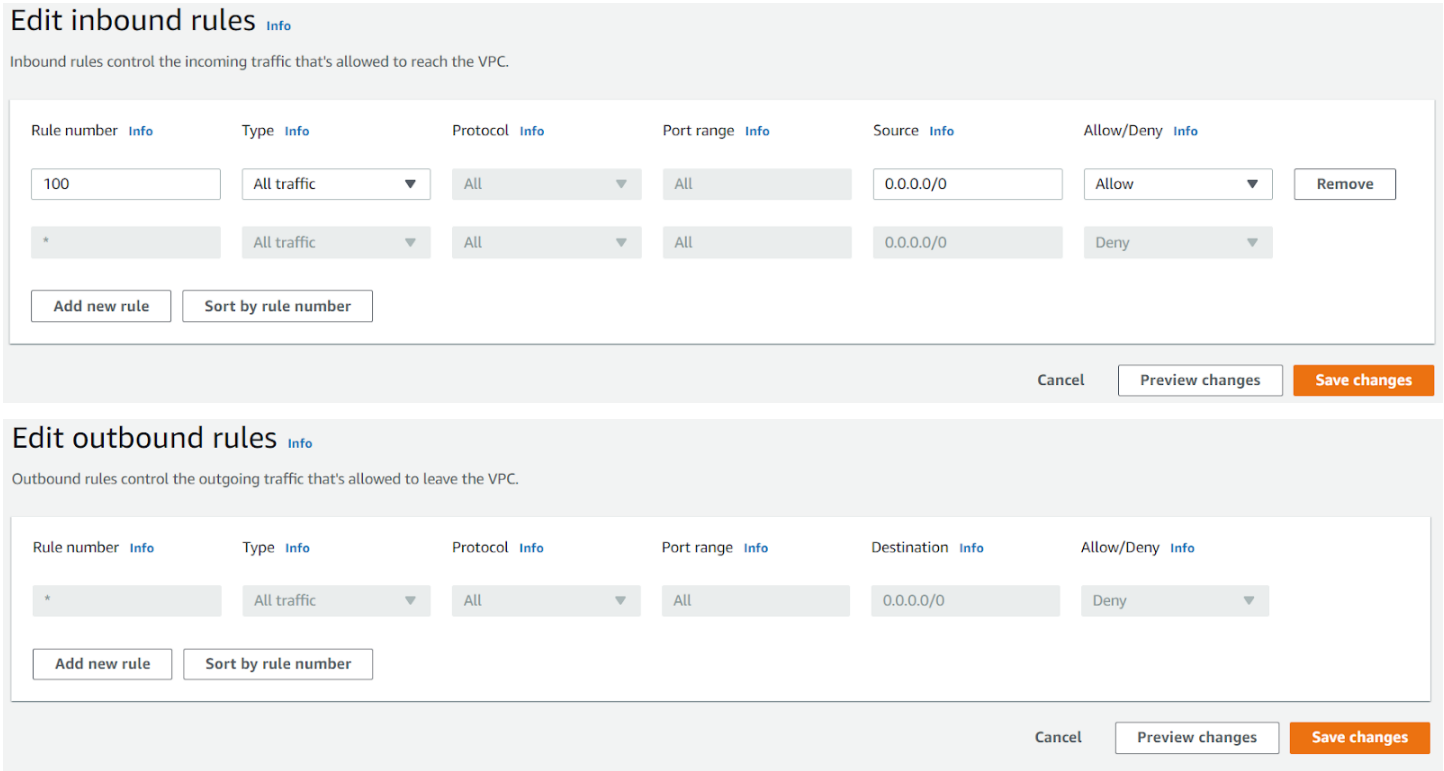


Network ACLs operate on the subnet layer, which means they protect your whole subnet rather than individual instances. Similar to security groups, traffic is managed through the use of rules. A network ACL rule consists of a rule number, traffic type, protocol, port range, source of the traffic for inbound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

Amazon EC2 Auto Scaling

The Amazon EC2 Auto Scaling service helps you ensure that you have the right number of EC2 instances available to handle the load for your web applications. This is a type of horizontal scaling, where you scale out or scale in your applications by dynamically launching or terminating EC2 instances.

