JON BONSO

AWS CERTIFIED SOLUTIONS ARCHITECT PROFESSIONAL

Tutorials Dojo Study Guide

TABLE OF CONTENTS

INTRODUCTION

In the fast-paced IT industry today, there will always be a growing demand for certified IT Professionals that can design highly available, fault-tolerant, resilient and cost-effective solutions. Companies are spending millions of dollars to optimize the performance of their applications and scale their infrastructure globally to serve customers around the world. They need a reliable and skillful IT staff to migrate their on-premises workload to AWS, reduce their total operating costs, effectively manage complex organizational accounts globally and design new solutions to meet customer demands.

This Study Guide eBook aims to equip you with the necessary knowledge and practical skill sets needed to pass the latest version of the AWS Certified Solutions Architect – Professional exam. We included the essential concepts, exam domains, exam tips, sample questions, cheat sheets, and other relevant information about the latest AWS Certified Solutions Architect – Professional SAP-C02 exam. It begins with the presentation of the exam structure to give you an insight into the question types, exam domains, scoring scheme, and the list of benefits you'll receive once you pass the exam.

We used the official SAP-C02 exam guide for the AWS Certified Solutions Architect Professional exam to structure the contents of this guide, covering all the relevant AWS topics for every exam domain. Various AWS concepts, related AWS services, and technical implementations are covered to provide you with an idea of what to expect on the actual exam.

> ℹ️ **Solutions Architect Professional Exam Notes:**
>
> Don't forget to read the boxed "**exam tips**" (like this one) scattered throughout the eBook, as these are the key concepts that you will likely encounter on your test. The last part of this guide includes a collection of articles that compares two or more similar AWS services to supplement your knowledge.

The AWS Certified Solutions Architect - Professional certification exam is a difficult test to pass; therefore, anyone who wants to take it must allocate ample time for review. The exam registration costs hundreds of dollars, which is why we spent considerable time and effort to ensure that this study guide provides you with the essential and relevant knowledge to increase your chances of passing the Solutions Architect Professional exam.

*** **Note:** This eBook is meant to be just a supplementary resource when preparing for the exam. We highly recommend working on your hands-on labs and practice exams to further expand your knowledge and improve your test taking skills.*

AWS CERTIFIED SOLUTIONS ARCHITECT PROFESSIONAL EXAM OVERVIEW

SAP-C02 Exam Details

The AWS Certified Solutions Architect – Professional (SAP-C02) exam is one of the two Professional-level certification tests of the AWS Certification program. This particular exam is meant for individuals who perform a solutions architect role in their current company/organization. The SAP-C02 exam validates the person's general IT knowledge, advanced technical skills, and experience in designing optimized AWS solutions that are based on the AWS Well-Architected Framework.

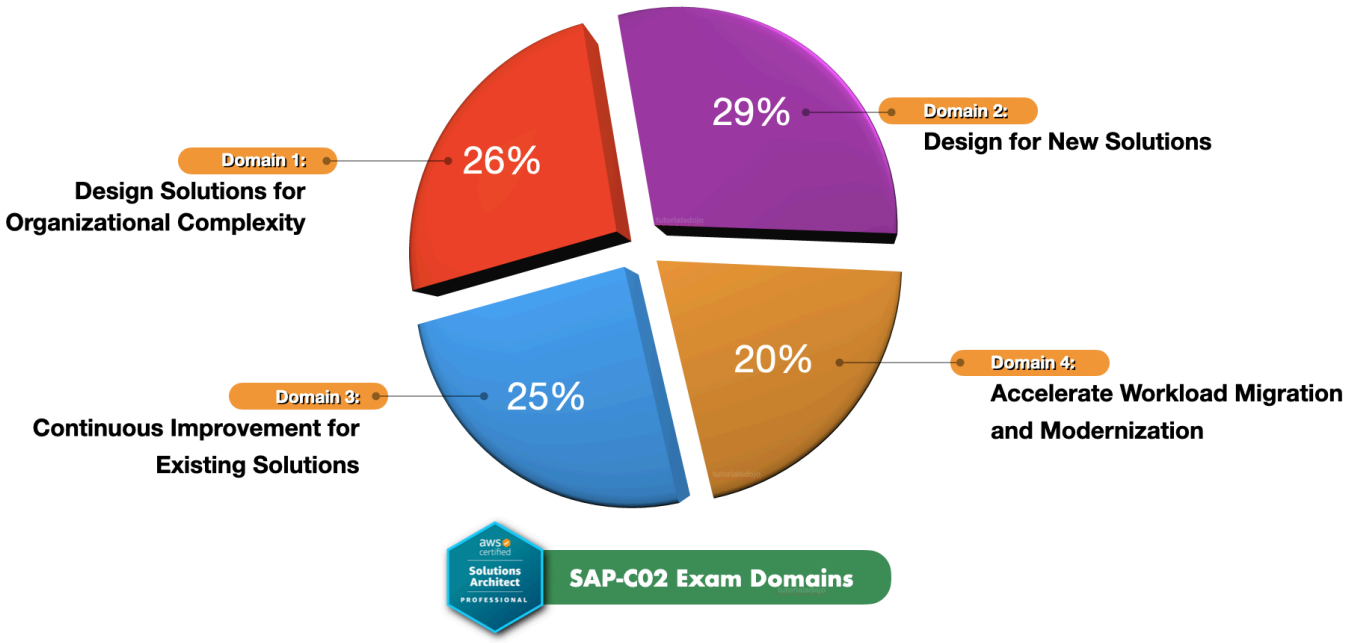
This Pro-level certification exam is composed of 75 multiple-choice or multiple-response scenario-based questions that you must complete within 180 minutes or 3 hours. The "multiple-choice" question type has one correct answer and three incorrect responses, while the "multiple response" item has two or more right responses out of five or more options. Like other AWS certification exams, you can take the SAP-C02 exam from a local testing center or online from your home.

Exam Code:	SAP-C02
Release Date:	November 2022
Prerequisites:	None
No. of Questions:	75
Score Range:	100 - 1000
Cost:	300 USD
Passing Score:	750/1000
Time Limit:	3 hours (180 minutes)
Format:	Scenario-based. Multiple choice/multiple answers.
Delivery Method:	Testing center or online proctored exam.

The AWS Certified Solutions Architect – Professional (SAP-C02) exam has no prerequisites, so you can take this exam directly. The score range is from 100 to 1000 and you need to score at least 750 to pass the test. Thus, the passing score is at 75% (750/1000) compared with just 72% for the Associate-level exams.

Exam Domains

The AWS Certified Solutions Architect Professional (SAP-C02) exam has 4 different domains, each with corresponding weight and topic coverage. The exam domains are as follows: Design Solutions for Organizational Complexity (26%), Design for New Solutions (29%), Continuous Improvement for Existing Solutions (25%) and lastly, Accelerate Workload Migration and Modernization (20%):



The list of exam domains can be found on the official Exam Guide for the AWS Certified Solutions Architect - Professional exam. Each exam domain is comprised of several task statements. A task statement is a sub-category of the exam domain that contains the necessary topics, knowledge, concepts, and skills for you to accomplish a particular task or activity in AWS.

As seen in the above pie graph, the first domain covers 26% of the overall test while the second domain covers 29%, which represents the biggest chunk of the exam. This is followed by the third and fourth domains with 25% and 20% coverage, respectively.

Let's look at each of these domains one by one.

Domain 1: Design Solutions for Organizational Complexity

The first exam domain is all about checking your knowledge in properly setting up cloud architectures for large organizations with multiple business units. You have to be knowledgeable in launching and maintaining a centralized cloud organization with multiple AWS accounts across several geographical regions using the AWS Organizations service as well the other governance and management services. This includes the skill of setting up cost-effective, resilient and reliable network connectivity for your hybrid cloud and shared AWS resources.

This is the second biggest domain in the exam with 26% percent coverage; therefore, you must allocate significant time to study the various concepts covered in this domain. It also includes the security posture of your cloud solutions to ensure that they conform with the guardrails and regulations of your organization.

The series of scenarios that you will encounter in this domain checks your know-how in doing these tasks:

- Architect network connectivity strategies.
- Prescribe security controls.
- Design reliable and resilient architectures.
- Design a multi-account AWS environment.
- Determine cost optimization and visibility strategies.

Domain 2: Design for New Solutions

The second exam domain (“Design Resilient Architectures”) is all about designing resilient architectures in AWS. It is the biggest domain in the exam, with 29% percent coverage so you must also allocate a lot of time to understand the different concepts covered in this domain.

The questions that you will encounter in this domain will challenge your knowledge in:

- Design a deployment strategy to meet business requirements.
- Design a solution to ensure business continuity.
- Determine security controls based on requirements.
- Design a strategy to meet reliability requirements.
- Design a solution to meet performance objectives.
- Determine a cost optimization strategy to meet solution goals and objectives.

Domain 3: Continuous Improvement for Existing Solutions

The third domain is all about continuously improving your existing cloud solutions in AWS. This has an exam coverage of 25 percent and revolves around improving the security, performance, reliability and overall operational excellence of your solutions. It also includes the topic of cost optimization and the skill involved to easily identify them on an existing cloud architecture. You should prepare for:

- Determine a strategy to improve overall operational excellence.
- Determine a strategy to improve security.
- Determine a strategy to improve performance.
- Determine a strategy to improve reliability
- Identify opportunities for cost optimizations.

Domain 4: Accelerate Workload Migration and Modernization

The last exam domain revolves around designing cost-optimized architectures. It comprises 20% of the exam coverage, so you have to limit the time you spend reviewing the concepts under this domain. As you might have guessed, this domain is all about the costs of your cloud architecture and the different ways to reduce operational expenditures. This domain checks if you possess the knowledge in doing following tasks:

- Select existing workloads and processes for potential migration
- Determine the optimal migration approach for existing workloads.
- Determine a new architecture for existing workloads.
- Determine opportunities for modernization and enhancements.

These are the four exam domains that you should be familiar with when you start your exam preparations. Again, the SAP-C02 exam is primarily focused on security, so make sure that you focus on the “Design Solutions for Organizational Complexity” domain and all the related knowledge areas in its task statements.

I highly recommend that you read the official exam guide for the AWS Certified Solutions Architect Professional exam from cover to cover. Pay close attention to the topics included, and don't forget to read the Appendix section, which contains a list of related AWS services that will appear in the exam.

The Old SAP-C01 and New SAP-C02 Exam Difference

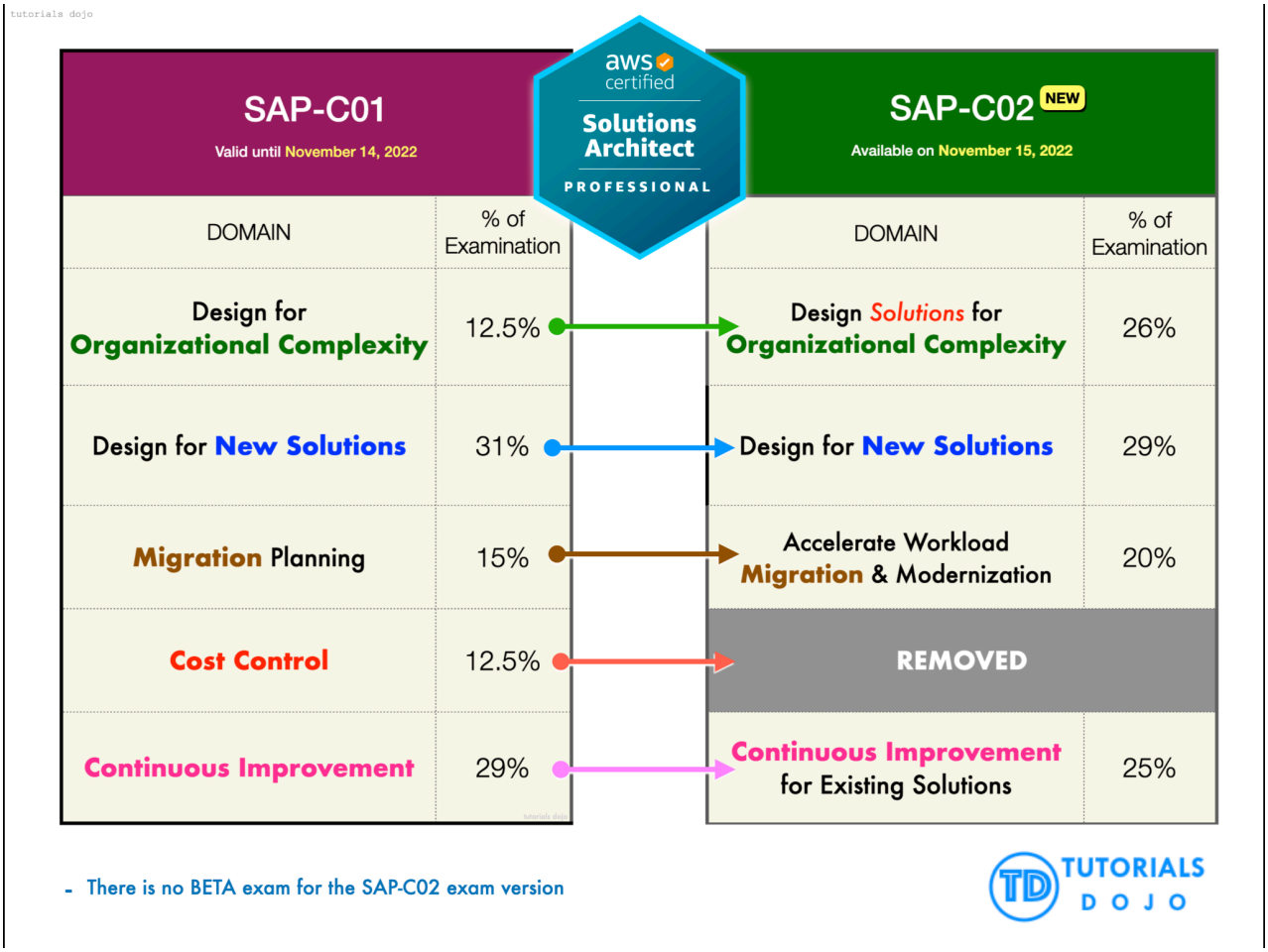
The SAP-C02 is the 3rd iteration of the AWS Certified Solutions Architect Professional exam, which was initially launched about a decade ago. The very first version of this certification test was released on May 2014 with an exam code of SAP-C00. AWS released the second version on February 2019, which is 5 years from the first one, with an exam code of SAP-C01. The third, and also the latest, exam version of the AWS Certified Solutions Architect – Professional exam has an exam code of SAP-C02 which became available on November 15, 2022.

I've already taken and passed the previous versions of this exam in the past, but this week, I took the latest exam version to ensure that our SAP-C02 reviewers here at Tutorials Dojo are still on par with the actual exam. I took the first version of this test (SAP-C00) in April 2018, which is almost 5 years from today, and then the second exam version (SAP-C01) in February 2019. I took the recently launched SAP-C02 exam version on November 2022.

It's noticeable that the length of time for the SAP-C01 version to be replaced with the new SAP-C02 is 3 years. Based on this trend, I'm assuming that the 4th iteration of this certification test will be coming in 2025, with an exam code of SAP-C03. This is just a rough estimation as it depends on the number of feature changes and brand-new services that AWS releases.

Through all these exam iterations, I would say that there were significant changes from the first SAP-C00 exam compared with the second version (SAP-C01). Back in 2018, when I took the SAP-C00 exam, the questions were long, but the number of AWS services included was not that extensive. It was when AWS released the SAP-C01 exam version where the AWS Certified Solutions Architect Professional exam topics covered a wide range of AWS services and other 3rd party systems.

The SAP-C02 still resembles the old format of the recently decommissioned SAP-C01 exam. It has a mix of long questions (with 3-4 paragraphs) and short ones (1-2 paragraphs). The same goes for the answers as well, with multiple-response questions where you have to select 2 or 3 items out of 5 or 6 options. In comparison with the SAA-C03 and other Associate-level exams, the AWS Certified Solutions Architect Professional exam has significantly more multiple-response questions than multiple-choice items.



The table above shows the difference between the SAP-C01 exam domains versus the new SAP-C02 exam. From 5 exam domains, it is now down to 4, with changes in the coverage percentage on each domain. The Design for Organizational Complexity domain was renamed to Design Solutions for Organizational Complexity, and its exam coverage was bumped up from 12.5% to 26% while the Migration Planning domain was upgraded from 15% to 20% and was also renamed to Accelerate Workload Migration & Modernization. Speaking of Modernization, expect to see a lot of containerized architectures in the exam, which involves Kubernetes, Amazon ECS, Amazon EKS, Serverless, and other related services.

You can also see here that the Cost Control domain was removed. Don't think that there will be no cost-related questions in the exam. It only means that the cost factor has been distributed to other exam domains. So, for example, the Design Solutions for Organizational Complexity domain includes cost management and billing setup topics for multi-account AWS environments using AWS Organizations. The exam coverage for the Design for New Solutions domain was reduced from 31% to 29% only. The same goes for the Continuous Improvement domain, which was at 29% before but is now at 25%, including a slight name change to "Continuous Improvement for Existing Solutions" as well.

Exam Topics for SAP-C02

This list of the AWS services covered for the AWS Certified Solutions Architect Professional exam is quite long which simply proves how wide the topics are covered in the test. Don't get scared by the sheer length of the list but rather, start by reviewing the AWS services based on its section. With proper training and study strategy, you would be able to adequately cover these topics on your review:

Analytics:
- Amazon Athena
- AWS Data Exchange
- Amazon EMR
- AWS Glue
- Amazon Managed Service for Apache Flink Studio
- Amazon Data Firehose
- Amazon Kinesis Data Streams
- AWS Lake Formation
- Amazon Managed Streaming for Apache Kafka (Amazon MSK)
- Amazon OpenSearch Service
- Amazon Quick

Application Integration:
- Amazon AppFlow
- AWS AppSync
- Amazon EventBridge (Amazon CloudWatch Events)
- Amazon MQ
- Amazon Simple Notification Service (Amazon SNS)
- Amazon Simple Queue Service (Amazon SQS)
- AWS Step Functions

Business Applications:
- Alexa for Business
- Amazon Simple Email Service (Amazon SES)

Blockchain:
- Amazon Managed Blockchain

Machine Learning:
- Amazon Comprehend
- Amazon Fraud Detector
- Amazon Kendra
- Amazon Lex
- Amazon Personalize
- Amazon Polly
- Amazon Rekognition
- Amazon SageMaker AI
- Amazon Textract
- Amazon Transcribe
- Amazon Translate

Management and Governance:
- AWS CloudFormation
- AWS CloudTrail
- Amazon CloudWatch
- Amazon CloudWatch Logs
- AWS Command Line Interface (AWS CLI)
- AWS Compute Optimizer
- AWS Config
- AWS Control Tower
- AWS License Manager
- Amazon Managed Grafana
- Amazon Managed Service for Prometheus
- AWS Management Console
- AWS Organizations
- AWS Personal Health Dashboard
- AWS Proton
- AWS Service Catalog
- Service Quotas
- AWS Systems Manager
- AWS Trusted Advisor
- AWS Well-Architected Tool

Cloud Financial Management:
AWS Budgets
AWS Cost and Usage Report
AWS Cost Explorer
Savings Plans

Compute:
AWS App Runner
AWS Auto Scaling
AWS Batch
Amazon EC2
Amazon EC2 Auto Scaling
AWS Elastic Beanstalk
Amazon Elastic Kubernetes Service (Amazon EKS)
Elastic Load Balancing
AWS Fargate
AWS Lambda
Amazon Lightsail
AWS Outposts
AWS Wavelength

Containers:
Amazon Elastic Container Registry (Amazon ECR)
Amazon Elastic Container Service (Amazon ECS)
Amazon ECS Anywhere
Amazon Elastic Kubernetes Service (Amazon EKS)
Amazon EKS Anywhere
Amazon EKS Distro

Database:
Amazon Aurora
Amazon Aurora Serverless
Amazon DocumentDB (with MongoDB compatibility)
Amazon DynamoDB
Amazon ElastiCache
Amazon Keyspaces (for Apache Cassandra)
Amazon Neptune
Amazon RDS
Amazon Redshift
Amazon Timestream

Migration and Transfer:
AWS Application Discovery Service
AWS Application Migration Service (AWS MGN)
AWS Database Migration Service (AWS DMS)
AWS DataSync
AWS Migration Hub
AWS Schema Conversion Tool (AWS SCT)
AWS Transfer Family

Security, Identity, and Compliance:
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)
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 Security Token Service (AWS STS)
AWS Shield
AWS IAM Identity Center
AWS WAF

Media Services:
- Amazon Elastic Transcoder
- Amazon Kinesis Video Streams

Developer Tools:
AWS CodeArtifact
AWS CodeBuild
AWS CodeDeploy
Amazon CodeGuru
AWS CodePipeline
AWS X-Ray

Frontend Web and Mobile:
AWS Amplify
Amazon API Gateway
AWS Device Farm
Amazon Pinpoint

Internet of Things:
AWS IoT Core
AWS IoT Device Defender
AWS IoT Device Management
AWS IoT Events
AWS IoT Greengrass
AWS IoT SiteWise
AWS IoT Things Graph
AWS IoT 1-Click

Networking and Content Delivery:
- Amazon CloudFront
- AWS Direct Connect
- Elastic Load Balancing (ELB)
- AWS Global Accelerator
- AWS PrivateLink
- Amazon Route 53
- AWS Transit Gateway
- Amazon VPC
- AWS VPN

Storage:
- AWS Backup
- Amazon Elastic Block Store (Amazon EBS)
- AWS Elastic Disaster Recovery (CloudEndure Disaster Recovery)
- Amazon Elastic File System (Amazon EFS)
- Amazon FSx (for all types)
- Amazon S3
- Amazon S3 Glacier
- AWS Storage Gateway

End User Computing:
- Amazon AppStream 2.0
- Amazon WorkSpaces

Exam Scoring System

You can get a score from 100 to 1,000 with a minimum passing score of **750** when you take the Solutions Architect Professional exam. AWS uses a scaled scoring model to equate scores across multiple exam types that may have different difficulty levels. The complete score report will be sent to you by email after a few days.

Individuals who unfortunately do not pass the AWS exam must wait 14 days before they are allowed to retake the exam. Fortunately, there is no hard limit on exam attempts until you pass the exam. Take note that on each attempt, the full registration price of the AWS exam must be paid. Within 5 business days of completing your exam, your AWS Certification Account will have a record of your complete exam results.

		Score Performance	
Section	**% of Scored Items**	**Needs Improvement**	**Meets Competencies**
Domain 1: Design Solutions for Organizational Complexity	26%		
Domain 2: Design for New Solutions	29%		
Domain 3: Continuous Improvement for Existing Solutions	25%		
Domain 4: Accelerate Workload Migration and Modernization	20%		

Disclaimer: AWS Certification exams are designed to make pass/fail decisions based on the total exam score. Section level results are designed to provide direction on areas where a candidate may be weak. Candidates should exercise caution when interpreting the above section level score information as it is less reliable than the total exam score and not intended to guide future test performance.

The score report, as shown above, contains a table of your performance at each section/domain, which indicates whether you met the competency level required for these domains or not. AWS uses a compensatory scoring model, which means that you do not necessarily need to pass each and every individual section, only the overall examination. Each section has a specific score weighting that translates to the number of questions; hence, some sections have more questions than others. The Score Performance table highlights your strengths and weaknesses that you need to improve on.

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.

- **Certification Digital Badges** - You can showcase your achievements to your colleagues and employers with digital badges on your email signatures, Linkedin profile, or on your social media accounts. You can also show your Digital Badge to gain exclusive access to Certification Lounges at AWS re:Invent, regional Appreciation Receptions, and select AWS Summit events. To view your badges, simply go to the "Digital Badges" section of your AWS Certification Account.

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

AWS CERTIFIED SOLUTIONS ARCHITECT PROFESSIONAL EXAM - STUDY GUIDE AND TIPS

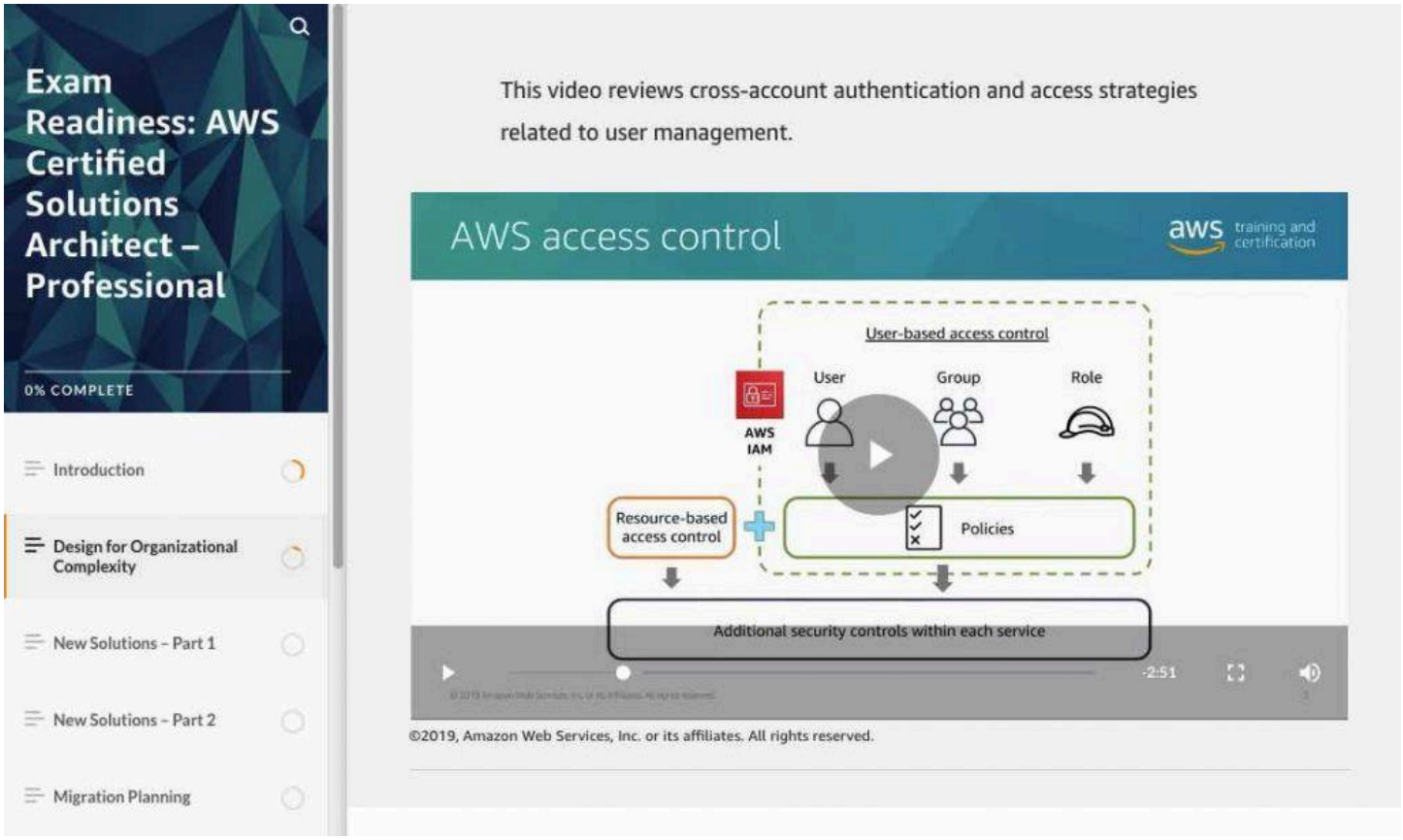
Few years ago, before you can take the AWS Certified Solutions Architect Professional exam (or SA Pro for short), you would first have to pass the associate level exam of this track. This is to ensure that you have sufficient knowledge and understanding on architecting in AWS, before tackling the more difficult certification. In October 2018, AWS removed this ruling so that there are no more prerequisites for taking the Professional level exams. You now have the freedom to directly pursue this certification if you wish to.

This certification is truly a levelled-up version of the AWS Solutions Architect Associate certification. It examines your capability to create well-architected solutions in AWS, but on a grander scale and with more difficult requirements. Because of this, we recommend that you go through our exam preparation guide for the AWS Certified Solutions Architect Associate and even the AWS Certified Cloud Practitioner if you have not done so yet. These study guides contain important materials that will be crucial for passing the SAP-C02 exam.

Study Materials

The official AWS sample questions, Whitepapers, FAQs, AWS Documentation, Re:Invent videos, forums, labs, AWS cheat sheets, AWS practice exams, and your own personal technical experiences are what you will need to pass the exam. Since the SA Pro is one of the most difficult AWS certification exams out there, you have to prepare yourself with every study material you can get your hands on. To learn more details regarding your exam, go through this AWS exam blueprint as it discusses the various domains they will test you on.

AWS has a digital course called Exam Readiness: AWS Certified Solutions Architect – Professional, which is a short video lecture that discusses what to expect on the AWS Certified Solutions Architect – Professional exam. It should sufficiently provide an overview of the different concepts and practices that you'll need to know about. Each topic in the course will also contain a short quiz right after you finish its lecture to help you lock in the important information.



For whitepapers, aside from the ones listed down in our Solutions Architect Associate and Cloud Practitioner exam guides, you should also study the following:

- Security best practices for AWS Key Management Service
- Encryption of Data at Rest
- Web Application Hosting in the AWS Cloud
- Practicing Continuous Integration and Continuous Delivery on AWS
- Microservices architecture on AWS
- AWS Well-Architected Framework
- Using Amazon Web Services for Disaster Recovery
- AWS Architecture Center architecture whitepapers

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

AWS Services to Focus On

Generally, as a soon-to-be AWS Certified SA Pro, you should already have a thorough understanding of every service and feature in AWS. But for the purpose of this review, give more attention on the following services since they are common topics in the SA Pro exams:

1. AWS Organizations
    1. Know how to create organizational units (OUs), service control policies (SCPs), and any additional parameters in AWS Organizations.
    2. There might be scenarios where the master account needs access to member accounts. Your options can include setting up OUs and SCPs, delegating an IAM role, or providing cross account access.
    3. Differentiate SCP from IAM policies.
    4. You should also know how to integrate AWS Organizations with other services such as CloudFormation, Service Catalog, and IAM to manage resources and user access.
    5. Lastly, read how you can save on costs by enabling consolidated billing in your organizations, and what would be the benefits of enabling all features.
2. AWS Application Migration Service (AWS MGN)
    1. Study the different ways to migrate on-premises servers to the AWS Cloud.
    2. Also study how you can perform the migration in a secure and reliable manner.
    3. You should be aware of the types of objects that AWS MGN can migrate for you, such as live applications, databases, web servers, and other workloads.
3. AWS Database Migration Service + Schema Conversion Tool
    1. Aside from server and application migration, you should also know how you can move on-premises databases to AWS, and not just to RDS but to other services as well as Aurora and RedShift.
    2. Read over what schemas can be converted by SCT.
4. AWS Serverless Application Model
    1. The AWS SAM has a syntax of its own. Study the syntax and how AWS SAM is used to deploy serverless applications through code.
    2. Know the relationship between SAM and CloudFormation. **Hint:** You can use these two together.
5. AWS Systems Manager
    1. Study the different features under Systems Manager and how each feature can automate EC2-related processes. Patch Manager and Maintenance Windows are often used together to perform automated patching. It allows for easier setup and better control over patch baselines, rather than using a cron job within an EC2 instance or using Amazon EventBridge.
    2. It is also important to know how you can troubleshoot EC2 issues using Systems Manager.
    3. Parameter Store allows you to securely store a string in AWS, which can be retrieved anywhere in your environment. You can use this service instead of AWS Secrets Manager if you don't need to rotate your secrets.

6. AWS CI/CD - Study the different CI/CD tools in AWS, from function to features to implementation. It would be very helpful if you can create your own CI/CD pipeline as well using the services below.
    1. CodeBuild
    2. CodeDeploy
    3. CodePipeline
7. AWS Service Catalog
    1. This service is also part of the automation toolkit in AWS. Study how you can create and manage portfolios of approved services in Service Catalog, and how you can integrate these with other technologies such as AWS Organizations.
    2. You can enforce tagging on services using service catalog. This way, users can only launch resources that have the tags you defined.
    3. Know when Service Catalog is a better option for resource control rather than AWS Cloudformation. A good example is when you want to create a customized portfolio for each type of user in an organization and selectively grant access to the appropriate portfolio.
8. AWS Direct Connect (DX)
    1. You should have a deep understanding of this service. Questions commonly include Direct Connect Gateway, public and private VIFs and LAGs.
    2. Direct Connect is commonly used for connecting on-premises networks to AWS, but it can also be used to connect different AWS Regions to a central datacenter. For these kinds of scenarios, take note of the benefits of Direct Connect such as dedicated bandwidth, network security, multi-Region and multi-VPC connection support.
    3. Direct Connect is also used along with a failover connection, such as a secondary DX line or IPsec VPN. The correct answer will depend on specific requirements like cost, speed, ease of management, etc.
    4. Another combination that can be used to link different VPCs is Transit Gateway + DX.
9. AWS CloudFormation - Your AWS exam might include a lot of scenarios that involve Cloudformation, so take note of the following:
    1. You can use CloudFormation to enforce tagging by requiring users to only use resources that CloudFormation launched.
    2. CloudFormation can be used for managing resources across different AWS accounts in an Organization using StackSets.
    3. CloudFormation is often compared to AWS Service Catalog and AWS SAM. The way to approach this in the exam is to know what features are supported by CloudFormation that cannot be performed in a similar fashion with Service Catalog or SAM.
10. Amazon VPC (in depth)
    1. Know the ins and outs of NAT Gateways and NAT instances, such as supported IP protocols, which types of packets are dropped in a cut connection, etc.
    2. Study about transit gateway and how it can be used together with Direct Connect.
    3. Remember longest prefix routing.
    4. Compare VPC peering to other options such as Site to Site VPN. Know what components are in use: Customer gateway, Virtual Private Gateway, etc.

11. Amazon ECS
    1. Differentiate task role from task execution role.
    2. Compare using ECS compute instances from the Fargate serverless model.
    3. Study how to link together ECS and ECR with CI/CD tools to automate deployment of updates to your Docker containers.
12. Elastic Load Balancer (in depth)
    1. Differentiate the internet protocols used by each type of ELB for listeners and target groups: HTTP, HTTPS, TCP, UDP, TLS.
    2. Know how you can configure load balancers to forward client IP to target instances.
    3. Know how you can secure your ELB traffic through the use of SSL and WAF. SSL can be offloaded on either the ELB or CloudHSM.
13. Elastic Beanstalk
    1. Study the different deployment options for Elastic Beanstalk.
    2. Know the steps in performing a blue/green deployment.
    3. Know how you can use traffic splitting deployment to perform canary testing
    4. Compare Elastic Beanstalk's deployment options to CodeDeploy.
14. WAF and Shield
    1. Know at what network layer WAF and Shield operate in
    2. Differentiate security capabilities of WAF and Shield Advanced, especially with regards to DDoS protection. A great way to determine which one to use is to look at the services that need the protection and if cost is a factor. You may also visit this AWS documentation for additional details.
15. Amazon Workspaces vs Amazon Appstream
    1. Workspaces is best for virtual desktop environments. You can use it to provision either Windows or Linux desktops in just a few minutes and quickly scale to provide thousands of desktops to workers across the globe.
    2. Appstream is best for standalone desktop applications. You centrally manage your desktop applications on AppStream 2.0 and securely deliver them to any computer.
16. Amazon Workdocs - It is important to determine what features makes Workdocs unique compared to using S3 and EFS. Choose this service if you need a secure document storage where you can collaborate in real-time with others and manage access to the documents.
17. Elasticache vs DAX vs Aurora Read Replicas
    1. Know your caching options especially when it comes to databases.
    2. If there is a feature that is readily integrated with the database, it would be better to use that integrated features instead for less overhead.
18. S3 Transfer Acceleration vs Direct Connect vs VPN - These three services are heavily used for data migration purposes. Read the exam scenario properly to determine which service is best used. Factors in choosing the correct answer are cost, time allotted for the migration, and how much data is needed to be transported.

19. Using Resource Tags with IAM - Study how you can use resource tags to manage access via IAM policies.

We also recommend checking out **Tutorials Dojo's AWS Cheat Sheets** which provides a summarized but highly informative set of notes and tips for your review on these services. These **cheat sheets** are presented mostly in bullet points which will help you retain the knowledge much better vs reading the lengthy FAQs.

We expect that you already have vast knowledge on the AWS services that a Solutions Architect commonly use, such as those listed in our SA Associate review guide. It is also not enough to just know the service and its features. You should also have a good understanding on how to integrate these services with one another to build large-scale infrastructures and applications. It's why it is generally recommended to have hands-on experience managing and operating systems on AWS.

Common Exam Scenarios

Scenario	Solution
Design Solutions for Organizational Complexity	
You are managing multiple accounts and you need to ensure that each account can only use resources that it is supposed to. What is a simple and reusable method of doing so?	AWS Organizations is a given here. It simplifies a lot of the account management and controls that you would use for this scenario. For resource control, you may use AWS CloudFormation Stacksets to define a specific stack and limit your developers to the created resources. You may also use AWS Service Catalog if you like to define specific product configurations or CloudFormation stacks, and give your developers freedom to deploy them. For permission controls, a combination of IAM policies and SCPs should suffice.
You are creating a CloudFormation stack and uploading it to AWS Service Catalog so you may share this stack with other AWS accounts in your organization. How can your end-users access the product/portfolio while still granting the least privilege?	Your end-users require appropriate IAM permissions to access AWS Service Catalog and launch a CloudFormation stack. The **AWSServiceCatalogEndUserFullAccess** and **AWSServiceCatalogEndUserReadOnlyAccess** policies grant access to the AWS Service Catalog end-user console view. When a user who has either of these policies chooses AWS Service Catalog in the AWS Management Console, the end-user console view displays the products that they have permission to launch. You should also provide the user the permission to pass IAM role to CloudFormation so that the

	CloudFormation stack can launch the necessary resources.
How can you provide access to users in a different account to resources in your account?	Use cross-account IAM roles and attach the permissions necessary to access your resources. Have the users in the other account reference this IAM role.
How do you share or link two networks together? (VPCs, VPNs, routes, etc) What if you have restrictions on your traffic e.g. it cannot traverse through the public Internet?	Sharing networks or linking two networks is a common theme in a very large organization. This ensures that your networks adhere to the best practices all the time. For VPCs, you can use VPC sharing, VPC Peering, or Transit Gateways. VPNs can utilize Site-to-Site VPN for cross-region or cross-account connections. For strict network compliance, you can access some of your AWS resources privately through shared VPC endpoints. This way, your traffic does not need to traverse through the public Internet. More information on that can be found in this article:
You have multiple accounts under AWS Organizations. Previously, each account can purchase their own RIs, but now, they have to request it from one central account for procurement. What steps should be done to satisfy this requirement in the most secure way?	Ensure that all AWS accounts are part of an AWS Organizations structure operating in all features mode. Then create an SCP that contains a deny rule to the ec2:PurchaseReservedInstancesOffering and ec2:ModifyReservedInstances actions. Attach the SCP to each organizational unit (OU) of the AWS Organizations' structure.
Can you connect multiple VPCs that belong to different AWS accounts and have overlapping CIDRs? If so, how can you manage your route tables so that the correct traffic is routed to the correct VPC?	You can connect multiple VPCs together even if they have overlapping CIDRs. What is important is that you are aware of how routing works in AWS. AWS uses longest prefix matching to determine where traffic is delivered to. So to make sure that your traffic is routed properly, be as specific as possible with your routes.
Members of a department will need access to your AWS Management Console. Without having to create IAM Users for each member, how can you provide long-term access?	You can use your on-premises SAML 2.0-compliant identity provider to grant your members federated access to the AWS Management Console via the AWS IAM Identity Center endpoint. This will provide them long term access to the console as long as they can authenticate with the IdP.

Is it possible for one account to monitor all API actions executed by each member account in an AWS Organization? If so, how does it work?	You can configure AWS CloudTrail to create a trail that will log all events for all AWS accounts in that organization. When you create an organization trail, a trail with the name that you give it will be created in every AWS account that belongs to your organization. Users with CloudTrail permissions in member accounts will be able to see this trail. However, users in member accounts will not have sufficient permissions to delete the organization trail, turn logging on or off, change what types of events are logged, or otherwise alter the organization trail in any way. When you create an organization trail in the console, or when you enable CloudTrail as a trusted service in the Organizations, this creates a service-linked role to perform logging tasks in your organization's member accounts. This role is named AWSServiceRoleForCloudTrail, and is required for CloudTrail to successfully log events for an organization. Log files for an account removed from the organization that were created prior to the account's removal will still remain in the Amazon S3 bucket where log files are stored for the trail.
You have 50 accounts joined to your AWS Organizations and you will require a central, internal DNS solution to help reduce the network complexity. Each account has its own VPC that will rely on the private DNS solution for resolving different AWS resources (servers, databases, AD domains, etc). What is the least complex network architecture that you can create?	Create a shared services VPC in your central account, and connect the other VPCs to yours using VPC peering or AWS Transit Gateway. Set up a private hosted zone in Amazon Route 53 on your shared services VPC and add in the necessary domains/subdomains. Associate the rest of the VPCs to this private hosted zone.
How can you easily deploy a basic infrastructure to different AWS regions while at the same time allowing your developers to optimize (but not delete) the launched infrastructures?	Use CloudFormation Stacksets to deploy your infrastructure to different regions. Deploy the stack in an administrator account. Create an IAM role that developers can assume so they can optimize the infrastructure. Make sure that the IAM role has a policy that denies deletion for cloudformation-launched resources.

You have multiple VPCs in your organization that are using the same Direct Connect line to connect back to your corporate datacenter. This setup does not account for line failure which will affect the business greatly if something were to happen to the network. How do you make the network more highly available? What if the VPCs span multiple regions?	Utilize Site-To-Site VPN between the VPCs and your datacenter and terminate the VPN tunnel at a virtual private gateway. Setup BGP routing. An alternative solution is to provision another Direct Connect line in another location if you require constant network performance, at the expense of additional cost. If the VPCs span multiple regions, you can use a Direct Connect Gateway.
Design for New Solutions	
You have production instances running in the same account as your dev environment. Your developers occasionally mistakenly stop/terminate production instances. How can you prevent this from happening?	You can leverage resource tagging and create an explicit deny IAM policy that would prevent developers from stopping or terminating instances tagged under production.
If you have documents that need to be collaborated upon, and you also need strict access controls over who gets to view and edit these documents, what service should you use?	AWS has a suite of services similar to Microsoft Office or Gsuite, and one of those services is called Amazon Workdocs. Amazon Workdocs is a fully managed, secure content creation, storage, and collaboration service.
You have objects in an S3 bucket that have different retrieval frequencies. To optimize cost and retrieval times, what change should you make?	S3 has a new storage class called "S3 Intelligent-Tiering". S3 IT moves 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. What makes this relatively cost-effective is that there are no retrieval fees in S3 Intelligent-Tiering, unlike the S3 IA storage class.
How can you quickly scale your applications in AWS while keeping costs low?	While EC2 instances are perfectly fine compute option, they tend to be pricey if they are not right-sized or if the capacity consumption is fluctuating. If you can, re-architect your applications to use Containers or Serverless compute options such as ECS, Fargate, Lambda and API Gateway.