Auto Scaling has three major components:

- The Auto Scaling Group
- The configuration templates
- Scaling Options

The Auto Scaling service works by organizing your EC2 instances into groups. An Auto Scaling group is treated as a logical unit for scaling and management purposes. A group must have a minimum, maximum, and desired number of EC2 instances.

A configuration template can either be a launch template, or a launch configuration. This acts as a template for your Auto Scaling Group, containing the AMI ID, the instance type, the key pair, the security groups, block device mapping, et cetera. All of this information is used to launch and configure the new EC2 instances. It is also recommended to use a launch template, rather than a launch configuration, as the latter only offers limited features.

The Scaling Option allows you to choose the suitable scaling behavior of your Auto Scaling Group. You can configure an Auto Scaling Group to scale based on certain conditions, such as the CPU Utilization of your EC2 instances or based on a particular date and time. The scaling option can be set to be dynamic, predictive, or scheduled.

You should also be aware that the process of launching or terminating your instances is not done in an instant. There's a certain lead time when you are launching brand new EC2 instances since AWS has to fetch the AMI, do the required configuration, run the user data that you included and install the custom applications that you specify. All of this must be completed before the instance can accept live incoming requests. This duration is also called "instance warm-up".

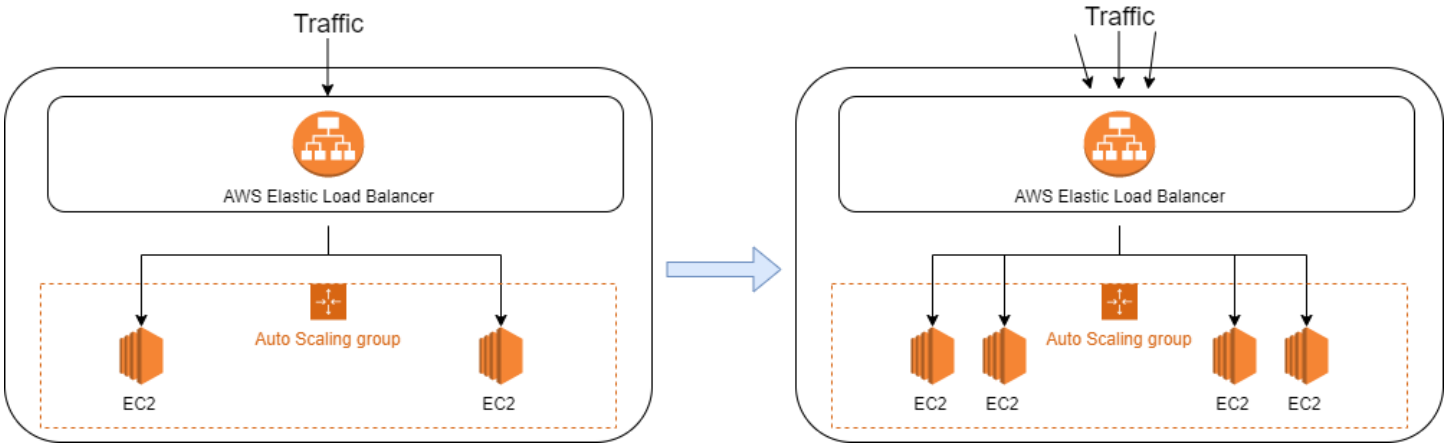
An Auto Scaling group also has a setting called "cool down", which is technically the interval between two scaling actions. This is the number of seconds that must pass before another scaling activity can be executed. This prevents conflicts in your Auto Scaling Group, where one activity is adding an instance while the other one is terminating your resources. You can also set up a termination policy to control which EC2 instances will be terminated first when a scale-in event occurs.

This feature also allows you to do certain actions while the scale-in or scale-out action is being done. The Amazon EC2 Auto Scaling service provides the ability to add lifecycle hooks to your Auto Scaling groups, which allows you to specify the amount of time to complete the lifecycle action before the EC2 instance transitions to the next state. A lifecycle hook enables you to suspend or resume your scaling process to do a variety of tasks, such as sending application logs, doing a system health check, or executing a custom shell script, before an instance gets launched or terminated.

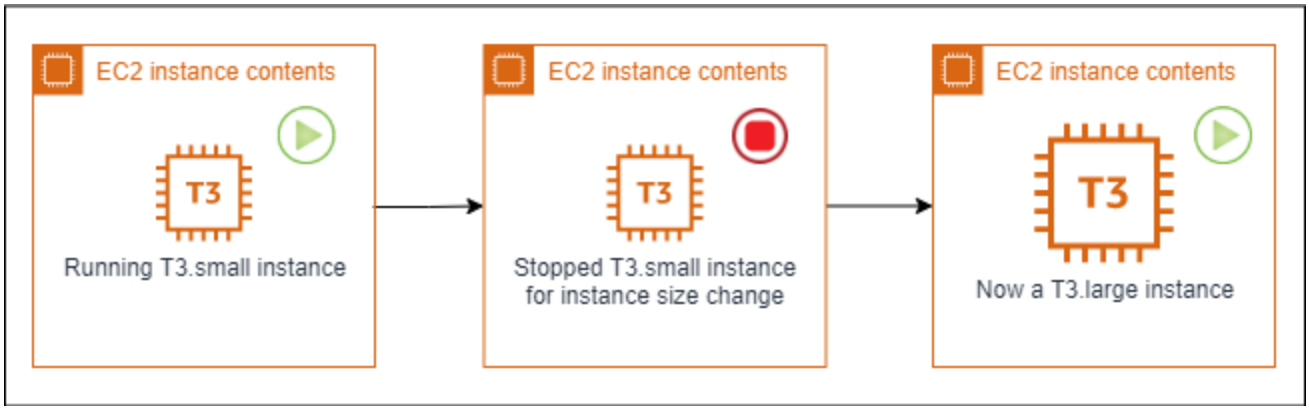
Horizontal Scaling and Vertical Scaling

When you have insufficient capacity for a workload, let's say for example serving a website, there are two ways to scale your resources to accommodate the increasing demand: scale horizontally or scale vertically.

When scaling horizontally, you are adding more servers to the system. More servers mean that workload is distributed to a greater number of workers, which thereby reduces the burden on each server. When you scale horizontally, you need a service such as EC2 auto scaling to manage the number of servers running at a time. You also need an Elastic Load Balancer to intercept and distribute the total incoming requests to your fleet of auto scaling servers. Horizontal scaling is a great way for stateless servers, such as public web servers, to meet varying levels of workloads.



Compared to scaling horizontally, scaling vertically refers to increasing or decreasing the resources of a single server, instead of adding new servers to the system. Vertical scaling is suited for resources that are stateful or have operations difficult to manage in a distributed manner, such as write queries to databases and IOPS sizing in storage volumes. For example, if your EC2 instance is performing slowly, then you can scale up its instance size to obtain more compute and memory capacity. Or when your EBS volumes are not hitting the required IOPS, you can increase their size or IOPS capacity by modifying the EBS volume. Note that for some services such as EC2 and RDS, the instance needs to be stopped before modifying the instance size.