You would like to automate your application deployments and use blue-green deployment to properly test your updates. Code updates are submitted to an S3 bucket you own. You wish to have a consistent environment where you can test your changes. Which services will help you fulfill this scenario?	Create a deployment pipeline using CodePipeline. Use AWS Lambda to invoke the stages in your pipeline. Use AWS CodeBuild to compile your code, before sending it to AWS Elastic Beanstalk in a blue environment. Have AWS Codebuild test the update in the blue environment. Once testing has succeeded, trigger AWS Lambda to swap the URLs between your blue and green Elastic Beanstalk environments. More information here.
Your company only allows the use of pre-approved AMIs for all your teams. However, users should not be prevented from launching unauthorized AMIs as it might affect some of their automation. How can you monitor all EC2 instances launched to make sure they are compliant with your approved AMI list, and that you are informed when someone uses an incompliant AMI?	Utilize AWS Config to monitor AMI compliance across all AWS accounts. Configure Amazon SNS to notify you when an EC2 instance was launched using an un-approved AMI. You can also use Amazon EventBridge to monitor each RunInstance event. Use it to trigger a Lambda function that will cross check the launch template to your AMI list and send you a notification via SNS if the AMI used was un-approved. This will give you more information such as who launched the instance.
How can you build a fully automated call center in AWS?	Utilize Amazon Connect, Amazon Lex, Amazon Polly, and AWS Lambda.
You have a large number of video files that are being processed locally by your custom AI application for facial detection and recognition. These video files are kept in a tape library for long term storage. Video metadata and timestamps of detected faces are stored in MongoDB. You decided to use AWS to further enhance your operations, but the migration procedure should have minimal disruption to the existing process. What should be your setup?	Use Amazon Storage Gateway Tape Gateway to store your video files in an Amazon S3 bucket. Start importing the video files to your tape gateway after you've configured the appliance. Create a Lambda function that will extract the videos from Tape Gateway and forward them to Amazon Rekognition. Use Amazon Rekognition for facial detection and timestamping. Once finished, have Rekognition trigger a Lambda function that will store the resulting information in Amazon DynamoDB.
You have a requirement to enforce HTTPS for all your connections but you would like to offload the SSL/TLS to a separate server to reduce the impact on application performance. Unfortunately, the region you are using does not support AWS ACM. What can be your alternative?	You cannot use ACM in another region for this purpose since ACM is a regional service. Generate your own certificate and upload it to AWS IAM. Associate the imported certificate with an elastic load balancer. More information here.
Accelerate Workload Migration and Modernization	

You are using a database engine on-premises that is not currently supported by RDS. If you wish to bring your database to AWS, how do you migrate it?	AWS has two tools to help you migrate your database workloads to the cloud: database migration service and schema conversion tool. First, collect information on your source database and have SCT convert your database schema and database code. You may check the supported source engines here. Once the conversion is finished, you can launch an RDS database and apply the converted schema, and use database migration service to safely migrate your database.
You have thousands of applications running on premises that need to be migrated to AWS. However, they are too intertwined with each other and may cause issues if the dependencies are not mapped properly. How should you proceed?	Use AWS Application Discovery Service to collect server utilization data and perform dependency mapping. Then send the result to AWS Migration Hub where you can initiate the migration of the discovered servers.
You have a custom-built application that you'd like to migrate to AWS. Currently, you don't have enough manpower or money to rewrite the application to be cloud-optimized, but you would still like to optimize whatever you can on the application. What should be your migration strategy?	Rehosting is out of the question since there are no optimizations done in a lift-and-shift scenario. Re-architecting is also out of the question since you do not have the budget and manpower for it. You cannot retire nor repurchase since this is a custom production application. So your only option would be to re-platform it to utilize scaling and load balancing for example.
How can you leverage AWS as a cost-effective solution for offsite backups of mission-critical objects that have short RTO and RPO requirements?	For hybrid cloud architectures, you may use AWS Storage Gateway to continuously store file backups onto Amazon S3. Since you have short RTO and RPO, the best storage type to use is File Gateway. File Gateway allows you to mount Amazon S3 onto your server, and by doing so you can quickly retrieve the files you need. Volume Gateway does not work here since you will have to restore entire volumes before you can retrieve your files. Enable versioning on your S3 bucket to maintain old copies of an object. You can then create lifecycle policies in Amazon S3 to achieve even lower costs.

You have hundreds of EC2 Linux servers concurrently accessing files in your local NAS. The communication is kept private by AWS Direct Connect and IPsec VPN. You notice that the NAS is not able to sufficiently serve your EC2 instances, thus leading to huge slowdowns. You consider migrating to an AWS storage service as an alternative. What should be your service and how do you perform the migration?	Since you have hundreds of EC2 servers, the best storage for concurrent access would be Amazon EFS. To migrate your data to EFS, you may use AWS DataSync. Create a VPC endpoint for your EFS so that the data migration is performed quickly and securely over your Direct Connect.
If you have a piece of software (e.g. CRM) that you want to bring to the cloud, and you have an allocated budget but not enough manpower to re-architect it, what is your next best option to make sure the software is still able to take advantage of the cloud?	Check in the AWS Marketplace and verify if there is a similar tool that you can use -- Repurchasing strategy.
A company wants AWS applications to authenticate users using its existing on-premises Microsoft Active Directory without storing directory data in AWS.	Deploy AWS Directory Service AD Connector to proxy authentication requests to the on-premises Active Directory environment.
Continuous Improvement for Existing Solutions	
You have a running EMR cluster that has erratic utilization and task processing takes longer as time goes on. What can you do to keep costs to a minimum?	Add additional task nodes, but use instance fleets with the master node in on-Demand mode and a mix of On-Demand and Spot Instances for the core and task nodes. Purchase Reserved Instances for the master node.
A company has multiple AWS accounts in AWS Organizations that has full features enabled. How do you track AWS costs in Organizations and alert if costs from a business unit exceed a specific budget threshold?	Use Cost Explorer to monitor the spending of each account. Create a budget in AWS Budgets for each OU by grouping linked accounts, then configure SNS notification to alert you if the budget has been exceeded.
You have a Serverless stack running for your mobile application (Lambda, API Gateway, DynamoDB). Your Lambda costs are getting expensive due to the long wait time caused by high network latency when communicating with the SQL database in your	If possible, migrate your database to AWS for lower latency. If this is not an option, consider purchasing a Direct Connect line with your VPN on top of it for a secure and fast network. Consider caching frequently retrieved results on API Gateway. Continuously monitor your

on-premises environment. Only a VPN solution connects your VPC to your on-premises network. What steps can you make to reduce your costs?	Lambda execution time and reduce it gradually up to an acceptable duration.
You have a set of EC2 instances behind a load balancer and an autoscaling group, and they connect to your RDS database. Your VPC containing the instances uses NAT gateways to retrieve patches periodically. Everything is accessible only within the corporate network. What are some ways to lower your cost?	If your EC2 instances are production workloads, purchase Reserved instances. If they are not, schedule the autoscaling to scale in when they are not in use and scale out when you are about to use them. Consider a caching layer for your database reads if the same queries often appear. Consider using NAT instances instead, or better yet, remove the NAT gateways if you are only using them for patching. You can easily create a new NAT instance or NAT gateway when you need them again.
You need to generate continuous database and server backups in your primary region and have them available in your disaster recovery region as well. Backups need to be made available immediately in the primary region while the disaster region allows more leniency, as long as they can be restored in a few hours. A single backup is kept only for a month before it is deleted. A dedicated team conducts game days every week in the primary region to test the backups. You need to keep storage costs as low as possible.	Store the backups in Amazon S3 Standard and configure cross-region replication to the DR region S3 bucket. Create a lifecycle policy in the DR region to move the backups to S3 Glacier Flexible Retrieval (formerly Glacier). S3 IA is not applicable since you need to wait fo 30 days before you can transition to IA from Standard.
Determine the most cost-effective infrastructure: a) Data is constantly being delivered to a file storage at a constant rate. Storage should have enough capacity to accommodate growth. b) The data is extracted and worked upon by worker nodes. A job can take a few hours to finish. c) This is not a mission-critical workload, so interruptions are acceptable as long as they are reprocessed. d) The jobs only need to run during evenings.	You may use Amazon Kinesis Firehose to continuously stream the data into Amazon S3. Then configure AWS Batch with spot pricing for your worker nodes. Use Amazon EventBridge (Amazon CloudWatch Events) to schedule your jobs at night. More information here.
If you are cost-conscious about the charges incurred by external users who frequently access your S3 objects, what change can you introduce to shift the charges to the users?	Ensure that the external users have their own AWS accounts. Enable S3 Requester Pays on the S3 buckets. Create a bucket policy that will allow these users read/write access to the buckets.

You have a Direct Connect line from an AWS partner data center to your on-premises data center. Webservers are running in EC2, and they connect back to your on-premises databases/data warehouse. How can you increase the reliability of your connection?	There are multiple ways to increase the reliability of your network connection. You can order another Direct Connect line for redundancy, which AWS recommends for critical workloads. You may also create an IPSec VPN connection over public Internet, but that will require additional configuration since you need to monitor the health of both networks.
You have a set of instances behind a Network Load Balancer and an autoscaling group. If you are to protect your instances from DDoS, what changes should you make?	Since AWS WAF does not integrate with NLB directly, you can create a CloudFront and attach the WAF there, and use your NLB as the origin. You can also enable AWS Shield Advanced so you get the full suite of features against DDoS and other security attacks.
You have a critical production workload (servers + databases) running in one region, and your RTO is 5 minutes while your RPO is 15 minutes. What is your most cost-efficient disaster recovery option?	If you have the option to choose warm standby, make sure that the DR infrastructure is able to automatically detect failure on the primary infrastructure (through health checks), and it can automatically scale up/scale out (autoscaling + scripts) and perform an immediate failover (Route 53 failover routing) in response. If your warm standby option does not state that it can do so then you might not be able to meet your RTO/RPO, which means you must use multi-site DR solution instead even though it is costly.
You use RDS to store data collected by hundreds of IoT robots. You know that these robots can produce up to tens of KBs of data per minute. It is expected that in a few years, the number of robots will continuously increase, and so database storage should be able to scale to handle the amount of data coming in and the IOPS required to match performance. How can you re-architect your solution to better suit this upcoming growth?	Instead of using a database, consider using a data warehousing solution such as Amazon Redshift instead. That way, your data storage can scale much larger and the database performance will not take that much of a hit.
You have a stream of data coming into your AWS environment that is being delivered by multiple sensors around the world. You need real-time processing for these data and you have to make sure that they are processed in the order in which they came in. What should be your architecture?	One might consider using SQS FIFO for this scenario, but since it also requires you to have real-time processing capabilities, Amazon Kinesis is a better solution. You can configure the data to have a specific partition key so that it is processed by the same Kinesis shard, thereby giving you similar FIFO capabilities.

You want to use your AWS Direct Connect to access S3 and DynamoDB endpoints while using your Internet provider for other types of traffic. How should you configure this?	Create a public interface on your AWS Direct Connect link. Advertise specific routes for your network to AWS, so that S3 traffic and DynamoDB traffic pass through your AWS Direct Connect.
You have a web application leveraging Cloudfront for caching frequently accessed objects. However, parts of the application are reportedly slow in some countries. What cost-effective improvement can you make?	Utilize Lambda@edge to run parts of the application closer to the users.
If you are running Amazon Redshift and you have a tight RTO and RPO requirement, what improvement can you make so that your Amazon Redshift is more highly available and durable in case of a regional disaster?	Amazon Redshift allows you to copy snapshots to other regions by enabling cross-region snapshots. Snapshots to S3 are automatically created on active clusters every 8 hours or when an amount of data equal to 5 GB per node changes. Depending on the snapshot policy configured on the primary cluster, the snapshot updates can either be scheduled, or based upon data change, and then any updates are automatically replicated to the secondary/DR region.
You have multiple EC2 instances distributed across different AZs depending on their function, and each of the AZ has its own m5.large NAT instance. A set of EC2 servers in one AZ occasionally cannot reach an API that is external to AWS when there is a high volume of traffic. This is unacceptable for your organization. What is the most cost-effective solution for your problem?	It would be better if you transition your NAT Instances to NAT Gateways since they provide faster network speeds. Resizing the NAT instance to something higher is not cost-effective anymore since the network speed increase is gradual as you go up. Adding more NAT instances to a single AZ makes your environment too complex.
Most of your vendors' applications use IPv4 to communicate with your private AWS resources. However, a newly acquired vendor will only be supporting IPv6. You will be creating a new VPC dedicated for this vendor, and you need to make sure that all of your private EC2 instances can communicate using IPv6. What are the configurations that you need to do?	Provide your EC2 instances with IPv6 addresses. Create security groups that will allow IPv6 addresses for inbound and outbound. Create an egress-only Internet gateway to allow your private instances to reach the vendor.

Validate Your Knowledge

After your review, you should take some practice tests to measure your preparedness for the real exam. AWS offers a sample practice test for free which you can find here. You can also opt to buy the longer AWS sample practice test at aws.training, and use the discount coupon you received from any previously taken certification exams. Be aware though that the sample practice tests do not mimic the difficulty of the real SA Pro exam. You should not rely solely on them to gauge your preparedness. It is better to take more **practice tests** to fully understand if you are prepared to pass the certification exam.

Fortunately, **Tutorials Dojo** also offers a great set of practice questions for you to take **here**. It is kept updated by the creators to ensure that the questions match what you'll be expecting in the real exam. The practice tests will help fill in any important details that you might have missed or skipped in your review. You can pair our practice exams with this study guide eBook to further help in your exam preparations.

Sample Practice Test Questions:

Question 1

A company has multiple AWS resources in its production account that are shared among various business units. Each business unit may have one or more AWS accounts which have resources in the production account. There were a lot of incidents in which the developers from a specific business unit accidentally terminated the EC2 instances owned by another business unit. You are tasked to come up with a solution to only allow a specific business unit who own the EC2 instances, and other AWS resources, to terminate their own resources.

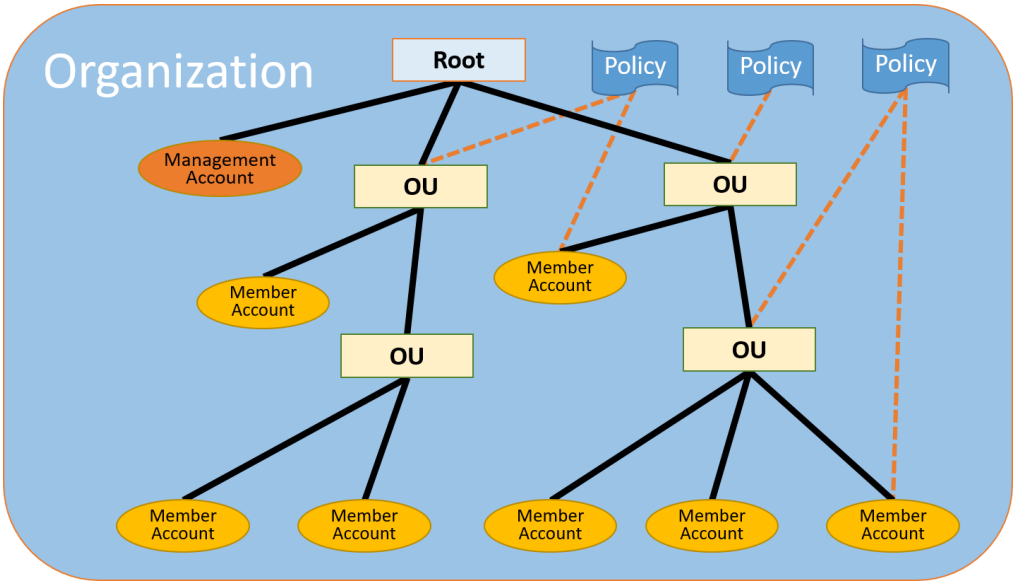
Which of the following is the most suitable multi-account strategy that you should implement?

1. Use AWS Organizations to centrally manage all of your accounts. Group your accounts, which belongs to a specific business unit, to individual Organization Unit (OU). Create an IAM Role in the production account for each business unit which has a policy that allows access to the EC2 instances including a resource-level permission to terminate the instances that it owns. Create an AWSServiceRoleForOrganizations service-linked role to the individual member accounts of the OU to enable **trusted access**.
2. Use AWS Organizations to centrally manage all of your accounts. Group your accounts, which belongs to a specific business unit, to individual Organization Unit (OU). Create a Service Control Policy in the production account for each business unit which has a policy that allows access to the EC2 instances including a resource-level permission to terminate the instances that it owns. Provide the cross-account access and the SCP to the individual member accounts to tightly control who can terminate the EC2 instances.
3. Use AWS Organizations to centrally manage all of your accounts. Group your accounts, which belong to a specific business unit, to individual Organization Units (OU). Create an IAM Role in the production account which has a policy that allows access to the EC2 instances including a resource-level permission to terminate the instances owned by a particular business unit. Provide the cross-account access and the IAM policy to every member accounts of the OU.

4. Use AWS Control Tower to centrally manage all of your accounts. Group your accounts, which belongs to a specific business unit, to individual Organization Unit (OU). Create a Service Control Policy in the production account which has a policy that allows access to the EC2 instances including a resource-level permission to terminate the instances owned by a particular business unit. Provide the cross-account access and the SCP to the OUs, which will then be automatically inherited by its member accounts.

Correct Answer: 3

AWS Organizations is an account management service that enables you to consolidate multiple AWS accounts into an *organization* that you create and centrally manage. AWS Organizations includes account management and consolidated billing capabilities that enable you to better meet the budgetary, security, and compliance needs of your business. As an administrator of an organization, you can create accounts in your organization and invite existing accounts to join the organization.



You can use organizational units (OUs) to group accounts together to administer as a single unit. This greatly simplifies the management of your accounts. For example, you can attach a policy-based control to an OU, and all accounts within the OU automatically inherit the policy. You can create multiple OUs within a single organization, and you can create OUs within other OUs. Each OU can contain multiple accounts, and you can move accounts from one OU to another. However, OU names must be unique within a parent OU or root.

Resource-level permissions refers to the ability to specify which resources users are allowed to perform actions on. Amazon EC2 has partial support for resource-level permissions. This means that for certain Amazon EC2 actions, you can control when users are allowed to use those actions based on conditions that

have to be fulfilled, or specific resources that users are allowed to use. For example, you can grant users permissions to launch instances, but only of a specific type, and only using a specific AMI.

The scenario on this question has a lot of AWS Accounts that need to be managed. AWS Organization solves this problem and provides you with control by assigning the different business units as individual Organization Units (OU). Service control policies (SCPs) are a type of organization policy that you can use to manage permissions in your organization. SCPs offer central control over the maximum available permissions for all accounts in your organization. However, SCPs alone are not sufficient for allowing access in the accounts in your organization. Attaching an SCP to an AWS Organizations entity just defines a guardrail for what actions the principals can perform. You still need to attach identity-based or resource-based policies to principals or resources in your organization's accounts to actually grant permission to them.
Since SCPs only allow or deny the use of an AWS service, you don't want to block OUs from completely using the EC2 service. Thus, you will need to provide cross-account access and the IAM policy to every member accounts of the OU.

Hence, the correct answer is: **Use AWS Organizations to centrally manage all of your accounts. Group your accounts, which belong to a specific business unit, to individual Organization Units (OU). Create an IAM Role in the production account which has a policy that allows access to the EC2 instances including a resource-level permission to terminate the instances owned by a particular business unit. Provide the cross-account access and the IAM policy to every member accounts of the OU.**

The option that says: **Use AWS Organizations to centrally manage all of your accounts. Group your accounts, which belongs to a specific business unit, to individual Organization Unit (OU). Create an IAM Role in the production account for each business unit which has a policy that allows access to the EC2 instances including a resource-level permission to terminate the instances that it owns. Create an AWSServiceRoleForOrganizations service-linked role to the individual member accounts of the OU to enable trusted access** is incorrect because **AWSServiceRoleForOrganizations** service-linked role is primarily used to only allow AWS Organizations to create service-linked roles for other AWS services. This service-linked role is present in all organizations and not just in a specific OU. SCPs are similar to IAM permission policies except that they don't grant any permissions.

The following options are incorrect because an SCP policy simply specifies the services and actions that users and roles can use in the accounts:

1. Use AWS Organizations to centrally manage all of your accounts. Group your accounts, which belongs to a specific business unit, to individual Organization Unit (OU). Create a Service Control Policy in the production account for each business unit which has a policy that allows access to the EC2 instances including a resource-level permission to terminate the instances that it owns. Provide the cross-account access and the SCP to the individual member accounts to tightly control who can terminate the EC2 instances.

2. Use AWS Control Tower to centrally manage all of your accounts. Group your accounts, which belongs to a specific business unit, to individual Organization Unit (OU). Create a Service Control Policy in the production

account which has a policy that allows access to the EC2 instances including a resource-level permission to terminate the instances owned by a particular business unit. Provide the cross-account access and the SCP to the OUs, which will then be automatically inherited by its member accounts.

References:
https://docs.aws.amazon.com/organizations/latest/userguide/orgs_manage_ous.html
https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-supported-iam-actions-resources.html
https://docs.aws.amazon.com/IAM/latest/UserGuide/tutorial_cross-account-with-roles.html

Check out this AWS Organizations Cheat Sheet:
https://tutorialsdojo.com/aws-organizations/

Service Control Policies (SCP) vs IAM Policies:
https://tutorialsdojo.com/service-control-policies-scp-vs-iam-policies/

Comparison of AWS Services Cheat Sheets:
https://tutorialsdojo.com/comparison-of-aws-services/

Question 2

A multinational manufacturing company has multiple AWS accounts in multiple AWS regions across North America, Europe, and Asia. The solutions architect has been tasked to set up AWS Organizations to centrally manage policies and have full administrative control across the multiple AWS accounts owned by the company.

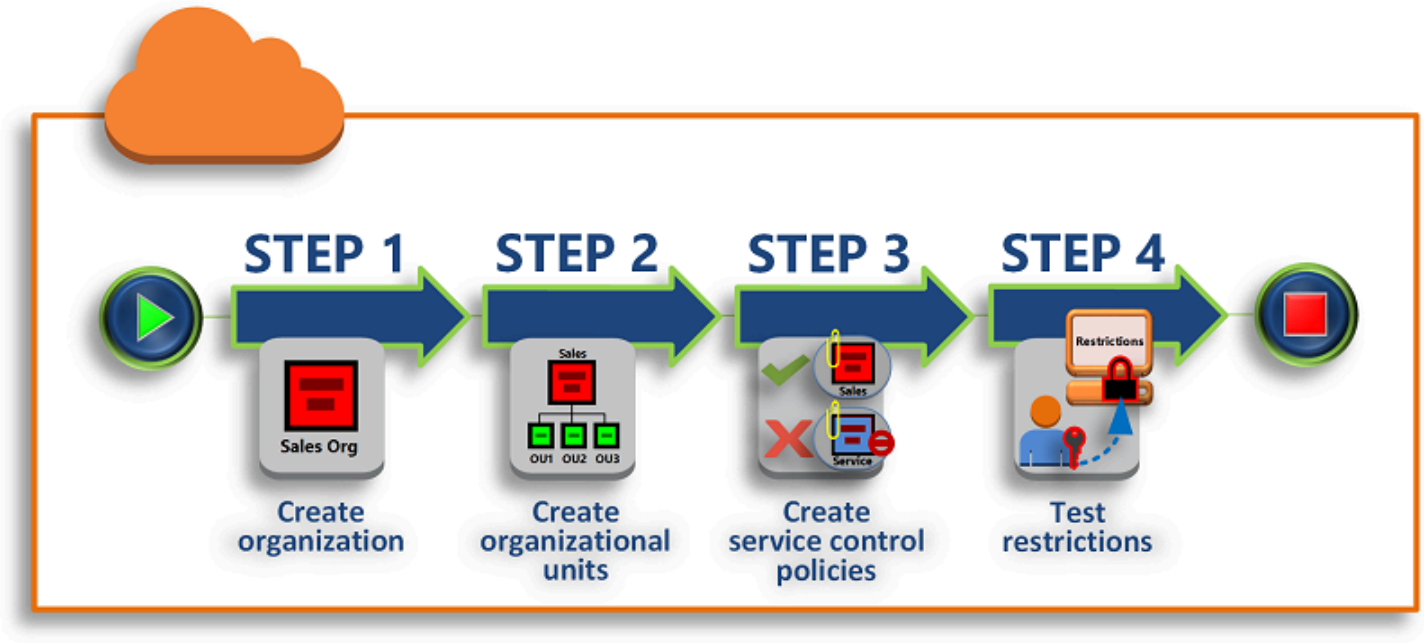
Which of the following options is the recommended implementation to achieve this requirement with the LEAST effort?

1. Set up AWS Organizations by establishing cross-account access from the master account to all member AWS accounts of the company. The master account will automatically have full administrative control across all member accounts.
2. Set up AWS Organizations by sending an invitation to the master account of your organization from each of the member accounts of the company. Create an `OrganizationAccountAccessRole` IAM role in the member account and grant permission to the master account to assume the role.
3. Use AWS Control Tower from the master account and enroll all the member AWS accounts of the company. AWS Control Tower will automatically provision the needed IAM permissions to have full administrative control across all member accounts.
4. Set up AWS Organizations by sending an invitation to all member accounts of the company from the master account of your organization. Create an `OrganizationAccountAccessRole` IAM role in the member account and grant permission to the master account to assume the role.

Correct Answer: 4

After you create an **Organization** and verify that you own the email address associated with the master account, you can invite existing AWS accounts to join your organization. When you invite an account, AWS Organizations sends an invitation to the account owner, who decides whether to accept or decline the invitation. You can use the AWS Organizations console to initiate and manage invitations that you send to other accounts. You can send an invitation to another account only from the master account of your organization.

If you are the administrator of an AWS account, you also can accept or decline an invitation from an organization. If you accept, your account becomes a member of that organization. Your account can join only one organization, so if you receive multiple invitations to join, you can accept only one.



When an invited account joins your organization, you *do not* automatically have full administrator control over the account, unlike created accounts. If you want the master account to have full administrative control over an invited member account, you must create the `OrganizationAccountAccessRole` IAM role in the member account and grant permission to the master account to assume the role.

Therefore, the correct answer is: **Set up AWS Organizations by sending an invitation to all member accounts of the company from the master account of your organization. Create an `OrganizationAccountAccessRole` IAM role in the member account and grant permission to the master account to assume the role.**

The option that says: **Set up AWS Organizations by establishing cross-account access from the master account to all member AWS accounts of the company. The master account will automatically have full administrative control across all member accounts** is incorrect. Cross-account access is primarily used for

scenarios where you need to grant your IAM users permission to switch to roles within your AWS account or to roles defined in other AWS accounts that you own.

The option that says: **Set up AWS Organizations by sending an invitation to the master account of your organization from each of the member accounts of the company. Create an `OrganizationAccountAccessRole` IAM role in the member account and grant permission to the master account to assume the role** is incorrect. It entails a lot of effort to send an individual invitation to the master account from each of the member accounts of the company. It's stated in the scenario that you should achieve this requirement with the LEAST effort, and you can do this by sending an invitation to all member accounts of the company from the master account of your organization.

The option that says: **Use AWS Control Tower from the master account and enroll all the member AWS accounts of the company. AWS Control Tower will automatically provision the needed IAM permissions to have full administrative control across all member accounts** is incorrect. AWS Control Tower can be typically used to set up and manage multiple AWS accounts. However, it will not automatically provision IAM permissions for all member accounts.

References:

https://docs.aws.amazon.com/organizations/latest/userguide/orgs_manage_accounts_invites.html

https://docs.aws.amazon.com/organizations/latest/userguide/orgs_manage_accounts.html

https://docs.aws.amazon.com/organizations/latest/userguide/orgs_manage_accounts_create.html

Check out this AWS Organizations Cheat Sheet:

https://tutorialsdojo.com/aws-organizations/

Click **here** for more **AWS Certified Solutions Architect Professional practice exam questions**.

More AWS reviewers can be found **here**:

Additional Training Materials: High-Quality Video Courses

There are a few top-rated AWS Certified Solutions Architect Professional 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:

- AWS Certified Solutions Architect Professional by Adrian Cantrill

Based on user feedback, any of these video courses plus our **practice test course** and this study guide eBook were enough to pass this tough exam.

In general, what you should have learned from your review are the following:

- Features and use cases of the AWS services and how they integrate with each other
- AWS networking, security, billing and account management
- The AWS CLI, APIs and SDKs
- Automation, migration planning, and troubleshooting
- The best practices in designing solutions in the AWS Cloud
- Building CI/CD solutions using different platforms
- Resource management in a multi-account organization
- Multi-level security

All these factors are essentially the domains of your certification exam. It is because of this difficult hurdle that AWS Certified Solutions Architect Professionals are highly respected in the industry. They are capable of architecting ingenious solutions that solve customer problems in AWS. They are also constantly improving themselves by learning all the new services and features that AWS produces each year to make sure that they can provide the best solutions to their customers. Let this challenge be your motivation to dream high and strive further in your career as a Solutions Architect!

Final notes regarding your exam

The SA Professional exam questions always ask for highly available, fault tolerant, cost-effective and secure solutions. Be sure to understand the choices provided to you, and verify that they have accurate explanations. Some choices are very misleading such that they seem to be the most natural answer to the question, but actually contain incorrect information, such as the incorrect use of a service. Always place accuracy above all else.

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

Since an SA Professional has responsibilities in creating large-scale architectures, be wary of the different ways AWS services can be integrated with one another. Common combinations include:

- AWS Security Hub, AWS Control Tower, AWS Organizations and AWS Resource Access Manager
- Amazon ECS Anywhere, Amazon ECS and AWS Fargate
- Amazon EKS Anywhere, Amazon EKS and AWS Fargate
- Lambda, API Gateway, Amazon SNS, and DynamoDB
- EC2, EBS/EFS/Elasticache, Auto Scaling, ELB, and SQS
- Amazon S3, Amazon Cloudfront, AWS WAF
- Amazon S3, Kinesis
- On-premises servers with Direct Connect/VPN/VPC Endpoints/Transit Gateway
- On-premises DNS servers and Amazon Route 53 with inbound/outbound DNS resolvers
- AWS Organizations, AWS IAM Identity Center, IAM roles, Config, Cloudformation and Service Catalog
- Mobile apps with Cognito, API Gateway, and DynamoDB
- CodePipeline, CodeBuild, CodeDeploy
- ECR, ECS/Fargate and S3
- EMR + Spot Fleets/Combinations of different instance types for master node and task nodes
- Amazon Connect + Alexa + Amazon Lex

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

Domain 1: Design Solutions for Organizational Complexity

Overview

The first domain of the AWS Certified Solutions Architect Professional exam evaluates your capability to implement solutions that allow different accounts and business units to operate in an AWS environment securely and reliably. As you become part of a larger organization, the number of users and stakeholders involved in your cloud architecture becomes more complex. You need to be able to segregate these user groups and business units according to their respective purpose and simplify their responsibilities within your AWS environments. Consequently, you also need to make sure that each group is given access to what they should and what they only need. This is to avoid any unnecessary access that could result in security leaks for the organization.

26% of questions in the actual SAP-C02 exam revolve around designing an organizational setup that involves the use of multiple AWS accounts, AWS Regions, VPCs, and billing configuration. This is the second biggest domain in the AWS Certified Solutions Architect Professional exam so expect to see a lot of questions that involve a variety of management and governance services in AWS. You have to focus on the following AWS services: AWS Organizations, AWS Control Tower, AWS Direct Connect, Amazon Route 53, AWS Security Hub, and others. Make sure that you know how Consolidated Billing works for multiple AWS accounts that are under an AWS Organization, especially the right configuration of enabling or disabling the Reserved Instance (RI) Sharing option for one or more AWS accounts.

This domain checks your know-how in doing these tasks:

- Architecting network connectivity strategies
- Prescribing security controls
- Designing reliable and resilient architectures.
- Designing a multi-account AWS environment
- Determining cost optimization and visibility strategies.

In this chapter, we will cover the related topics for organizational designs and strategies in AWS that will likely show up in your Solutions Architect Professional exam.

Managing of Multiple AWS Accounts in an Organization

As a company grows larger and the number of AWS users and resources increase, it becomes extraordinarily difficult to manage such a huge, complex ecosystem in just a single AWS account. Various teams will have different workloads, different stakeholders will have different objectives, and different environments will have different priorities. And much like in a software development lifecycle wherein you have a dedicated environment for development, for QA or staging, for UAT, and for production, AWS allows you to set up a similar structure at no cost through account organizations.

In an ideal scenario, you should be using one account per development lifecycle environment. You should also have a separate account for centrally storing logs, another separate account for facilitating security between the different accounts under your organization, and a separate account for billing and administration tasks. This is known as a **Landing Zone** setup.

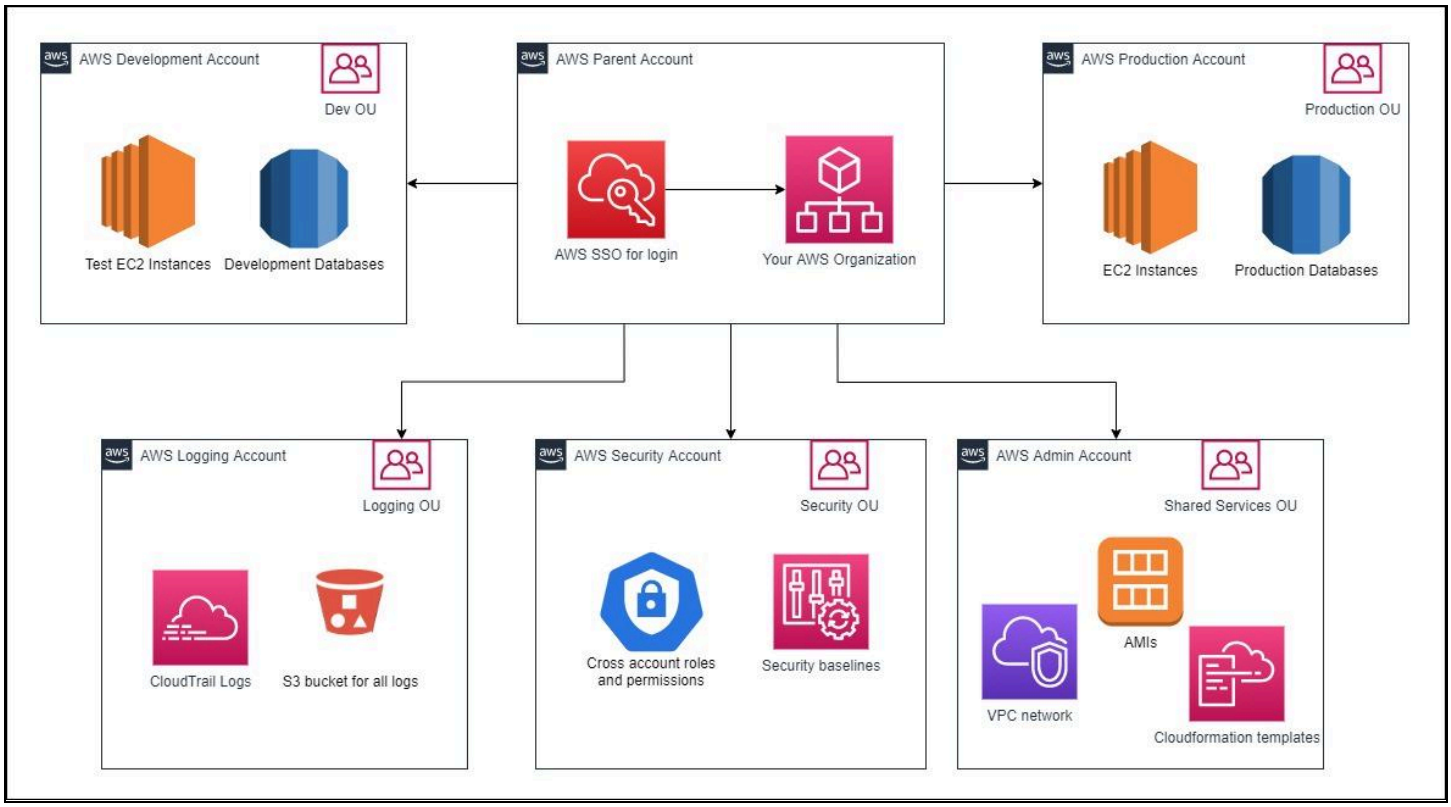


Figure: Example Setup of Landing Zone

Through this structure, you can experiment and develop faster since you have achieved a degree of isolation and flexibility. You can create an exact copy of an environment setup of another account and not have to worry about affecting the other account's processes while you define your own.

There are many ways to manage multiple accounts in AWS, but the most common and simplest method is by using AWS Organizations. This service allows you to govern and centrally manage your different AWS accounts under one account. It also provides many features for implementing security, cost management, and infrastructure compliance which we will also discuss along the way. The main components in AWS Organizations are the **master account** and the **member accounts**. As the name implies, the master account is where you'll be creating your organization. You can then invite other accounts to join your organization as members and manage them from your end. Take note that a member account can be a part of only one AWS Organization at a time. Once you have all your necessary accounts joined to the organization, you can start grouping them together into **Organization Units** (OUs), which will allow you to create a hierarchy. Making use of OUs will not only simplify account management, but also enable you to easily deploy security policies and shared resources across multiple accounts in an OU at the same time.

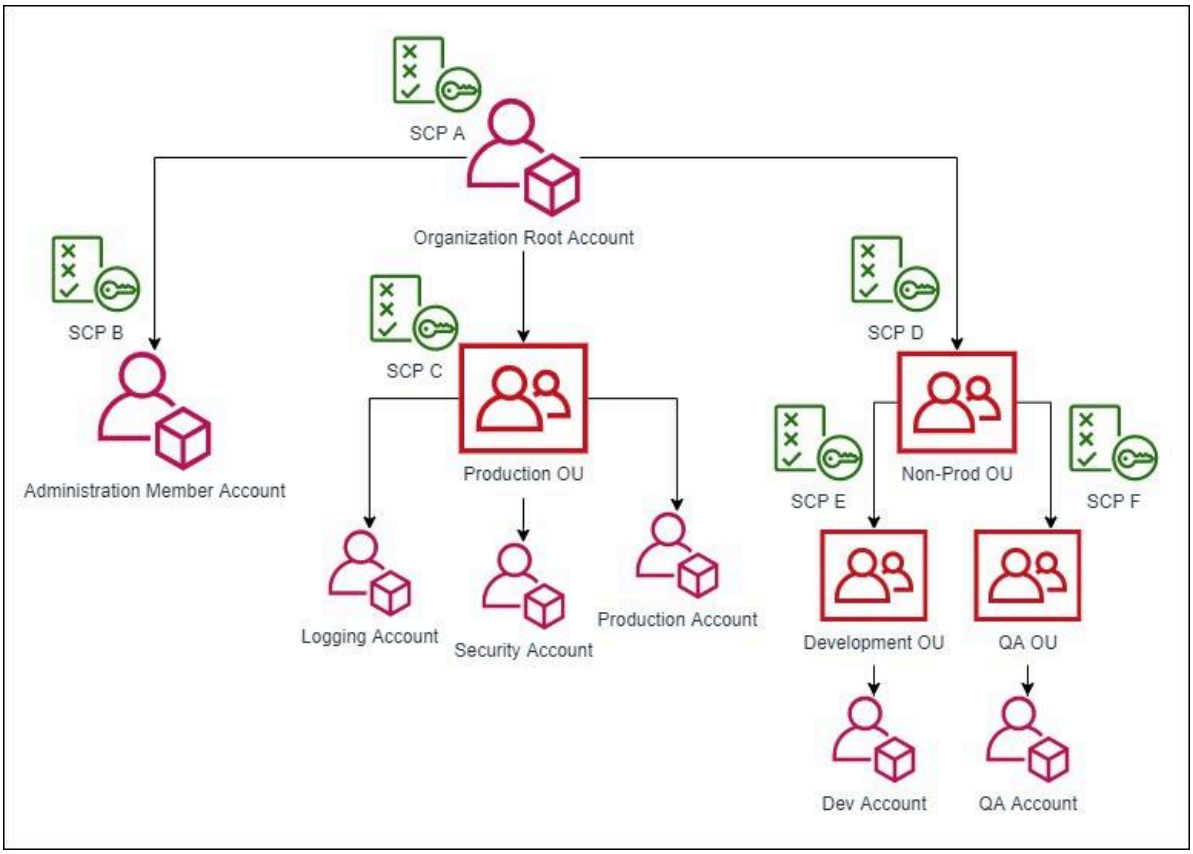


Figure: Example Structure of an AWS Organization with Multiple Accounts, OUs and SCPs

References:
https://aws.amazon.com/solutions/implementations/aws-landing-zone/
https://aws.amazon.com/blogs/mt/tag/aws-multi-account-management/
https://aws.amazon.com/organizations/

Security and Access Controls for a Multi-Account Structure

Since there can be hundreds of users and services interacting with one another in a multi-account structure, configuring security properly is vital in ensuring that you adhere to the principle of least privilege. There are different strategies that you can implement for multi-account security, depending on your business needs. There are also a few best practices that we will be discussing while you leverage these strategies. Sometimes, there can be questions in your exam that utilize more than one strategy for implementing security. The best way to know which to choose is to determine the assets involved in the accounts.

Cross-account roles

In a standalone account, IAM roles are a great way to provide access to your resources without having to create dedicated user credentials. They can also be attached to AWS services to allow interaction with one another in a secure manner. But what you might not have known is that you can also use IAM roles to provide access to users in another account. These are known as **cross-account roles**. Cross-account roles save you from the tedious task of creating and managing dedicated IAM Users in each account.

To get started with cross-account roles, you need to go to the IAM service and create a role meant for cross-account access. For convenience, imagine that you administer account A and the one requiring access to your environment is account B. During role creation, you input the Account ID of account B. At this point, you can also require users of account B to be MFA authenticated before they can assume the role.

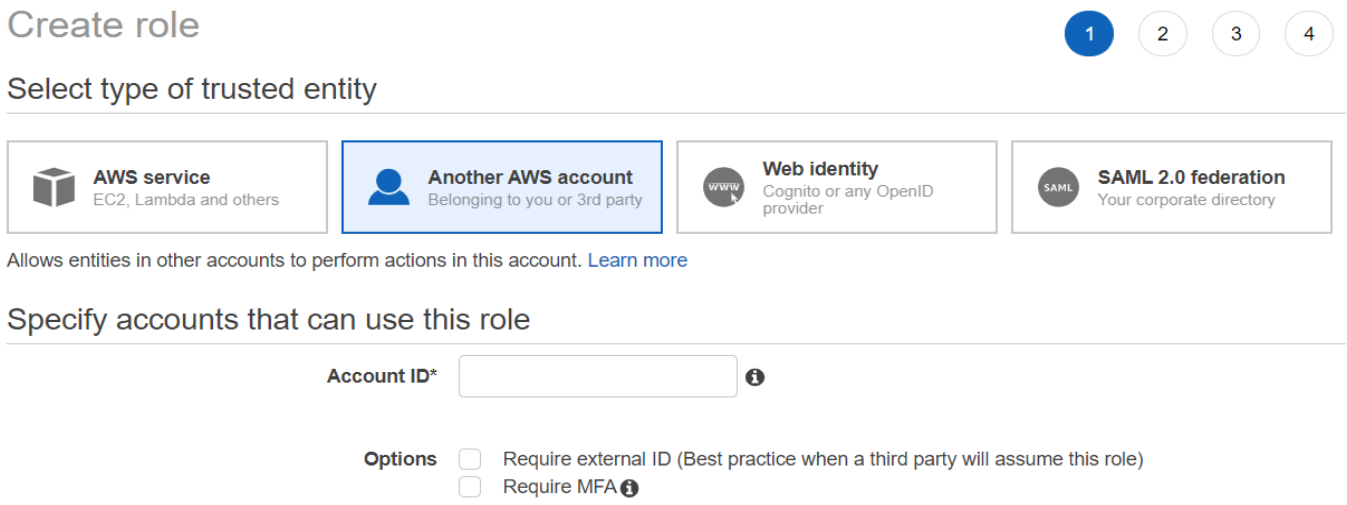


Figure: Create a cross-account role in IAM

On the final step, you provide the role with the necessary permissions to your account A via IAM Policies. Once this cross-account role has been created, IAM Users in account B can switch to or assume this role and gain

the permissions to do what they need to do in your account. To limit who can assume this role in account B, the admin of account B can create a policy that allows only specific users to assume the cross-account role.

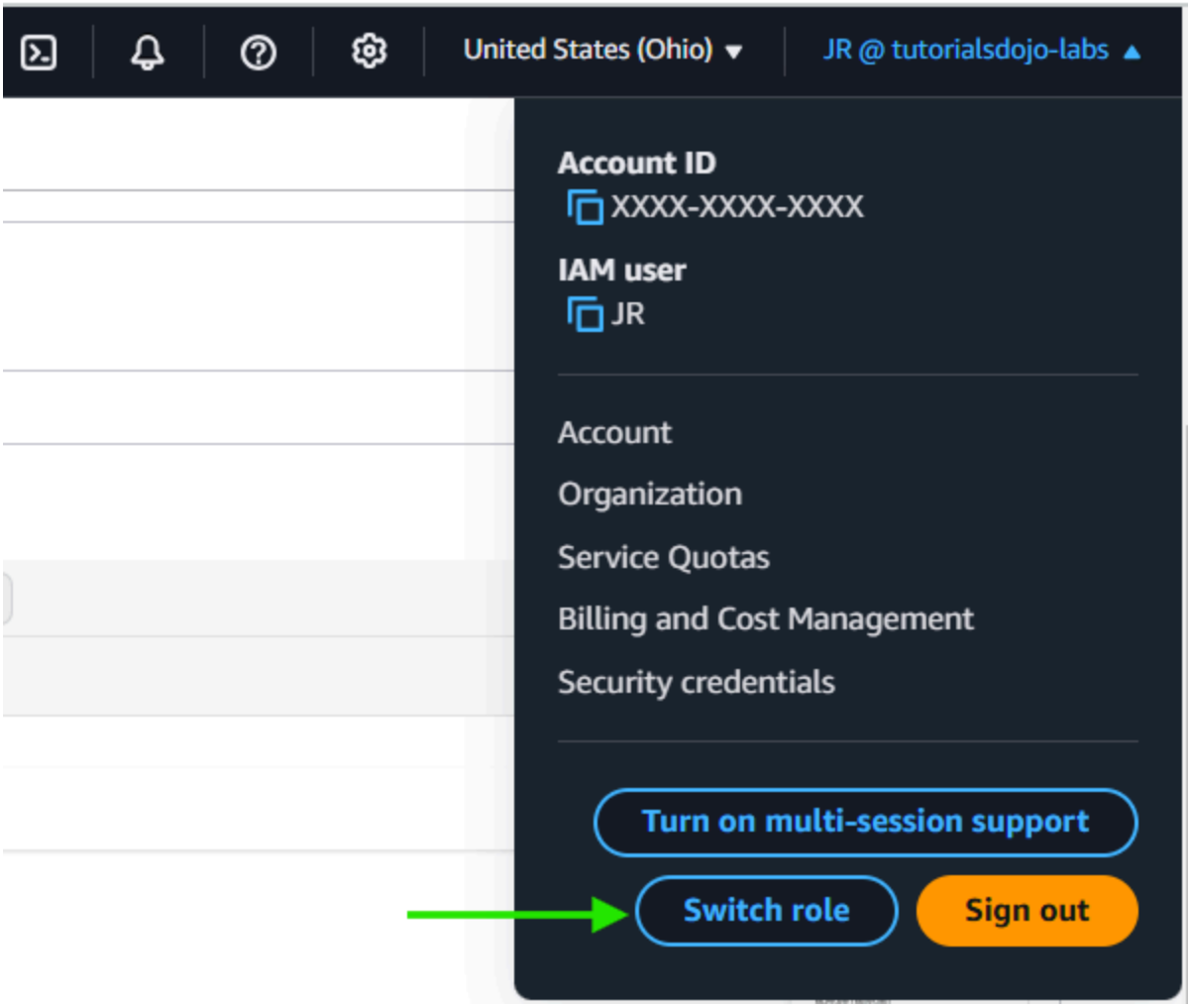


Figure: How to switch role in the AWS Console

AWS Organizations Service Control Policies

When you are the administrator of multiple accounts in an AWS Organization, you need to make sure that each account will function the way they are intended to. You can use Service Control Policies (SCPs) to restrict the actions that entities can perform in an account. SCPs are written in similar syntax as IAM Policies. SCPs apply account-wide, so it affects both IAM Users and IAM Roles managed by that account, but it does not affect resource-based policies/service-linked roles. Do keep in mind that you can only use SCPs if you have enabled **all features** in your AWS Organization.

SCPs can be attached to individual accounts and OUs. Attaching an SCP to an OU cascades the policy to member accounts of that OU. This means that any SCP you attach at the root of a hierarchy also applies to everything below it. An explicit deny overrules an explicit allow. An explicit allow overrules an implicit deny. By default, an SCP named *FullAWSAccess* is attached to every organization root, OU, and account. This default SCP allows all actions and all services.

ℹ️ Solutions Architect Professional Exam Notes:

Remember that an SCP can only define what actions are available in an account. It does not delegate the actual permissions unlike IAM Policies. If you need to do something in your environment, you need to have the necessary policies attached first. In terms of permission hierarchy, the *deny* rule always takes precedence. This means that even if you have an IAM Policy that allows you to perform an action, if the SCP attached to that account implicitly or explicitly denies this action then you cannot perform it. Same goes with having an allow in the SCP but being implicitly or explicitly denied in the IAM Policy.

There are two common approaches to SCPs: *whitelisting* and *blacklisting*.

- **Blacklisting** applies the FullAWSAccess SCP, which doesn't filter out any AWS service APIs, then filters out specific APIs by blacklisting them in subsequent SCPs attached to OUs at various points in your organization's structure.
- **Whitelisting** is about modifying your SCPs to be more restrictive in allow permissions. All other actions are therefore implicitly denied. Users and roles in the affected accounts can then exercise only that level of access, even if their IAM policies allow all actions.

Shared Directory Services

If you are using AWS Managed Microsoft AD Directory, you can share this directory to other VPCs and AWS accounts within the same Region. This makes it convenient to manage different directory-aware services such as EC2 instances or local Windows servers across different VPCs and accounts. To share your directory, you first need to configure the network between the VPCs that will be communicating. You have multiple options on how to do this, such as VPC Peering, Direct Connect, Transit Gateway, VPN and so on. Once you have configured your route tables and security groups, you have two ways to share your directory:

1) If you are in an AWS Organization, you only need to select the accounts that you want to share the directory to. Your AWS Organization must have all features enabled and the directory must be in the organization's master account for this to work.
2) If you are sharing the directory to an external account, you need to initiate a handshake request and the recipient needs to accept your request.

If you have an external AD that you want to use as an authentication method for your AWS account (which is common in a hybrid environment), you can do so by using **SAML Federation**. Federation is the practice of establishing trust between a system acting as an identity provider and other systems, often called service providers, that accept authentication tokens from that identity provider. You have options on how to implement federation in AWS:

1) You may use AWS IAM Identity Center which works with your identity provider to handle access for your federated users and roles.
2) You may use IAM identity providers instead of creating IAM users in your AWS account. IAM supports providers that are compatible with OpenID Connect (OIDC) or SAML 2.0. OIDC calls the *AssumeRoleWithWebIdentity* API to trade the authentication token you get from those IdPs for AWS temporary security credentials. SAML calls AWS STS *AssumeRoleWithSAML* API, passing the ARN of the SAML provider, the ARN of the role to assume, and the SAML assertion from IdP.
3) You may use third-party SAML solution providers and manually configure a solution to work with AWS federation.

References:
https://docs.aws.amazon.com/pdfs/whitepapers/latest/organizing-your-aws-environment/organizing-your-aws-environment.pdf#organizing-your-aws-environment
https://docs.aws.amazon.com/IAM/latest/UserGuide/tutorial_cross-account-with-roles.html
https://docs.aws.amazon.com/organizations/latest/userguide/orgs_manage_policies_type-auth.html
https://docs.aws.amazon.com/directoryservice/latest/admin-guide/ms_ad_directory_sharing.html

Using S3 Requester Pays and Bucket Policies

If you have partner accounts that need to access your S3 objects, you can have them shoulder the costs of these requests via S3 Requester Pays. You can enable this feature by simply going to your bucket properties and turning on Requester Pays. Once enabled, you must authenticate all requests involving Requester Pays buckets. Also, after you configure a bucket to be a Requester Pays bucket, requesters must include x-amz-request-payer in their requests either in the header, for POST, GET and HEAD requests, or as a parameter in a REST request.

To provide cross-account access to objects that are in your S3 buckets, configure the bucket policy to allow API access for the other account. See example policy below:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::OtherAccount:user/OtherAccountUserName"
            },
            "Action": [
                "s3:GetObject",
                "s3:PutObject"
            ],
            "Resource": [
                "arn:aws:s3:::YourBucketName/*"
            ]
        }
    ]
}
```

References:
https://docs.aws.amazon.com/AmazonS3/latest/dev/RequesterPaysBuckets.html
https://aws.amazon.com/premiumsupport/knowledge-center/cross-account-access-s3/

Multi-Account Infrastructure Management

Managing multiple accounts requires you to monitor what resources need to be launched in each account and how they should be launched. You will have to employ different administration and devops techniques to make sure that users are launching resources that they only require, and that these resources comply with the organization's compliance requirements. You can perform proactive monitoring using AWS Config deployed in each account via CloudFormation script, but a better solution would be to define the accounts' infrastructures right from the get go. The key services to use for this purpose are AWS Organizations, AWS CloudFormation Stack Sets, and AWS Service Catalog.

CloudFormation Stack Sets

AWS CloudFormation StackSets is similar to your usual CloudFormation stack operation but it extends this by enabling you to create, update, or delete stacks across multiple accounts and regions with a single operation. Using an administrator account, you define and manage an AWS CloudFormation template, and use the template as the basis for provisioning stacks into selected target accounts across specified regions. As a result, you can deploy standardized infrastructures onto multiple accounts without having to log in to each of these accounts. Through the standard template, you can specify how resources will be configured at launch (such as enforcing tags) and restrict users to only use resources launched by the CloudFormation stack (via IAM policy).

Before you start using CloudFormation Stack Sets, you must first configure a trust relationship between your administrator account and the accounts you will be deploying the CloudFormation stack to. There are two methods for this:

1) If the target accounts are not part of your AWS Organization, you need to establish a trust relationship between the administrator and target accounts by creating IAM roles in each account.
    a) In each target account, create a service role named **AWSCloudFormationStackSetExecutionRole** that trusts the administrator account.
    b) Grant the service role the required permissions to perform the operations that are specified in your AWS CloudFormation template.
2) If the target accounts are part of your AWS Organization, you just need to enable trusted access. Note that this requires your AWS Organization to have all features enabled.
    a) After trusted access is enabled, StackSets creates the necessary IAM roles in the administrator (AWS Organizations master) account and target accounts when you create stack sets with service-managed permissions.

Your stack set can be deployed to your entire organization or specific OUs. If your stack set targets your organization, it also targets all accounts in all OUs in the organization. If your stack set targets specified OUs, it also targets all accounts in those OUs and all child OUs. StackSets does not deploy stack instances to the

organization master account, even if the master account is in your organization or in an OU in your organization.

Service Catalog

AWS Service Catalog lets you create a portfolio of services that are approved for use in your AWS account. It helps you centrally manage commonly deployed services, achieve consistent governance, and meet compliance requirements. Users can view the approved services and select the configuration that they wish to use for deployment. In short, *what you see is what you get*. Before your end users can use your products, you must grant them permissions that allow them to access the AWS Service Catalog console, launch products, and manage launched products as provisioned products.

Catalog administrators can define these products through a CloudFormation template, which is convenient if you have a full stack of services. They can add constraints and resource tags to be used at provisioning, and then grant access to the portfolio through IAM users and groups. Updating a product in a portfolio is also very easy and can be done by updating the submitted template. You can also employ versioning to a product. When you create a new version of a product, the update is automatically distributed to all users who have access to the product, allowing the user to select which version of the product to use.

To make your AWS Service Catalog products available to users who are not in your AWS account, such as users who belong to other organizations or to other AWS accounts in your organization, you share your portfolios with them. This can be done in several ways, including **account-to-account sharing**, **organizational sharing via AWS Organizations**, and **deploying catalogs using CloudFormation stack sets**. When you share a portfolio using account-to-account sharing or organizational sharing, you allow the Service Catalog administrator of the target AWS account to import your portfolio into his or her account and distribute the products to end users in that account. The products and constraints in the imported portfolio stay in sync with changes that you make to your shared portfolio. The products in the portfolio cannot be modified by recipients, but only designate who can access the products.

- For account-to-account sharing, if a Service Catalog administrator from another AWS account shares a portfolio with you, you can import that portfolio into your account by getting its URL from the administrator.
- In AWS Organizations, you can share portfolios to an organization account, a single OU or to the whole organization which is every account in that organization.
- You can use AWS CloudFormation StackSets to launch AWS Service Catalog products across multiple regions and accounts. You can specify the order in which products deploy sequentially within regions. Across accounts, products are deployed in parallel.

ℹ️ **Solutions Architect Professional Exam Notes:**

Should I Use CloudFormation Stack Sets? Service Catalog? Or Both?

Generally, one can see the similarities between these options. The end goal is to make sure that your users only launch what they need to and use the configuration that complies with your organization. However, there are a few pros and cons that come in between.

Stack Sets is a great option if you already have a CloudFormation template ready to deploy to multiple accounts and multiple regions. They make sure that the resulting infrastructure is similar in each location. The issue, however, is you need to have a great understanding of what each team needs for their infrastructure. It is difficult to standardize something if users have different objectives. Another issue is compliance monitoring. The resources in a CloudFormation stack can be modified after provisioning, as long as the user has the permissions to do so. This might be unacceptable for some companies.

Service Catalog is a great tool if you have specific restrictions for some of the basic services, such as EC2 or RDS. It also makes sure that end users can only deploy these services the way you designed them, so they won't need to worry about code. One issue is that products created in Service Catalog are regional. They are only visible and usable in the region you deployed them in. This can be difficult to manage when you have multiple accounts using multiple regions. Some administrators might consider creating a CI/CD pipeline, but this is additional overhead. Service Catalog also reduces the flexibility for your users with their work.

A nifty solution to cover each other's weaknesses is to combine Stack Sets with Service Catalog. That way, you can govern the infrastructure of multiple accounts and multiple regions. Users can also customize the resulting infrastructure using desired parameters. The only downside to this method is that your users need to be familiar with how products are configured.

References:

https://aws.amazon.com/blogs/aws/use-cloudformation-stacksets-to-provision-resources-across-multiple-aws-accounts-and-regions/
https://aws.amazon.com/blogs/mt/managing-aws-organizations-accounts-using-aws-config-and-aws-cloudformation-stacksets/
https://aws.amazon.com/blogs/mt/simplify-sharing-your-aws-service-catalog-portfolios-in-an-aws-organizations-setup/
https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/what-is-cfnstacksets.html
https://aws.amazon.com/servicecatalog/

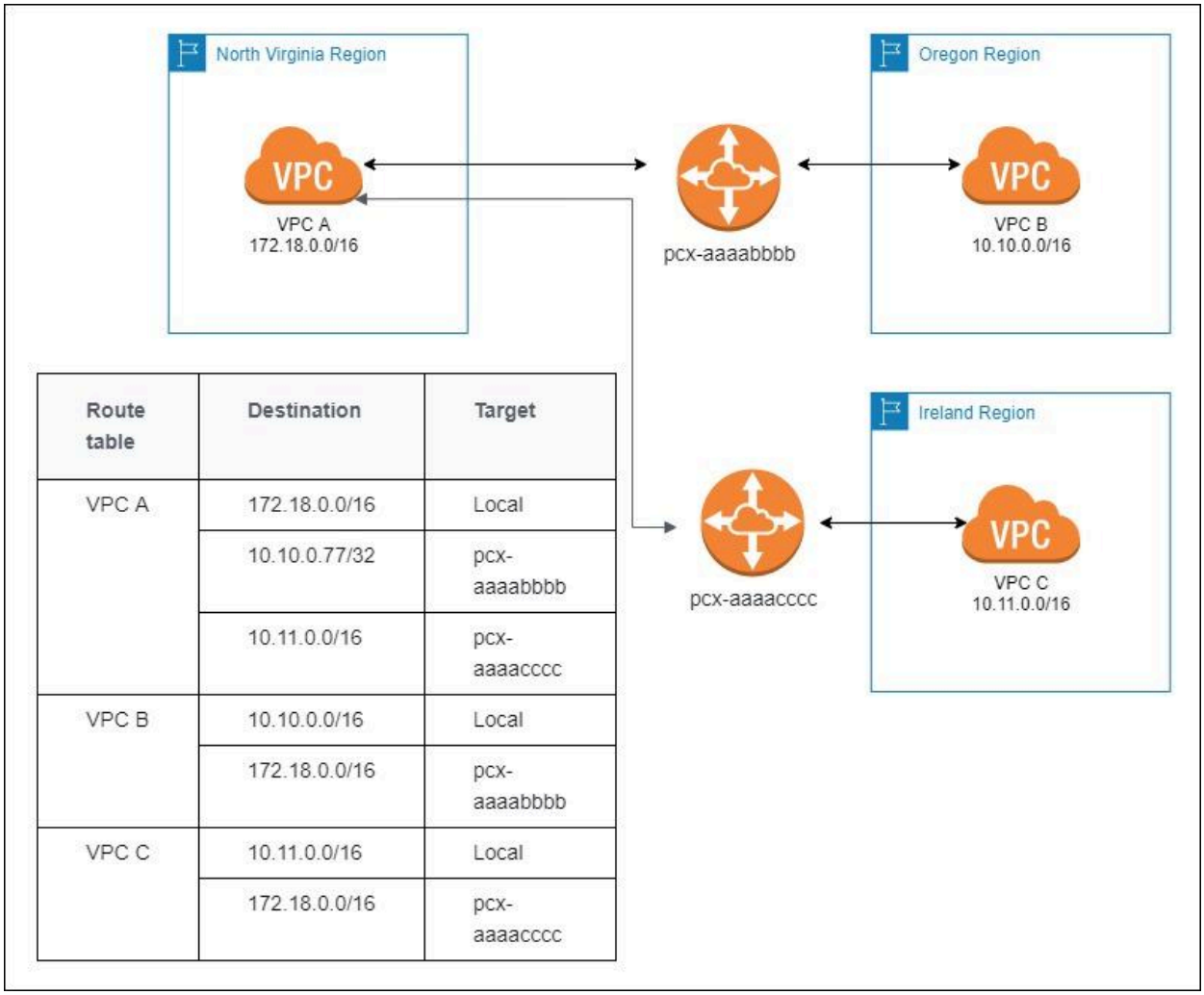
Multi-Account Network Configuration

Managing networks is a huge part of your work as an AWS Solutions Architect. As an organization grows, the network requirements become more complex and intertwined. This requires careful modelling so that traffic flow won't get disrupted. It is also common for users to have more than one VPC in their environment. Oftentimes, these VPCs are placed in different locations or regions, requiring you to figure out the most appropriate solution to make sure they can communicate with each other. Some customers may also opt for a hybrid environment, wherein their AWS VPCs need to be connected to their own corporate network. AWS has a ton of options provided for these scenarios which we will discuss in the sections below.

Multi-VPC Connection and Routing

The simplest way to connect two VPCs together is to use **VPC Peering**. VPC Peering allows bidirectional communication, so once you have peered two VPCs and modified their route tables (and network ACLs if you are security-conscious) then you are good to go. VPC Peering allows you to peer VPCs in the same region, in different regions, or in different accounts. There are a few things to consider though when using VPC Peering:

1) The connection is not transitive. This means that you cannot peer VPC A and VPC B, where VPC B is peered with VPC C, and expect VPC A to communicate with VPC C through this connection. If you have this requirement, either peer VPC A and VPC C directly or use another solution.
2) When you are peered to a VPC, remember that your route table uses *longest prefix matching* to know which destination to send traffic to.
3) VPC Peering is **not** a great option for a hub-and-spoke model.



Route table	Destination	Target
VPC A	172.18.0.0/16	Local
	10.10.0.77/32	pcx-aaaabbbb
	10.11.0.0/16	pcx-aaaacccc
VPC B	10.10.0.0/16	Local
	172.18.0.0/16	pcx-aaaabbbb
VPC C	10.11.0.0/16	Local
	172.18.0.0/16	pcx-aaaacccc

Figure: VPC Peering Between Three VPCs in Different Regions

Transit VPC allows you to build a hub-and-spoke model by using a hub VPC to connect to multiple spoke VPCs through a VPN connection leveraging BGP over IPsec. This solution can also handle transitive routing thanks to the VPN overlay. Take note that VPCs using this solution will need a VPN Gateway.

A fairly new solution that lets you build a hub-and-spoke model without a custom VPN appliance is to use **AWS Transit Gateway**. Transit Gateways enable you to connect multiple VPCs together, and beyond this, you can also link VPN solutions and AWS Direct Connect connections to a transit gateway. Transit gateways are regional services, which means you can peer transit gateways if you have VPCs in other regions, but you cannot peer transit gateways belonging to the same region. Unlike VPC Peering, you cannot reference the security group of a VPC in another VPC's security group. You have to use IP addresses or IP ranges for your rules.

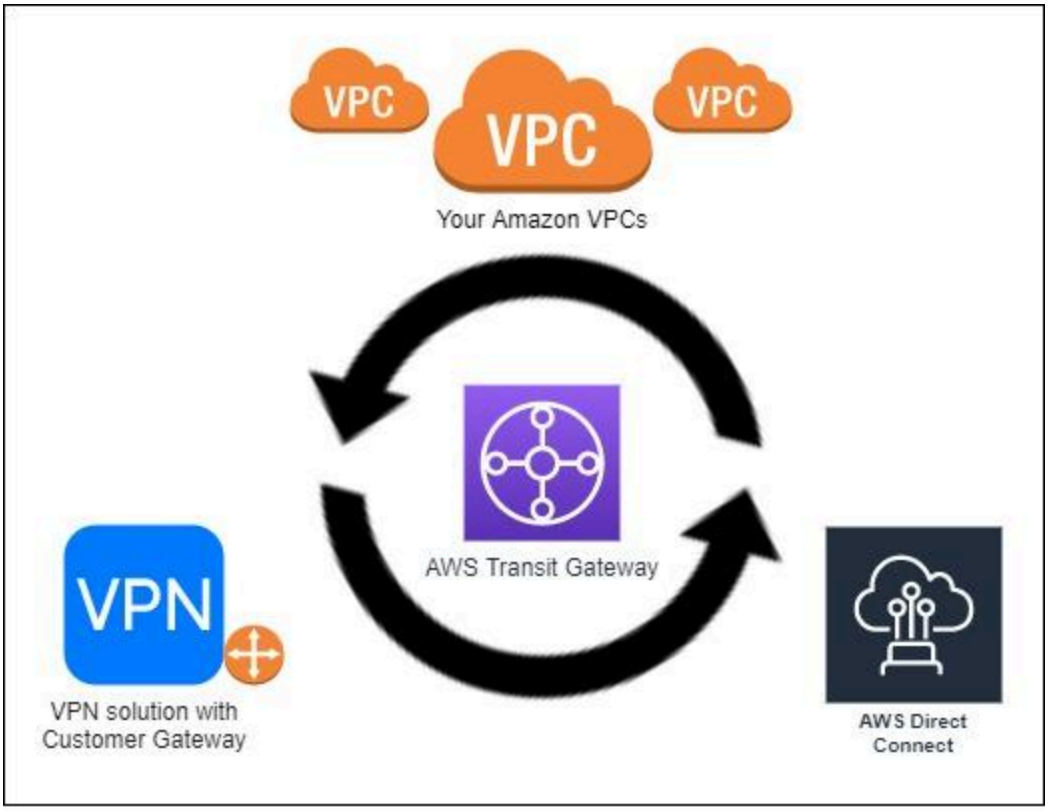


Figure: AWS Transit Gateway as a Hub-and-Spoke Solution

In case you already own an AWS Direct Connect connection on your on-premises center, you can **divide the physical connection into multiple logical connections**, one for each VPC. You can then use these logical connections for routing traffic between VPCs in the same region. You can also connect AWS Direct Connect locations in other regions using your existing WAN providers and leverage AWS Direct Connect to route traffic between regions over your WAN backbone network.

Setting up Network for Hybrid Environment

A very simple way to connect your on-premises network to AWS is through the use of a VPN. All you need is a VPN appliance in your on-premises location or use the managed AWS VPN service and a gateway in AWS to which the VPN will connect to. More often than not, customers will leverage an AWS Direct Connect line along with an IPsec VPN to ensure that traffic does not pass through the public network and they have a dedicated line for their network demands. Nonetheless, there are other solutions that you can apply with a VPN without having to purchase a pricey Direct Connect, at the expense of slower transfer speeds of course.

1. **AWS Transit Gateway supports IPsec termination for site-to-site VPN**. Register the VPCs that you need connection to in Transit Gateway and connect your VPN on the other end of the Transit Gateway service. Transit Gateway supports both Static and BGP-based Dynamic VPN connections.
2. **You can launch an EC2 instance and terminate the VPN on the instance** if you have a vendor that you or your company uses. This is a good alternative if you have compliance or compatibility issues. Afterwards, you can also peer your VPCs together to ensure that your VPN can reach your VPCs.
3. If you need one-to-one connectivity to your VPCs, **you can provision a Virtual Private Gateway (VGW)**. This is a good option if you only have a few VPCs to manage. As your environment grows larger, this might not be feasible anymore so you should instead adapt AWS Transit Gateway.

If you need a dedicated line for your traffic, provision an AWS Direct Connect from a provider and have it linked to your network. AWS Direct Connect provides many benefits compared to a VPN solution, such as a private connection to AWS, lower latency, and a higher network bandwidth. There are different ways to leverage Direct Connect:

1. If you need access to resources located inside a VPC, **create a private virtual interface (VIF) to a VGW attached to the VPC**. You can create 50 VIFs per Direct Connect connection, enabling you to connect to a maximum of 50 VPCs. Connectivity in this setup restricts you to the AWS Region that the Direct Connect location is homed to. This is not the best solution if you need to connect to a bunch of VPCs.
2. If your VPCs are located in different AWS Regions, **create a private VIF to a Direct Connect gateway associated with multiple VGWs**, where each VGW is attached to a VPC. You can attach multiple private virtual interfaces to your Direct Connect gateway from connections at any Direct Connect location. You have one BGP peering per Direct Connect Gateway per Direct Connect connection. This solution will not work if you need VPC-to-VPC connectivity.
3. You can associate a Transit Gateway to a Direct Connect gateway over a dedicated or hosted Direct Connect connection running at 1 Gbps or more. To do so, you need to create a **transit VIF to a Direct Connect gateway associated with Transit Gateway**. You can connect up to 3 transit gateways across different AWS Regions and AWS accounts over one VIF and BGP peering. This is the most scalable and manageable option if you have to connect to multiple VPCs in multiple locations.
4. If you need access to AWS public endpoints or services reachable from a public IP address (such as public EC2 instances, Amazon S3, and Amazon DynamoDB), **create a VPN connection to Transit Gateway over Direct Connect public VIF.** You can connect to any public AWS service and AWS Public IP in any AWS Region. When you create a VPN attachment on a Transit Gateway, you get two public IP addresses for VPN termination at the AWS end. These public IPs are reachable over the public VIF. You can create as many VPN connections to as many Transit Gateways as you want over public VIF. When you create a BGP peering over the public VIF, AWS advertises the entire AWS public IP range to your router.

AWS Direct Connect supports both IPv4 and IPv6 on public and private VIFs. You will be able to add an IPv6 peering session to an existing VIF with IPv4 peering session (or vice versa). You can also create 2 separate VIFs – one for IPv4 and another one for IPv6.

References:
https://d1.awsstatic.com/whitepapers/building-a-scalable-and-secure-multi-vpc-aws-network-infrastructure.pdf
https://d1.awsstatic.com/whitepapers/aws-amazon-vpc-connectivity-options.pdf
https://docs.aws.amazon.com/vpc/latest/peering/what-is-vpc-peering.html
https://docs.aws.amazon.com/directconnect/latest/UserGuide/direct-connect-gateways.html

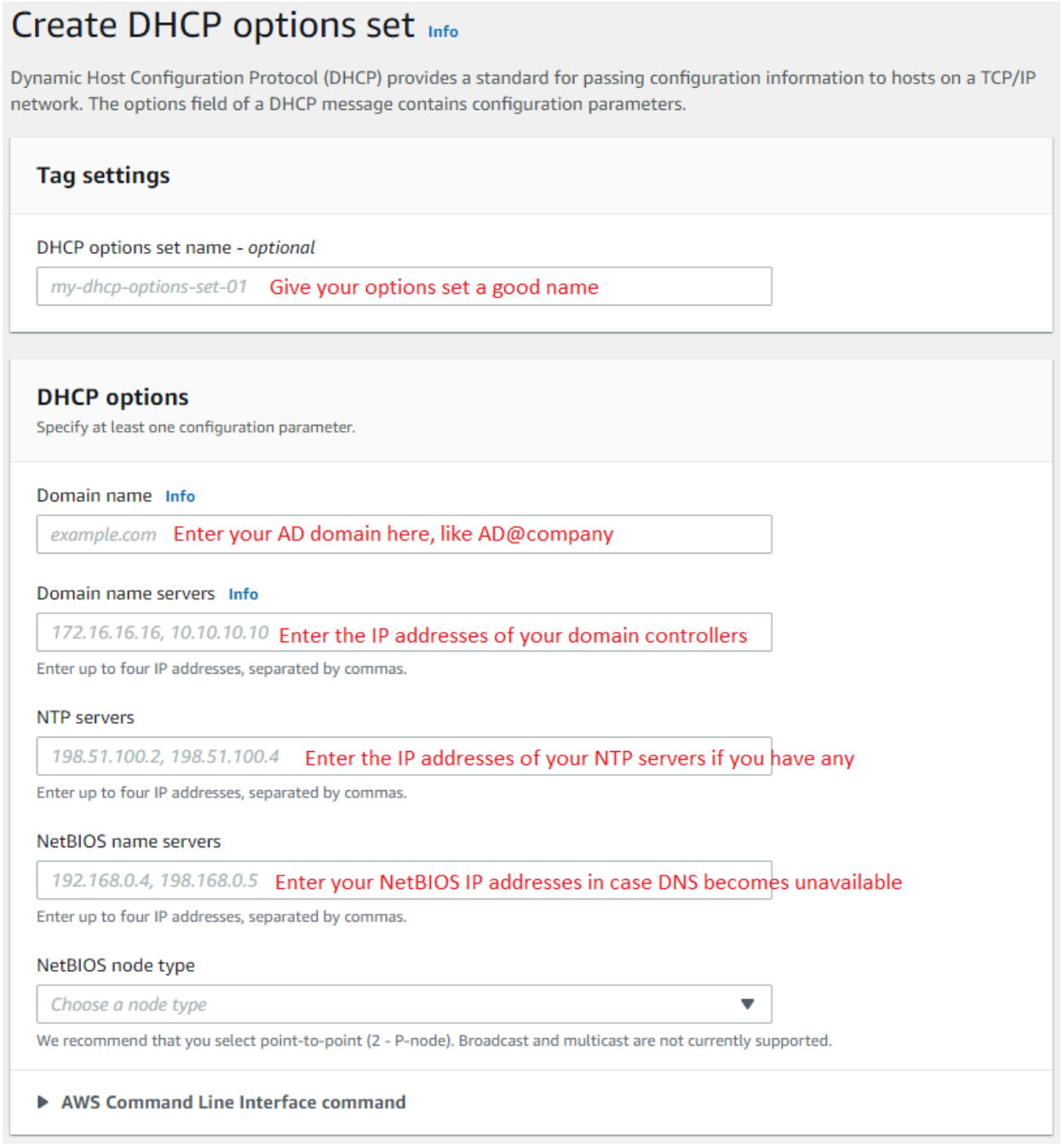
Configuring DNS Resolution for your Servers

In large organizations, it is common to use an Active Directory to manage all your users, computers, and policies. Having an Active Directory also allows you to separate these identities into different groups and domains. You can then apply Group Policy Objects or GPOs to properly secure and manage your environment to meet requirements. Most of the time, computers within a domain only talk to each other within a private network. This requires having a single source of truth as to which computer owns which IP address. Therefore, you must also configure your Active Directory domain controllers to become DNS providers for your domain members. Once the DNS is up, you go into each of your domain computers and configure the network to use your domain controllers as your DNS servers. However, when you are managing multiple computers at the same time, it can be difficult to keep track if they are properly communicating with your DNS servers, especially if you're in AWS handling multiple accounts and multiple environments. To simplify this task, you can instead create DHCP option sets for your VPCs.

The Dynamic Host Configuration Protocol (DHCP) provides a standard for passing configuration information to hosts on a TCP/IP network. When you launch private instances in a non-default VPC, these instances receive an unresolvable hostname from AWS. You can assign your own domain name to your instances, and use up to four of your own DNS servers. To do so, you must specify a set of DHCP options to use with the VPC. The DHCP options set will help resolve your desired domain name to your DNS servers, which reduces the chances of misconfiguration. You can also specify your NTP servers and NetBIOS name servers if you are using any. These information, once attached to the selected VPC, are made available to all EC2 instances running in that VPC.

After you create a set of DHCP options, you can't modify them. If you want your VPC to use a different set of DHCP options, you must create a new set and associate them with your VPC. You can also set up your VPC to use no DHCP options at all. You can have multiple sets of DHCP options, but you can associate only one set of DHCP options with a VPC at a time. By default, when you create a new VPC, AWS automatically creates a set of DHCP options and associates them with the VPC. This set includes two options:

domain-name-servers=*AmazonProvidedDNS*, and domain-name=*domain-name-for-your-region*. **AmazonProvidedDNS** is an Amazon Route 53 Resolver server, and this option enables DNS for instances that need to communicate **over the VPC's Internet gateway**.



Your VPC has attributes that determine whether instances launched in the VPC receive public DNS hostnames that correspond to their public IP addresses, and whether DNS resolution through the Amazon DNS server is

supported for the VPC. These attributes are **enableDnsHostnames** and **enableDnsSupport**. Values for these two attributes are true and false. By default, both of these are true.

VPC Rules:

- If both attributes are set to true:
  - Instances with a public IP address receive corresponding public DNS hostnames.
  - The Amazon Route 53 Resolver server can resolve Amazon-provided private DNS hostnames.
- If either or both of the attributes is set to false:
  - Instances with a public IP address do not receive corresponding public DNS hostnames.
  - The Amazon Route 53 Resolver cannot resolve Amazon-provided private DNS hostnames.
- Instances receive custom private DNS hostnames if there is a custom domain name in the DHCP options set. If you are not using the Amazon Route 53 Resolver server, your custom domain name servers must resolve the hostname.
- If you want to access the resources in your VPC using custom DNS domain names, you can create a private hosted zone in Route 53. Using custom DNS domain names defined in a private hosted zone in Route 53, or using private DNS with interface VPC endpoints (AWS PrivateLink) requires you to set both VPC attributes to true.
- The Amazon Route 53 Resolver can resolve private DNS hostnames to private IPv4 addresses for all address spaces.

ℹ️ **Solutions Architect Professional Exam Notes:**

How do I use both Active Directory and VPC Resolver for private DNS resolution in my VPC?

Once you have set up your DHCP options set with your own DNS servers, all your instances that are joined to the domain will rely on these servers for DNS-related functions. For local VPC domain names that need to be resolved to their respective IP addresses, you can configure your Active Directory to forward these types of queries to the Route 53 resolver instead via an inbound endpoint. The Route 53 resolver will then return the private IP address of those instances for you.

References:
https://docs.aws.amazon.com/vpc/latest/userguide/vpc-dns.html
https://docs.aws.amazon.com/vpc/latest/userguide/VPC_DHCP_Options.html
https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/resolver.html

Domain 2: Design for New Solutions

Overview

The second domain of the AWS Certified Solutions Architect Professional exam focuses on designing solutions for specific business objectives. The majority of your work as a Solutions Architect Professional lies in building concrete solutions that follow industry best practices to meet certain regulatory compliances and to serve your customers better. You need to be very knowledgeable about the different AWS services and other external tools that you can use to complement your existing cloud architectures. Having an in-depth understanding of what you're working with will allow you to design a strategy that works best for your customer's business objectives while maintaining the costs in check.

This domain is the biggest one among all other SAP-C02 exam domains, with 29% coverage. Ensure that you know the concept of Infrastructure as code (IaC) and how to deploy your cloud stack across different AWS accounts using AWS CloudFormation. Setting up Continuous Integration/Continuous Delivery (CI/CD) and Change Management processes in AWS Elastic Beanstalk, AWS CodeDeploy, Amazon ECS Anywhere, Amazon EKS Anywhere, and deployment strategies are included. You should also know how to use the Configuration Management tools available in AWS Cloud, such as AWS Systems Manager, AWS Service Catalog, AWS Config, et cetera.

The questions in the actual exam revolve around the following tasks:

- Designing a deployment strategy to meet business requirements
- Designing a solution to ensure business continuity
- Determining security controls based on requirements
- Designing a strategy to meet reliability requirements
- Designing a solution to meet performance objectives.
- Determining a cost optimization strategy to meet solution goals and objectives

In this chapter, we will cover the related topics for designing solutions centered around security, performance, cost optimization and reliability, as well as different deployment strategies in AWS that will likely show up in your Solutions Architect Professional exam.

Using Amazon AppStream 2.0 / Amazon Workspaces for Remote Desktop Operations

Amazon AppStream and Amazon Workspaces are both great services if you need to stream virtual workstations and desktop applications without worrying about hardware. Compared to running virtual machines, virtual workstations can scale and load balance automatically to match user demand. They are made globally available so they can be accessed almost anywhere. Both these services also integrate seamlessly with active directory so you can apply your organization policies and protocols.

When connecting to EC2 virtual machines, you do so via SSH or RDP connection. The method of accessing your Workspaces workstations is different, since you will need a Workspaces client for this. Multiple devices can access the workstation as long as they have the client. During configuration, Workspaces offers multiple bundles as a base, which include a mix of hardware and software options for you to choose from. With regards to overall expenses, larger corporations benefit more from Amazon Workspaces due to the service requiring you to use an active directory for managing your users.

If you only need to share access to your applications and not a whole desktop environment then Amazon Appstream is the service to use. You can stream your applications to any computer without having to provision and operate hardware. All you need is a HTML5-capable browser. Appstream allows you to restrict the applications that are available for use to your users and prevent them from using any other unrelated applications. Appstream fleets can be linked to an active directory, but this is not mandatory. Appstream also allows you to share your applications on a massive scale, which is normally constrained by hardware, thanks to the capacity made available by AWS infrastructure.

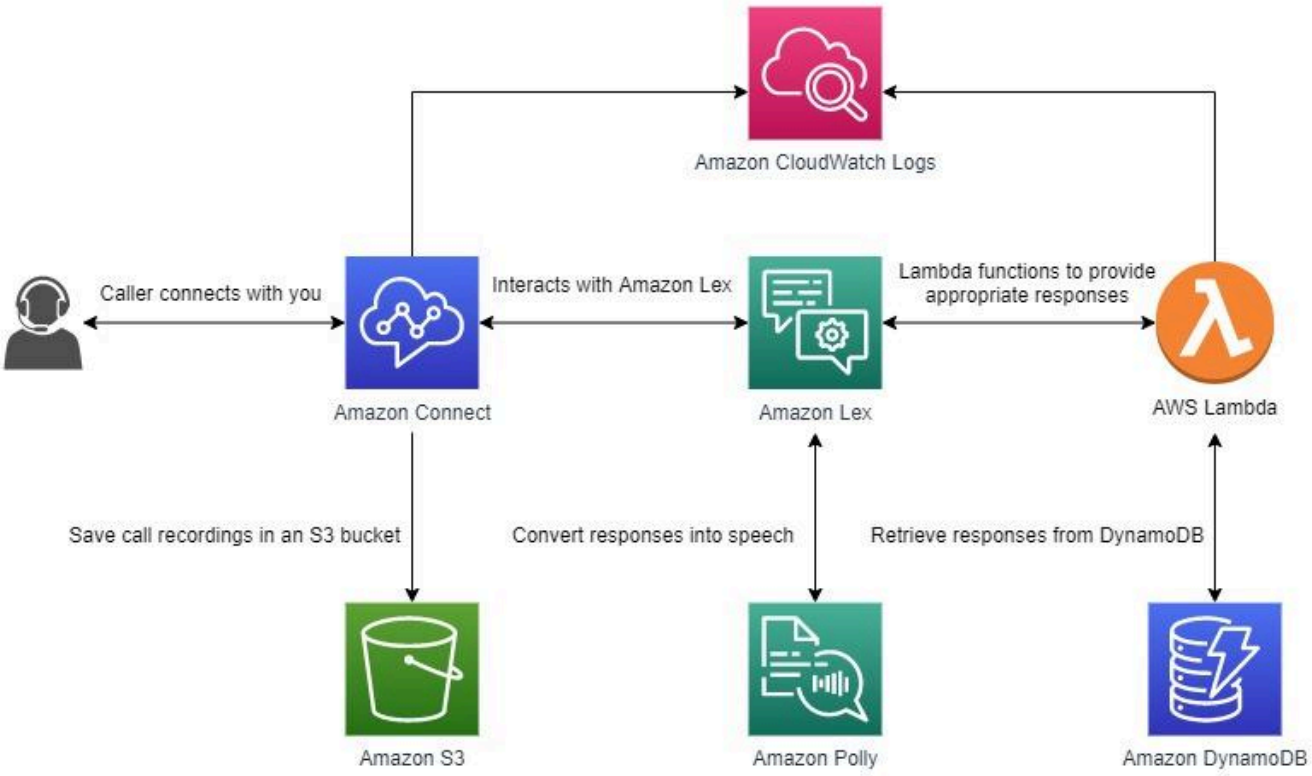
References:
https://docs.aws.amazon.com/whitepapers/latest/aws-overview/desktop-app-streaming-services.html
https://aws.amazon.com/workspaces/features/?nc=sn&loc=2
https://aws.amazon.com/appstream2/features/?nc=sn&loc=3
AWS re:Invent 2018: Desktops & Applications to AWS with Amazon WorkSpaces & AppStream 2.0

Using Amazon Connect, Amazon Lex, and Amazon Polly For Chat and Call Functionality

If your business involves taking calls and communicating with customers via phone, you can quickly set up a contact center using Amazon Connect, Amazon Lex, and Amazon Polly. Amazon Connect is an easy to use omnichannel cloud contact center that supports many useful features for voice and chat operations. Customers can easily call into your Amazon Connect contact center using any phone and speak to an agent. You can also set up web and voice chat via APIs. You can design workflows that route calls and messages to the appropriate responder. If you need to perform language processing when you receive calls, Amazon Connect can be integrated with Amazon Lex for *Natural Language Understanding* and automated customer interactions. For text-to-speech messages, you may also use Amazon Polly. Calls can also be recorded and stored in Amazon S3 for future analytics. Lastly, it is common nowadays to have interactive chat boxes running on your website for visitors. You can easily configure your automated responses for some expected queries and have them invoke Lambda functions to handle any external processing.

Amazon Connect, Amazon Lex, and Amazon Polly make it very convenient for call centers and businesses that offer phone and chat support to quickly set up a production-ready contact center. These services allow you to respond back quickly to your customers and offer top quality support. Amazon Connect provides out-of-the-box integrations for Salesforce and Zendesk if you are using these tools already. You can also run machine learning workloads on your recordings which will further improve your engagement with your customers. If you have a chatbot running on your website, the recording data will definitely serve useful in training your bot.

Amazon Lex is a machine learning service that you can use to develop client-facing chatbots for your customers. You can launch either a Text-based or Voice-based chatbot with Amazon Lex and then integrate it on your website or to the other AWS services. This minimizes the dependency of having a virtual agent for your conversational Interactive Voice Response (IVR) system, corporate website, or other customer-facing system. It can also lower down the costs of companies in maintaining its contact center.



References:
https://aws.amazon.com/blogs/aws/amazon-connect-customer-contact-center-in-the-cloud/
https://aws.amazon.com/blogs/aws/new-omnichannel-contact-center-web-and-mobile-chat-for-amazon-connect/

Using Amazon WorkDocs for Secure Document Management and Collaboration

Most of the users in AWS use more well-known storage services such as Amazon S3, Amazon EBS, and Amazon EFS for file storage and file sharing purposes. Others may use third-party tools such as Google Drive, DropBox, and Microsoft OneDrive, which may offer more functionality for the user, or even local storage drives in case there are compliance requirements to meet. While it is perfectly fine to use these services, don't forget that AWS also offers its own product for secure content storage and collaboration through Amazon Workdocs. Compared to storage services such as Amazon S3 or your own storage devices, Amazon WorkDocs provides the following additional benefits:

1) Amazon WorkDocs offers unlimited versioning. A new version of a file is created every time you save it.
2) You can invite other AWS users to view, contribute to, or co-own your files by entering user names, group names, and email addresses. You can also request specific feedback with a personal message and set a deadline.
3) Amazon WorkDocs lets you use your Active Directory to manage your users.
4) Amazon WorkDocs lets you control who can access, comment, and download or print your files. You can "lock" files to ensure that edits are not overwritten by other contributors, eliminating the need to coordinate changes.
5) You can create an approval workflow to route documents and other files stored in WorkDocs to one or more users for their approval. This allows you to track and manage your documents easier.
6) *Amazon WorkDocs Drive* is a desktop application that connects your computer to your Workdocs filesystem so that all of your files are available on-demand from your device, eliminating the need to use network shares.
7) You can use AWS API to interact with your WorkDocs filesystem programmatically.

ℹ️ Solutions Architect Professional Exam Notes:

So when should I choose Amazon WorkDocs over other options?

Since you will be taking an AWS exam, third-party options are already out of the question. But when you are made to choose between the different AWS Storage Services and Amazon WorkDocs, keep an eye out for key terms such as "content management", "collaboration" and "document sharing". These usually indicate that the scenario requires a convenient document collaboration tool which is what Amazon WorkDocs is. Lastly, be sure to read through the benefits listed above, since they will be useful in letting you evaluate if Amazon WorkDocs is the better choice.

Reference:
https://aws.amazon.com/workdocs/

Using AWS Data Exchange for Proprietary Data Access in Amazon Redshift

AWS Data Exchange allows data providers to share data products with subscribers, who can then use this data for their own purposes, such as training a machine learning model. For businesses that process and store proprietary data in Amazon Redshift, the challenge often lies in securely and sharing this data with external customers. If you want to monetize your data, making them available to users and managing the access takes a lot of work.

AWS Data Exchange simplifies this process significantly. To share data processed within Amazon Redshift, the steps are as follows:

1. Create a Datashare within Amazon Redshift: As a data provider, you start by creating a datashare in your Amazon Redshift cluster. This involves selecting the specific schemas, tables, views, and user-defined functions that you intend to share. The datashare acts as a container for the data you wish to monetize, allowing you to control exactly what is shared and ensuring that your proprietary data is handled securely.
2. Import the Datashare into AWS Data Exchange: Once the datashare is created, the next step is to import it into AWS Data Exchange. This process involves creating a new dataset within AWS Data Exchange, into which your Redshift datashare is incorporated. This dataset forms the basis of the product you will offer to potential subscribers.
3. Create and Publish Your Product: With your dataset in place, you can now create a product in AWS Data Exchange. This product encapsulates the data you wish to share and can include additional details such as pricing, terms of use, and a description of the data. Once your product is ready, you publish it on AWS Data Exchange, making it available to potential subscribers.
4. Subscription by Customers: After publication, customers can subscribe to your product. Subscribers are granted read-only access to the data you have shared, allowing them to view this data within their without the ability to modify the original data.

References:
https://docs.aws.amazon.com/data-exchange/
https://aws.amazon.com/blogs/aws/new-aws-data-exchange-for-amazon-redshift/

Implementing DDoS Resiliency in AWS

Websites and web applications are always under threat of security attacks. Today, there are many ways for people to exploit a vulnerability to steal data, cause huge amounts of downtime, and prevent your applications from recovering swiftly and properly. One infamous security attack that affects thousands of websites and

endpoints each day is known as Denial of Service. A Denial of Service (DoS) attack is a deliberate attempt to make your website or application unavailable to users, by flooding it with network traffic for example. To achieve this, attackers use a variety of techniques that consume large amounts of network bandwidth or tie up other system resources, disrupting access for legitimate users.

A more extreme version of this is DDoS, or distributed denial of service, where the attacker uses multiple computers to perform the attack. DDoS attacks are most common at layers 3, 4, 6, and 7 of the OSI model. To address this, we will be taking a look at some of the tools that AWS provides to you as their customer to defend against these types of attacks.

AWS WAF is a web application firewall that helps protect your web applications or APIs against common web exploits such as flood attacks, XSS, and SQL injection attacks. In AWS WAF, you can use a web access control list to protect your AWS resources. You create a web ACL and define its protection strategy by adding rules. Rules define criteria for inspecting web requests and specify how to handle requests that match the criteria. You set a default action for the web ACL that indicates whether to block or allow those requests that pass the rules inspections. To help you get started, AWS and AWS Marketplace vendors offer preconfigured WAF rule groups. These rule groups contain a set of rules that will help protect you from common security threats and exploits. The resource types that you can protect using WAF web ACLs are Amazon CloudFront distributions, Amazon API Gateway REST APIs, and Application Load Balancers.

To mitigate a potential layer 7 DDoS attack, create conditions in AWS WAF that match the unusual behavior. Configure the web ACL to count the requests that match the rules. If the volume of requests continues to be unusually high, change your web ACL to block those requests.

AWS Shield is a service that protects your resources from DDoS attacks. AWS Shield Standard provides protection against common and most frequently occurring infrastructure (layer 3 and 4) attacks like SYN/UDP floods, reflection attacks, and others to support high availability of your applications on AWS. AWS Shield Standard is automatically included in every AWS account at no extra cost. AWS Shield Advanced is an optional paid subscription that provides enhanced protections for your applications running on EC2 with EIPs, ELB load balancers, CloudFront distributions, AWS Global Accelerator, and Route 53 resources against more sophisticated and larger attacks. Using AWS Shield Advanced with EIPs allows you to protect Network Load Balancer (NLBs). If you are using Amazon CloudFront and Amazon Route 53, these services receive comprehensive availability protection against all known infrastructure attacks.

Amazon Route 53 and Amazon CloudFront are two services that you should employ to maximize your defense against DoS attacks in AWS. Evident from the previous security services, Route 53, and CloudFront both easily integrate with AWS WAF and Shield. AWS edge locations also provide an additional layer of network infrastructure that provides these benefits to any web application that uses CloudFront and Route 53.

Persistent connections and variable time-to-live (TTL) settings can be used to offload traffic from your origin, even if you are not serving cacheable content. These features mean that using CloudFront reduces the number of requests and TCP connections back to your origin which helps protect your web application from HTTP floods. Amazon CloudFront only accepts well-formed connections, which helps prevent many common DDoS attacks, like SYN floods and UDP reflection attacks, from reaching your origin. DDoS attacks are also geographically isolated close to the source which prevents the traffic from impacting other locations.

Amazon Route 53 is a highly available and scalable domain name system. It uses shuffle sharding and anycast striping to prevent a DDoS attack from affecting your website's availability. With shuffle sharding, each name server in your delegation set corresponds to a unique set of edge locations and internet paths. If one of these name servers becomes unavailable, users can retry their request and receive a response from another name server at a different edge location. Anycast striping allows each DNS request to be served by the most optimal location, spreading the network load and reducing DNS latency. Additionally, Route 53 can detect anomalies in the source and volume of DNS queries, and prioritize requests from users that are known to be reliable.

ℹ️ **Solutions Architect Professional Exam Notes:**

In the exam, it is common to see AWS WAF being used in conjunction with CloudFront or ALB to defend against security attacks. Remember that WAF does not support integration with NLB or CLB, nor does it support direct EC2 integration. You can use WAF with Amazon ECS, as long as your ECS cluster has an ALB in front. Also, take note that AWS WAF is not the best service to use against DDoS attacks, due to the fact that a lot of the mitigation effort requires manual activity. If you are given the option to select AWS Shield in a DDoS scenario, you should lean more on this option since the mitigation effort is automated.

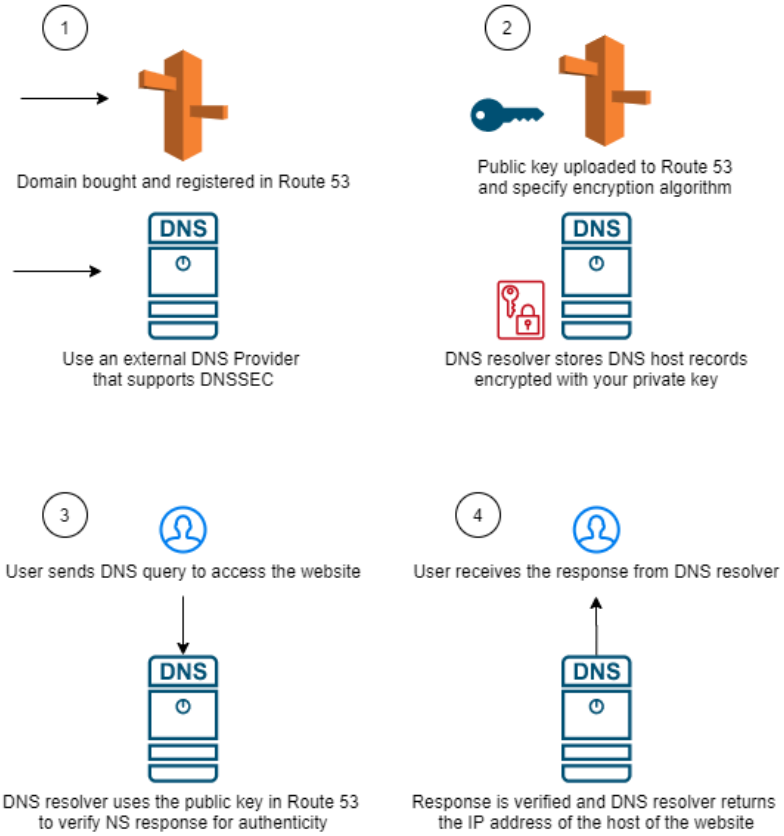
Other options you might encounter include using load balancers and auto scaling groups, or enforcing rules in security groups and network ACLs. Although these services can help prolong your website's availability and keep a few attacker IP addresses at bay, they do not protect your web servers completely from the different security threats. You won't be able to filter out all of the IP addresses if it is a large scale attack, and your instances can only scale out so much until you hit your max size.

References:
https://d1.awsstatic.com/whitepapers/Security/DDoS_White_Paper.pdf
https://docs.aws.amazon.com/waf/latest/developerguide/waf-chapter.html
https://docs.aws.amazon.com/waf/latest/developerguide/ddos-overview.html

Configuring DNSSEC for a Domain in Route 53

Sometimes, dodgy people just feel like doing dodgy stuff. It is up to you to keep yourself and your resources protected from their maliciousness. Everything you expose on the Internet needs to be properly secured. And yes, that includes your own web domain. Attackers sometimes hijack traffic to Internet endpoints such as web servers by forging DNS data and fooling DNS resolvers to route users to the IP addresses provided by the attackers, for example, to fake websites. This type of attack is known as DNS spoofing or a man-in-the-middle attack. Not only does this mean that you lose reputation for your websites, but also the fact that your other Internet endpoints might be affected as well.

You can protect your domain from this type of attacks by configuring **Domain Name System Security Extensions (DNSSEC)**, a protocol for securing DNS traffic. DNSSEC works by using asymmetric encryption to secure your DNS records and verify domain requests sent to the DNS resolver. Amazon Route 53 supports DNSSEC for domain registration as well as DNSSEC for its DNS service.



Reference:
https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/domain-configure-dnssec.html

Configuration Management in AWS with AWS Systems Manager

When managing servers, the traditional approach for applying changes or performing checks involves manually logging into each server to update configurations or run diagnostics. This method is manageable for a single server or a small setup. However, at scale, with potentially hundreds or thousands of servers spread across different environments, this approach quickly becomes impractical. It's not only time-consuming but also increases the risk of inconsistencies and errors.

Imagine you're managing hundreds of servers and one employee at your work accidentally misconfigures a server, perhaps by mistakenly altering firewall settings. This single misstep could expose your organization to significant security risks, potentially leading to data breaches.

AWS Systems Manager State Manager for Consistency

Systems Manager State Manager allows you to define a desired configuration state for your servers, covering everything from security policies and software installations to system settings. Once defined, State Manager continually enforces these configurations across your entire server fleet. For example, you could use State Manager to ensure that firewall rules across all servers only permit traffic from trusted IP addresses and block all unauthorized access attempts. If there are changes to these settings on any server, State Manager would automatically detect this deviation from the desired state and revert the settings back to the correct configuration.

Responding to Emergencies with Run Command

For those times when urgent interventions are required—be it deploying emergency patches or executing critical commands across multiple servers, Systems Manager Run Command is helpful. It allows you to execute commands on your servers at scale, without the need to log into each one individually. This capability is crucial for quickly responding to vulnerabilities or ensuring that all servers are running the latest software, providing a layer of agility and security that is indispensable in modern server management.

References:
https://docs.aws.amazon.com/systems-manager/latest/userguide/systems-manager-state.html
https://docs.aws.amazon.com/systems-manager/latest/userguide/run-command.html

Using Lambda@Edge for Low Latency Access to your Applications

Perhaps you have products that are catering to a global audience, or your traffic is being served worldwide via the vast network channel of Amazon CloudFront. Although you are able to serve your content very quickly thanks to the CDN service, this will only be useful if all the content you serve is cacheable content. If perhaps you have an application that runs in one region while your customers are in other locations then surely they will experience the latency in the responsiveness of your app. Running multiple copies of your infrastructure and applications in different regions can be a solution to this problem, but it is costly and has a lot of management overhead. A better option would be to actually bring your applications closer to your customers using the multiple Edge Locations placed around the globe using Lambda@Edge.

Re-architecting can be scary for some people, since it involves a lot of money, effort and time to successfully migrate a working application to serverless. Though in many cases, re-architecting actually gives a lot of advantages in return. AWS Lambda functions specifically offer a rich set of features and integrations that you can use to improve the performance of your web applications, such as Lambda@Edge with Amazon CloudFront. With Lambda@Edge, your Node JS and Python Lambda functions are executed nearest to your customer's location, and it easily scales as well to keep up with the demand. This significantly reduces latency and improves the user experience.

When you associate a CloudFront distribution with a Lambda@Edge function, CloudFront intercepts requests and responses at CloudFront edge locations. You can execute Lambda functions when the following CloudFront events occur:

- When CloudFront receives a request from a viewer (viewer request)
- Before CloudFront forwards a request to the origin (origin request)
- When CloudFront receives a response from the origin (origin response)
- Before CloudFront returns the response to the viewer (viewer response)

Steps to deploy a Lambda@Edge function:

1. To deploy your Lambda functions at CloudFront Edge Locations, you must first create and publish a Lambda function in the US-East-1 (N. Virginia) Region. Remember that the language should be either Node JS or Python.
2. After publishing it, choose (or create) the CloudFront distribution to be associated with and modify the cache behavior.
3. Select the event type or the *trigger* for your function. After the trigger is created, your Lambda function is now replicated around the world.
4. Verify that your function runs properly. If you receive an error saying that CloudFront cannot execute your Lambda function, be sure to add a policy that allows this action in the IAM role of your function. You can also consult CloudWatch Logs for further information.

5. To update your function, you must edit the $LATEST version of the function in the US-East-1 (N. Virginia) Region. Then, before you set it up to work with CloudFront, you publish a new numbered version.

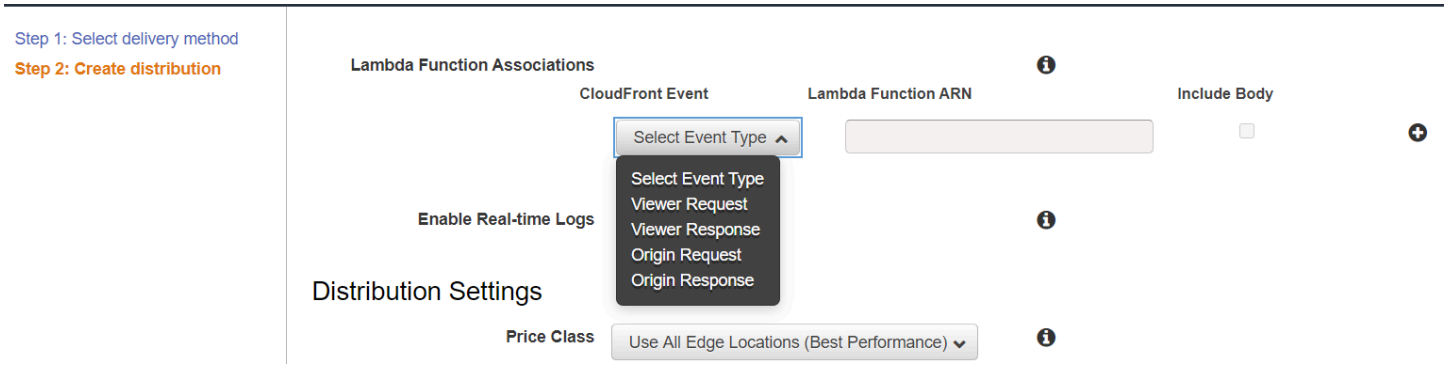


Figure: Configuring your CloudFront to add a Lambda function

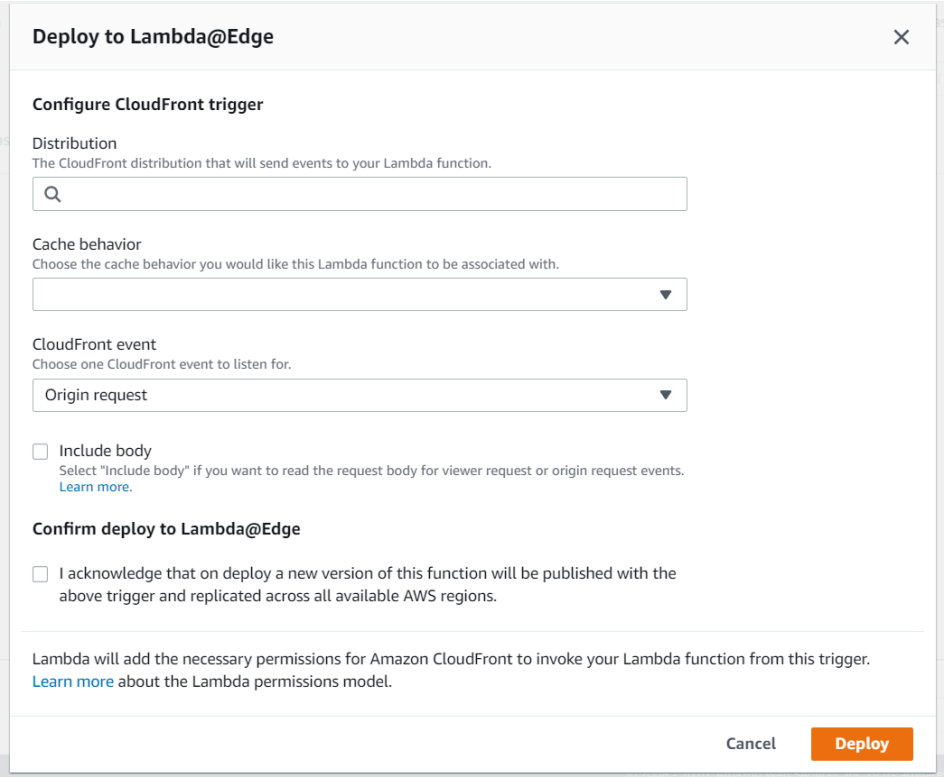


Figure: Adding a trigger on your Lambda function and specifying the CloudFront distribution

Example use cases of Lambda@Edge include:

- Work as an extension of or replacement for your origin. This enables you to do everything from simple HTTP request and response processing at the edge to more advanced functionalities, such as website security, real-time image transformation, intelligent bot mitigation, search engine optimization, and more.
- Adding HTTP security headers on all origin responses without having to modify your application code on your origin. This helps improve security and privacy for your users and content providers, while using CloudFront to deliver the content at low latencies.
- Working with Amazon Cognito to provide user authentication for your applications based on location. You can also filter out unauthorized requests before they reach your origin infrastructure.

References:
https://aws.amazon.com/blogs/networking-and-content-delivery/reducing-latency-and-shifting-compute-to-the-edge-with-lambdaedge/
https://aws.amazon.com/lambda/edge/
https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/lambda-at-the-edge.html

Setting Up an ELK (ElasticSearch, Logstash and Kibana) Stack Using Amazon OpenSearch

So what is an ELK stack, and what makes it popular in log analytics? ELK is an acronym for three popular open-source projects — ElasticSearch, Logstash, and Kibana. The ELK stack allows you to aggregate logs from all your systems and applications (logstash), analyze these logs (elasticsearch), and create visualizations for application and infrastructure monitoring, faster troubleshooting, security analytics, and more (kibana). You can imagine that in a public cloud space where there are lots of applications and services producing large amounts of log data, one will need a robust and scalable solution to manage these logs and obtain value-adding information.

	Elasticsearch	Logstash	Kibana
Definition	An open-source, RESTful, distributed search and analytics engine built on Apache Lucene.	An open-source data ingestion tool that allows you to collect data from a variety of sources, transform it, and send it to your desired destination.	An open-source data visualization and exploration tool for reviewing logs and events.
How it works	You forward data in the form of JSON documents to Elasticsearch using the API or ingestion tools such	Logstash ingests data from multiple sources, then the data is transformed through a series of filters	Logs and documents sent to Elasticsearch can be visualized in Kibana for graphical views and

	as Logstash and Amazon Kinesis Firehose. Elasticsearch automatically stores the original document and adds a searchable reference to the document in the cluster's index. You can then search and retrieve the document using the Elasticsearch API.	which you design, and finally outputs your data into a "stash".	aggregated representations of your data.
Use cases	Log search, document indexing, log storage, and more.	Data transformation to make the output usable by your receiving applications.	Log and time-series analytics, application monitoring, and operational intelligence use cases.

So why use Amazon OpenSearch Service if you can deploy your own ELK stack? Amazon OpenSearch Service is a fully managed service, cost-effective, and can run at petabyte-scale. You can deploy your own stack onto an EC2 instance or your own servers for that matter, but you do get that additional management overhead as well as configuring scaling to meet demand. Amazon OpenSearch already offers support for Elasticsearch APIs, built-in Kibana, and integration with Logstash, so transitioning between an AWS and a non-AWS ELK deployment is very easy to do. Lastly, Amazon OpenSearch integrates with other AWS services such as Amazon Data Firehose, Amazon CloudWatch Logs, and AWS IoT, giving you the flexibility to select the data ingestion tool that meets your use case requirements.

ℹ️ Solutions Architect Professional Exam Notes:

Whenever you encounter Elasticsearch in AWS, always consider the options that discuss the Amazon OpenSearch service. It is the most convenient, scalable, and cost-effective solution that you can use to run your ELK stack on AWS. If you need to migrate an on-premises ELK stack to AWS, deploy your Amazon OpenSearch domain first and configure your new cluster. After that, you can utilize AWS Database Migration Service (AWS DMS) to migrate data to Amazon OpenSearch Service from all AWS DMS–supported sources, which currently are:

- Oracle DB
- MS SQL Server
- MySQL
- MariaDB
- PostgreSQL
- MongoDB

- SAP ASE
- IBM Db2
- Azure SQL Database
- Amazon Aurora
- Amazon S3

If your log source is not in this list, you can instead pause the ingestion and export your data and indexes from your existing Elasticsearch. Then import it into your new OpenSearch cluster. Once done, you can repoint your log ingestion to use your new cluster. The open-source Elasticsearch is a great tool to use, but it still involves some additional learning from the user. So unless the scenario calls for Elasticsearch explicitly, consider your other options for log management first such as Cloudwatch Logs or S3.

Reference:
https://aws.amazon.com/elasticsearch-service/the-elk-stack/
https://aws.amazon.com/blogs/database/introducing-amazon-elasticsearch-service-as-a-target-in-aws-database-migration-service/

Data Analytics and Visualization Using Amazon Athena and Amazon Quick

Performing data analytics in AWS has never been easier thanks to the wide array of services at your disposal. With Amazon S3, you can cost-effectively build and scale a data lake of any size in a secure environment where data is durably stored. Amazon S3 is an object storage solution so it supports almost all kinds of file types. Once you have built your own data lake, you can use the different services that readily integrate with Amazon S3, such as Amazon Athena, to perform data analytics and data processing.

Amazon Athena is a service that lets you run queries on your S3 objects using SQL. If the files stored in your bucket have a common format (let's say load balancer logs for example), you can create a table in Amazon Athena pointing to your S3 bucket and define the schema used by your files. Once the table is generated, you can run your standard SQL queries and Amazon Athena will handle the parsing based on the schema you defined. The results of your queries are saved in a different S3 bucket in case you need them later on. Below is an example of creating a table to parse application load balancer logs:

```
CREATE EXTERNAL TABLE IF NOT EXISTS myalblogs (
    type string,
    time string,
    elb string,
    client_ip string,
    client_port int,
    target_ip string,
    target_port int,
    request_processing_time double,
    target_processing_time double,
    response_processing_time double,
    elb_status_code string,
```

```
    target_status_code string,
    received_bytes bigint,
    sent_bytes bigint,
    request_verb string,
    request_url string,
    request_proto string,
    user_agent string,
    ssl_cipher string,
    ssl_protocol string,
    target_group_arn string,
    trace_id string,
    domain_name string,
    chosen_cert_arn string,
)
ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.RegexSerDe'
WITH SERDEPROPERTIES (
     'serialization.format' = '1',
     'input.regex' = '([^ ]*) ([^ ]*) ([^ ]*) ([^ ]*):([0-9]*) ([^ ]*)[:-]([0-9]*) ([-.0-9]*) ([-.0-9]*)
([-.0-9]*) (|[-0-9]*) (-|[-0-9]*) ([-0-9]*) ([-0-9]*) \"([^ ]*) ([^ ]*) (- |[^ ]*)\" \"([^\"]*)\"
([A-Z0-9-]+) ([A-Za-z0-9.-]*) ([^ ]*) \"([^\"]*)\" \"([^\"]*)\" \"([^\"]*)\" ([-.0-9]*) ([^ ]*)
\"([^\"]*)\" \"([^\"]*)\" \"([^ ]*)\" \"([^\s]+?)\" \"([^\s]+)\" \"([^ ]*)\" \"([^ ]*)\"')
LOCATION 's3://alblogbucket/AWSLogs/1234567890/elasticloadbalancing/us-east-1/';
```

Amazon Athena also supports Amazon Quick for interactive data visualization of your query results. Before you try to read files from S3 buckets, make sure that you grant Amazon Quick access to them. You also need to grant access to Amazon Athena to run queries.

To start visualizing the data, follow the steps below:

1) Go to the homepage of Amazon Quick and choose *Manage Data*.
2) Create a new data set.
3) For the data source, choose Athena and fill in additional details such as a data source name.
4) On the *Choose your table* screen, you can write your own SQL script or choose an existing database and table that you have created in Athena.
5) On the Finish data set creation page, choose how you want Amazon Quick to handle your data.
    a) You can load your data into memory with *Importing Data into SPICE*
    b) You can query your data directly without using SPICE. With this option, you rerun the query each time you open the analysis or dashboard.
6) Select *Visualize* to create the dataset and analyze your data.

References:
https://aws.amazon.com/products/storage/data-lake-storage/
https://docs.aws.amazon.com/whitepapers/latest/big-data-analytics-options/amazon-athena.html
https://aws.amazon.com/quick/

Using AWS Transfer Family for FTP Use Cases

Multiple industries rely on secure channels to transfer their data back and forth between different servers and storages in AWS. Some might also have compliance requirements to follow, which often requires strong encryption for data in-transit. Having to manage your own servers and secure channels to transfer sensitive files to AWS can add unnecessary overhead to your IT team. So instead, the AWS Transfer Family is a set of solutions that removes the operational overhead so your team can focus on your file transfers.

The AWS Transfer Family is the aggregated name of **AWS Transfer for SFTP (Secure Shell File Transfer Protocol), AWS Transfer for FTPS (File Transfer Protocol over SSL), and AWS Transfer for FTP**. The AWS Transfer Family offers fully managed support for the transfer of files over SFTP, FTPS, and FTP directly into and out of Amazon S3. You can seamlessly migrate your file transfer workflows by maintaining existing client-side configurations for authentication, access, and firewalls, so no changes need to be made for your customers, partners, and internal teams, or their applications. Data stored in Amazon S3 can be processed for multiple types of workloads, and can be moved around in your internal AWS network securely.

FTP	FTPS	SFTP
FTP is a network protocol used for the transfer of data. FTP uses a separate channel for control and data transfers. FTP uses cleartext and does not support encryption of traffic, which is also why the server does not allow you to use FTP over public networks. If traffic needs to traverse the public network, secure protocols such as SFTP or FTPS should be used.	FTPS is an extension of FTP that uses Transport Layer Security (TLS) and Secure Sockets Layer (SSL) cryptographic protocols to encrypt traffic. FTPS allows encryption of both the control and data channel connections either concurrently or independently.	SFTP is a network protocol used for secure transfer of data over the Internet. The protocol supports the full security and authentication functionality of SSH, such as the use of SSH keys.

The AWS Transfer Family provides you with a fully managed, highly available file transfer service with auto-scaling capabilities. Your end users' workflows remain the same, while data uploaded and downloaded over the different FTP protocols are stored in your S3 bucket. You set up your users by integrating an existing identity provider like Microsoft AD or LDAP for authentication. You should also assign IAM Roles to your users to provide access to your S3 buckets. A VPC is required to host FTP server endpoints.

Reference:
https://aws.amazon.com/aws-transfer-family/

A Single Interface for Querying Multiple Data Sources with AWS AppSync

Suppose you have multiple data sources that use different APIs to communicate with your applications. Your infrastructure might turn out to be very complex which can make it difficult for your users to fetch the data from all these sources, especially if they need it in real time. With AWS AppSync, you can simplify the process by having a single interface that users can interact with, and the interface will take care of fetching the data from multiple sources for you through the help of GraphQL technology.

For those who are unfamiliar with GraphQL, it is a data query and manipulation language that enables client apps to fetch, change, and subscribe to data from servers. The client specifies exactly what data it needs, and GraphQL aggregates the data from multiple sources and returns it to the client in JSON format. GraphQL also includes a feature called "*introspection*" which lets new developers on a project discover the data available without requiring knowledge of the backend.

The following are the features of AWS AppSync:

- Real-time data access and updates through subscriptions. When there are changes in the data, the results can be passed down to subscribed clients immediately using either MQTT over WebSockets or pure WebSockets.
- Offline data synchronization with Amplify DataStore that provides a queryable on-device datastore for web, mobile, and IoT developers.
- Data querying, filtering, and search in apps. AWS AppSync supports AWS Lambda, Amazon Aurora Serverless, Amazon DynamoDB, Amazon Elasticsearch, and HTTP endpoints as data sources.
- Server-side data caching capabilities. It reduces the need to directly access data sources all the time. Frequently accessed data are stored in high speed in-memory managed caches, and delivered at low latency.
- Several levels of data access management and authorization through AWS IAM Roles, integration with Amazon Cognito User Pools for email and password functionality, social identity providers (Facebook, Google+, and Login with Amazon), and enterprise federation with SAML.

Use Cases of AWS AppSync include:

- Create dashboards and web and mobile applications that need collective real-time data from multiple sources.
- Access and combine data from microservices running in containers in a VPC, behind a REST API endpoint, a GraphQL API endpoint, and more in a single interface in AppSync.
- Retrieve or modify data from multiple data sources (SQL, NoSQL, search data, REST endpoints, and serverless backends) with a single query.
- Automatically synchronize data between mobile/web apps and the cloud with AWS AppSync and AWS Amplify DataStore.

AWS AppSync SDKs support iOS, Android, and JavaScript, and span web frameworks such as React and Angular as well as React Native and Ionic. You can also use open source clients to connect to the AppSync GraphQL endpoint such as generic HTTP libraries or simple CURL commands.

Solutions Architect Professional Exam Tips:

If you look at it, AWS AppSync sounds awfully similar to Amazon API Gateway. While they do provide API functionalities for your applications, they differ in the kind of APIs provided. AppSync has GraphQL while API Gateway has RESTful and WebSocket APIs. There are also numerous other distinctions such as throttling features, integration features, request validation and custom response features, latency requirements, security features, etc. So do be careful in reading your exam scenario so you can discern what is the best solution for the item.

References:
https://aws.amazon.com/appsync/
https://aws.amazon.com/blogs/mobile/appsync-microservices/
Data Driven Applications with AWS AppSync and GraphQL

Domain 3: Continuous Improvement for Existing Solutions

Overview

The third domain of the AWS Certified Solutions Architect Professional SAP-C02 exam focuses on continuously improving your current cloud solutions by leveraging on new AWS services and features. Each month, AWS releases a set of updates on their services which are most of the time new features but can sometimes be freshly minted services themselves. By keeping yourself up-to-date with the latest technologies and best practices available, you can help you and your customers improve the existing cloud architectures that you manage.

You should know how to properly analyze and refactor certain components of your current cloud solutions to further improve your services. This entails the use of automation, deployment tools, data replication methods, scaling methodologies, logging & monitoring strategies, and other components in improving your existing stack.

This domain covers a quarter (25%) of the questions in the actual SAP-C02 exam and is the third largest domain. The topics for this domain are focused on these task statements.

- Determining a strategy to improve overall operational excellence
- Determining a strategy to improve security.
- Determining a strategy to improve performance
- Determining a strategy to improve reliability.
- Identifying opportunities for cost optimizations.

Using Amazon Cognito for Web App Authentication

Instead of building your own user management system for your websites and web applications, AWS offers a much simpler alternative with Amazon Cognito. Amazon Cognito is a service that allows you to add user sign-up, sign-in, and access control to your web and mobile apps. You can construct your user pool or use social identity providers to provide users a convenient method of signing up and logging in. Amazon Cognito also supports enterprise identity providers such as Microsoft Active Directory using SAML.

Amazon Cognito offers two types of pools for your business applications — **user pools** and **identity pools**. The main difference between the two is that user pools are used for authentication (identify verification) while identity pools are for authorization (access control). For authentication, Amazon Cognito uses multiple identity management standards including OpenID Connect, OAuth 2.0, and SAML 2.0.

Users Pools

With a user pool, your users can sign in through the user pool or federate through a third-party identity provider. It essentially acts as a directory. Use cases include:

- Be able to add sign-up and sign-in features for your app.

- Be able to access and manage user data.
- Be able to track user device, location, and IP address, and adapt to sign-in requests of different risk levels.
- Be able to use a custom authentication flow for your app.
- Be able to access resources with Amazon API Gateway and AWS Lambda.

Identity Pools

Identity pools provide tokens that can be exchanged for temporary AWS credentials in AWS STS after a successful authorization. The permissions for each user's credentials are controlled through IAM roles that you create. You can use identity pools to create unique identities for users and give them access to your AWS services. Use cases include:

- Giving your users access to AWS resources, such as an Amazon S3 bucket or an Amazon DynamoDB table.
- Generating temporary AWS credentials for unauthenticated users.

Users Pools + Identity Pools

There is no rule stating that you cannot use these two services together. An example of a use case is when you want to manage your users in Amazon Cognito and you would like to provide them temporary access to your AWS services. After a successful user pool authentication, the user's app will receive user pool tokens from Amazon Cognito. The user can then exchange them for temporary access to AWS services with an identity pool.

AWS AppSync (Newer service than Cognito Sync)

AWS AppSync is a service that lets you manage and synchronize mobile app data in real time across different devices and users, but still allows the data to be accessed and altered when the mobile device is offline. To tighten security around using AWS AppSync, you can grant your users access to AppSync resources with tokens from a successful Amazon Cognito authentication.

Scenario: Accessing Resources with Amazon API Gateway and AWS Lambda After Sign-in

You should make sure users accessing your API through Amazon API Gateway are authorized to do so. You can configure API Gateway to validate the tokens from a successful user pool authentication in Amazon Cognito, and use them to grant your users access to resources including Lambda functions, or your own API. Token verification is usually performed by an Amazon Cognito authorizer Lambda function.

References:
https://aws.amazon.com/cognito/
https://aws.amazon.com/premiumsupport/knowledge-center/cognito-user-pools-identity-pools/

https://aws.amazon.com/appsync/

Using AWS Systems Manager for Patch Management

We patch servers regularly during each of our maintenance periods to make sure that our operating systems are always kept up-to-date with the latest bug fixes and security fixes. This is especially crucial for production workloads since there are always new security vulnerabilities being discovered each day, and most, if not all of them, are too risky to simply leave unresolved. But of course, patching activity also presents its own set of challenges, particularly when there are multiple servers involved and each have a different patch baseline. Manually connecting to your instances and running Windows Update or *sudo yum update* is not feasible, so you will need to automate your patching activities. The second challenge here is coordinating your patching window with your server availability. There are many approaches to solving these challenges, but for this section we will be focusing on using AWS Systems Manager.

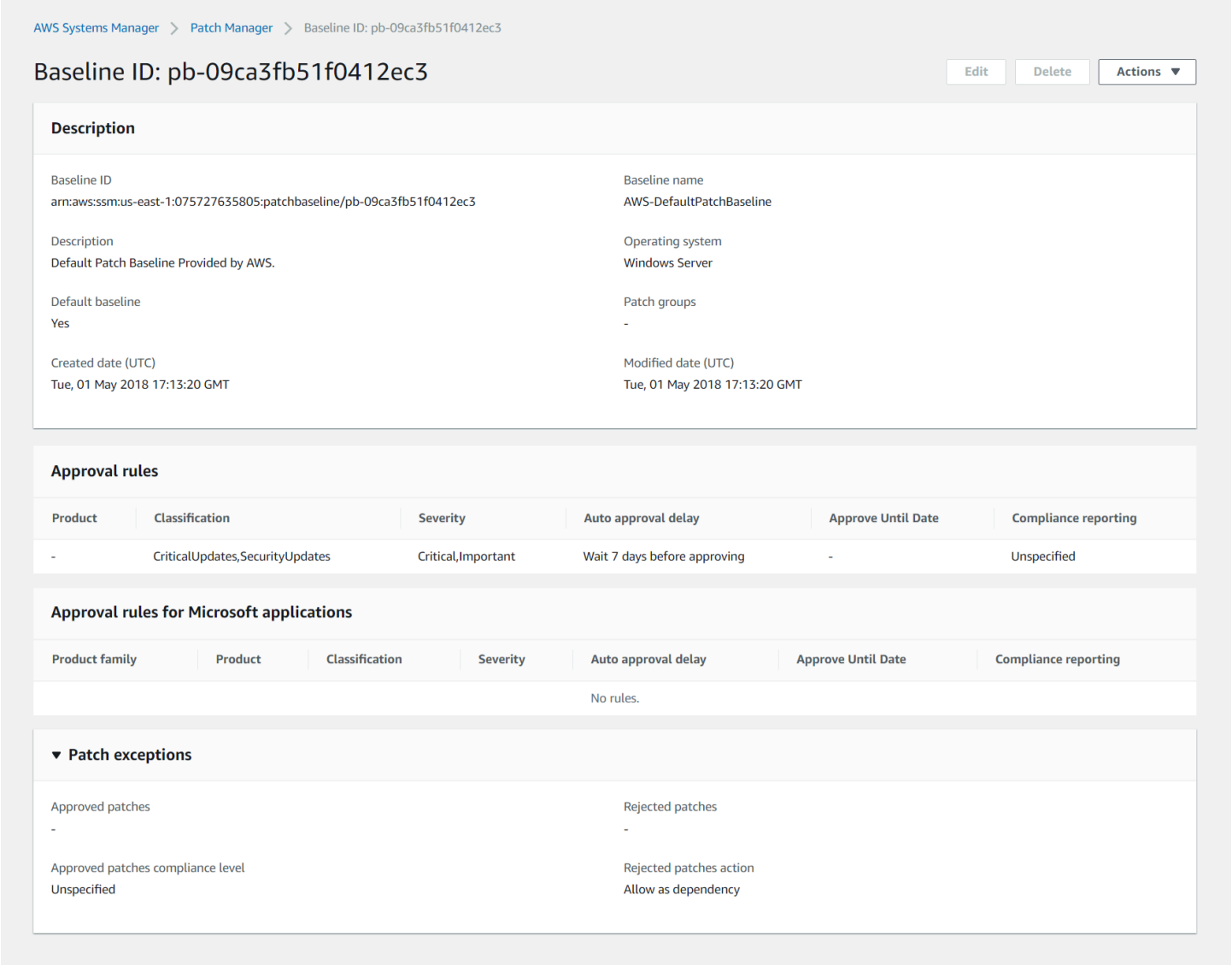
Note: Since you are letting AWS Systems Manager handle your instances, you will need to assign the appropriate IAM role to your instances that would grant permissions for AWS Systems Manager to perform patching and other related tasks. You also need to make sure that your instances have SSM Agent installed and the agents are able to communicate back with AWS Systems Manager. One way to verify this is to go to AWS Systems Manager Managed Instances and check if your desired instances are in the list.

There are two key services under AWS Systems Manager that we will use to build our fully-automated patching solution, namely:

1) Patch Manager
2) Maintenance Windows

Patch Manager automates the process of patching managed instances with both security related and other types of updates. You can use Patch Manager to apply patches for both operating systems and applications. You can patch fleets of EC2 instances or your on-premises servers and virtual machines (VMs) by the type of operating system. You can also scan instances to see only a report of missing patches, or you can scan and automatically install all missing patches.

Patch Manager uses **patch baselines** to let you specify which patches to install and apply a delay after patches are released before auto-approving them. AWS already provides you a set of preconfigured patch baselines for different OS types, but you can also configure your own if you wish.

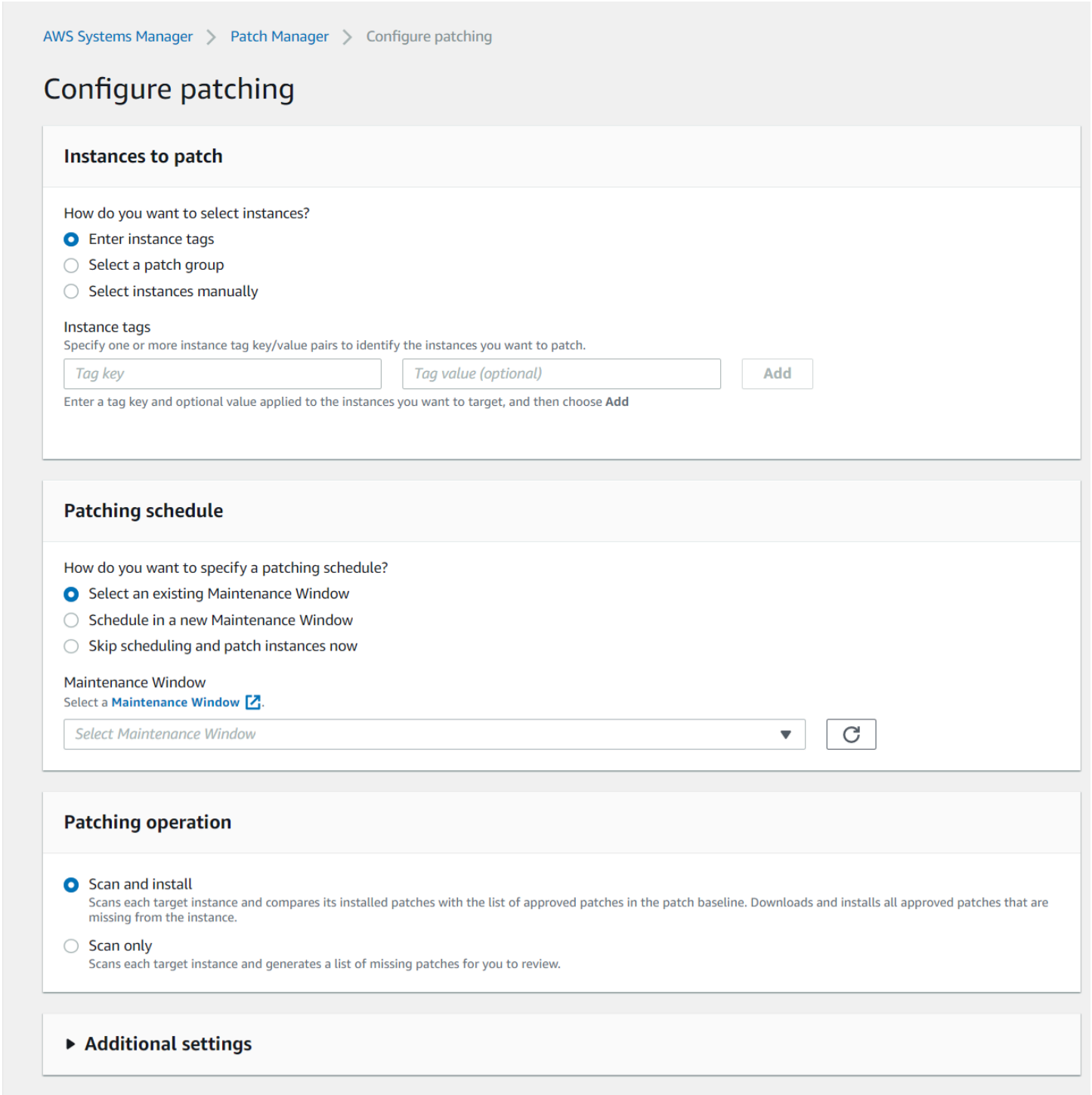


Maintenance Windows, on the other hand, lets you define a **schedule** for when to perform potentially disruptive actions on your instances such as patching an operating system, updating drivers, or installing software or patches. Each maintenance window has a schedule, a maximum duration, a set of registered targets (the instances or other AWS resources that are acted upon), and a set of registered tasks.

In summary, a maintenance window for patching works like this:

1. You create a maintenance window with a schedule for your patching operations.
2. Then choose the targets for the maintenance window by specifying either a **Patch Group** tag or your own tag key, or by choosing the instances manually.
3. Finally, create a new maintenance window task, and specify the **AWS-RunPatchBaseline** document or any other document you would like to use for your patching operation.

These steps can also be done conveniently in the Patch Manager Window.



When creating a schedule for your maintenance window, you have two ways to build it.

a) Scheduling via cron expression - With cron, you can schedule your window at a specific rate (for example, every 30 minutes or every hour) or at a specific day and time (for example, run everyday at 12 am or run every Sunday at 1 am). You can also specify at what date should your tasks start executing and until when. Lastly, you can change your preferred time zone in case you follow a different one.
b) Scheduling via rate expression - With rate, your available options are similar to cron except that you cannot specify at which day and time you want your tasks to execute. As the name implies, you can only specify rates (for example, every 30 minutes, every 1 hour, or everyday)

ℹ️ **Solutions Architect Professional Exam Notes:**

So what should I look out for before choosing these two services for an item in the exam?

The answer is convenience. Remember that as a Solutions Architect, you are supposed to build solutions that would make your business operations more convenient and simple. Patch Manager and Maintenance Windows help you out in this area since they are less disruptive than other options, while making sure the job gets done. You are offered a lot of flexibility in the configurations for the scheduling and the patch baselines. And since this is automation, your solution can be reused multiple times and the results will always be predictable.

Some other options that might throw you off include Amazon EventBridge for the scheduling or Systems Manager State Manager for patch compliance. If you have a good understanding of these two services then you should also know why they aren't the best choices. Scripting your own cron scheduler and patching automation are also ruled out since they are not the most convenient to do and maintain.

References:
https://docs.aws.amazon.com/systems-manager/latest/userguide/sysman-patch-mw-console.html
Patching Windows Servers using AWS Systems Manager

Implementing CI/CD using AWS CodeDeploy, AWS CodeBuild, and AWS CodePipeline

It is highly likely that in your exam, you will also encounter scenarios involving CI/CD. What you need to learn for these kinds of scenarios are the CI/CD tools in AWS and how to build your own pipelines using them. To start with, we will first briefly define each AWS CI/CD tool and in what situations will you use them for.

1) **AWS CodeDeploy**

AWS CodeDeploy is a fully managed deployment service that automates software deployments to Amazon EC2, AWS Fargate, AWS Lambda, and your on-premises servers. The steps for initiating a deployment involves:

- Choosing your compute platform for your deployment
- Creating a deployment group
- Providing CodeDeploy with the necessary permissions via service role
- Specifying the targets for that compute platform
- Configuring deployment settings such as choosing your deployment strategy, setting up alarms and notifications, creating deployment triggers, defining rollback settings and adding tags.

AWS CodeDeploy offers multiple deployment strategies for each compute platform. For Amazon EC2 instances and on-premises servers, you can select if you would like to use **in-place** or **blue-green** deployment. An in-place deployment updates your instances/servers right as they are. The application on each instance in the deployment group is stopped, the latest application revision is installed, and the new version of the application is started and validated. These updates can be deployed to your instances/servers one at a time, in batches, or all at once. In-place deployments can be used if your applications have redundant copies to which traffic can failover to, or if the applications being updated are not critical to the overall availability of your system.

Blue-green deployment is a better strategy for workloads that cannot tolerate interruptions, such as an active website. Furthermore, transitioning to the new application version is more gradual and controlled, and rolling back to a previous working version is easier and quicker. In a blue-green deployment, the instances in a deployment group are replaced by a different set of instances containing your updates. New instances will be registered to your load balancer so that it can start accepting traffic, while old instances are deregistered. Deregistered instances can be kept alive for rollback scenarios, or can be terminated immediately. Similarly, these updates can be deployed to your instances/servers one at a time, in batches, or all at once.

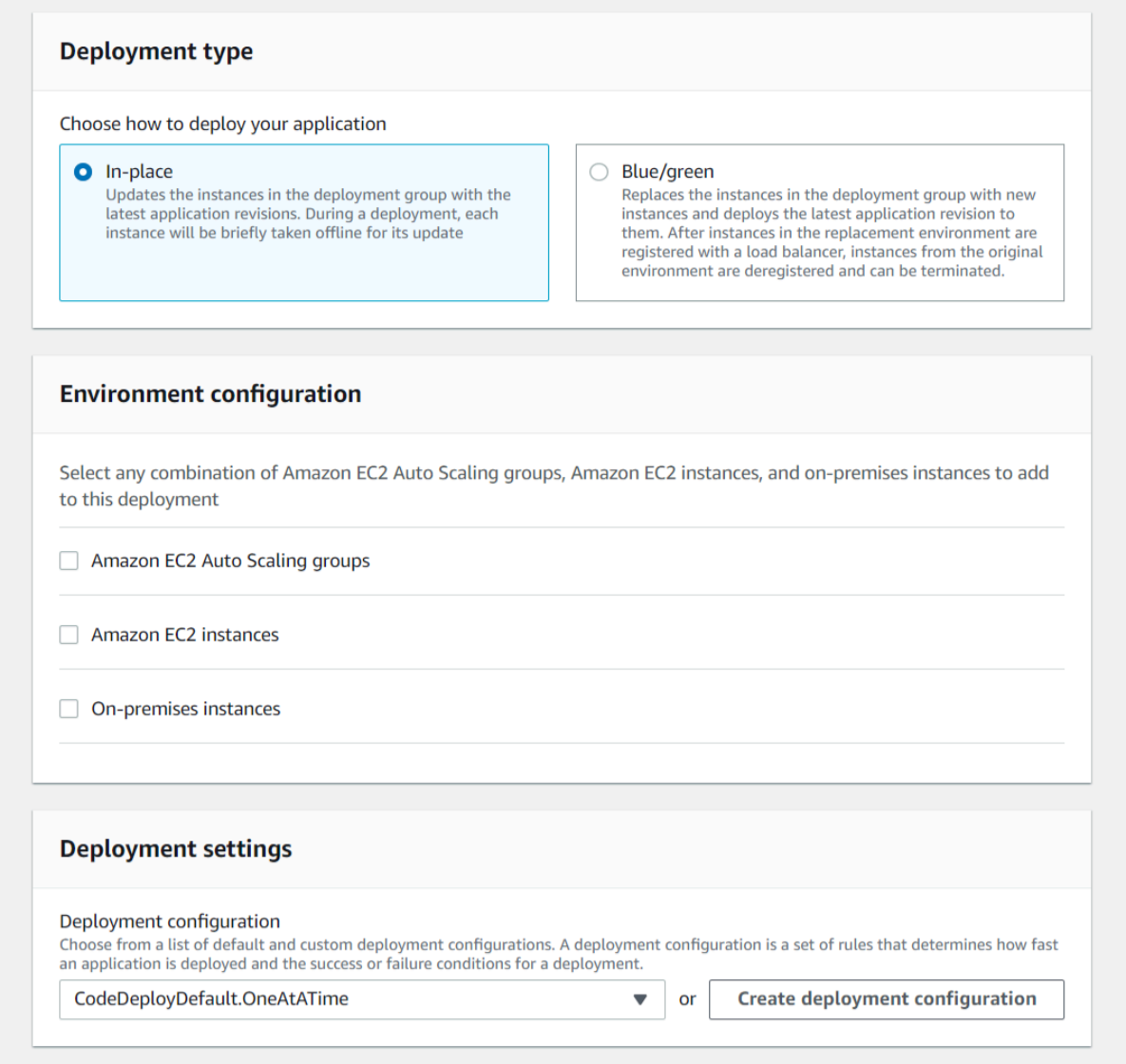
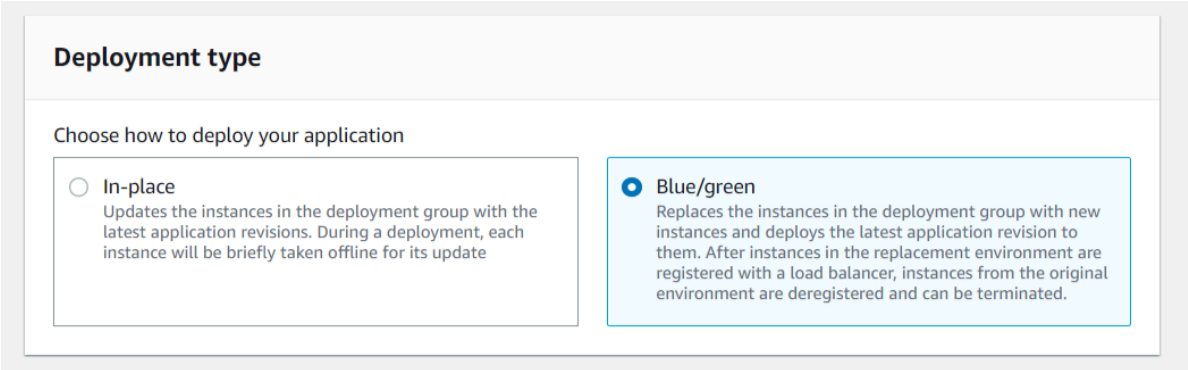


Figure: In Place Deployment in AWS CodeDeploy



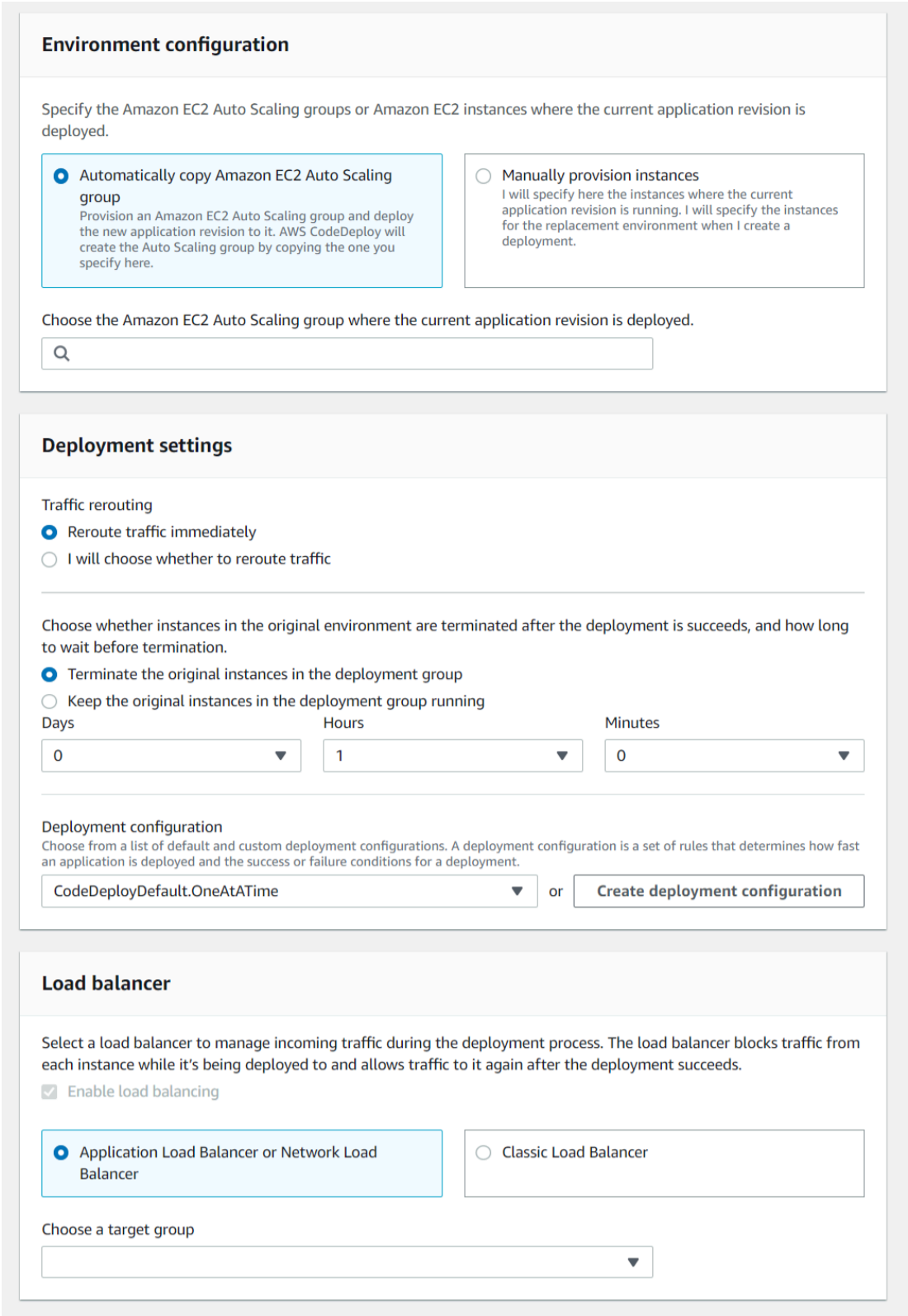


Figure: Blue Green Deployment in AWS CodeDeploy

For Lambda and ECS/Fargate platforms, you only have blue-green deployment as your available deployment strategy. You can have your updates deployed all at once, gradually in a linear fashion (meaning a percentage of the traffic is shifted from the old to the new at a rate until all traffic is shifted), or in a two-step order using canary (meaning a small percentage of traffic is given to the new, and the rest of the traffic is shifted to the new environment after a specified wait time)

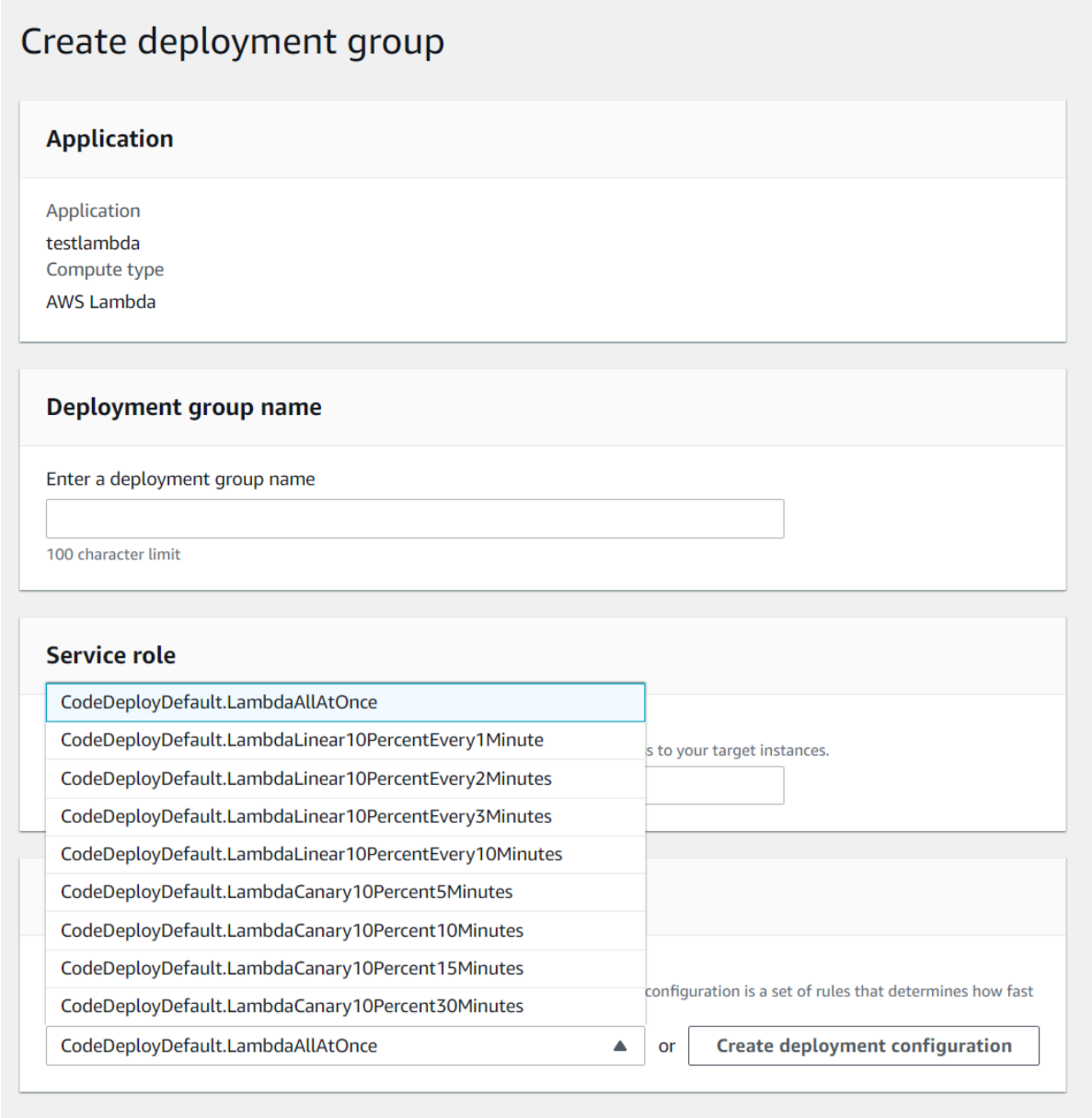
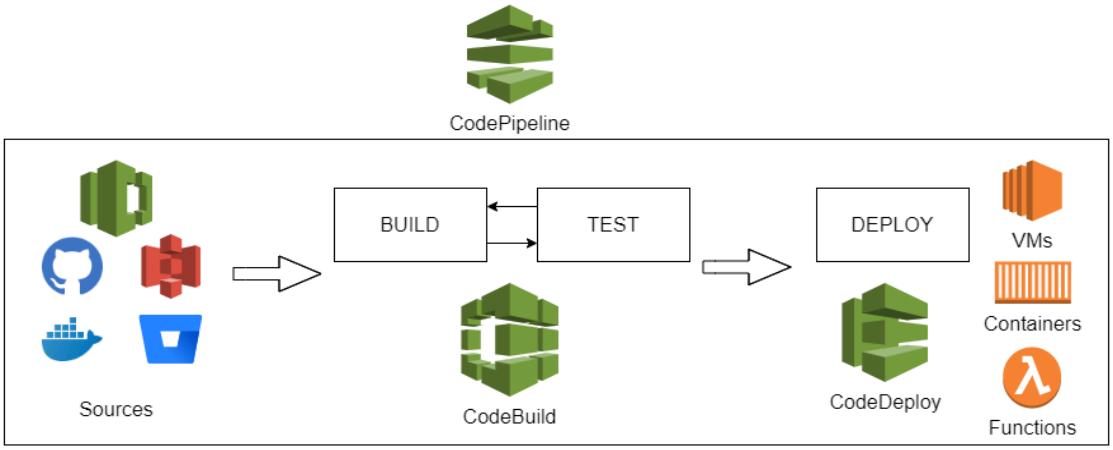


Figure: Lambda Deployment in AWS CodeDeploy

Reference:
https://aws.amazon.com/codedeploy/

2) **AWS CodeBuild**

AWS CodeBuild is a continuous integration service that **compiles source code, runs tests, and produces software packages** that are ready to deploy. AWS CodeBuild does not retrieve source code from your repositories, nor does it deploy packages to your machines.



AWS CodeBuild is often used together with a pipeline, either with AWS CodePipeline or third party software such as Jenkins.

To use CodeBuild, you first create a project. In it, you specify the source of your builds, the environment and operating system you want your code to compile and get tested in, an IAM service role to allow CodeBuild to run, a *buildspec* YAML file to define how to compile and test your code, and lastly, an S3 bucket to store your *artifact* or "final product". You can also store CodeBuild logs in Cloudwatch logs or in S3. Specifying an S3 bucket for artifacts is not required if you are only conducting testing or if you are compiling a Docker image, which of course should be uploaded to a Docker repository.

References:
https://aws.amazon.com/codebuild
Setting up CI/CD for containers

3) **AWS CodePipeline**

AWS CodePipeline is a continuous delivery service that helps you automate the build, test, and deploy phases of your release process every time there is a code change. To understand AWS CodePipeline better, we will need to define a few terminologies:

- A *pipeline* is a workflow construct that describes how software changes go through a release process. You define the workflow with a sequence of stages and actions.
- A *stage* is a group of one or more actions. A pipeline can have two or more stages.
- An *action* is a task performed on a revision. Pipeline actions occur in a specified order, in serial or in parallel, as determined in the configuration of the stage.
- A *revision* is a change made to the source location defined for your pipeline. It can include source code, build output, configuration, or data.
- The stages in a pipeline are connected by *transitions*. Revisions that successfully complete the actions in a stage will be automatically sent on to the next stage as indicated by the transition.
- When an action runs, it acts upon a file or set of files called *artifacts*. These artifacts can be worked upon by later actions in the pipeline.

Reference:
https://aws.amazon.com/codepipeline/
https://aws.amazon.com/blogs/devops/build-a-continuous-delivery-pipeline-for-your-container-images-with-amazon-ecr-as-source/

4) **Amazon ECS / AWS Fargate**

Amazon ECS and AWS Fargate are both compute services for containers. It is common for containers to be used in a CI/CD deployment. In the exam, what you have to know is that an Amazon ECS and AWS Fargate deployment requires four (4) components:

- An ECR Repository where you will store versioned container images (source of your pipeline).
- An ECS Cluster which will be your cluster of container instances. This will include a load balancer and auto scaling configurations.
- ECS Task Definition which specifies your container image and environment configurations.
- ECS Service which specifies how your task definition will be deployed onto underlying compute resources.

There are also two IAM roles that you need to distinguish: The ECS **task execution role** grants the Amazon ECS container and Fargate agents permission to make AWS API calls on your behalf. The ECS **task role** grants applications running in your containers permission to make AWS API calls.

References:
https://aws.amazon.com/ecs/
https://aws.amazon.com/fargate/

5) **AWS X-Ray**

AWS X-Ray provides an end-to-end view of requests as they travel through your application, and shows

a map of your application's underlying components. With X-Ray, you can understand how your application and its underlying services are performing to identify and troubleshoot the root cause of performance issues and errors. In the exam, if the scenario presents multiple components in an application or a complex microservice architecture (e.g. APIs, functions, containers, etc), use AWS X-Ray to pinpoint and debug HTTP errors.

You can get started with X-Ray by including the X-Ray language SDK in your application and installing the X-Ray agent. X-Ray can be used with distributed applications of any size to trace and debug both synchronous requests and asynchronous events. You can use X-Ray with applications running on EC2, ECS, Lambda, Amazon SQS, Amazon SNS, and Elastic Beanstalk.

Reference:
https://aws.amazon.com/xray/

Using Federation to Manage Access

When you are in an organization with multiple users and multiple accounts, one way to provide your users access to AWS in a secure and centrally manageable manner is through federation. You can use two AWS services to federate into AWS: AWS IAM Identity Center (formerly AWS SSO) and AWS IAM. Use the AWS IAM Identity Center to help you define federated access permissions for your users based on their group memberships in a single centralized directory. If you use multiple directories, or want to manage the permissions based on user attributes, use AWS IAM instead.

AWS IAM Identity Center (formerly AWS SSO)

AWS IAM Identity Center works with identity provider (IdP) services such as Okta Universal Directory or Azure Active Directory via the SAML 2.0. You can add any AWS account managed using AWS Organizations to AWS IAM Identity Center, but you need to enable all features in your organizations first. The AWS IAM Identity Center service leverages IAM permissions and policies for federated users and roles to help you manage federated access centrally across all AWS accounts in your AWS Organization. With AWS IAM Identity Center, you can assign permissions based on the group membership in your IdP's directory, and then control the access for your users by simply modifying users and groups in the IdP.

You can also control who can have access to your cloud applications as it can securely communicates with your applications through a trusted relationship between the AWS IAM Identity Center and the application's service provider. This trust is created when you add the application from the AWS IAM Identity Center console and configure it with the appropriate metadata for both the IAM Identity Center and the service provider. Take note that this service supports only SAML 2.0–based applications, so using OIDC-based applications will not work.

AWS IAM