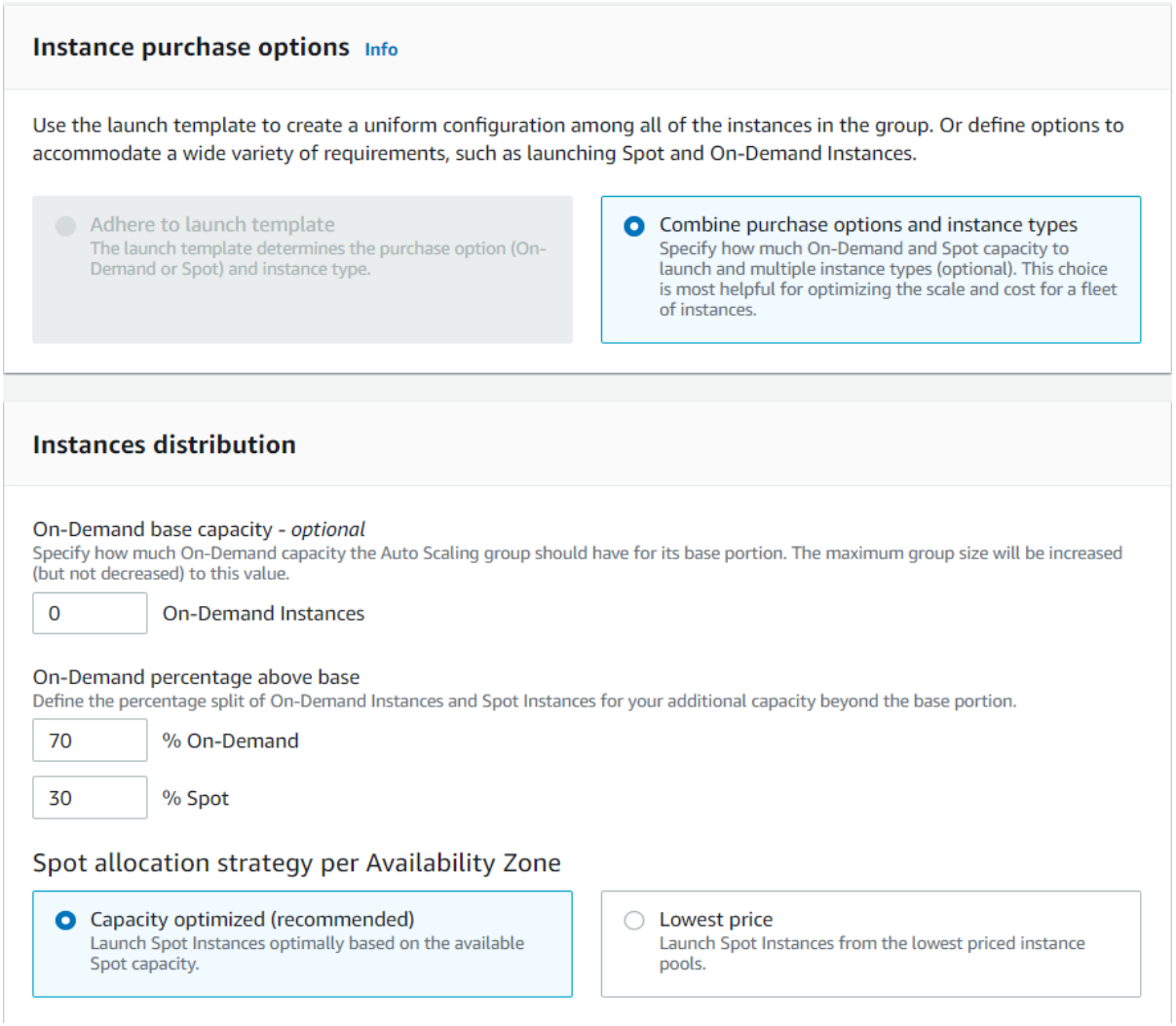
Components of an AWS EC2 Auto Scaling Group