AWS IAM allows you to enable a separate SAML 2.0 or an Open ID Connect (OIDC) IdP for each AWS account you manage and use federated user attributes for access control. Instead of creating IAM users, you can use IAM identity providers to manage your user identities outside of AWS and give these external user identities permissions to use AWS resources in your account. This is useful if your organization already has its own identity system, such as a corporate user directory. It is also useful if you are creating a mobile app or web application that requires access to AWS resources.

To use an IdP, you create an IAM identity provider entity to establish a trust relationship between your AWS account and the IdP.

- With **web identity federation**, you don't need to create custom sign-in code or manage your own user identities. Instead, users of your app can sign in using an OpenID Connect (OIDC)-compatible IdP. They will receive an authentication token, and then your application calls the `AssumeRoleWithWebIdentity` API to exchange that token for temporary security credentials in AWS. The credential is mapped to an IAM role with permissions to use the resources in your AWS account. For convenience, use Amazon Cognito as your identity broker for almost all web identity federation scenarios.
- With **SAML 2.0-based federation**, a user in your organization requests authentication from your organization's IdP through an app. The IdP authenticates the user against your organization's identity store. Then the IdP constructs a SAML assertion with information about the user and sends the assertion to the app. The app calls the AWS STS `AssumeRoleWithSAML` API, passing the ARN of the SAML provider, the ARN of the role to assume, and the SAML assertion from IdP. The API response to the app includes temporary security credentials, which the user can use to access your AWS resources.

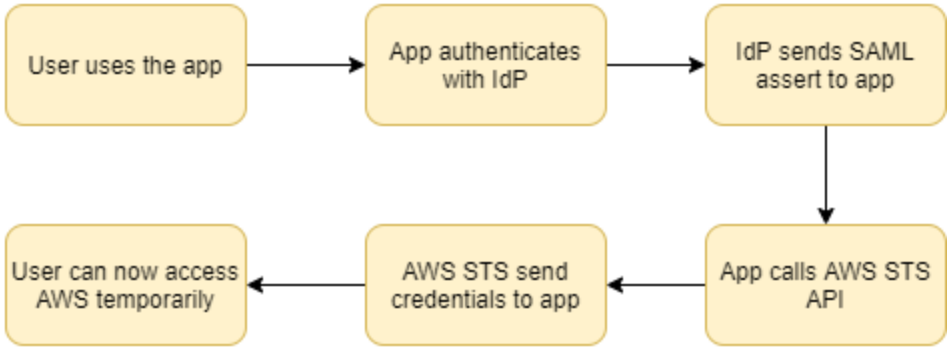


Figure: AWS IAM SAML federation

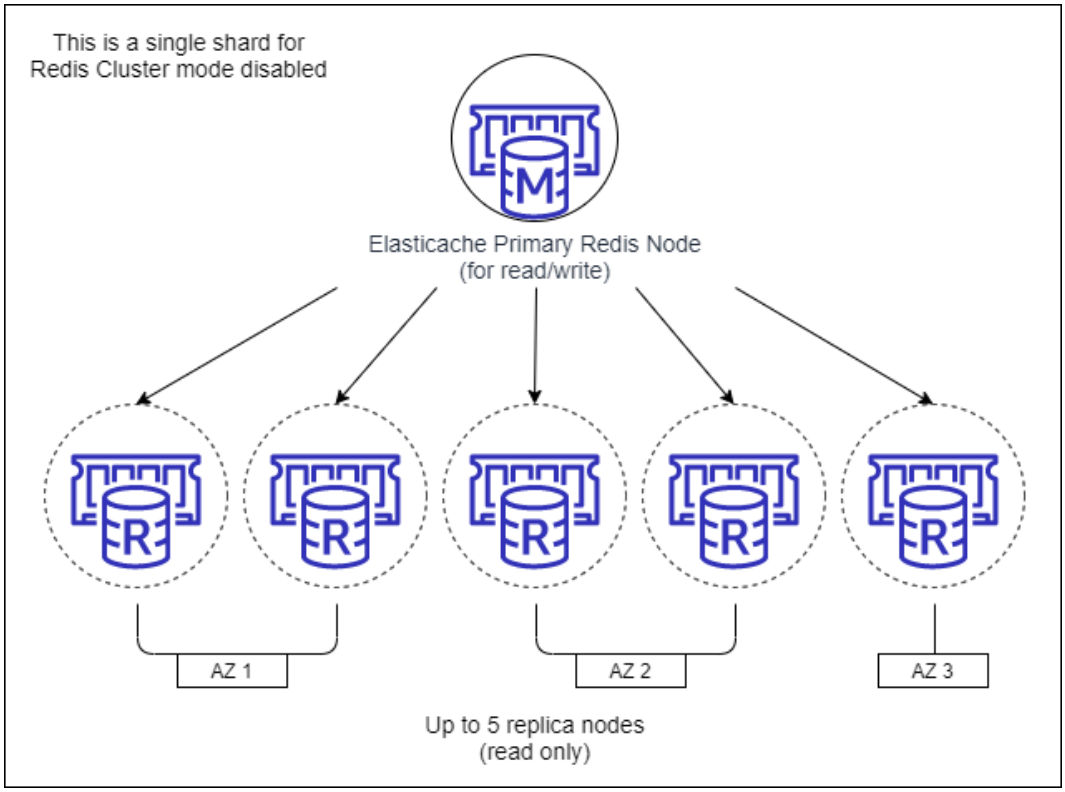
References:
https://aws.amazon.com/identity/federation/
https://docs.aws.amazon.com/singlesignon/latest/userguide/what-is.html
https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles_providers.html

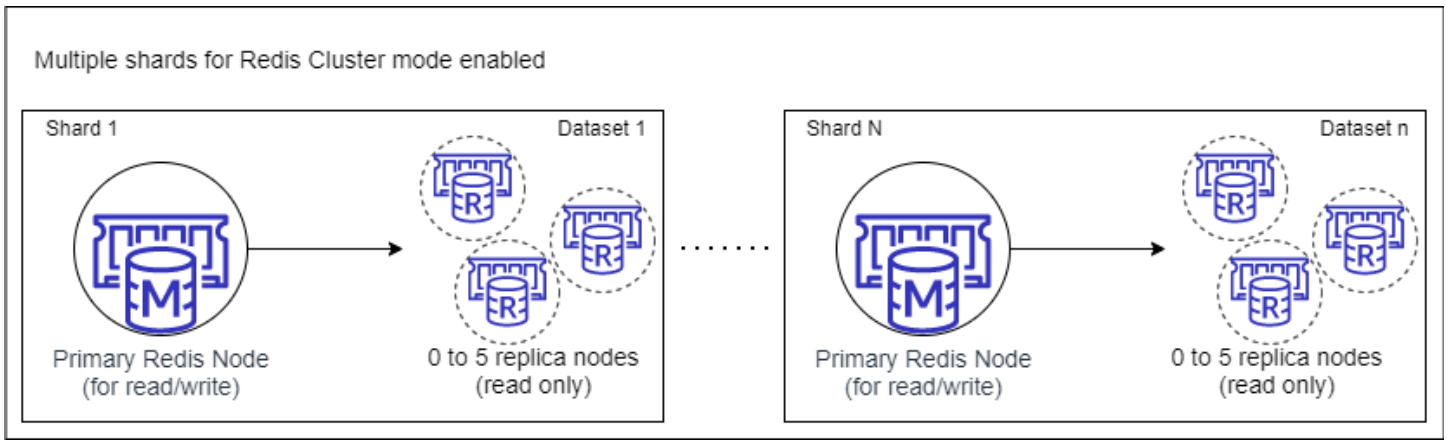
Setting Up a Fault Tolerant Cache Layer with Amazon Elasticache

When adding a caching layer to your infrastructure, similar to databases, you should also make sure that your cache is highly available to avoid any issues. If you only deploy single nodes for example, a sudden outage or reboot on this node can incur large amounts of data loss. Your applications will also take a hit since performance will be greatly impacted, and unless you have configured a fault-tolerant system, might even cause downtime. To protect your cache from a total outage, you can configure **Replication Groups** or set up **Append Only Files.**

Redis Replication Groups

In Amazon Elasticache for Redis, you can have 2 to 6 nodes in a cluster where 1 - 5 nodes are read-only replicas. In this scenario, if one node were to fail, you do not lose all your data because of the replication. But since the replication mode is asynchronous, some data may be lost if it is the primary read/write node that fails. Recall that for Redis cluster mode disabled, clusters always have one shard. On the other hand, Redis cluster mode enabled clusters can have up to 90 shards. Redis cluster mode enabled provides you the flexibility to create a cluster with your desired number of shards and number of replicas with up to 5 replicas (as long as the total is 90 nodes). If the cluster with replicas has Multi-AZ enabled and the primary node fails, the primary fails over to a read replica. **Multi-AZ is required for all Redis (cluster mode enabled) clusters**!





When a failover does occur, ElastiCache also propagates the DNS name of the promoted replica. No endpoint change is required in your application if it is using the primary endpoint. Applications using individual endpoints need to change the read endpoint of the replica promoted to primary to the new replica's endpoint.

Append Only Files

If you cannot use replication groups due to some constraint, but still need data durability, you may use Redis append-only file feature (AOF). When this feature is enabled, your Elasticache Redis node writes all of the commands that change cache data to an append-only file. If the node is rebooted, the AOF is "replayed", much like a database recovery process. This ensures that your cache data remains intact.

To enable AOF for a cluster running Redis, you must create a parameter group with the *appendonly* parameter set to yes. You then assign that parameter group to your cluster. You can also modify the *appendfsync* parameter to control how often Redis writes to the AOF file.

Between the two fault tolerance solutions, the better option to implement would be the replication group with multi-AZ.

References:
https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/Replication.html
https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/RedisAOF.html

Improving the Cache Hit Ratio of your CloudFront Distribution

One of the main purposes of using CloudFront is to reduce the number of requests that your origin server must respond to directly. CloudFront caching allows you to serve objects from CloudFront edge locations, which are closer to your users. This effectively reduces the load on your origin server and reduces latency. If you notice that your CloudFront distribution is not doing a good job caching your objects, and that your origin server is responding too frequently, you can optimize your cache settings to encompass a larger subset of cacheable objects.

The proportion of requests that are served from caches to all requests is called the *cache hit ratio*. There are a number of changes you can do to improve your cache hit ratio.

1) Increase the duration that your objects stay cached in CloudFront edge locations. You can configure your origin to add a Cache-Control max-age header to your objects, and specify the longest practical value for max-age. The shorter the cache duration, the more frequently CloudFront checks if the object has changed to get the latest version.
2) Configure CloudFront to forward only the query string parameters for which your origin will return unique objects.
3) Configure CloudFront to forward only specific cookies instead of all cookies to your origin. Create separate cache behaviors for static and dynamic content, and forward cookies to your origin only for dynamic content.
4) Configure CloudFront to forward and cache objects based on specific headers only instead of forwarding and caching objects based on all headers.
5) Remove Accept-Encoding Header when compression is not needed. When you use this configuration, CloudFront removes the header from the cache key and doesn't include the header in origin requests.
    - Header name: Accept-Encoding
    - Header value: (Keep blank)

To check if any of these changes has helped you improve your cache hit ratio, you may visit the CloudFront Cache Statistics Reports page and review the metrics.

References:
https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/ConfiguringCaching.html
https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/cache-statistics.html

Other Ways of Combining Route 53 Records for High Availability and Fault Tolerance

To build a fully highly-available and fault tolerant infrastructure, it's not only your EC2 instances and RDS databases that you should worry about. You also need to make sure that your users are properly routed to a working origin in case of a failover with least amount (and if possible, zero) downtime. Protection from a single point of failure typically includes having health checks continuously monitoring your endpoints, and distributing your endpoints in different locations. In AWS, you get less headaches if you don't place all your eggs in one basket.

Though Route 53 has made failover routing possible with active-active and active-passive failover solutions, it is not always the case that failover records are the best to use for your environment. For example, you might want to route your users to the servers closest to them, or serve specific content based on where your customers are. In these scenarios, you might consider other, more beneficial routing policies.

Route 53 latency and weighted records

If your web application is running on EC2 instances in more than one Region, and if you have more than one instance running in one or more of these Regions, you can use latency-based routing to route traffic to the correct region and then use weighted records to route traffic to instances within the region based on weights that you specify. To use latency and weighted records in Amazon Route 53 together:

1) Create a group of weighted records for your EC2 instances in each region.
    a) Give each weighted record the same value for *Record Name* and *Record Type*.
    b) For *Value/Route traffic to*, choose IP address or another value depending on the record type, and specify the value of one of the EC2 IP addresses.
    c) If you want the EC2 instances to weigh equally, specify the same value for *Weight*.
    d) Specify a unique value for *Set ID* for each record.
    e) Associate a Route 53 health check for your instances under this weighted record.
2) If you have multiple EC2 instances in other regions, repeat Step 1 for the other regions, but specify a different value for *Name* in each region.
3) For each region in which you have multiple EC2 instances, create a latency alias record. For *Value/Route traffic to*, choose *Alias to another record in this hosted zone,* and specify the value of the *Record Name* of your weighted records in that region. Set the value of *Evaluate Target Health* to Yes.
4) For each region in which you have a single EC2 instance, create a latency record. For *Record Name*, specify the same value that you specified for the latency alias records created in Step 3. For *Value/Route traffic to*, choose *IP address or another value depending on the record type*, and specify the IP address of your EC2 instance in the Region.

Latency alias records allow you to utilize different regions that are close to your users for low latency performance. Together with weighted records, you ensure that your applications are protected from the failure of a single endpoint or an Availability Zone. Lastly, enabling *Evaluate Target Health* in your latency alias records lets Route 53 determine whether there are any healthy resources in a region before trying to route traffic there. If there are none, Route 53 chooses a healthy resource in the other region where you also have a latency alias record set up.

Route 53 weighted multi-record answers

A Route 53 weighted record can only be associated with one record, meaning a combination of one name and one record type. But it is often desirable to define weights for DNS responses that contain multiple records. In the event of an endpoint failure, providing multiple IP addresses in DNS responses provides users with alternative endpoints. You can even protect against the failure of an availability zone if you configure responses to contain a mix of IPs hosted in two or more availability zones.

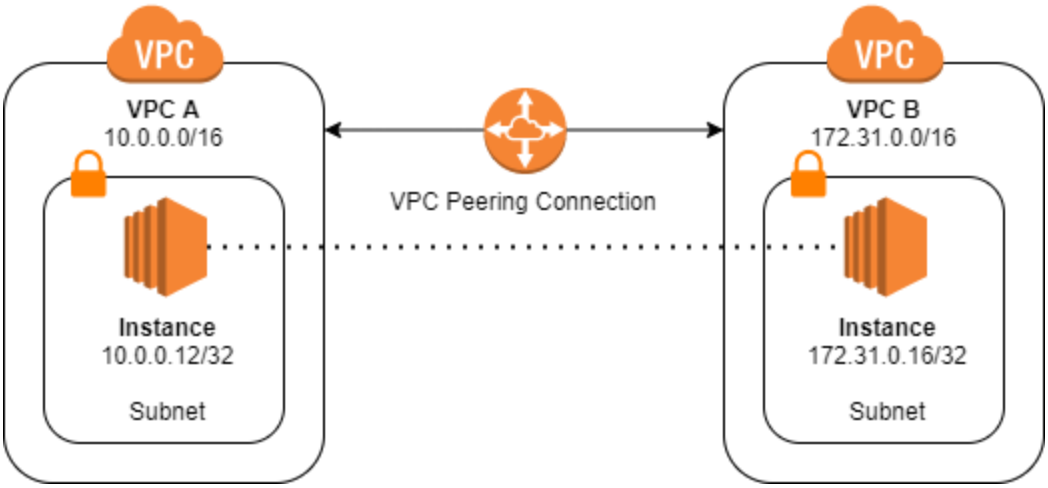
These types of weighted multi-record answers can be achieved by using a combination of records and weighted alias records. You can group multiple endpoints into distinct record sets with each containing a subset of the IP addresses. You can then create a weighted alias record that points to each group and assign a corresponding weight. Weighted multi-record answer routing is different from Route 53 multi-value answer routing.

References:
https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/TutorialLBRMultipleEC2InRegion.html
https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/TutorialWeightedFTMR.html
https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/dns-failover-complex-configs.html

Longest Prefix Match: Understanding Advanced Concepts in VPC Peering

VPC Peering Basics

In AWS, a Virtual Private Cloud (VPC) peering connection is a networking connection between two VPCs which allows you to route specific traffic between them using either private IPv4 addresses or IPv6 addresses.



A VPC peering connection can be created between your own VPCs, or alternatively, a VPC in another AWS account. You can also create an inter-region VPC peering connection where the VPCs are located in different AWS Regions. Amazon EC2 Instances in either VPC can communicate with each other freely as if they are within the same network.

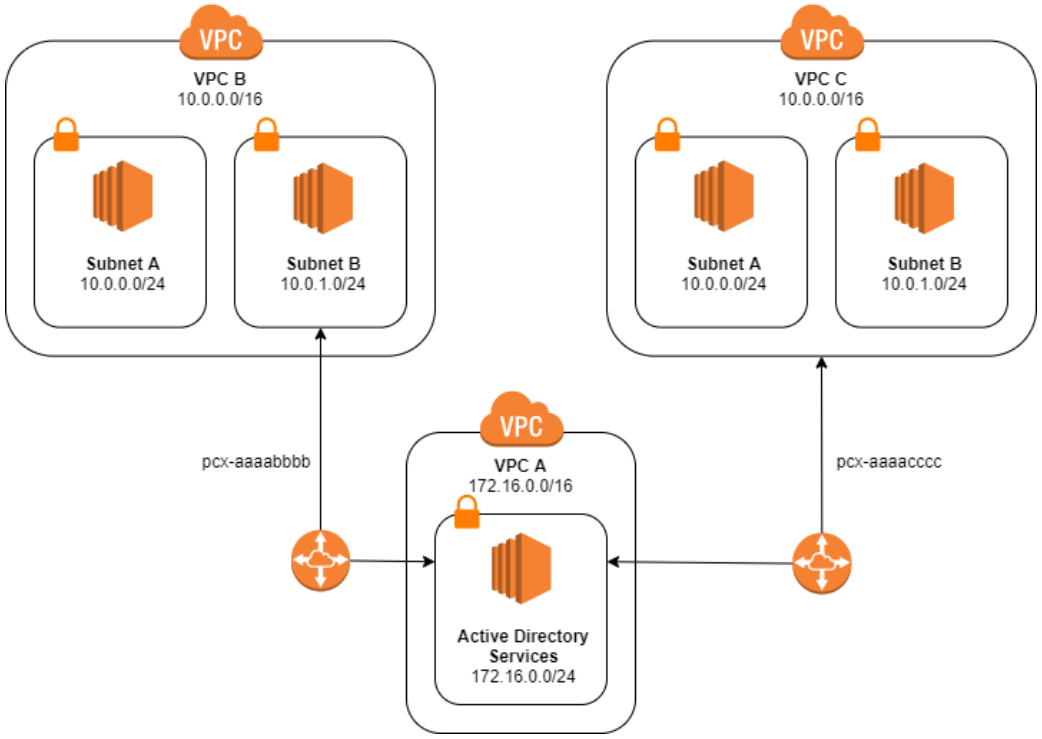
One example of VPC Peering is the integration of third-party services to your AWS account. Say you want to utilize a cloud database service from an external provider, like MongoDB Atlas which provides AWS, GCP, and Azure-backed clusters. In order for your EC2 instances to communicate with your external MongoDB cluster, you need to establish a VPC Peering connection to the MongoDB Atlas VPC first.

There are various VPC Peering setups that you can put up such as configurations with ClassicLink, configurations with routes to an entire CIDR block (VPCs don't have subnets) and lastly, configurations with **specific** routes (VPCs have two or more subnets). This article will focus on the latter type in which your VPC has peered with 2 VPCs and configured with specific routes on their route table, leveraging on the *longest prefix match* algorithm.

Longest Prefix Match — what is it?

As an overview, let say that you have a central VPC (*labeled as VPC A below)* with one subnet. This has a VPC peering connection between VPC A and VPC B (pcx-aaaabbbb) and also between VPC A and VPC C (pcx-aaaacccc).

The key point of this scenario is: Both VPC B and VPC C have **matching** CIDR blocks of 10.0.0.0/16 as shown in the diagram below:



At this point, if there is incoming traffic from VPC A which is intended to 10.0.0.66, which VPC will it go to? Will it be VPC B or VPC C, considering that these two VPCs have exactly the same prefix (CIDR block)? What's your guess?

This scenario is an advanced topic regarding VPC Peering where you have one VPC Peered with 2 VPCs in which it uses *Longest Prefix Match* for directing traffic based on your route table configuration. Let's go over the basics first in order for us to better understand this scenario:

The term "longest prefix match" is basically an algorithm used by routers in Internet Protocol (IP) networking used for choosing an entry from a forwarding route table. It is possible that each entry in a forwarding table may specify a **sub-network** in which one destination address may match more than one forwarding table entry. In this case, 10.0.0.0/24 is a sub-network of 10.0.0.0/16 CIDR block. At this point, the "**longest** prefix" actually refers to the one with the **longest** subnet mask which is the most specific of the matching table entries where the traffic will be forwarded.

Route Table	Destination	Target
VPC A	172.16.0.0/16	Local
	10.0.0.0/24	pcx-aaaabbbb
	10.0.1.0/24	pcx-aaaacccc
VPC B	10.0.0.0/16	Local
	172.16.0.0/16	pcx-aaaabbbb
VPC C	10.0.0.0/16	Local
	172.16.0.0/16	pcx-aaaacccc

Now that we get the context of "Longest Prefix Match", we can now better understand how this works. Each VPC has a CIDR Block of 10.0.0.0/16 with two subnets: 10.0.0.0/24 (Subnet A) and 10.0.1.0/24 (Subnet C).

Adding an entry of **10.0.0.0/24** to pcx-aaaabbbb on your route table is the actual implementation of the prefix match we discussed earlier. Since 10.0.0.0/24 is a sub-network (Subnet A) of 10.0.0.0/16, we can better control the flow of traffic. The CIDR block of 10.0.0.0/24 has a total of 256 IP addresses with a range starting from 10.0.0.0 to 10.0.0.255.

The same goes for Subnet B which has a CIDR block of 10.0.1.0/24. Since its prefix is the same with Subnet A, it also has a total of 256 IP addresses with a range starting from 10.0.**1**.0 to 10.0.**1**.255 address.

Remember that the IP address in the question is 10.0.0.66 which is within the range of the 10.0.0.0/24 sub-network (Subnet A). Since we have a specific entry to pcx-aaaabbbb for this, the router's behavior will forward the traffic to VPC B.

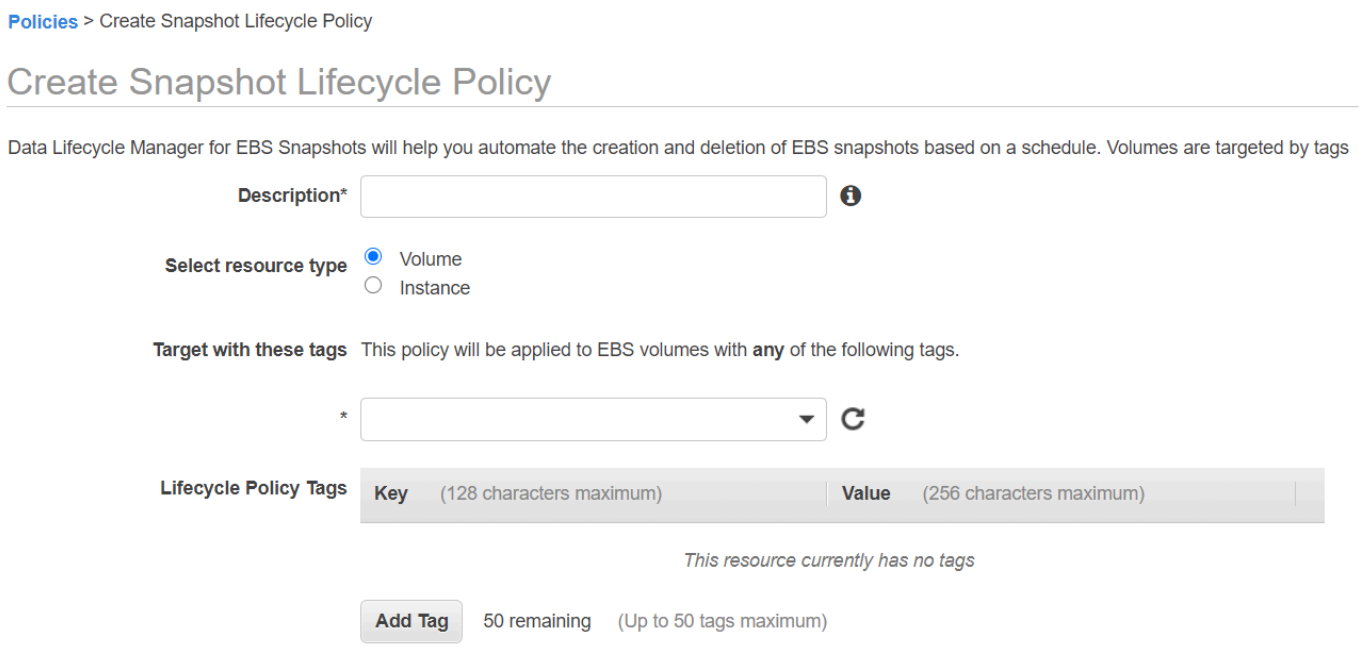
Reference:
https://docs.aws.amazon.com/vpc/latest/peering/peering-configurations-partial-access.html#one-to-two-vpcs-lpm

Automate your EBS Snapshots using Amazon Data Lifecycle Manager (Amazon DLM)

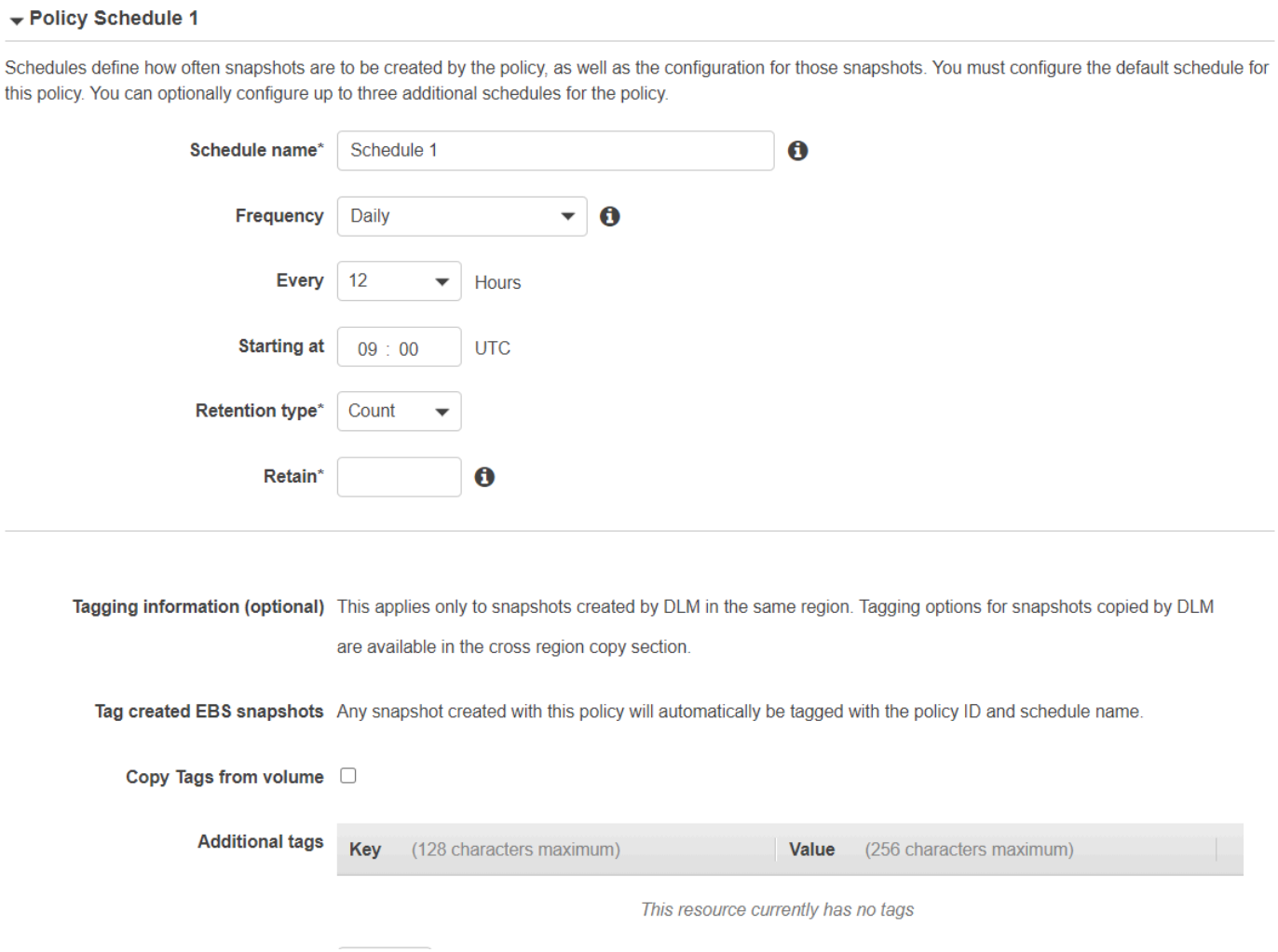
Amazon Data Lifecycle Manager brings a ton of convenience to customers who take regular EBS snapshots. You won't need to script your own Lambda function/Cloudwatch Event anymore just to backup your important files. Amazon DLM features a scheduler which you can configure to create a regular schedule for taking EBS snapshots. You can also define a retention period for EBS snapshots by creating lifecycle policies based on tags. Amazon DLM can back up select EBS volumes or all EBS volumes attached to an EC2 instance that has your specific tags. Any future EBS volume that needs to be included in your snapshot policy can be easily included by adding the identifying tag. What's more, this service is currently available at no additional cost.

To get started with Amazon DLM,

1) Navigate to the EC2 console, and click on **Lifecycle Manager** under **Elastic Block Store** on the left pane.
2) Select **Create Snapshot Lifecycle Policy.**
3) Enter a **description,** select if the resource type should be **EBS volumes** or **EC2 instances**, then enumerate the **tags** that would identify which resources will be included in your lifecycle policy.



4) You can also add your own tags to the lifecycle policy for easier identification.
5) Next, specify the IAM role that would allow the service to manage your volumes. AWS provides a default role, **AWSDataLifecycleManagerDefaultRole**, or you can create a custom IAM role.
6) You can define up to four (4) policy schedules. Each policy schedule describes how often snapshots are to be created by the policy, as well as the configuration for those snapshots.



a) Add your **Schedule name**
b) Define the **frequency**. Valid values are daily, weekly, monthly, yearly, or a custom cron expression.
c) The **Retention Type** specifies how often snapshots are cleaned. Values are count and age. Use count if there is a max number of snapshots that you would like to keep. Use age if there is a lifespan for your snapshots. You can copy each snapshot to up at three additional Regions.

7) You also have the option to enable **Fast Snapshot Restore** and **Cross Region Copy**. If you enable fast snapshot restore, you must choose the Availability Zones in which to enable it. You are billed for each minute that fast snapshot restore is enabled for a snapshot in a particular Availability Zone.
8) You may review your configuration in the **Policy Summary** section. Once everything is good to go, click **Create Policy.**

References:
https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/snapshot-lifecycle.html

Real-time Log Processing using CloudWatch Logs Subscription Filters

You use CloudWatch Logs subscriptions to gain access to a real-time feed of log events from CloudWatch Logs and have it delivered to other services such as an Amazon Kinesis stream, an Amazon Data Firehose, or AWS Lambda. A **subscription filter** defines the filter pattern to use for filtering which log events get delivered to your AWS resource, as well as information about where to send matching log events to. The logs sent to the destination resource are Base64 encoded and compressed with the gzip format. A log group can have two subscription filters at most.

Key elements of a subscription filter include:

- **log group name** - indicates which log group is associated with the subscription filter.
- **filter pattern** - defines how CloudWatch Logs interpret the data for each log event. The filtering expression will control what is delivered to the destination service.
- **destination arn** - could be the ARN of a Kinesis stream, a Data Firehose stream, and/or a Lambda function.
- **role arn** - to put the data in the chosen destination, you must create an IAM Role and grant CloudWatch Logs the necessary permissions.
- **distribution** - this element is only applicable for Amazon Kinesis Stream. By default, log data is grouped by a log stream. This log data can be distributed more evenly to your Kinesis stream by grouping them at random.

Cross-Account Log Data Sharing with Subscriptions

To collaborate with a different AWS account, you can use *cross-account data sharing*. It will allow you to receive the log events from their accounts to your AWS resources. To start receiving log events from cross-account users, the log data recipient must first create a CloudWatch Logs destination. The log group and destination must be in the same Region, but the AWS resource that the destination points to can be in a different region. Lastly, the only supported destination resource for cross-account subscriptions is Kinesis Streams.

There are two parties involved in cross-account data sharing:

- **Log data sender** - retrieves the destination information from the recipient and prepares CloudWatch Logs to send its log events to the specified destination.
- **Log data receiver** - sets up a CloudWatch Logs destination that encapsulates a Kinesis stream and prepares CloudWatch Logs to receive log data. The recipient then shares the information about his destination with the sender.

Key elements of the destination:

- **Destination name** - a user friendly identifier of the destination.
- **Target ARN** - the ARN of the destination resource for the subscription feed.
- **Role ARN** - grants CloudWatch Logs the necessary permissions to put data into the chosen Kinesis stream.
- **Access policy** - an IAM policy that defines who is allowed to send log data to the recipient's destination.

References:
https://docs.aws.amazon.com/AmazonCloudWatch/latest/logs/Subscriptions.html
https://docs.aws.amazon.com/AmazonCloudWatch/latest/logs/SubscriptionFilters.html
https://docs.aws.amazon.com/AmazonCloudWatch/latest/logs/CrossAccountSubscriptions.html

Scaling Memory-Intensive Applications in AWS

Scaling is an important practice in AWS to make sure your applications are always available to meet demand. To scale an EC2 instance based on the `CPUUtilization` metric, we can rely on Amazon CloudWatch for metric monitoring since the metric is available by default for EC2 instances. However, if the EC2 instance is memory-intensive and needs to scale based on `MemoryUtilization`, we will need to install a CloudWatch agent on the instance first. The CloudWatch agent will collect system-level metrics such as memory and store them in Amazon CloudWatch Logs.

A CloudWatch agent is different from the SSM agent. The CloudWatch agent allows you to collect more system-level metrics from your EC2 and on-premises servers than just the standard CloudWatch metrics; while SSM Agent processes requests from the AWS Systems Manager and configures your machine as specified in the request. Although you can configure a document that runs a script to log system-level metrics and have the SSM agent perform this task, it is not a very efficient solution. Instead of using SSM Agent to gather log files on each instance, you can use the Amazon CloudWatch agent to collect additional metrics and logs for you at a more convenient process.

An Auto Scaling Group uses a launch template to determine how it will provision additional EC2 instances for you. Remember that you can only associate one launch template at a time and you are also not allowed to modify an existing launch template settings. If you wish to update your autoscaling group's launch template settings, you must create a new launch template from scratch or copy and modify the existing launch template, then you modify the ASG to use the newly created launch template.

An example scenario:

An organization is running a memory-intensive application on compute-optimized instances. The instances are launched through an autoscaling group and have Cloudwatch Agent installed already. When the workload increases, the autoscaling group is configured to scale based on CPU usage. They noticed that the performance

of the application was still slow so they decided to increase the number of instances further. From a Solutions Architect's perspective, this is not the best approach since they are scaling on the wrong metric. To improve the application's performance while saving costs, they should instead create a new launch template and use a memory-optimized instance as the instance type. Afterwards, they should modify the scaling policy to scale based on memory usage.

References:
https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/metrics-collected-by-CloudWatch-agent.html
https://docs.amazonaws.cn/en_us/systems-manager/latest/userguide/monitoring-cloudwatch-agent.html
https://docs.aws.amazon.com/autoscaling/ec2/userguide/change-launch-config.html

Using AWS Step Functions For Orchestrating Serverless Workflows

Step Functions lets you orchestrate the components of an application into a serverless workflow. A **workflow** is a series of steps, with the output of one step acting as input into the next step. A workflow defines the process of how your application accomplishes its goal from start to end.

In Step Functions, we refer to the workflow as **State machines**, and each step in a State machine is called a **state**.

A State Machine is written using the Amazon States Language or ASL, which has a similar syntax to a JSON document. If you don't want to write in ASL, Step Functions also provides a drag-and-drop interface that further simplifies how you build your state machine. Step Functions will write the state machine definition for you.

States

There are eight state types that you can use to build a State machine:

- **Task state** - represents a single unit of work in your State machine. Application code that needs to be executed by your Lambda function is done by the Task state. A Task state can only include a single Lambda function, an ECS task, an SNS notification, a DynamoDB action, etc.

- **Choice state** - adds a branching logic to your workflow. You can think of it as an if-then-else statement where there is more than one possible outcome that can occur when evaluating a parameter against a condition that you set.

- **Parallel state** - performs two branches of execution at the same time. A use case for this is when you want to run two independent activities that perform different processes to an input from the same node.

- **Wait state** - adds a delay to your State machine before it continues executing states. This is similar to how a Sleep function in Python works.

- **Fail State** - stops the execution of the state machine and marks it as a failure. One example where you could use this is when you have a registration process, and you want to deny any invalid form that a user submitted. You can mark the registration as a failure and have a custom error message like 'InvalidUsername'.

- **Succeed State -** stops the execution successfully.

- **Pass state -** can use the Pass state to pass the input from the previous state to its output without performing work. This is useful when you're debugging a portion of your state machine and you want to have visibility of the output of a particular state.

Rearchitecting Serverless Applications

When it comes to serverless applications, AWS Lambda often stands out as the first choice due to its flexibility and ease of use. It's the go-to solution for executing code in response to events without the need to manage servers. However, Lambda's 15-minute execution limit can be a bottleneck for more extended operations. This limitation becomes apparent in scenarios like processing application forms, where sequential steps such as parsing data, validating information, and performing background checks are required. These operations demand a more flexible approach to handle the complexity and duration of the tasks involved.

AWS Step Functions offers a robust solution for orchestrating workflows that exceed Lambda's time constraints. By leveraging Step Functions, you can design workflows that split the logic into multiple Lambda functions, each handling a part of the process. This approach not only circumvents Lambda's execution limit but also organizes the workflow into a series of discrete, manageable tasks. Optionally, you can use AWS Step Functions' integration with AWS SDK to call API calls directly without the need for additional Lambda functions as intermediaries for API calls.

Built in retry and errors

Furthermore, Step Functions offers built-in error handling and retry mechanisms. If the background check API fails or times out, Step Functions can automatically retry the task according to predefined policies. Once all checks are completed, Step Functions can update the application status in the database, either directly or through another Lambda function, depending on the task's complexity and requirements.

References:
https://docs.aws.amazon.com/step-functions/latest/dg/concepts-states.html
https://aws.amazon.com/blogs/aws/new-aws-step-functions-workflow-studio-a-low-code-visual-tool-for-building-state-machines/

AWS Pricing Models

Lowering the customers' overall infrastructure and management costs is always the objective of an AWS Solutions Architect. When we talk about reducing costs, we do not mean that we will compensate performance, security, and availability just to reach that lower level of spending. The process is much more complex than that. When we want to lower costs, we as Solutions Architects will find ways to optimize and refine the existing infrastructure. We weigh all of our available options and design new solutions that are centered around those options. We make sure that these designs still achieve the business objectives, performance baselines, security requirements, and all of the important bits while bringing in higher savings for our customers *in the long term*. That is why, to become an AWS Solutions Architect Professional, you must be aware of all the pricing models in AWS.

In AWS, there are many pricing models to choose from. You have:

1) Pay-as-you-go or on-demand
2) Long term capacity reservations
3) Purchase based on available capacity or Spot
4) Volume discounts

There are three fundamental drivers of cost with AWS: compute, storage, and outbound data transfer. For compute capacity, you often associate it with the first three models. For storage and data transfers, you associate them with the last model.

Amazon EC2 and services that use EC2 as the backend, such as Amazon ElastiCache, Amazon RDS, and Amazon Redshift, offer greater savings if you purchase Reserved Instances rather than On-Demand Instances. A reservation allows you to commit to one full year or three full years of continuous usage. You may also choose to pay a portion of the total bill or the full bill upfront, which will give you even higher discount rates. There are many ways to reserve compute capacity in AWS, each with their own advantages and disadvantages. Though one thing is for certain, if you expect workloads to run continuously for long periods of time then you should always consider using Reserved Instances.

Conversely, if you have workloads running only for short bursts or tasks that are only handled as they arrive in a queue, then consider using Spot Instances. Spot Instances let you take advantage of unused EC2 capacity in the AWS cloud, and you just pay the Spot price that's in effect for the current hour for the instances that you launch. Since you can never expect when your Spot instances will be terminated, only use them if your workload can handle interruptions. Another way to use Spot instances is when you just need "extra capacity", such as stateless servers moving data from one place to another. You can maintain a fixed amount of compute capacity that's always running and have spot reduce the burden when there is a heavier workload.

ℹ️ **Solutions Architect Professional Exam Notes:**

What about using mixed pricing models?

In the exam, you will encounter questions that ask you how to provision your compute resources in the most cost-effective way. A common service to use as an example is Amazon Redshift. You can combine Reserved, On-Demand, and Spot instances in one cluster, and delegate each model according to its own area of strength. For example, you can have one reserved leader node and a mix of on-demand and spot compute nodes.

Lastly, when you are managing multiple accounts through AWS Organizations, you can benefit from lower total costs through volume discounts. Some services, such as AWS Data Transfer Out and Amazon S3, have volume pricing tiers across certain usage dimensions that give you lower prices the more you use the service. So with consolidated billing, AWS considers the usage across all accounts for the pricing tier eligibility.

References:
https://d1.awsstatic.com/whitepapers/aws_pricing_overview.pdf
https://d1.awsstatic.com/whitepapers/cost-optimization-reservation-models.pdf
https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/instance-purchasing-options.html
https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/useconsolidatedbilling-discounts.html

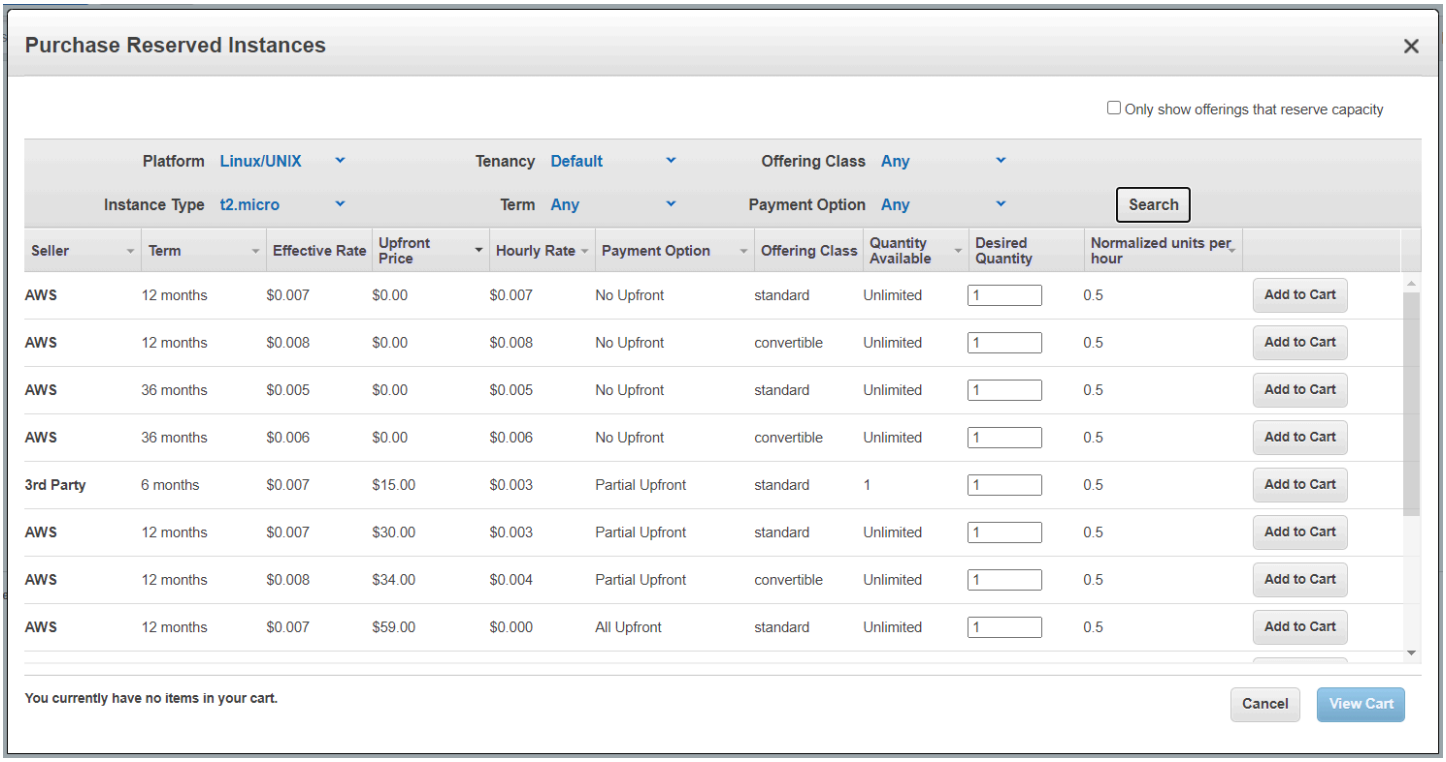
Reserved Instances and Savings Plan

As mentioned in the previous section, there are many options to save costs with Amazon EC2 instances. You choose the arrangement that works best for your needs.

Purchasing Reserved Instances is the commonly known method of receiving discounts for EC2 pricing. You select how many years you would like to use the reserved instance and specify if you will be paying an upfront cost for greater discount benefits. Services that support RI include Amazon EC2, Amazon RDS, Amazon Elasticache, and Amazon Redshift. There are two types of reserved instances to choose from:

1) Standard RI
    - Can be purchased to apply to instances in a specific Availability Zone (zonal Reserved Instances), or to instances in a Region (regional Reserved Instances).
    - Enables you to modify instance attributes such as Availability Zone, scope (from zonal to regional and vice versa), network platform (EC2 Classic to VPC and vice versa), and instance size (**within the same instance type e.g. C-type**) of your Reserved Instance.
    - Standard Reserved Instances typically provide the highest discount levels.
2) Convertible RI
    - Can be purchased to apply to instances in a specific Availability Zone (zonal Reserved Instances), or to instances in a Region (regional Reserved Instances).

- Enables you to exchange one or more Convertible Reserved Instances for another Convertible Reserved Instance **with new attributes**. These attributes include instance family, instance type, platform, scope, and tenancy, if the exchange results in the creation of a Reserved Instance of equal or greater value.
- Useful when workloads are likely to change or you do not have forecasted data on your usage patterns.



Savings Plans is a fairly new offering from AWS that works similar to RIs. They give you pricing discounts in exchange for a long-term usage commitment. Savings Plans is not only limited to EC2 instances, but can also be applied to Lambda and Fargate usage. AWS offers two types of Savings Plans:

1) Compute Savings Plans
   - Provide the most flexibility and help to reduce your costs by up to 66%.
   - These plans automatically apply to EC2 instance usage **regardless** of instance family, size, AZ, region, OS or tenancy, and also apply to Fargate and Lambda usage.
   - For example, you can change from C4 to M5 instances, shift a workload from EU (Ireland) to EU (London), or move a workload from EC2 to Fargate or Lambda at any time and automatically continue to pay the Savings Plans price.
2) EC2 Instance Savings Plans
   - Provide the lowest prices, offering savings up to 72% in exchange for commitment to usage of **individual instance families** in a region (e.g. M5 usage in N. Virginia).

- This automatically reduces your cost on the selected instance family in that region regardless of AZ, size, OS, or tenancy.
- EC2 Instance Savings Plans give you the flexibility to change your usage between instances within a family in that region. For example, you can move from c5.xlarge running Windows to c5.2xlarge running Linux and automatically benefit from the Savings Plans prices.

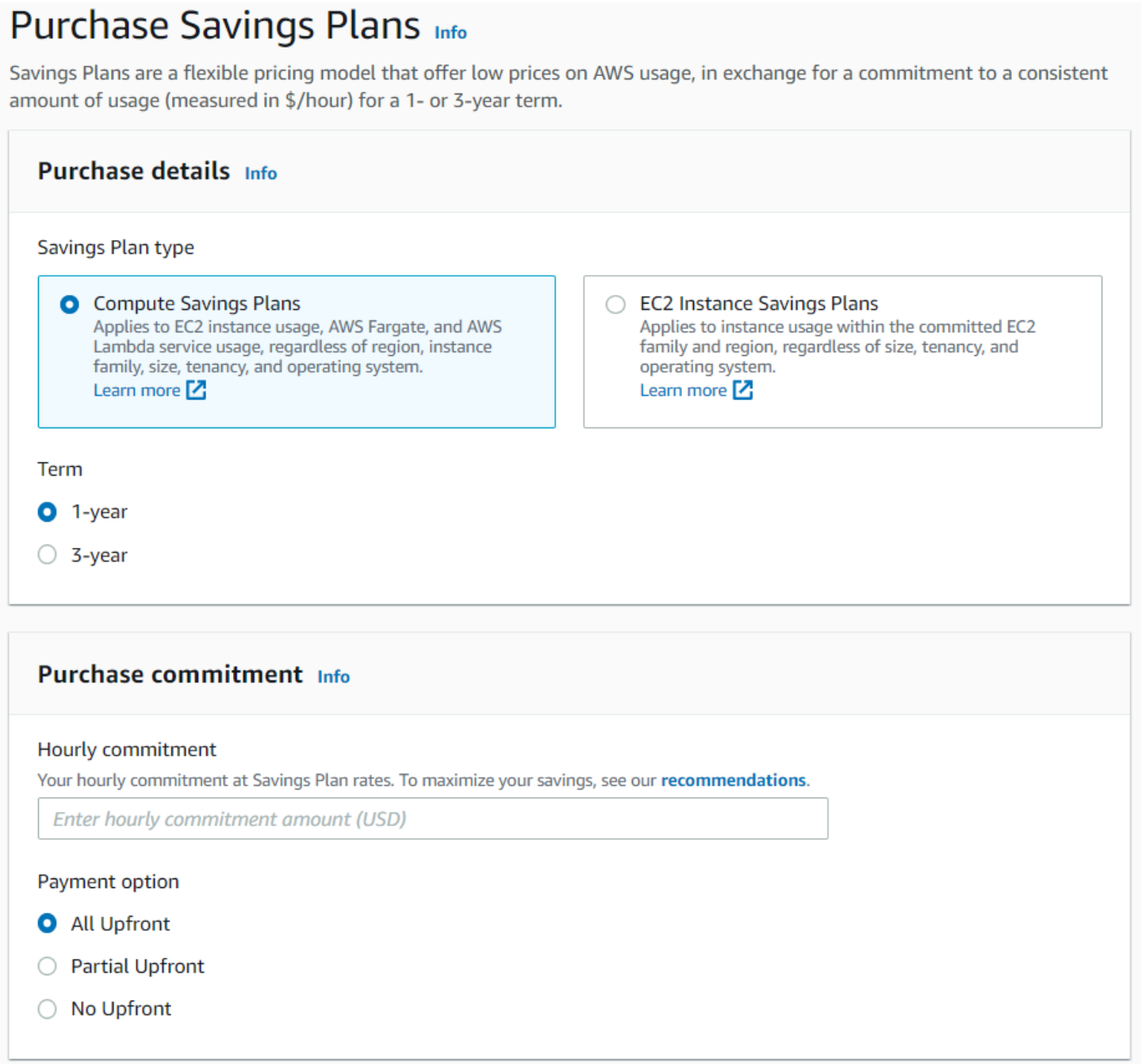


Figure: AWS Compute Savings Plans

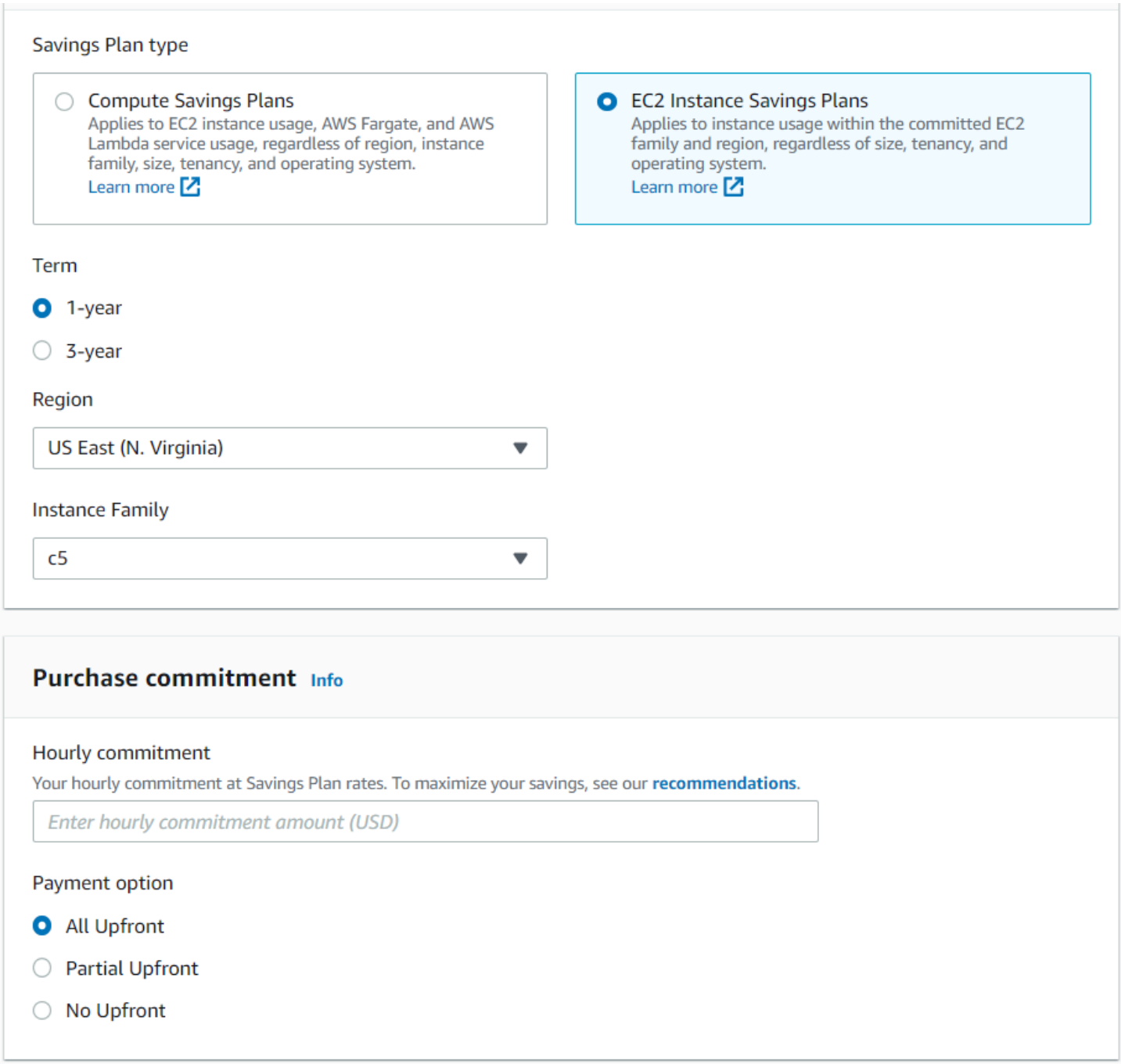


Figure: AWS EC2 Instance Family Savings Plans

Savings Plans also works with AWS Organizations, similar to RIs. By default, the benefit provided by Savings Plans is applicable to usage across all accounts within an AWS Organization/Consolidated billing family. However, you can also choose to restrict this benefit to only the account that purchased them. Your RIs will continue to work alongside Savings Plans if you wish to use both.

A few notable differences between Savings Plans and Reserved Instances are:

- You cannot directly reserve compute capacity with Savings Plans. You can, however, reserve capacity with On Demand Capacity Reservations and pay lower prices on them with Savings Plans.
- You also cannot sell unused Savings Plans capacity in the AWS Marketplace, unlike RIs.

References:
https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/reserved-instances-types.html
https://docs.aws.amazon.com/whitepapers/latest/cost-optimization-reservation-models/standard-vs.-convertible-offering-classes.html
https://aws.amazon.com/savingsplans/

Amazon EC2 Auto Scaling

Amazon EC2 Auto Scaling helps maintain application availability by automatically adding or removing EC2 instances based on workload demand. It improves fault tolerance, performance, and cost optimization by ensuring the correct number of instances are running at all times.

With Amazon EC2 Auto Scaling, you can configure scaling policies that automatically adjust capacity according to metrics such as CPU utilization, network traffic, or custom CloudWatch metrics.

Common benefits include:

- Improved application availability
- Automatic scaling during traffic spikes
- Reduced operational overhead
- Cost savings by removing unused capacity
- Better fault tolerance across Availability Zones

Dynamic Scaling

Dynamic scaling automatically adjusts the number of EC2 instances in response to real-time demand.

Amazon EC2 Auto Scaling continuously monitors Amazon CloudWatch metrics and scales resources in or out whenever thresholds are breached.

There are several dynamic scaling policy types:

Scaling Type	Description
Target Tracking Scaling	Maintains a specific metric target such as 50% CPU utilization
Step Scaling	Adds or removes instances based on alarm thresholds
Simple Scaling	Performs a scaling action after a CloudWatch alarm triggers

Example:

- Add 2 EC2 instances when CPU utilization exceeds 70%
- Remove 1 EC2 instance when CPU utilization drops below 30%

Dynamic scaling is ideal for workloads with unpredictable or fluctuating traffic patterns.

Scheduled Scaling

Scheduled scaling allows scaling actions to occur at predefined times.

This is useful when traffic patterns are predictable, such as:

- Business hours
- Weekly events
- Payroll processing
- Marketing campaigns

Administrators define schedules that increase or decrease capacity automatically.

Example:

- Increase capacity to 10 instances every weekday at 8:00 AM
- Reduce capacity to 2 instances every weekday at 8:00 PM

Scheduled scaling helps ensure sufficient resources are available before expected traffic increases occur.

Predictive Scaling

Predictive scaling uses machine learning to forecast future traffic patterns and proactively launch EC2 instances before demand increases.

Amazon EC2 Auto Scaling analyzes historical utilization data and predicts future capacity requirements. This helps applications maintain performance during predictable traffic spikes.

Predictive scaling is particularly useful for applications with recurring usage patterns such as:

- Daily traffic peaks
- Weekly workloads
- Seasonal demand
- E-commerce events

Instead of reacting to traffic increases after they occur, predictive scaling launches instances in advance to reduce latency and improve user experience.

Example:

An e-commerce platform experiences increased traffic every evening at 7:00 PM. Predictive scaling learns this pattern and automatically launches additional EC2 instances before the traffic spike begins.

Using Different AWS Cost Management Services

Your AWS Billing is one thing that you should always keep an eye out for. One of the best benefits the cloud can give you is lowering your CapEx so you can direct it to other more valuable targets. Therefore, you should also be knowledgeable in the different cost monitoring and cost control services in AWS. These services will be your main go-to tools to monitor your spending, as well as to discover opportunities to further reduce costs.

1) **AWS Cost Allocation Tags**

   AWS Cost Allocation Tags allow you to generate cost allocation reports as CSV files with your usage and costs grouped by your active tags. Cost tagging is a very useful strategy if you need to group different AWS resources according to defined categories, such as business cases for example. It also allows you to check if the performance gains from your usage are proportional to the amount you are paying for. This allows you to make informed decisions on how to manage your own AWS resources cost-effectively.

2) **AWS Cost Explorer + AWS Cost and Usage Report**

   AWS Cost Explorer and AWS Cost and Usage Report are two very commonly used services for understanding your spending in AWS. Visualization is the key aspect of these two services. They allow you to construct meaningful diagrams that easily show you the trends in your spending. You can filter through the data by specifying your parameters (such as which services to include, which regions, which time period, etc), and this can provide you either a high level overview or a more granular view of your spending patterns. AWS Cost Explorer also analyzes your billing history and provides you a forecast of your future costs and usages. Lastly, once you have configured a cost and usage dashboard that you would like to review continuously, you can save it as a report and return to it anytime you wish. This is very useful if you like to track your EC2 Reserved Instance usage for example.

3) **AWS Budgets + Amazon SNS**

   Once you have a pretty good idea of your cost and usage patterns in AWS, you can start configuring your own AWS Budgets alerts to make sure you don't overspend. Having an alert watching over your spending, especially when you manage multiple accounts, will give you the peace of mind that you need. The alerts also work with Amazon SNS to notify relevant personnel when your budget is about to be exceeded. This gives you and your business the opportunities to refine your operations and remove any unnecessary resource consumption. Budget tracking may also reflect business growth, since the higher your budget is, the more you can innovate and expand your operations.

4) **AWS Trusted Advisor**

AWS Trusted Advisor is an indispensable tool for ensuring your account is as cost-effective as possible. The Cost Optimization feature under AWS Trusted Advisor makes use of the well-architected best practices for cost-efficiency, so you have a centralized monitoring solution that continuously reviews your account for any items that can incur you unnecessary expenses. How AWS Trusted Advisor does this is by having multiple checks that scan for underutilized (e.g. idle instances) and unoptimized (e.g. oversized instances) resources that are running in your account. The number of Trusted Advisor checks that will be available to you will depend on your support plan. Nevertheless, you should often review your AWS Trusted Advisor to ensure all your resources are well-utilized and right-sized.

5) **Consolidated Billing for AWS Organizations**

If you are running AWS Organizations, you should enable consolidated billing to enjoy some of the benefits of volume discounts. Services such as S3 and Data Transfer Out offer pricing tiers that lower the cost the more you use the service. Aside from volume discounts, if you are the master account (payer account), you should also centrally track and monitor spending of each of your payee accounts. Do note, however, that if you do not have all AWS Organizations features enabled, you cannot control how each account handles their spending. The consolidated billing feature treats all the accounts in the organization as one account. Therefore, all accounts in the organization can receive the hourly cost benefit of Reserved Instances that are purchased by any other account. Reserved Instance discount sharing can be disabled if you do not wish to share RIs.

References:
https://aws.amazon.com/aws-cost-management/
https://docs.aws.amazon.com/pdfs/whitepapers/latest/cost-management/cost-management.pdf

Domain 4: Accelerate Workload Migration and Modernization

Overview

The fourth and last exam domain of the AWS Certified Solutions Architect Professional SAP-C02 certification exam is all about migration in AWS. As a Solutions Architect, the onus is on you to properly migrate your on-premises applications and services onto the AWS Cloud. This is a daunting task as there are many intricacies involved; not just with technologies, but also with infrastructure design, personnel, and budget. Some companies would often perform a lift and shift migration (also known as rehosting) onto AWS while other organizations re-platform or re-architect their existing systems. AWS provides a wealth of services, tools, and features that you can leverage for your migration workloads and to accelerate the whole process.

This domain has the least amount of questions in the SAP-C02 with 20% coverage so therefore, you have to limit the time you spent studying the topics here. The questions in the actual exam revolve around these topics:

- Select existing workloads and processes for potential migration.
- Determine the optimal migration approach for existing workloads
- Determine a new architecture for existing workloads.
- Determine opportunities for modernization and enhancements.

In this chapter, we will cover the related topics for migration in AWS that will likely show up in your Solutions Architect Professional exam.

Planning Out a Migration

When planning to do a migration, there are a lot of factors that you need to carefully assess. It is common for companies to skip the planning stage and go right into migration, which in the end just becomes a lift-and-shift process. Although lift-and-shift is a legitimate migration strategy, it is not always the best one to go with. You are missing out on a lot of potential improvements and cost savings to your operations which stems from not being able to fully take advantage of the cloud. To establish a proper migration plan, we will answer the basic questions of who, what, where, when, why, and how. These are some of the questions that you need to answer to properly designate tasks and objectives to your people.

Who — Who are the stakeholders involved? Who will oversee the migration (SME)? Who will decide what to migrate and what migration strategy to use? Who will be responsible for which part of the migration? Who will manage the new infrastructure after migration?

What — What is the business objective? What will you be migrating? What is the strategy for the migration? What tools will you be using for the migration? What is the timeline for the migration? What possible issues can come up before, during, and after migration?

Where — Where will you migrate your applications to? Will it be on a managed service or on a virtual machine? Will there be refactoring involved and where exactly? Where will you start?

When — When will the migration be performed? When is the migration expected to end? When can you test your new infrastructure for production workloads? When will the old infrastructure be expected to retire?

Why — Why will you migrate these applications? Why will you migrate these data? Why migrate this first before the others? Why aren't some applications being migrated? Why choose AWS over other platforms? Why migrate to EC2 or to RDS?

How — How will you perform the migration? How long will it take for you to migrate one application? One database? One complete service? How will you monitor your progress? How do you measure success after migration? How will you rollback after a failed migration? How will you manage the new infrastructure after migration? How much will this all cost you?

Migration Strategies

There are seven migration strategies, which are also known as the 7 R's, that we can perform in AWS. Each one has its own advantages and disadvantages.

Retire

This strategy is very easy to understand. If you don't need a certain application anymore then disregard it. Start by determining which components are essential to your business and which are not. This way, your migration process will only involve the important parts of your system, thereby reducing cost, effort, and time for everyone.

Relocate

Basically, this migration strategy is the act of relocating your proprietary virtualization platform from your on-premises network to the cloud. This is perfect if the existing license of your on-premises platform is supported in the cloud and you want to maintain a low to medium operational complexity in managing your infrastructure. You have to assess both the cost and the operational efficiency gains first via technology replatform against the required effort to re-platform or re-architect your stack while considering migration deadlines. For example: if your on-premises servers are running in a VMWare vSphere platform, you can consider relocating your virtualized servers to AWS using the VMWare Cloud on AWS service.

Rehost

The simplest method of migrating to AWS is to move your applications without changing them, essentially a "lift-and-shift" scenario. A common example for this is when you are moving your legacy web servers from on-prem onto EC2 instances. You treat the EC2 instances as if they were your own servers, thereby not

modifying any aspect of your application. This strategy is a quick and easy way to get things running in the cloud without much repercussion. Although you do not make use of all of AWS's advantages, you still receive some such as cheaper infrastructure pricing options, some elasticity and scalability for your VMs, as well as basic security and network services.

> ℹ️ **Solutions Architect Professional Exam Notes:**
>
> Another not so common but possible application for re-host is when a customer wants to move a certain application to AWS but AWS currently doesn't support it as a native service. For example, you want to move a database to AWS but RDS does not support its engine or engine version, then you will have to use EC2 to run your database.

Replatform

With re-platform, you are utilizing new services to host your applications without changing the core of it. These services usually provide some form of support or feature that reduces management overhead. Examples would be AWS Elastic Beanstalk for hosting web applications and Amazon RDS for hosting your databases. The main benefit of re-platforming is to achieve increased savings and agility for your workloads.

> ℹ️ **Solutions Architect Professional Exam Notes:**
>
> You might encounter a lot of PaaS scenarios in your exam, including services such as Elastic Beanstalk, RDS, ECS and many more. If the scenario requests that there should be less management overhead for the customer's application or database, look for services that give you that benefit. For example, with Elastic Beanstalk and ECS, they can quickly provision the resources you need with just a few clicks and they also support CI/CD deployment. They make it convenient for developers to apply their changes to production and quickly rollback if they encounter any issues. For RDS, these are the common maintenance procedures such as patching, automated backups, scaling, monitoring, etc.

Refactor / Re-architect

Refactoring is often the most expensive strategy for customers and is also the most complex. Although this strategy allows you to reap the most out of AWS, there are a lot of factors and decision-making involved which can prolong your migration process. Sometimes, it is easier to just start fresh rather than modifying legacy systems to fit into AWS. Another option that you can do is to first rehost or replatform as much of your system as you can, and slowly but surely refactor them to fit your desired environment.

It takes a lot of expertise and understanding of different technologies, not just AWS, to properly architect a system in the cloud. This is why Solutions Architects are paid handsomely in the industry. AWS is continuously growing and innovating new products, and as a Solutions Architect, you need to be up-to-date with these

offerings to bring the best value to your customers. Hence, certifications are a must if you want to reach the highest level of this craft.

Repurchase

This strategy discusses moving from perpetual licenses to a software-as-a-service model. Figure out what products are available out there that you can adopt instead of using your own systems. This reduces the chances of incompatibility and allows you to focus on your value-adding operations. By purchasing or adopting well-known alternatives, you also gain access to a larger user base with better support and more consistent updates from the developers of the product.

Retain

Although this strategy can be a bit counterintuitive, performing a migration doesn't necessarily mean you have to move *everything immediately*. Sometimes, it can be better to leave an application behind or to hold off its migration temporarily. For example, if you only allocated a specific budget for this task, you can start off with migrating crucial applications first, and leave the rest on-prem. This way, you can strictly control the whole migration procedure and design better implementations as you go. The only downside to this strategy is that it can prolong your whole migration journey, which in effect can increase the total cost and effort than what is necessary.

> ℹ️ **Solutions Architect Professional Exam Notes:**
>
> It can sometimes be impossible to move a system to AWS without fully re-architecting it. An application can be very old and too deeply rooted in one's current operations to properly move it to the cloud. Other customers may cite that they need their applications to stay on-premises due to compliance reasons, or because their (expensive) software licenses are tied to their servers. There are also a lot of customers who go for a hybrid environment because they do not want to fully commit to the cloud. Be sure to take note of your scenarios to know when to retain and to not retain applications.

Strategy (increasing complexing)	**Effort and Cost**	**Opportunity to Optimize**
Retire	N/A	N/A
Retain	Minimal effort and cost	N/A
Relocate	Minimal effort and cost	Small
Rehost	Minimal to Average effort and cost	Small
Repurchase	Average effort and cost	Small

Replatform	Above average effort and cost	Average
Refactor/Re-architect	High effort and cost	High

References:
https://docs.aws.amazon.com/prescriptive-guidance/latest/migration-retiring-applications/overview.html
https://docs.aws.amazon.com/prescriptive-guidance/latest/application-portfolio-assessment-guide/iterating-7-rs-migration-strategy-selection.html

Analyzing Your Workloads Using AWS Application Discovery Service

When you have hundreds to thousands of servers, virtual machines, and applications running in your on-premises infrastructure, it can be tedious to do an inventory and analyze all their usage patterns. Data collection and dependency mapping are very important tasks during the migration planning phase, as it will influence your decision-making and the outcome of your migration process. To conduct this analysis in a simpler manner, you can use AWS Application Discovery Service to perform these tasks for you.

AWS Application Discovery Service is an automated solution that collects and presents configuration, usage, and behavior data from your servers to help you better understand your workloads. These data are then stored in the AWS Application Discovery Service local data store, and can be exported in csv format. The data will help you estimate the Total Cost of Ownership (TCO) of running on AWS. When paired with AWS Migration Hub, you can use the resulting data to migrate the discovered servers and applications using an AWS or partner migration tool, and track their progress as they get migrated to AWS.

AWS Application Discovery Service works in two ways depending on the environment to be scanned. If you are a VMware user, AWS Application Discovery Service uses an agentless discovery process to collect VM configuration and performance profiles. Users in a non-VMware environment or those that need additional information, such as network dependencies and information about running processes, will need to install the Application Discovery Agent on each of your servers and VMs.

AWS Application Discovery Service is able to collect the following information:

- Server hostnames,
- IP addresses,
- MAC addresses,
- CPU, network, memory, and disk utilization
- Disk and network performance (e.g., latency and throughput)

AWS Application Discovery Service agents record inbound and outbound network activity for each server. This data can then be used to understand the dependencies across servers. For VMware environments, you won't be able to record if you do not install the agent first.

Do note that the AWS Application Discovery Service does not perform any type of migration. It is purely a discovery service that integrates well with other AWS migration services.

Reference:
https://aws.amazon.com/application-discovery
https://aws.amazon.com/migration-hub/
How AWS Migration Hub Helps You Plan, Track, and Complete Your Application Migrations

Performing Data Migration

ℹ️ **Solutions Architect Professional Exam Notes:**

How much data can move from your on-premises data center to AWS through your current network connection? For a best case scenario, you can use this formula:

```
No. of days = (Total Bytes) /  ( Megabits per second * 125 * 1000 )
                               * Network Utilization
                               * (60 seconds * 60 minutes * 24 hours )
```

For example:

- You have a T1 connection (1.544Mbps) with a network utilization of 80% and 1 TB of data to move in or out of AWS. 1 TB is equivalent to 1,099,511,627,776 bytes. Using the formula above, we'll get the following result:

```
= (Total Bytes) / (Megabits per second *125*1000) * Network Utilization * Time
= ( 1,099,511,627,776 ) / ( 1.544 * 125 * 1000 ) * ( 0.80 ) * ( 60 * 60 * 24 )
= 1,099,511,627,776 / 13,340,160,000
= 82.42
```

- As calculated above, the theoretical minimum time it would take to load over your network connection at 80% network utilization is **82** days. Sometimes, you might not have a calculator on-hand, so you can always go for rough estimates instead.

AWS Storage Gateway lets you connect and extend your on-premises applications to AWS storage services such as S3, S3 Glacier and EBS.

- File Gateway uses **SMB or NFS file shares** for on-premises applications to store files as **S3 objects** and access them with traditional file interfaces.

- Volume Gateway stores or caches block volumes locally, with point-in-time backups as **EBS snapshots**. These snapshots can also be recovered in the cloud.
- Tape Gateway **virtual tape library** (VTL) configuration integrates with existing backup software for cost effective tape replacement in **Amazon S3** and long term archival in **S3 Glacier Flexible Retrieval** and **S3 Glacier Deep Archive**.

Use AWS Storage Gateway if you need to sync appliances with Amazon S3 or generate Amazon EBS volumes. Transfer speeds will depend on your network connection speed.

AWS Direct Connect is a dedicated physical connection to accelerate network transfers between your datacenters and AWS datacenters. This dedicated connection can be partitioned into multiple virtual interfaces. Partitioning enables you to use the same connection to access public resources such as objects stored in Amazon S3 using public IP address space, and private resources such as Amazon EC2 instances running within a VPC using private IP space, while maintaining network separation between the public and private environments. **1Gbps and 10Gbps ports** are available. You can order lines with transfer speeds of **50Mbps, 100Mbps, 200Mbps, 300Mbps, 400Mbps, and 500Mbps**. Direct Connect works with both **IPsec VPN** using public VIF and **AWS Transit Gateway** with transit VIF if you need to connect to multiple VPCs.

AWS S3 Transfer Acceleration makes public Internet transfers to Amazon S3 faster by taking advantage of **Amazon CloudFront's globally distributed edge locations**. There is **no guarantee** that you'll experience better transfer speeds, so if you need consistent, fast transfers, use other options. If you need a low cost option in speeding up transfers whenever acceleration is available, this will do.

AWS Data Sync is a data transfer service that **automates** moving data between on-premises storage and **Amazon S3, Amazon EFS, or Amazon FSx for Windows File Server.** You can use DataSync to copy data over AWS Direct Connect or Internet links to AWS for data migrations, recurring data processing workflows, and automated replication for data protection and recovery.

Amazon Data Firehose is the easiest way to capture, modify, and deliver data streams into various AWS services such as **Amazon S3**, **Amazon Redshift**, **Amazon OpenSearch Service**, **Splunk**, **Snowflake**, and other 3rd party analytics services. Additionally, it simplifies the process of maintaining streaming data delivery pipelines by managing the provisioning and scaling of resources for you.

References:
https://aws.amazon.com/cloud-data-migration/
https://docs.aws.amazon.com/whitepapers/latest/aws-vpc-connectivity-options/aws-direct-connect-vpn.html
Migrating Data to AWS: Understanding Your Options - AWS Online Tech Talks

Performing Database Migration

For database migration, you will be using AWS Database Migration Service if you need a managed solution without giving your database downtime. **AWS Database Migration Service** (AWS DMS) can migrate your data to and from relational databases, data warehouses, NoSQL databases, and other types of data stores. You can perform homogeneous migrations such as Oracle to Oracle, and heterogeneous migrations between different database platforms, such as Oracle to Amazon Aurora or Microsoft SQL Server to MySQL. It also allows you to stream data to Amazon Redshift from any of the supported sources including Amazon Aurora, PostgreSQL, MySQL, MariaDB, Oracle, SAP ASE, and SQL Server.

AWS Database Migration Service can also be used for continuous data replication with high availability through change data capture (CDC) to keep your databases in-sync even after migration has completed.

To perform a database migration, AWS DMS connects to the source data store, reads the source data, and formats the data for consumption by the target data store. It then loads the data into the target data store. Cached transactions and log files are also written to disk.

At a high level, you perform the following to initiate a migration:

- Create a replication server
- Create source and target endpoints that have connection information about your data stores
- Create one or more migration tasks to migrate data between the source and target data stores

Once migration has started:

- AWS DMS will first do a full migration where existing data from the source is moved to the target. While this is in progress, any changes made to the tables being loaded are cached on the replication server.
- Once the full migration is finished, AWS DMS will apply the cached changes stored in the replication server.
- When all tables have been loaded on the target, AWS DMS begins to collect changes as transactions for the ongoing replication phase.

If your migration is heterogeneous (between two databases that use different engine types), you can use the **AWS Schema Conversion Tool** (AWS SCT) to generate a complete target schema for you. If you use the tool, any dependencies between tables, such as foreign key constraints, need to be disabled during the migration's "full load" and "cached change apply" phases.

Oftentimes, we associate migration with moving things into AWS, but this is not always the case. There are cases when you are asked to migrate or sync a database outside of AWS, and with the least possible downtime. If you are hosting your database in Amazon RDS for MySQL and you also have an on-premises MySQL server, you can migrate data from Amazon RDS for MySQL to your on-premises database server.

To do so:

- Create an Amazon RDS for MySQL read replica
- Switch the replication target from the read replica to the on-premises server
- Once replication is finished and the pair are in-sync, you may delete the read replica

If none of these options work for you, there is always the traditional backup and restore route. However, it will be difficult to keep your databases in-sync as new changes come in and experience little downtime during the switchover.

References:
https://docs.aws.amazon.com/whitepapers/latest/aws-overview/migration-services.html
https://docs.aws.amazon.com/dms/latest/userguide
https://aws.amazon.com/premiumsupport/knowledge-center/replicate-amazon-rds-mysql-on-premises/

AWS CHEAT SHEETS

The following is a compilation of the most relevant AWS services cheat sheets, which are among the core topics in the Solutions Architect Professional Exam. Head over to the **Tutorials Dojo website** to view our complete library of AWS cheat sheets.

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

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

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.

- **AWS Site-to-Site Virtual Private Network** (VPN) connections can be moved from a virtual private gateway to an **AWS Transit Gateway** without having to make any changes on your customer gateway. Transit Gateways enable you to easily scale connectivity across thousands of Amazon VPCs, AWS accounts, and on-premises networks.
- **AWS PrivateLink** enables you to privately connect your VPC to supported AWS services, services hosted by other AWS accounts (VPC endpoint services), and supported AWS Marketplace partner services. You do not require an internet gateway, NAT device, public IP address, AWS Direct Connect connection, or VPN connection to communicate with the service. Traffic between your VPC and the service does not leave the Amazon network.
- You can create a **VPC peering connection** between your VPCs, or with a VPC in another AWS account, and enable routing of traffic between the VPCs using private IP addresses. You cannot create a VPC peering connection between VPCs that have overlapping CIDR blocks.
- Applications in an Amazon VPC can securely access AWS PrivateLink endpoints across VPC peering connections. The support of VPC peering by AWS PrivateLink makes it possible for customers to privately connect to a service even if that service's endpoint resides in a different Amazon VPC that is connected using VPC peering.
- AWS PrivateLink endpoints can now be accessed across both intra- and inter-region VPC peering connections.

VPC Use Case Scenarios

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

Subnets

- When you create a VPC, you must specify a range of IPv4 addresses for the VPC in the form of a Classless Inter-Domain Routing (CIDR) block (example: 10.0.0.0/16). This is the **primary CIDR block** for your VPC.
- You can add one or more subnets in each Availability Zone of your VPC's region.
- You specify the CIDR block for a subnet, which is a subset of the VPC CIDR block.
- A CIDR block must not overlap with any existing CIDR block that's associated with the VPC.
- Types of Subnets
    - Public Subnet - has an internet gateway
    - Private Subnet - doesn't have an internet gateway

    - VPN-only Subnet - has a virtual private gateway instead
- IPv4 CIDR block size should be between a /16 netmask (65,536 IP addresses) and /28 netmask (16 IP addresses).
- The **first four IP addresses and the last IP address in each subnet CIDR block** are **NOT available** for you to use, and cannot be assigned to an instance.
- You cannot increase or decrease the size of an existing CIDR block.
- When you associate a CIDR block with your VPC, a route is automatically added to your VPC route tables to enable routing within the VPC (the destination is the CIDR block and the target is *local*).
- You have a limit on the number of CIDR blocks you can associate with a VPC and the number of routes you can add to a route table.
- The following rules apply when you add IPv4 CIDR blocks to a VPC that's part of a **VPC peering connection**:
    - If the VPC peering connection is active, you can add CIDR blocks to a VPC provided they do not overlap with a CIDR block of the peer VPC.
    - If the VPC peering connection is pending-acceptance, the owner of the requester VPC cannot add any CIDR block to the VPC. Either the owner of the accepter VPC must accept the peering connection, or the owner of the requester VPC must delete the VPC peering connection request, add the CIDR block, and then request a new VPC peering connection.
    - If the VPC peering connection is pending-acceptance, the owner of the accepter VPC can add CIDR blocks to the VPC. If a secondary CIDR block overlaps with a CIDR block of the requester VPC, the VPC peering connection request fails and cannot be accepted.
- If you're using AWS Direct Connect to connect to multiple VPCs through a direct connect gateway, the VPCs that are associated with the direct connect gateway must not have overlapping CIDR blocks.
- The CIDR block is ready for you to use when it's in the *associated* state.
- You can disassociate a CIDR block that you've associated with your VPC; however, you cannot disassociate the primary CIDR block.

Subnet Routing

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

Subnet Security

- Security Groups — control inbound and outbound traffic for your instances
    - You can associate one or more (up to five) security groups to an instance in your VPC.

    - If you don't specify a security group, the instance automatically belongs to the default security group.
    - When you create a security group, it has no inbound rules. By default, it includes an outbound rule that allows all outbound traffic.
    - Security groups are associated with network interfaces.
- Network Access Control Lists — control inbound and outbound traffic for your subnets
    - Each subnet in your VPC must be associated with a network ACL. If none is associated, automatically associated with the default network ACL.
    - You can associate a network ACL with multiple subnets; however, a subnet can be associated with only one network ACL at a time.
    - A network ACL contains a numbered list of rules that is evaluated in order, starting with the lowest numbered rule, to determine whether traffic is allowed in or out of any subnet associated with the network ACL.
    - The default network ACL is configured to **allow all traffic to flow in and out** of the subnets to which it is associated.
    - 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.
- Flow logs can help you with a number of tasks, such as:
    - Diagnosing overly restrictive security group rules
    - Monitoring the traffic that is reaching your instance
    - Determining the direction of the traffic to and from the network interfaces
- Flow log data is collected outside of the path of your network traffic, and therefore does not affect network throughput or latency. You can create or delete flow logs without any risk of impact to network performance.
- After you've created a flow log, it can take several minutes to begin collecting and publishing data to the chosen destinations. Flow logs do not capture real-time log streams for your network interfaces.
- VPC Flow Logs can be sent directly to an Amazon S3 bucket which allows you to retrieve and analyze these logs yourself.
- Amazon security groups and network ACLs don't filter traffic to or from link-local addresses or AWS-reserved IPv4 addresses. Flow logs do not capture IP traffic to or from these addresses.

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 with it.	Applies to all instances in the subnet it is associated with.
Has separate rules for inbound and outbound traffic.	Has separate rules for inbound and outbound traffic.
A newly created security group denies all inbound traffic by default.	A newly created nACL denies all inbound traffic by default.
A newly created security group has an outbound rule that allows all outbound traffic by default.	A newly created nACL denies all outbound traffic by 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.

VPC Networking Components

- Network Interfaces
  - a virtual network interface that can include:
    - a primary private IPv4 address
    - one or more secondary private IPv4 addresses
    - one Elastic IP address per private IPv4 address
    - one public IPv4 address, which can be auto-assigned to the network interface for eth0 when you launch an instance
    - one or more IPv6 addresses
    - one or more security groups
    - a MAC address
    - a source/destination check flag
    - a description
  - Network interfaces can be attached and detached from instances, however, you cannot detach a primary network interface.
- Route Tables
  - contains a set of rules, called *routes*, that are used to determine where network traffic is directed.
  - A subnet can only be associated with one route table at a time, but you can associate multiple subnets with the same route table.
  - You cannot delete the main route table, but you can replace the main route table with a custom table that you've created.
  - You must update the route table for any subnet that uses gateways or connections.
  - Uses the most specific route in your route table that matches the traffic to determine how to route the traffic (longest prefix match).
- Internet Gateways
  - Allows communication between instances in your VPC and the internet.
  - Imposes no availability risks or bandwidth constraints on your network traffic.
  - Provides a target in your VPC route tables for internet-routable traffic, and performs network address translation for instances that have been assigned public IPv4 addresses.
  - The following table provides an overview of whether your VPC automatically comes with the components required for internet access over IPv4 or IPv6.
  - To enable access to or from the Internet for instances in a VPC subnet, you must do the following:
    - Attach an Internet Gateway to your VPC
    - Ensure that your subnet's route table points to the Internet Gateway.
    - Ensure that instances in your subnet have a globally unique IP address (public IPv4 address, Elastic IP address, or IPv6 address).

- Ensure that your network access control and security group rules allow the relevant traffic to flow to and from your instance

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

- Egress-Only Internet Gateways
  - VPC component that allows outbound communication over IPv6 from instances in your VPC to the Internet, and prevents the Internet from initiating an IPv6 connection with your instances.
  - An egress-only Internet gateway is stateful.
  - You cannot associate a security group with an egress-only Internet gateway.

- You can use a network ACL to control the traffic to and from the subnet for which the egress-only Internet gateway routes traffic.

- NAT
  - Enable instances in a private subnet to connect to the internet or other AWS services, but prevent the internet from initiating connections with the instances.
  - NAT Gateways
    - You must specify the **public subnet** in which the NAT gateway should reside.
    - You must specify an **Elastic IP address** to associate with the NAT gateway when you create it.
    - Each NAT gateway is created in a specific Availability Zone and implemented with redundancy in that zone.
    - Deleting a NAT gateway disassociates its Elastic IP address, but does not release the address from your account.
    - A NAT gateway supports the following protocols: TCP, UDP, and ICMP.
    - You cannot associate a security group with a NAT gateway.
    - A NAT gateway can support up to 55,000 simultaneous connections to each unique destination.
    - A NAT gateway cannot send traffic over VPC endpoints, VPN connections, AWS Direct Connect, or VPC peering connections.
    - A NAT gateway uses ports 1024-65535. Make sure to enable these in the inbound rules of your network ACL.
    - NAT gateways do not support IPv6 traffic—use an outbound-only (egress-only) internet gateway instead.
  - NAT Instance vs NAT Gateways

Attribute	NAT gateway	NAT instance
Availabiity	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	Managed by AWS.	Managed by you.
Performance	Software is optimized for handling NAT traffic.	A generic Amazon Linux AMI that's configured to perform NAT.
Cost	Charged depending on the number of NAT gateways you use, duration of usage, and amount of data that you send through the NAT gateways.	Charged depending on the number of NAT instances that you use, duration of usage, and instance type and size.

Type and Size	Uniform offering: you don't need to decide on the type or size.	Choose a suitable instance type and size, according to your predicted workload.
Public IP Addresses	Choose the Elastic IP address to associate with a NAT gateway at creation.	Use an elastic IP address or a public IP address with a NAT instance. You can change the public IP address at any time by associating a new elastic IP address with the instance.
Private IP Addresses	Automatically selected from the subnet's IP address range when you create the gateway.	Assign a specific private IP address from the subnet's IP address range when you launch the instance.
Security Groups	Cannot be associated with a NAT gateway.	Associate with your NAT instance and the resources behind your NAT instance to control inbound and outbound traffic.
Network ACLs	Use a network ACL to control the traffic to and from the subnet in which your NAT gateway resides.	Use a network ACL to control the traffic to and from the subnet in which your NAT instance resides.
Flow Logs	Use flow logs to capture the traffic.	Use flow logs to capture the traffic.
Port Forwarding	Not supported.	Manually customize the configuration to support port forwarding.
Bastion Servers	Not supported.	Use as a bastion server.
Traffix 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 the resources behind the NAT instance to close the connection.
IP Fragmentation	Supports forwarding of IP fragmented packets for the UDP protocol. Does not support fragmentation for the TCP and ICMP protocols. Fragmented packets for these protocols will get dropped.	Supports reassembly of IP fragmented packets for the UDP, TCP, and ICMP protocols.

- DHCP Options Sets

    - **Dynamic Host Configuration Protocol (DHCP)** provides a standard for passing configuration information to hosts on a TCP/IP network.
    - You can assign your own domain name to your instances, and use up to four of your own DNS servers by specifying a special set of DHCP options to use with the VPC.
    - Creating a VPC automatically creates a set of DHCP options, which are domain-name-servers=AmazonProvidedDNS, and domain-name=domain-name-for-your-region, and associates them with the VPC.
    - After you create a set of DHCP options, you can't modify them. Create a new set and associate a different set of DHCP options with your VPC, or use no DHCP options at all.
- DNS
    - AWS provides instances launched in a default VPC with public and private DNS hostnames that correspond to the public IPv4 and private IPv4 addresses for the instance.
    - AWS provides instances launched in a non-default VPC with private DNS hostname and possibly a public DNS hostname, depending on the DNS attributes you specify for the VPC and if your instance has a public IPv4 address.
    - Set VPC attributes *enableDnsHostnames* and *enableDnsSupport* to true so that your instances receive a public DNS hostname and Amazon-provided DNS server can resolve Amazon-provided private DNS hostnames.
        - If you use custom DNS domain names defined in a private hosted zone in Route 53, the *enableDnsHostnames* and *enableDnsSupport* attributes must be set to true.
- VPC Peering
    - A networking connection between two VPCs that enables you to route traffic between them privately. Instances in either VPC can communicate with each other as if they are within the same network.
- Elastic IP Addresses
    - A **static, public IPv4 address**.
    - You can associate an Elastic IP address with any instance or network interface for any VPC in your account.
    - You can mask the failure of an instance by rapidly remapping the address to another instance in your VPC.
    - Your Elastic IP addresses remain associated with your AWS account until you explicitly release them.
    - AWS imposes a small hourly charge when EIPs aren't associated with a running instance, or when they are associated with a stopped instance or an unattached network interface.
    - You're limited to five Elastic IP addresses.
- VPC Endpoints
    - Privately connect your VPC to supported AWS services and VPC endpoint services powered by PrivateLink without requiring an internet gateway, NAT device, VPN connection, or AWS Direct Connect connection.
    - Endpoints are virtual devices.
    - Two Types

- **Interface Endpoints**
  - An elastic network interface with a private IP address that serves as an entry point for traffic destined to a supported service.
  - Can be accessed through AWS VPN connections or AWS Direct Connect connections, through intra-region VPC peering connections from Nitro instances, and through inter-region VPC peering connections from any type of instance.
  - For each interface endpoint, you can choose only one subnet per Availability Zone. Endpoints are supported within the same region only.
  - Interface endpoints do not support the use of endpoint policies.
  - An interface endpoint supports IPv4 TCP traffic only.

Gateway Endpoints

- A gateway that is a target for a specified route in your route table, used for traffic destined to a supported AWS service.
- You can create multiple endpoints in a single VPC, for example, to multiple services. You can also create multiple endpoints for a single service, and use different route tables to enforce different access policies from different subnets to the same service.
- You can modify the endpoint policy that's attached to your endpoint, and add or remove the route tables that are used by the endpoint.
- Endpoints are supported within the same region only. You cannot create an endpoint between a VPC and a service in a different region.
- Endpoints support IPv4 traffic only.
- You must enable DNS resolution in your VPC, or if you're using your own DNS server, ensure that DNS requests to the required service (such as S3) are resolved correctly to the IP addresses maintained by AWS.

You can create your own application in your VPC and configure it as an AWS PrivateLink-powered service (referred to as an *endpoint service*). You are the *service provider*, and the AWS principals that create connections to your service are *service consumers*.

VPN Connections

VPN connectivity option	Description
AWS managed VPN	You can create an IPsec VPN connection between your VPC and your remote network. On the AWS side of the VPN connection, a *virtual private gateway* provides two VPN endpoints (tunnels) for automatic failover. You

	configure your *customer gateway* on the remote side of the VPN connection.
AWS VPN CloudHub	If you have more than one remote network, you can create multiple AWS managed VPN connections via your virtual private gateway to enable communication between these networks.
Third party software VPN appliance	You can create a VPN connection to your remote network by using an Amazon EC2 instance in your VPC that's running a third party software VPN appliance. AWS does not provide or maintain third party software VPN appliances; however, you can choose from a range of products provided by partners and open source communities.
AWS Direct Connect	You can also use AWS Direct Connect to create a dedicated private connection from a remote network to your VPC. You can combine this connection with an AWS managed VPN connection to create an IPsec-encrypted connection.

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

Pricing

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

References:
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/

Amazon CloudFront

- A web service that speeds up distribution of your static and dynamic web content to your users. A Content Delivery Network (CDN) service.
- It delivers your content through a worldwide network of data centers called **edge locations**. When a user requests content that you're serving with CloudFront, the user is routed to the edge location that provides the lowest latency, so that content is delivered with the best possible performance.
    - If the content is already in the edge location with the lowest latency, CloudFront delivers it immediately.
    - If the content is not in that edge location, CloudFront retrieves it from an origin that you've defined
- **How CloudFront Delivers Content**
    - You specify **origin servers**, like an S3 bucket or your own HTTP server, from which CloudFront gets your files which will then be distributed from CloudFront edge locations all over the world.
    - Upload your files to your origin servers. Your files, also known as **objects**.
    - Create a **CloudFront distribution**, which tells CloudFront which origin servers to get your files from when users request the files through your web site or application. At the same time, you specify details such as whether you want CloudFront to log all requests and whether you want the distribution to be enabled as soon as it's created.
    - CloudFront assigns a domain name to your new distribution that you can see in the CloudFront console.
    - CloudFront sends your distribution's configuration (but not your content) to all of its **edge locations**—collections of servers in geographically dispersed data centers where CloudFront caches copies of your objects.
- CloudFront supports the **WebSocket protocol** as well as the **HTTP protocol** with the following HTTP methods:
    - GET

    - HEAD
    - POST
    - PUT
    - DELETE
    - OPTIONS
    - PATCH.
- Using **Lambda@Edge** with CloudFront enables a variety of ways to customize the content that CloudFront delivers. It can help you configure your CloudFront distribution to serve private content from your own custom origin, as an option to using signed URLs or signed cookies.(See AWS Compute Services Lambda Lambda@Edge)
- CloudFront also has **regional edge caches** that bring more of your content closer to your viewers, even when the content is not popular enough to stay at a CloudFront edge location, to help improve performance for that content.
- You can use a zone apex name on CloudFront
- CloudFront supports wildcard CNAME
- Different CloudFront Origins
    - **Using S3 buckets for your origin** - you place any objects that you want CloudFront to deliver in an S3 bucket.
    - **Using S3 buckets configured as website endpoints for your origin**
    - **Using 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.
- Objects are cached for 24 hours by default. You can invalidate files in CloudFront edge caches even before they expire.
- You can configure CloudFront to automatically compress files of certain types and serve the compressed files when viewer requests include *Accept-Encoding: gzip* in the request header.
- CloudFront can cache different versions of your content based on the values of query string parameters.
- CloudFront Distributions
    - You create a **CloudFront distribution** to tell CloudFront where you want content to be delivered from, and the details about how to track and manage content delivery.
    - You create a distribution and choose the configuration settings you want:
        - Your content origin—that is, the Amazon S3 bucket, MediaPackage channel, or HTTP server from which CloudFront gets the files to distribute. You can specify any combination of up to 25 S3 buckets, channels, and/or HTTP servers as your origins.

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

    - Choose the price class that corresponds with the maximum price that you want to pay for CloudFront service. By default, CloudFront serves your objects from edge locations in all CloudFront regions.
- **Performance and Availability**
    - CloudFront also allows you to set up multiple origins to enable redundancy with **Origin Failover**. To set up origin failover, you must have a distribution with at least two origins. Next, you create an origin group for your distribution that includes the two origins, setting one as the primary. Finally, you define a cache behavior in which you specify the origin group as your origin.
        - The two origins in the origin group can be any combination of the following: AWS origins, like Amazon S3 buckets or Amazon EC2 instances, or custom origins, like your own HTTP web server.
        - When you create the origin group, you configure CloudFront to failover to the second origin for GET, HEAD, and OPTIONS HTTP methods when the primary origin returns specific status codes that you configure.
    - CloudFront is optimized for both dynamic and static content, providing extensive flexibility for optimizing cache behavior, coupled with network-layer optimizations for latency and throughput.
- **Using HTTPS with CloudFront**
    - You can choose HTTPS settings both for communication between viewers and CloudFront, and between CloudFront and your origin.
    - If you want your viewers to use HTTPS and to use alternate domain names for your files, you need to choose one of the following options for how CloudFront serves HTTPS requests:
        - Use a dedicated IP address in each edge location
        - Use Server Name Indication (SNI)
- **Monitoring**
    - The billing report is a high-level view of all of the activity for the AWS services that you're using, including CloudFront.
    - The usage report is a summary of activity for a service such as CloudFront, aggregated by hour, day, or month. It also includes usage charts that provide a graphical representation of your CloudFront usage.
    - CloudFront console includes a variety of reports based on the data in CloudFront access logs:
        - CloudFront Cache Statistics Reports
        - CloudFront Popular Objects Report
        - CloudFront Top Referrers Report
        - CloudFront Usage Reports
        - CloudFront Viewers Reports
    - You can use AWS Config to record configuration changes for CloudFront distribution settings changes.
    - CloudFront integrates with Amazon CloudWatch metrics so that you can monitor your website or application.
    - Capture API requests with AWS CloudTrail. CloudFront is a global service. To view CloudFront requests in CloudTrail logs, you must update an existing trail to include global services.

- **Security**
  - CloudFront, AWS Shield, AWS WAF, and Route 53 work seamlessly together to create a flexible, layered security perimeter against multiple types of attacks including network and application layer DDoS attacks.
  - You can deliver your content, APIs or applications via SSL/TLS, and advanced SSL features are enabled automatically.
  - Through geo-restriction capability, you can prevent users in specific geographic locations from accessing content that you're distributing through CloudFront.
  - With the **Origin Access Control** feature, you can restrict access to an S3 bucket to only be accessible from CloudFront.
  - **Field-Level Encryption** is a feature of CloudFront that allows you to securely upload user-submitted data such as credit card numbers to your origin servers.
- **Pricing**
  - Charge for storage in an S3 bucket.
  - Charge for serving objects from edge locations.
  - Charge for submitting data to your origin.
    - Data Transfer Out
    - HTTP/HTTPS Requests
    - Invalidation Requests,
    - Dedicated IP Custom SSL certificates associated with a CloudFront distribution.
  - You also incur a surcharge for HTTPS requests, and an additional surcharge for requests that also have field-level encryption enabled.
- **Compliance**
  - CloudFront has been validated as being compliant with Payment Card Industry (PCI) Data Security Standard (DSS).
  - CloudFront is a HIPAA eligible service.
  - CloudFront is compliant with SOC measures.

References:
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 Direct Connect

- Using Direct Connect, data can now be delivered through a private network connection between AWS and your datacenter or corporate network.
- Direct Connect links your internal network to a Direct Connect location over a standard Ethernet fiber-optic cable. One end of the cable is connected to your router, the other to a Direct Connect router. With this connection, you can create *virtual interfaces* directly to public AWS services or to Amazon VPC.
- 1 Gbps, 10 Gbps, and 100 Gbps connections are available.
- Supports hosted connection capacities of 1, 2, 5 and 10 Gbps. 1, 2, 5 and 10 Gbps hosted connections will provide customers with higher capacities that were previously only available via dedicated connections.
- Amazon Direct Connect also supports AWS Transit Gateway, aside from configuring Site-to-Site VPN connections. With this feature, customers can connect thousands of Amazon VPCs in multiple AWS Regions to their on-premises networks using 1/2/5/10 Gbps AWS Direct Connect connections.
- **Beneficial Use Cases**
    - When transferring large data sets.
    - When developing and using applications that use real-time data feeds.
    - When building hybrid environments that satisfy regulatory requirements requiring the use of private connectivity.
- **Setting Up Methods**

Port speed	Method
1 Gbps or higher	Connect directly to an AWS device from your router at an AWS Direct Connect location.
1 Gbps or higher	Work with a partner in the AWS Partner Network or a network provider to connect a router from your data center, office, or colocation environment to an AWS Direct Connect location. The network provider does not have to be a member of the APN to connect you.

Less than 1 Gbps	Work with a partner in the AWS Partner Network who can create a hosted connection for you. Sign up for AWS and then follow the instructions to accept your hosted connection.

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

    - Prerequisites
        - Connection: The Direct Connect connection or link aggregation group for which you are creating the virtual interface.
        - Virtual interface name: A name for the virtual interface.
        - Virtual interface owner
        - (Private virtual interface only) Connection to
        - VLAN: A unique virtual local area network tag that's not already in use on your connection.
        - Address family: Whether the BGP peering session will be over IPv4 or IPv6.
        - Peer IP addresses: A virtual interface can support a BGP peering session for IPv4, IPv6, or one of each (dual-stack). You cannot create multiple BGP sessions for the same IP addressing family on the same virtual interface
        - BGP information: A public or private Border Gateway Protocol Autonomous System Number for your side of the BGP session, and an MD5 BGP authentication key.
        - (Public virtual interface only) Prefixes you want to advertise: Public IPv4 routes or IPv6 routes to advertise over BGP. You must advertise at least one prefix using BGP.
    - The maximum transmission unit (MTU) of a network connection is the size, in bytes, of the largest permissible packet that can be passed over the connection. The MTU of a virtual private interface can be either 1500 or 9001 (jumbo frames). The MTU of a transit virtual interface for VPC Transit Gateways associated with Direct Connect gateways can be either 1500 or 8500 (jumbo frames). A public virtual interface doesn't support jumbo frames.
    - Jumbo frames are supported on virtual private interfaces attached to a virtual private gateway or a Direct Connect gateway. Jumbo frames apply only to propagated routes from Direct Connect.
- **Link Aggregation Groups (LAG)**
    - A logical interface that uses the Link Aggregation Control Protocol to aggregate multiple connections at a single Direct Connect endpoint, allowing you to treat them as a single, managed connection.
    - All connections in the LAG must use the same bandwidth.
    - You can have a maximum of four connections in a LAG. Each connection in the LAG counts towards your overall connection limit for the Region.
    - All connections in the LAG must terminate at the same Direct Connect endpoint.
    - Can aggregate up to 4 Direct Connect ports into a single connection using LAG.
    - All connections in a LAG operate in Active/Active mode.
    - It will only be available for dedicated 1G and 10G connections.
- **Direct Connect Gateways**
    - Use a Direct Connect gateway to connect your Direct Connect connection over a private virtual interface to one or more VPCs in your account that are located in the same or different Regions.
    - It is a globally available resource.

    - Direct Connect gateway also enables you to connect between your on-premises networks and Amazon Virtual Private Cloud (Amazon VPC) in any commercial AWS Region except in China regions.
    - Prior to multi-account support, you could only associate Amazon VPCs with a Direct Connect gateway in the same AWS account. With the launch of multi-account support for Direct Connect gateway, you can associate up to 10 Amazon VPCs from multiple accounts with a Direct Connect gateway. The VPCs must be owned by AWS Accounts that belong to the same AWS payer account ID.
- **Security**
    - Use IAM for controlling access.
- **Monitoring**
    - You can optionally assign tags to your Direct Connect resources to categorize or manage them. A tag consists of a key and an optional value, both of which you define.
    - CloudTrail captures all API calls for AWS Direct Connect as events.
    - Set up CloudWatch alarms to monitor metrics.
- **Pricing**
    - You pay only for the network ports you use and the data you transfer over the connection.
    - Pricing is per port-hour consumed for each port type. Data transfer out over AWS Direct Connect is charged per GB. Data transfer IN is $0.00 per GB in all locations.

References:
https://docs.aws.amazon.com/directconnect/latest/UserGuide
https://aws.amazon.com/directconnect/features/
https://aws.amazon.com/directconnect/pricing/
https://aws.amazon.com/directconnect/faqs/

AWS Transit Gateway

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

Features:

- **Inter-region peering**
    - Transit Gateway leverages the AWS global network to allow customers to route traffic across AWS Regions.

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

Reference:
https://aws.amazon.com/transit-gateway/

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.

References:
https://docs.aws.amazon.com/organizations/latest/userguide/
https://aws.amazon.com/organizations/features/
https://aws.amazon.com/organizations/faqs/

AWS Control Tower

- A service for configuring and managing a multi-account AWS environment.

Concepts

- Landing zone
  - A multi-account environment that is well-architected and adheres to security and compliance best practices.
  - Each organization can have one landing zone.
  - A container that holds the following:
    - Organizational Units (OUs)
    - Accounts
    - Users
    - Other Resources
  - Structure of a landing zone:
    - Root – parent that contains all OUs.
    - Security OU – contains the shared accounts.
    - Sandbox OU – contain the registered accounts used by your users to carry out their AWS workloads.
    - IAM Identity Center directory – scope of permissions of each user.
    - IAM Identity Center users – identities that your users can use to perform AWS workloads.
- Guardrails
  - A high-level rule or policy that governs your AWS environment.
  - Applies to both OU and AWS accounts within the OU.
  - Guardrails are classified based on their behavior and guidance.
  - Behavior
    - Preventive
      - Prohibits actions that result in policy violations.
      - Implemented using AWS Organizations SCPs.
      - Status is either enforced or not enabled.
    - Detective
      - Detects noncompliance resources and provides alerts through the dashboard.

        - Implemented using AWS Config rules.
        - Status is either clear, in violation, or not enabled.
    - Guidance
        - Mandatory – always enforced.
        - Strongly recommended – enforce best practices.
        - Elective – track actions that are commonly restricted.

    - By default, mandatory guardrails are applied to top-level OUs.
    - The exception for guardrails is only for root or management accounts.
- Account Factory
    - Automates provisioning of new accounts.
    - It also helps you standardize the provisioning of new accounts by using pre-approved account configurations.
    - Shared accounts:
        - Management account – used for billing, provisioning of accounts, and managing OUs and guardrails.
        - Log Archive account – a repository for logs of API activities and resource configurations.
        - Audit account – a restricted account for security and compliance teams.
    - Member accounts are the accounts used by your users to perform AWS workloads.
    - You can also provision accounts using AWS Control Tower Account Factory for Terraform.
- Dashboard
    - Offers continuous oversight to the following:
        - Accounts across your enterprise.
        - Guardrails enabled for policy enforcement.
        - Guardrails enabled for continuous detection of policy non-conformance.
        - Non-compliant resources organized by accounts and OUs.

AWS Control Tower Networking Features

- AWS automatically creates an AWS-default VPC in every Region, even those not governed by AWS Control Tower, as part of the account creation process.
- The default VPC is not the same as the VPC created by AWS Control Tower for a provisioned account.
- You also have the option to remove AWS default VPCs in non-governed Regions.
- Each AWS Control Tower VPC has three Availability Zones.
- By default, an AZ has one public and two private subnets.
- Supports VPC-to-VPC peering for multiple VPCs.
- Region deny guardrail
    - Applies to a landing zone.
    - Blocks API calls to services in non-governed Regions.

- IAM users can still connect to an AWS default VPC in a Region where AWS Control Tower is not supported.
- If a guardrail is enabled, you will be unable to deploy resources in the denied Regions.

AWS Control Tower Monitoring

- A log archive account is dedicated to collecting all logs centrally.
- You can use AWS CloudTrail to capture the actions or events of AWS Control Tower.
- With CloudWatch Logs and CloudWatch Logs Insights, you can view and query AWS Control Tower lifecycle events.
- A lifecycle event is only recorded after a series of actions has been completed.
- The event log for each lifecycle event indicates whether the originating Control Tower action was successful or unsuccessful.
- Each lifecycle event is automatically recorded as a non-API AWS service event by AWS CloudTrail.
- Each lifecycle event is sent to Amazon EventBridge.

AWS Control Tower Pricing

- You are charged for AWS services that are configured to set up your landing zone and mandatory guardrails.
- You are charged by AWS Config for running ephemeral workloads as it records configuration changes related to the creation and deletion of temporary resources.

AWS CloudFormation

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

Features

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

- CloudFormation enables you to build custom extensions to your stack template using AWS Lambda.

CloudFormation vs Elastic Beanstalk

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

Concepts

- **Templates**
  - A JSON or YAML formatted text file.
  - CloudFormation uses these templates as blueprints for building your AWS resources.
- **Stacks**
  - Manage related resources as a single unit.
  - All the resources in a stack are defined by the stack's CloudFormation template.
- **Change Sets**
  - Before updating your stack and making changes to your resources, you can generate a change set, which is a summary of your proposed changes.
  - Change sets allow you to see how your changes might impact your running resources, especially for critical resources, before implementing them.
- With AWS CloudFormation and AWS CodePipeline, you can use continuous delivery to automatically build and test changes to your CloudFormation templates before promoting them to production stacks.
- *CloudFormation artifacts* can include a stack template file, a template configuration file, or both. AWS CodePipeline uses these artifacts to work with CloudFormation stacks and change sets.
  - **Stack Template File** - defines the resources that CloudFormation provisions and configures. You can use YAML or JSON-formatted templates.
  - **Template Configuration File** - a JSON-formatted text file that can specify template parameter values, a stack policy, and tags. Use these configuration files to specify parameter values or a stack policy for a stack.
- Through the AWS PrivateLink, you can use CloudFormation APIs inside of your Amazon VPC and route data between your VPC and CloudFormation entirely within the AWS network.

Stacks

- If a resource cannot be created, CloudFormation rolls the stack back and automatically deletes any resources that were created. If a resource cannot be deleted, any remaining resources are retained until the stack can be successfully deleted.
- Stack update methods
  - Direct update
  - Creating and executing change sets

- **Drift detection** enables you to detect whether a stack's actual configuration differs, or has drifted, from its expected configuration. Use CloudFormation to detect drift on an entire stack, or on individual resources within the stack.
    - A resource is considered to have drifted if any if its actual property values differ from the expected property values.
    - A stack is considered to have drifted if one or more of its resources have drifted.
- To share information between stacks, export a stack's output values. Other stacks that are in the same AWS account and region can import the exported values.
- You can nest stacks.

Templates

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

StackSets

- CloudFormation StackSets allow you to roll out CloudFormation stacks over multiple AWS accounts and in multiple Regions with just a couple of clicks. StackSets is commonly used together with AWS Organizations to centrally deploy and manage services in different accounts.
- Administrator and target accounts - An *administrator account* is the AWS account in which you create stack sets. A stack set is managed by signing in to the AWS administrator account in which it was created. A *target account* is the account into which you create, update, or delete one or more stacks in your stack set.
- In addition to the organization's management account, you can delegate other administrator accounts in your AWS Organization that can create and manage stack sets with service-managed permissions for the organization.
- Stack sets - A *stack set* lets you create stacks in AWS accounts across regions by using a single CloudFormation template. All the resources included in each stack are defined by the stack set's CloudFormation template. A stack set is a regional resource.
- Stack instances - A *stack instance* is a reference to a stack in a target account within a region. A stack instance can exist without a stack; for example, if the stack could not be created for some reason, the stack instance shows the reason for stack creation failure. A stack instance can be associated with only one stack set.
- Stack set operations - Create stack set, update stack set, delete stacks, and delete stack set.

- Tags - You can add tags during stack set creation and update operations by specifying key and value pairs.

Monitoring

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

Security

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

Pricing

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

References:
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 Service Catalog

- Allows you to create, manage, and distribute catalogs of approved products to end-users, who can then access the products they need in a personalized portal.
- Administrators can control which users have access to each product to enforce compliance with organizational business policies. Administrators can also set up adopted roles so that end users only require IAM access to AWS Service Catalog in order to deploy approved resources.
- This is a regional service.

Features

- Standardization of assets

- Self-service discovery and launch
- Fine-grain access control
- Extensibility and version control

Concepts

- Users
  - Catalog administrators – Manage a catalog of products, organizing them into portfolios and granting access to end users. Catalog administrators prepare AWS CloudFormation templates, configure constraints, and manage IAM roles that are assigned to products to provide for advanced resource management.
  - End users – Use AWS Service Catalog to launch products to which they have been granted access.
- Products
  - Can comprise one or more AWS resources, such as EC2 instances, storage volumes, databases, monitoring configurations, and networking components, or packaged AWS Marketplace products.