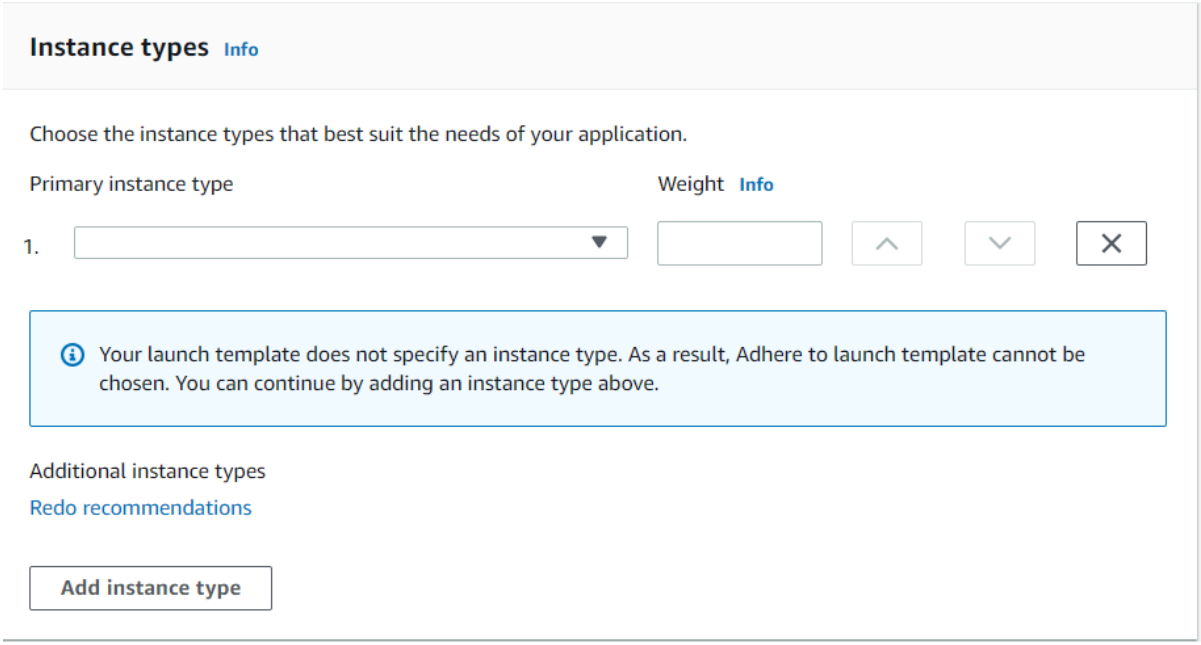
An EC2 Auto Scaling Group has two parts to it: a launch configuration or template that will define your auto scaling instances, and the auto scaling service that performs scaling and monitoring actions.

Creating a launch configuration is similar to launching an EC2 instance. Each launch configuration has a name that uniquely identifies it from your other launch configurations. You provide the AMI that it will use to launch your instances. You also get to choose the instance type and size for your auto scaling instances. You can request spot instances or just use the standard on-demand instances. You can also include an instance profile that will provide your auto scaling instances with permissions to interact with your other services.

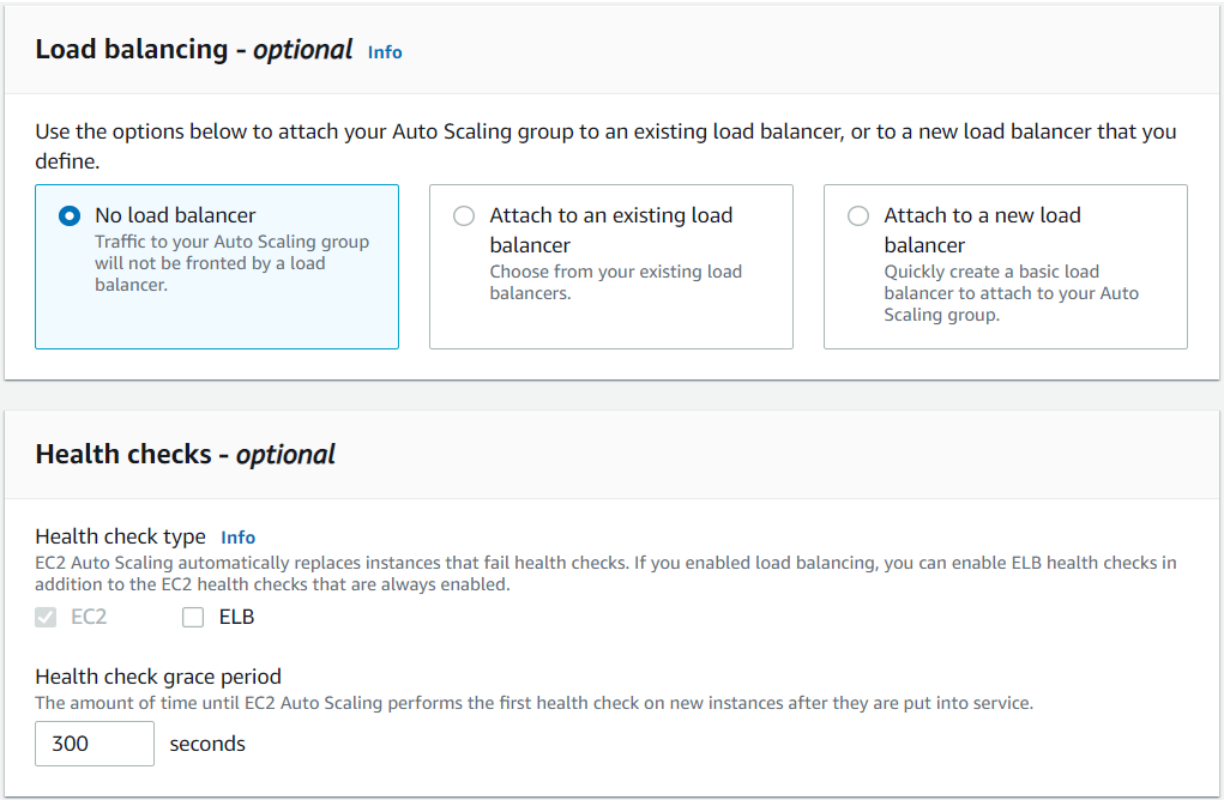
If you need Cloudwatch detailed monitoring, you can enable the option for a cost. Aside from that, you can include user data which will be executed every time an auto scaling instance is launched. You can also choose whether to assign public IP addresses to your instances or not. Lastly, you select which security groups you'd like to apply to your auto scaling instances, and configure EBS storage volumes for each of them. You also specify the key pair to be used to encrypt access.

A launch template is similar to a launch configuration, except that you can have multiple versions of a template. Also, with launch templates, you can create Auto Scaling Groups with multiple instance types and purchase options.


Instance purchase options Info
Use the launch template to create a uniform configuration among all of the instances in the group. Or define options to accommodate a wide variety of requirements, such as launching Spot and On-Demand Instances.
Adhere to launch template
The launch template determines the purchase option (On-Demand or Spot) and instance type.
Combine purchase options and instance types
Specify how much On-Demand and Spot capacity to launch and multiple instance types (optional). This choice is most helpful for optimizing the scale and cost for a fleet of instances.
Instances distribution
On-Demand base capacity - optional
Specify how much On-Demand capacity the Auto Scaling group should have for its base portion. The maximum group size will be increased (but not decreased) to this value.
0
On-Demand Instances
On-Demand percentage above base
Define the percentage split of On-Demand Instances and Spot Instances for your additional capacity beyond the base portion.
70
% On-Demand
30
% Spot
Spot allocation strategy per Availability Zone
Capacity optimized (recommended)
Launch Spot Instances optimally based on the available Spot capacity.
Lowest price
Launch Spot Instances from the lowest priced instance pools.



Once you have created your launch configuration or launch template, you can proceed with creating your auto scaling group. To start off, select the launch configuration/template you'd like to use. Next, you define the VPC and subnets in which the auto scaling group will launch your instances in. You can use multiple Availability Zones and let EC2 Auto Scaling balance your instances across the zones. You can optionally associate a load balancer to the auto scaling group, and the service will handle attaching and detaching instances from the load balancer as it scales. Note that when you do associate a load balancer, you should use the load balancer's health check for instance health monitoring so that when an instance is deemed unhealthy by the load balancer's health check, the load balancer will initiate a scaling event to replace the faulty instance.



Next, you define the size of the auto scaling group — the minimum, desired and the maximum number of instances that your auto scaling group should manage. Specifying a minimum size ensures that the number of running instances do not fall below this count at any time, and the maximum size prevents your auto scaling group from exploding in number. Desired size just tells the auto scaling group to launch this number of instances after you create it. Since the purpose of an auto scaling group *is to auto scale*, you can add cloudwatch monitoring rules that will trigger scaling events once a scaling metric passes a certain threshold. Lastly, you can optionally configure Amazon SNS notifications whenever a scaling event occurs, and add tags to your auto scaling group.

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

Types of EC2 Auto Scaling Policies

Amazon's EC2 Auto Scaling provides an effective way to ensure that your infrastructure is able to dynamically respond to changing user demands. For example, to accommodate a sudden traffic increase on your web application, you can set your Auto Scaling group to automatically add more instances. And when traffic is low, have it automatically reduce the number of instances. This is a cost-effective solution since it only provisions

EC2 instances when you need them. EC2 Auto Scaling provides you with several dynamic scaling policies to control the scale-in and scale-out events.

In this article, we'll discuss the differences between a simple scaling policy, a step scaling policy and a target tracking policy. And we'll show you how to create an Auto Scaling group with step scaling policy applied.

Simple Scaling

Simple scaling relies on a metric as a basis for scaling. For example, you can set a CloudWatch alarm to have a CPU Utilization threshold of 80%, and then set the scaling policy to add 20% more capacity to your Auto Scaling group by launching new instances. Accordingly, you can also set a CloudWatch alarm to have a CPU utilization threshold of 30%. When the threshold is met, the Auto Scaling group will remove 20% of its capacity by terminating EC2 instances.

When EC2 Auto Scaling was first introduced, this was the only scaling policy supported. It does not provide any fine-grained control to scaling in and scaling out.

Target Tracking

Target tracking policy lets you specify a scaling metric and metric value that your auto scaling group should maintain at all times. Let's say for example your scaling metric is the average CPU utilization of your EC2 auto scaling instances, and that their average should always be 80%. When CloudWatch detects that the average CPU utilization is beyond 80%, it will trigger your target tracking policy to scale out the auto scaling group to meet this target utilization. Once everything is settled and the average CPU utilization has gone below 80%, another scale in action will kick in and reduce the number of auto scaling instances in your auto scaling group. With target tracking policies, your auto scaling group will always be running in a capacity that is defined by your scaling metric and metric value.

A limitation though – this type of policy assumes that it should scale out your Auto Scaling group when the specified metric is above the target value. You cannot use a target tracking scaling policy to scale out your Auto Scaling group when the specified metric is below the target value. Furthermore, the Auto Scaling group scales out proportionally to the metric as fast as it can, but scales in more gradually. Lastly, you can use AWS predefined metrics for your target tracking policy, or you can use other available CloudWatch metrics (native and custom). Predefined metrics include the following:

- **ASGAverageCPUUtilization** – Average CPU utilization of the Auto Scaling group.
- **ASGAverageNetworkIn** – Average number of bytes received on all network interfaces by the Auto Scaling group.
- **ASGAverageNetworkOut** – Average number of bytes sent out on all network interfaces by the Auto Scaling group.
- **ALBRequestCountPerTarget** – If the auto scaling group is associated with an ALB target group, this is the number of requests completed per target in the target group.