  - You create your products by importing AWS CloudFormation templates. The templates define the AWS resources required for the product, the relationships between resources, and the parameters for launching the product to configure security groups, create key pairs, and perform other customizations.
  - You can see the products that you are using and their health state in the AWS Service Catalog console.
- Portfolio
  - A collection of products, together with configuration information. Portfolios help manage product configuration, determine who can use specific products and how they can use them.
  - When you add a new version of a product to a portfolio, that version is automatically available to all current users of that portfolio.
  - You can also share your portfolios with other AWS accounts and allow the administrator of those accounts to distribute your portfolios with additional constraints.
  - When you add tags to your portfolio, the tags are applied to all instances of resources provisioned from products in the portfolio.
- Versioning
  - Service Catalog allows you to manage multiple versions of the products in your catalog.
  - A version can have one of three statuses:
    - Active - An active version appears in the version list and allows users to launch it.
    - Inactive - An inactive version is hidden from the version list. Existing provisioned products launched from this version will not be affected.
    - Deleted - If a version is deleted, it is removed from the version list. Deleting a version can't be undone.
- Access control

    - You apply AWS IAM permissions to control who can view and modify your products and portfolios.
    - By assigning an IAM role to each product, you can avoid giving users permissions to perform unapproved operations, and enable them to provision resources using the catalog.
- Constraints
    - You use constraints to apply limits to products for governance or cost control.
    - Types of constraints:
        - Template constraints restrict the configuration parameters that are available for the user when launching the product. Template constraints allow you to reuse generic AWS CloudFormation templates for products and apply restrictions to the templates on a per-product or per-portfolio basis.
        - Launch constraints allow you to specify a role for a product in a portfolio. This role is used to provision the resources at launch, so you can restrict user permissions without impacting users' ability to provision products from the catalog.
        - Notification constraints specify an Amazon SNS topic to receive notifications about stack events.
        - Tag update constraints allow administrators to allow or disallow end users to update tags on resources associated with an AWS Service Catalog provisioned product.
- Stack
    - Every AWS Service Catalog product is launched as an AWS CloudFormation stack.
    - You can use CloudFormation StackSets to launch Service Catalog products across multiple regions and accounts. You can specify the order in which products deploy sequentially within regions. Across accounts, products are deployed in parallel.

Security

- Service Catalog uses Amazon S3 buckets and Amazon DynamoDB databases that are encrypted at rest using Amazon-managed keys.
- Service Catalog uses TLS and client-side encryption of information in transit between the caller and AWS.
- Service Catalog integrates with AWS CloudTrail and Amazon SNS.

Pricing

- The AWS Service Catalog free tier includes 1,000 API calls per month.
- You are charged based on the number of API calls made to Service Catalog beyond the free tier.

References:
https://aws.amazon.com/servicecatalog/
https://docs.aws.amazon.com/servicecatalog/latest/adminguide/introduction.html
https://docs.aws.amazon.com/servicecatalog/latest/userguide/end-user-console.html

https://aws.amazon.com/servicecatalog/pricing/
https://aws.amazon.com/servicecatalog/faqs/

AWS Systems Manager

- Allows you to centralize operational data from multiple AWS services and automate tasks across your AWS resources.

Features

- Create logical groups of resources such as applications, different layers of an application stack, or production versus development environments.
- You can select a resource group and view its recent API activity, resource configuration changes, related notifications, operational alerts, software inventory, and patch compliance status.
- Collects information about your instances and the software installed on them.
- Allows you to safely automate common and repetitive IT operations and management tasks across AWS resources.
- Provides a browser-based interactive shell and CLI for managing Windows and Linux EC2 instances, without the need to open inbound ports, manage SSH keys, or use bastion hosts. Administrators can grant and revoke access to instances through a central location by using IAM policies.
- Helps ensure that your software is up-to-date and meets your compliance policies.
- Lets you schedule windows of time to run administrative and maintenance tasks across your instances.

SSM Agent is the tool that processes Systems Manager requests and configures your machine as specified in the request. SSM Agent must be installed on each instance you want to use with Systems Manager. On newer AMIs and instance types, SSM Agent is installed by default. On older versions, you must install it manually.

Capabilities

- **Automation**
  - Allows you to safely automate common and repetitive IT operations and management tasks across AWS resources
  - A **step** is defined as an initiated action performed in the Automation execution on a per-target basis. You can execute the entire Systems Manager automation document in one action or choose to execute one step at a time.
  - Concepts
    - **Automation document** - defines the Automation workflow.
    - **Automation action** - the Automation workflow includes one or more steps. Each step is associated with a particular action or plugin. The action determines the inputs, behavior, and outputs of the step.
    - **Automation queue** - if you attempt to run more than 25 Automations simultaneously, Systems Manager adds the additional executions to a queue and

displays a status of *Pending*. When an Automation reaches a terminal state, the first execution in the queue starts.
    - You can schedule Systems Manager automation document execution.

- **Resource Groups**
    - A collection of AWS resources that are all in the same AWS region, and that match criteria provided in a query.
    - Use Systems Manager tools such as *Automation* to simplify management tasks on your groups of resources. You can also use groups as the basis for viewing monitoring and configuration *insights* in Systems Manager.

- **Built-in Insights**
    - Show detailed information about a single, selected resource group.
    - Includes recent API calls through CloudTrail, recent configuration changes through Config, Instance software inventory listings, instance patch compliance views, and instance configuration compliance views.

- **Systems Manager Activation**
    - Enable hybrid and cross-cloud management. You can register any server, whether physical or virtual to be managed by Systems Manager.

- **Inventory Manager**
    - Automates the process of collecting software inventory from managed instances.
    - You specify the type of metadata to collect, the instances from where the metadata should be collected, and a schedule for metadata collection.

- **Configuration Compliance**
    - Scans your fleet of managed instances for patch compliance and configuration inconsistencies.
    - View compliance history and change tracking for Patch Manager patching data and State Manager associations by using AWS Config.
    - Customize Systems Manager Compliance to create your own compliance types.

- **Run Command**
    - Remotely and securely manage the configuration of your managed instances at scale.
    - **Managed Instances** - any EC2 instance or on-premises server or virtual machine in your hybrid environment that is configured for Systems Manager.

- **Session Manager**
    - Manage your EC2 instances through an interactive one-click browser-based shell or through the AWS CLI.
    - Makes it easy to comply with corporate policies that require controlled access to instances, strict security practices, and fully auditable logs with instance access details, while still providing end users with simple one-click cross-platform access to your Amazon EC2 instances.

- You can use AWS Systems Manager Session Manager to tunnel SSH (Secure Shell) and SCP (Secure Copy) traffic between a client and a server.

- **Distributor**
  - Lets you package your own software or find AWS-provided agent software packages to install on Systems Manager managed instances.
  - After you create a package in Distributor, which creates an Systems Manager document, you can install the package in one of the following ways.
    - One time by using Systems Manager Run Command.
    - On a schedule by using Systems Manager State Manager.

- **Patch Manager**
  - Automate the process of patching your managed instances.
  - Enables you to scan instances for missing patches and apply missing patches individually or to large groups of instances by using EC2 instance tags.
  - For security patches, Patch Manager uses *patch baselines* that include rules for auto-approving patches within days of their release, as well as a list of approved and rejected patches.
  - You can use AWS Systems Manager Patch Manager to select and apply Microsoft application patches automatically across your Amazon EC2 or on-premises instances.
  - AWS Systems Manager Patch Manager includes common vulnerability identifiers (CVE ID). CVE IDs can help you identify security vulnerabilities within your fleet and recommend patches.
  - You can configure actions to be performed on a managed instance before and after installing patches.

- **Incident Manager**
  - Provides an incident management console to manage and monitor all incidents relating to the AWS resources that your applications are using.
  - Primarily used to mitigate and recover from production incidents affecting their applications that are hosted in AWS.
  - Expedites incident resolution by notifying responders of impact, highlighting relevant troubleshooting data, and providing collaboration tools to return normal operations quickly.
  - Automates response plans and allows responder team escalation.

- **Compliance**
  - Automatically scans your fleet of managed nodes in AWS for patch compliance and configuration inconsistencies.
  - Collects and aggregates data from multiple AWS accounts and AWS Regions then drill down into specific resources that aren't compliant.
  - Displays compliance data about Patch Manager patching and State Manager associations.
  - Allows you to create your own compliance types based on your technical requirements.

- **Fleet Manager**
  - A unified user interface (UI) experience that helps you remotely manage your nodes/servers in AWS which allows you to view the health and performance status of your entire fleet from a single UI console.
  - Gathers data from individual devices, external servers and Amazon EC2 instances to perform common troubleshooting and management tasks straight from the console without manually connecting to the resource.
  - Enables you to view the directory and file contents of your nodes/instances, Windows registry management, operating system user management et cetera.

- **State Manager**
  - A service that automates the process of keeping your EC2 and hybrid infrastructure in a state that you define.
  - A *State Manager association* is a configuration that is assigned to your managed instances. The configuration defines the state that you want to maintain on your instances. The association also specifies actions to take when applying the configuration.

- **Parameter Store**
  - Provides secure, hierarchical storage for configuration data and secrets management.
  - You can store values as plain text or encrypted data with *SecureString*.
  - Parameters work with Systems Manager capabilities such as Run Command, State Manager, and Automation.

- **OpsCenter**
  - OpsCenter helps you view, investigate, and resolve operational issues related to your environment from a central location.
  - OpsCenter complements existing case management systems by enabling integrations via Amazon Simple Notification Service (SNS) and public AWS SDKs. By aggregating information from AWS Config, AWS CloudTrail logs, resource descriptions, and Amazon EventBridge (Amazon CloudWatch Events), OpsCenter helps you reduce the mean time to resolution (MTTR) of incidents, alarms, and operational tasks.

- **Change Manager**
  - An enterprise change management framework for requesting, approving, implementing, and reporting on operational changes to your application configuration and infrastructure.
  - From a single delegated administrator account, if you use AWS Organizations, you can manage changes across multiple AWS accounts and across AWS Regions. Alternatively, using a local account, you can manage changes for a single AWS account.
  - Can be used for both AWS and on-premises resources.

- For each change template, you can add up to five levels of approvers. When it's time to implement an approved change, Change Manager runs the Automation runbook that is specified in the associated change request.

- **Maintenance Window**
  - Set up recurring schedules for managed instances to execute administrative tasks like installing patches and updates without interrupting business-critical operations.
  - Supports running four types of tasks:
    - Systems Manager Run Command commands
    - Systems Manager Automation workflows
    - AWS Lambda functions
    - AWS Step Functions tasks

- **Systems Manager Document (SSM)**
  - Defines the actions that Systems Manager performs.
  - Types of SSM Documents

Type	Use with	Details
Command document	Run Command, State Manager	Run Command uses command documents to execute commands. State Manager uses command documents to apply a configuration. These actions can be run on one or more targets at any point during the lifecycle of an instance.
Policy document	State Manager	Policy documents enforce a policy on your targets. If the policy document is removed, the policy action no longer happens.
Automation document	Automation	Use automation documents when performing common maintenance and deployment tasks such as creating or updating an AMI.
Package document	Distributor	In Distributor, a package is represented by a Systems Manager document. A package document includes attached ZIP archive files that contain software or assets to install on managed instances. Creating a package in Distributor creates the package document.

- Can be in JSON or YAML.
- You can create and save different versions of documents. You can then specify a default version for each document.
- If you want to customize the steps and actions in a document, you can create your own.
- You can tag your documents to help you quickly identify one or more documents based on the tags you've assigned to them.

Monitoring

- SSM Agent writes information about executions, scheduled actions, errors, and health statuses to log files on each instance. For more efficient instance monitoring, you can configure either SSM Agent itself or the CloudWatch Agent to send this log data to CloudWatch Logs.
- Using CloudWatch Logs, you can monitor log data in real-time, search and filter log data by creating one or more metric filters, and archive and retrieve historical data when you need it.
- Log System Manager API calls with CloudTrail.

Security

- Systems Managers is linked directly to IAM for access controls.

Pricing

- For your own packages, you pay only for what you use. Upon transferring a package into Distributor, you will be charged based on the size and duration of storage for that package, the number of Get and Describe API calls made, and the amount of out-of-Region and on-premises data transfer out of Distributor for those packages.
- You are charged based on the number and type of Automation steps.

References:
https://docs.aws.amazon.com/systems-manager/latest/userguide
https://aws.amazon.com/systems-manager/features/
https://aws.amazon.com/systems-manager/pricing/
https://aws.amazon.com/systems-manager/faq/

AWS Config

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

Features

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

Concepts

Configuration History

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

Configuration item

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

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

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

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

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

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

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

- Config deletes data older than your specified retention period. The default period is 7 years.
- Multi-Account Multi-Region Data Aggregation
  - An aggregator collects configuration and compliance data from the following:
    - Multiple accounts and multiple regions.
    - Single account and multiple regions.
    - An organization in AWS Organizations and all the accounts in that organization.

Monitoring

- Use Amazon SNS to send you notifications every time a supported AWS resource is created, updated, or otherwise modified as a result of user API activity.
- Use Amazon EventBridge (Amazon CloudWatch Events) to detect and react to changes in the status of AWS Config events.
- Use AWS CloudTrail to capture API calls to Config.

Security

- Use IAM to create individual users for anyone who needs access to Config and grant different permissions to each IAM user.

Compliances

- ISO
- PCI DSS
- HIPAA
- FedRAMP

Pricing

- You are charged based on the number of configuration items recorded and on the number of AWS Config rules evaluations recorded, instead of the number of active rules in your account per region.. You are charged only once for recording the configuration item.

References:
https://docs.aws.amazon.com/config/latest/developerguide
https://aws.amazon.com/config/features/
https://aws.amazon.com/config/pricing/
https://aws.amazon.com/config/faq/

Amazon CloudWatch

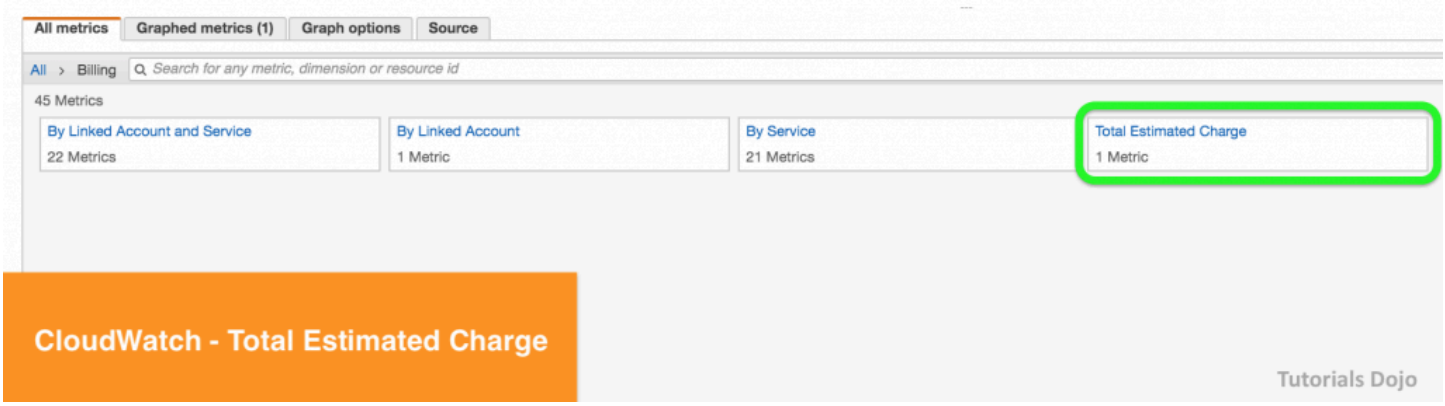
- Monitoring tool for your AWS resources and applications.
- Display metrics and create alarms that watch the metrics and send notifications or automatically make changes to the resources you are monitoring when a threshold is breached.
- CloudWatch is basically a metrics repository. An AWS service, 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.
        - **Metric math** enables you to query multiple CloudWatch metrics and use math expressions to create new time series based on these metrics.
        - **Important note for EC2 metrics:** CloudWatch does not collect memory utilization and disk space usage metrics right from the get go. You need to install CloudWatch Agent in your instances first to retrieve these metrics.
    - **Dimensions** - a name/value pair that uniquely identifies a metric.
        - You can assign up to 10 dimensions to a metric.
        - Whenever you add a unique dimension to one of your metrics, you are creating a new variation of that metric.
    - **Statistics** - metric data aggregations over specified periods of time.
        - Each statistic has a unit of measure. Metric data points that specify a unit of measure are aggregated separately.
        - You can specify a unit when you create a custom metric. If you do not specify a unit, CloudWatch uses *None* as the unit.

- A *period* is the length of time associated with a specific CloudWatch statistic. The default value is 60 seconds.
- CloudWatch aggregates statistics according to the period length that you specify when retrieving statistics.
- For large datasets, you can insert a pre-aggregated dataset called a *statistic set*.

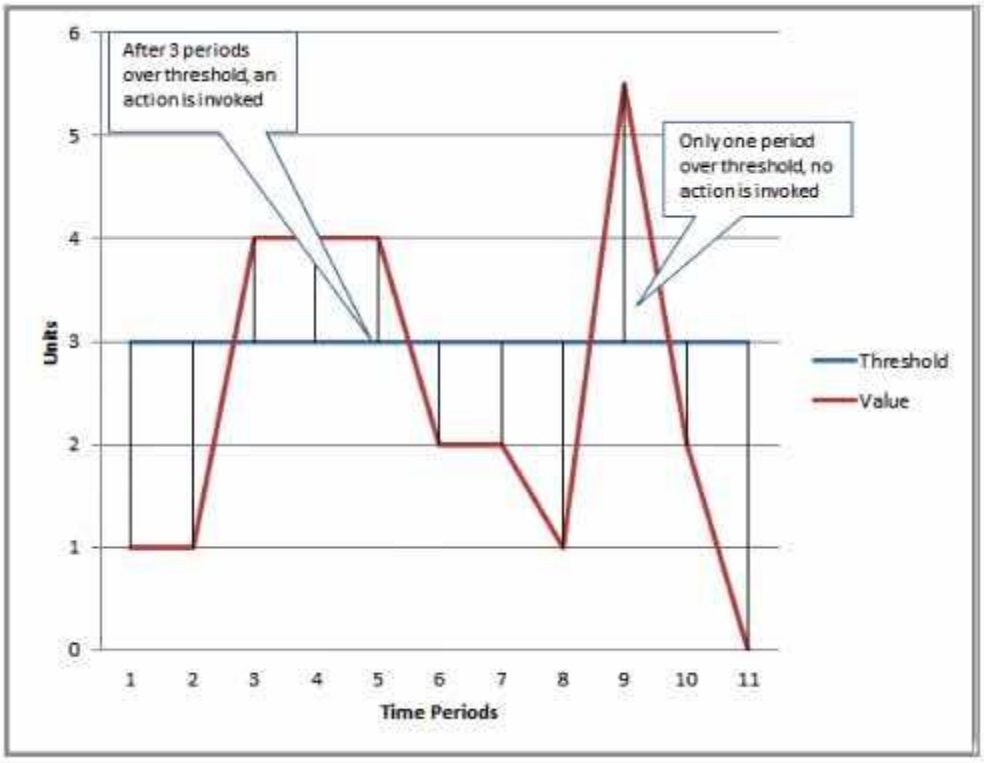
Statistic	Description
Minimum	The lowest value observed during the specified period. You can use this value to determine low volumes of activity for your application.
Maximum	The highest value observed during the specified period. You can use this value to determine high volumes of activity for your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

- **Percentiles** - indicates the relative standing of a value in a dataset. Percentiles help you get a better understanding of the distribution of your metric data.
- **Alarms** - watches a single metric over a specified time period, and performs one or more specified actions, based on the value of the metric relative to a threshold over time.

- You can create an alarm for monitoring CPU usage and load balancer latency, for managing instances, and for billing alarms.
- When an alarm is on a dashboard, it turns red when it is in the *ALARM* state.
- Alarms invoke actions for sustained state changes only.
- Alarm States
  - **OK**—The metric or expression is within the defined threshold.
  - **ALARM**—The metric or expression is outside of the defined threshold.
  - **INSUFFICIENT_DATA**—The alarm has just started, the metric is not available, or not enough data is available for the metric to determine the alarm state.
- 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.



- When you create an alarm, you specify three settings:
  - **Period** is the length of time to evaluate the metric or expression to create each individual data point for an alarm. It is expressed in seconds.
  - **Evaluation Period** is the number of the most recent periods, or data points, to evaluate when determining alarm state.
  - **Datapoints to Alarm** is the number of data points within the evaluation period that must be breaching to cause the alarm to go to the ALARM state. The breaching data points do not have to be consecutive, they just must all be within the last number of data points equal to **Evaluation Period**.



- For each alarm, you can specify CloudWatch to treat missing data points as any of the following:
  - *missing*—the alarm does not consider missing data points when evaluating whether to change state (default)
  - *notBreaching*—missing data points are treated as being within the threshold
  - *breaching*—missing data points are treated as breaching the threshold
  - *ignore*—the current alarm state is maintained
- You can now create tags in CloudWatch alarms that let you define policy controls for your AWS resources. This enables you to create resource level policies for your alarms.

CloudWatch Dashboard

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

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 Logs Insights** enables you to interactively search and analyze your log data in CloudWatch Logs using queries.
- **CloudWatch Vended logs** are logs that are natively published by AWS services on behalf of the customer. **VPC Flow logs** is the first Vended log type that will benefit from this tiered model.
- After the CloudWatch Logs agent begins publishing log data to Amazon CloudWatch, you can search and filter the log data by creating one or more metric filters. **Metric filters** define the terms and patterns to look for in log data as it is sent to CloudWatch Logs.
- Filters **do not** retroactively filter data. Filters only publish the metric data points for events that happen after the filter was created. Filtered results return the first 50 lines, which will not be displayed if the timestamp on the filtered results is earlier than the metric creation time.
- Metric Filter Concepts
  - filter pattern - you use the pattern to specify what to look for in the log file.
  - metric name - the name of the CloudWatch metric to which the monitored log information should be published.
  - metric namespace - the destination namespace of the new CloudWatch metric.
  - metric value - the numerical value to publish to the metric each time a matching log is found.
  - default value - the value reported to the metric filter during a period when no matching logs are found. By setting this to 0, you ensure that data is reported during every period.
- You can create two subscription filters with different filter patterns on a single log group.

CloudWatch Agent

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

Authentication and Access Control

- Use IAM users or roles for authenticating who can access
- Use Dashboard Permissions, IAM identity-based policies, and service-linked roles for managing access control.
- A *permissions policy* describes who has access to what.
  - Identity-Based Policies
  - Resource-Based Policies

- There are no CloudWatch Amazon Resource Names (ARNs) for you to use in an IAM policy. Use an * (asterisk) instead as the resource when writing a policy to control access to CloudWatch actions.

Pricing

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

References:
https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring
https://aws.amazon.com/cloudwatch/faqs/

AWS Lambda

- A serverless compute service.
- Lambda executes your code only when needed and scales automatically.
- Lambda functions are stateless - no affinity to the underlying infrastructure.
- You choose the amount of memory you want to allocate to your functions and AWS Lambda allocates proportional CPU power, network bandwidth, and disk I/O.
- AWS Lambda is SOC, HIPAA, PCI, ISO compliant.
- 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.
- **Layers** – Lambda layers are a distribution mechanism for libraries, custom runtimes, and other function dependencies. Layers let you manage your in-development function code independently from the unchanging code and resources that it uses.
- **Event source** – an AWS service or a custom service that triggers your function and executes its logic.
- **Downstream resources** – an AWS service that your Lambda function calls once it is triggered.
- **Log streams** – While Lambda automatically monitors your function invocations and reports metrics to CloudWatch, you can annotate your function code with custom logging statements that allow you to analyze the execution flow and performance of your Lambda function.
- AWS Serverless Application Model

Lambda Functions

- You upload your application code in the form of one or more *Lambda functions*. Lambda stores code in Amazon S3 and encrypts it at rest.
- To create a Lambda function, you first package your code and dependencies in a deployment package. Then, you upload the deployment package to create your Lambda function.
- After your Lambda function is in production, Lambda automatically monitors functions on your behalf, reporting metrics through Amazon CloudWatch.
- Configure **basic function settings** including the description, memory usage, execution timeout, and role that the function will use to execute your code.
- **Environment variables** are always encrypted at rest, and can be encrypted in transit as well.
- **Versions and aliases** are secondary resources that you can create to manage function deployment and invocation.
- A **layer** is a ZIP archive that contains libraries, a custom runtime, or other dependencies. Use layers to manage your function's dependencies independently and keep your deployment package small.
- You can configure a function to mount an Amazon EFS file system to a local directory. With Amazon EFS, your function code can access and modify shared resources securely and at high concurrency.

Invoking Functions

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

    - Amazon Simple Queue Service
- Your functions' *concurrency* is the number of instances that serve requests at a given time. When your function is invoked, Lambda allocates an instance of it to process the event. When the function code finishes running, it can handle another request. If the function is invoked again while a request is still being processed, another instance is allocated, which increases the function's concurrency.
- To ensure that a function can always reach a certain level of concurrency, you can configure the function with **reserved concurrency**. When a function has reserved concurrency, no other function can use that concurrency. Reserved concurrency also limits the maximum concurrency for the function.
- To enable your function to scale without fluctuations in latency, use **provisioned concurrency**. By allocating provisioned concurrency before an increase in invocations, you can ensure that all requests are served by initialized instances with very low latency.

Configuring a Lambda Function to Access Resources in a VPC

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

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

Lambda@Edge

- Lets you run Lambda functions to customize content that CloudFront delivers, executing the functions in AWS locations closer to the viewer. The functions run in response to CloudFront events, without provisioning or managing servers.
- You can use Lambda functions to change CloudFront requests and responses at the following points:
    - After CloudFront receives a request from a viewer (viewer request)
    - Before CloudFront forwards the request to the origin (origin request)
    - After CloudFront receives the response from the origin (origin response)
    - Before CloudFront forwards the response to the viewer (viewer response)
- You can automate your serverless application's release process using AWS CodePipeline and AWS CodeDeploy.
- Lambda will automatically track the behavior of your Lambda function invocations and provide feedback that you can monitor. In addition, it provides metrics that allows you to analyze the full function invocation spectrum, including event source integration and whether downstream resources perform as expected.

Pricing

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

References:
https://docs.aws.amazon.com/lambda/latest/dg
https://aws.amazon.com/lambda/faqs/

AWS Elastic Beanstalk

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

Environment Pages

- The **Configuration** page shows the resources provisioned for this environment. This page also lets you configure some of the provisioned resources.
- The **Health** page shows the status and detailed health information about the EC2 instances running your application.
- The **Monitoring** page shows the statistics for the environment, such as average latency and CPU utilization. You also use this page to create alarms for the metrics that you are monitoring.

- The **Events** page shows any informational or error messages from services that this environment is using.
- The **Tags** page shows tags — key-value pairs that are applied to resources in the environment. You use this page to manage your environment's tags.

Elastic Beanstalk Concepts

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

Environment Types

- Load-balancing, Autoscaling Environment - automatically starts additional instances to accommodate increasing load on your application.
- Single-Instance Environment - contains one Amazon EC2 instance with an Elastic IP address.

Environment Configurations

- Your environment contains:
    - Your **EC2 virtual machines** configured to run web apps on the platform that you choose.
    - An **Auto Scaling group** that ensures that there is always one instance running in a single-instance environment, and allows configuration of the group with a range of instances to run in a load-balanced environment.
    - When you enable load balancing, Elastic Beanstalk creates an **Elastic Load Balancing load balancer** to distributes traffic among your environment's instances.

- Elastic Beanstalk provides integration with **Amazon RDS** to help you add a database instance to your Elastic Beanstalk environment : **MySQL, PostgreSQL, Oracle, or SQL Server**. When you add a database instance to your environment, Elastic Beanstalk provides connection information to your application by setting environment properties for the database hostname, port, user name, password, and database name.
- You can use **environment properties** to pass secrets, endpoints, debug settings, and other information to your application. Environment properties help you run your application in multiple environments for different purposes, such as development, testing, staging, and production.
- You can configure your environment to use **Amazon SNS** to notify you of important events that affect your application.
- Your environment is available to users at a **subdomain of elasticbeanstalk.com**. When you create an environment, you can choose a unique subdomain that represents your application.

Monitoring

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

Security

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

Pricing

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

References:
https://docs.aws.amazon.com/elasticbeanstalk/latest/dg
https://aws.amazon.com/elasticbeanstalk/faqs/

AWS Storage Gateway

- The service enables **hybrid storage** between on-premises environments and the AWS Cloud.
- It integrates on-premises enterprise applications and workflows with Amazon's block and object cloud storage services through industry standard storage protocols.
- The service stores files as native S3 objects, archives virtual tapes in Amazon Glacier, and stores EBS Snapshots generated by the Volume Gateway with Amazon EBS.

  Storage Solutions

  - **File Gateway** - supports a file interface into S3 and combines a service and a virtual software appliance.
    - The software appliance, or gateway, is deployed into your on-premises environment as a virtual machine running on VMware ESXi or Microsoft Hyper-V hypervisor.
    - File gateway supports
      - S3 Standard
      - S3 Standard - Infrequent Access
      - S3 One Zone - IA
    - With a file gateway, you can do the following:
      - You can store and retrieve files directly using the NFS version 3 or 4.1 protocol.
      - You can store and retrieve files directly using the SMB file system version, 2 and 3 protocol.
      - You can access your data directly in S3 from any AWS Cloud application or service.
      - You can manage your S3 data using lifecycle policies, cross-region replication, and versioning.
    - File Gateway now supports Amazon S3 Object Lock, enabling write-once-read-many (WORM) file-based systems to store and access objects in Amazon S3.
    - Any modifications such as file edits, deletes or renames from the gateway's NFS or SMB clients are stored as new versions of the object, without overwriting or deleting previous versions.
  - **Volume Gateway** - provides cloud-backed storage volumes that you can mount as iSCSI devices from your on-premises application servers.

        - **Cached volumes** – you store your data in S3 and retain a copy of frequently accessed data subsets locally. Cached volumes can range from 1 GiB to 32 TiB in size and must be rounded to the nearest GiB. Each gateway configured for cached volumes can support up to 32 volumes.
        - **Stored volumes** – if you need low-latency access to your entire dataset, first configure your on-premises gateway to store all your data locally. Then asynchronously back up point-in-time snapshots of this data to S3. Stored volumes can range from 1 GiB to 16 TiB in size and must be rounded to the nearest GiB. Each gateway configured for stored volumes can support up to 32 volumes.
        - AWS Storage Gateway customers using the Volume Gateway configuration for block storage can detach and attach volumes, from and to a Volume Gateway. You can use this feature to migrate volumes between gateways to refresh underlying server hardware, switch between virtual machine types, and move volumes to better host platforms or newer Amazon EC2 instances.
    - **Tape Gateway** - archive backup data in Amazon Glacier.
        - Has a virtual tape library (VTL) interface to store data on virtual tape cartridges that you create.
        - Deploy your gateway on an EC2 instance to provision iSCSI storage volumes in AWS.
        - The AWS Storage Gateway service integrates Tape Gateway with Amazon S3 Glacier Deep Archive storage class, allowing you to store virtual tapes in the lowest-cost Amazon S3 storage class.
        - Tape Gateway also has the capability to move your virtual tapes archived in Amazon S3 Glacier Flexible Retrieval to Amazon S3 Glacier Deep Archive storage class, enabling you to further reduce the monthly cost to store long-term data in the cloud by up to 75%.

Storage Gateway Hosting Options

- As a VM containing the Storage Gateway software, run on VMware ESXi, Microsoft Hyper-V on premises
- As a VM in VMware Cloud on AWS
- As a hardware appliance on premises
- As an AMI in an EC2 instance

Storage Gateway stores volume, snapshot, tape, and file data in the AWS Region in which your gateway is activated. File data is stored in the AWS Region where your S3 bucket is located.

The local gateway appliance maintains a cache of recently written or read data so your applications can have low-latency access to data that is stored durably in AWS. The gateways use a **read-through and write-back** cache.

File Gateway File Share

- You can create an NFS or SMB file share using the AWS Management Console or service API.

- After your file gateway is activated and running, you can add additional file shares and grant access to S3 buckets.
- You can use a file share to access objects in an S3 bucket that belongs to a different AWS account.
- The AWS Storage Gateway service added support for Access Control Lists (ACLs) to Server Message Block (SMB) shares on the File Gateway, helping enforce data security standards when using the gateway for storing and accessing data in Amazon Simple Storage Service (S3).
- After your file gateway is activated and running, you can add additional file shares and grant access to S3 buckets.

Security

- You can use AWS KMS to encrypt data written to a virtual tape.
- Storage Gateway uses Challenge-Handshake Authentication Protocol (CHAP) to authenticate iSCSI and initiator connections. CHAP provides protection against playback attacks by requiring authentication to access storage volume targets.
- Authentication and access control with IAM.

Compliance

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

Pricing

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

References:
https://docs.aws.amazon.com/storagegateway/latest/userguide/
https://aws.amazon.com/storagegateway/faqs/

Amazon Elasticache

- ElastiCache is a distributed **in-memory cache** environment in the AWS Cloud.
- ElastiCache works with both the **Redis** and **Memcached** engines.

Components

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

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

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

- You can vertically scale up or scale down your sharded Redis Cluster on demand. Amazon ElastiCache resizes your cluster by changing the node type, while the cluster continues to stay online and serve incoming requests.
- You can set up automatic snapshots or initiate manual backups, and then seed new ElastiCache for Redis clusters. You can also export your snapshots to an S3 bucket of your choice for disaster recovery, analysis or cross-region backup and restore.
- Endpoints
  - **Single Node Redis** (**cluster mode disabled**) Endpoints - used to connect to the cluster for both reads and writes.
  - **Multi-Node Redis** (**cluster mode disabled**) Endpoints - use the primary endpoint for all writes to the cluster. The read endpoint points to your read replicas.
  - **Redis** (**cluster mode enabled**) Endpoints - has a single configuration endpoint. By connecting to the configuration endpoint, your application is able to discover the primary and read endpoints for each shard in the cluster.
- Parameter Groups
  - **Cache parameter group** is a named collection of engine-specific parameters that you can apply to a cluster.
  - Parameters are used to control memory usage, eviction policies, item sizes, and more.
- Redis Security
  - ElastiCache for Redis node access is restricted to applications running on whitelisted EC2 instances. You can control access of your cluster by using subnet groups or security groups. By default, network access to your clusters is turned off.
  - By default, all new ElastiCache for Redis clusters are launched in a VPC environment. Use subnet groups to grant cluster access from Amazon EC2 instances running on specific subnets.
  - ElastiCache for Redis supports TLS and in-place encryption for nodes running specified versions of the ElastiCache for Redis engine.
  - You can use your own AWS KMS Keys to encrypt data at rest in ElastiCache for Redis.
- Redis Backups
  - A point-in-time copy of a Redis cluster.
  - Backups consist of all the data in a cluster plus some metadata.
- Global Datastore
  - A new feature that provides fully managed, secure cross-region replication. You can now write to your ElastiCache for Redis cluster in one region and have the data available for reading in two other cross-region replica clusters.
  - In the unlikely event of regional degradation, one of the healthy cross-region replica clusters can be promoted to become the primary cluster with full read/write capabilities.

ElastiCache Memcached

- Features

    - Automatic detection and recovery from cache node failures.
    - Automatic discovery of nodes within a cluster enabled for automatic discovery, so that no changes need to be made to your application when you add or remove nodes.
    - Flexible Availability Zone placement of nodes and clusters.
    - **ElastiCache Auto Discovery** feature for Memcached lets your applications identify all of the nodes in a cache cluster and connect to them.
    - ElastiCache node access is restricted to applications running on whitelisted EC2 instances. You can control the instances that can access your cluster by using subnet groups or security groups.
    - It is not persistent.
    - Supports large nodes with multiple cores or threads.
    - Does not support multi-AZ failover or replication
    - Does not support snapshots
- Components
    - **Memcached cluster** - a logical grouping of one or more ElastiCache Nodes. Data is partitioned across the nodes in a Memcached cluster.
        - Memcached supports up to 100 nodes per customer for each Region with each cluster having 1–20 nodes.
        - When you partition your data, use *consistent hashing*.
    - **Endpoint** - the unique address your application uses to connect to an ElastiCache node or cluster.
        - Each node in a Memcached cluster has its own endpoint.
        - The cluster also has an endpoint called the *configuration endpoint*.
    - **ElastiCache parameter group** - a named collection of engine-specific parameters that you can apply to a cluster. Parameters are used to control memory usage, eviction policies, item sizes, and more.
    - ElastiCache allows you to control access to your clusters using **security groups**. By default, network access to your clusters is turned off.
    - A **subnet group** is a collection of subnets that you can designate for your clusters running in a VPC environment. If you create a cluster in a VPC, then you must specify a *cache subnet group*. ElastiCache uses that cache subnet group to choose a subnet and IP addresses within that subnet to associate with your cache nodes.
- Mitigating Failures
    - Node Failures
        - Spread your cached data over more nodes. Because Memcached does not support replication, a node failure will always result in some data loss from your cluster.
    - Availability Zone Failure
        - Locate your nodes in as many Availability Zones as possible. In the unlikely event of an AZ failure, you will lose the data cached in that AZ, not the data cached in the other AZs.

- ElastiCache uses DNS entries to allow client applications to locate servers (nodes). The DNS name for a node remains constant, but the IP address of a node can change over time.

Caching Strategies

- **Lazy Loading** - a caching strategy that loads data into the cache only when necessary.
  - Only requested data is cached.
  - Node failures are not fatal.
  - There is a cache miss penalty.
  - Stale data.
- **Write Through** - adds data or updates data in the cache whenever data is written to the database.
  - Data in the cache is never stale.
  - Write penalty vs. Read penalty. Every write involves two trips: A write to the cache and a write to the database.
  - Missing data.
  - Cache churn.
- By adding a time to live (TTL) value to each write, we are able to enjoy the advantages of each strategy and largely avoid cluttering up the cache with superfluous data.

Scaling ElastiCache for Memcached Clusters

- Scaling Memcached Horizontally
  - The Memcached engine supports partitioning your data across multiple nodes. Because of this, Memcached clusters scale horizontally easily. A Memcached cluster can have 1 to 20 nodes. To horizontally scale your Memcached cluster, just add or remove nodes.
- Scaling Memcached Vertically
  - When you scale your Memcached cluster up or down, you must create a new cluster. Memcached clusters always start out empty unless your application populates it.

Monitoring

- The service continuously monitors the health of your instances. In case a node experiences failure or a prolonged degradation in performance, ElastiCache will automatically restart the node and associated processes.
- ElastiCache provides both host-level metrics and metrics that are specific to the cache engine software. These metrics are measured and published for each Cache node in 60-second intervals.
- Monitor events with **ElastiCache Events**. When significant events happen on a cache cluster, including failure to add a node, success in adding a node, the modification of a security group, and others, ElastiCache sends a notification to a specific SNS topic.
- Monitor costs with tags.

Redis VS Memcached

- Memcached is designed for **simplicity** while Redis offers a **rich set of features** that make it effective for a wide range of use cases.

	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	No
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	• You need to partition your data across two to 250 or 500 nodes if the Redis engine version is 5.0.6 or higher.(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.

References:
https://aws.amazon.com/elasticache/redis-details/
https://aws.amazon.com/elasticache/redis-vs-memcached/
https://aws.amazon.com/elasticache/features/

Amazon DynamoDB

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

Core Components

- **Tables** - a collection of items

    - DynamoDB stores data in a table, which is a collection of data.
    - Are schemaless.
    - There is an initial limit of 256 tables per region.
- **Items** - a collection of attributes
    - DynamoDB uses **primary keys** to uniquely identify each item in a table and **secondary indexes** to provide more querying flexibility.
    - Each table contains zero or more items.
- **Attributes** - a fundamental data element
    - DynamoDB supports nested attributes up to 32 levels deep.
- **Primary Key** - uniquely identifies each item in the table, so that no two items can have the same key. Must be scalar.
    - **Partition key** - a simple primary key, composed of one attribute.
    - **Partition key and sort key** (*composite primary key*) - composed of two attributes.
    - DynamoDB uses the partition key value as input to an internal hash function. The output from the hash function determines the partition in which the item will be stored. All items with the same partition key are stored together, in sorted order by sort key value. If no sort key is used, no two items can have the same partition key value.
- **Secondary Indexes** - lets you query the data in the table using an alternate key, in addition to queries against the primary key.
    - You can create one or more secondary indexes on a table.
    - Two kinds of indexes:
        - **Global secondary index** – An index with a partition key and sort key that can be different from those on the table.
        - **Local secondary index** – An index that has the same partition key as the table, but a different sort key.
    - You can define up to 20 global secondary indexes and 5 local secondary indexes per table.
- **DynamoDB Streams** - an optional feature that captures data modification events in DynamoDB tables.
    - The naming convention for DynamoDB Streams endpoints is *streams.dynamodb..amazonaws.com*
    - Each event is represented by a *stream record*, and captures the following events:
        - A new item is added to the table: captures an image of the entire item, including all of its attributes.
        - An item is updated: captures the "before" and "after" image of any attributes that were modified in the item.
        - An item is deleted from the table: captures an image of the entire item before it was deleted.
    - Each stream record also contains the name of the table, the event timestamp, and other metadata.
    - Stream records are organized into groups, or ***shards***. Each shard acts as a container for multiple stream records, and contains information required for accessing and iterating through these records.

- Stream records have a lifetime of 24 hours; after that, they are automatically removed from the stream.
- You can use DynamoDB Streams together with AWS Lambda to create a *trigger*, which is a code that executes automatically whenever an event of interest appears in a stream.
- DynamoDB Streams enables powerful solutions such as data replication within and across Regions, materialized views of data in DynamoDB tables, data analysis using Kinesis materialized views, and much more.

Data Types for Attributes

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

Other Notes:

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

Throughput Management

- Provisioned throughput - manually defined maximum amount of capacity that an application can consume from a table or index. If your application exceeds your provisioned throughput settings, it is subject to request throttling. Free tier eligible.
    - DynamoDB auto scaling
        - Define a range (upper and lower limits) for **read and write capacity units**, and define a target utilization percentage within that range.
        - A table or a global secondary index can increase its **provisioned read and write capacity** to handle sudden increases in traffic, without request throttling.
        - DynamoDB auto scaling can decrease the throughput when the workload decreases so that you don't pay for unused provisioned capacity.
    - Reserved capacity - with reserved capacity, you pay a one-time upfront fee and commit to a minimum usage level over a period of time, for cost-saving solutions.
- Amazon DynamoDB on-demand is a flexible capacity mode for DynamoDB capable of serving thousands of requests per second without capacity planning. When you choose on-demand capacity mode, DynamoDB instantly accommodates your workloads as they ramp up or down to any previously reached traffic level. If a workload's traffic level hits a new peak, DynamoDB adapts rapidly to accommodate the workload. DynamoDB on-demand offers simple pay-per-request pricing for read and write requests so that you only pay for what you use, making it easy to balance costs and performance.

Capacity Unit Consumption

- CUC for Reads - strongly consistent read request consumes one read capacity unit, while an eventually consistent read request consumes 0.5 of a read capacity unit.
    - GetItem - reads a single item from a table.
    - BatchGetItem - reads up to 100 items, from one or more tables.
    - Query - reads multiple items that have the same partition key value.
    - Scan - reads all of the items in a table
- CUC for Writes
    - PutItem - writes a single item to a table.
    - UpdateItem - modifies a single item in the table.
    - DeleteItem - removes a single item from a table.
    - BatchWriteItem - writes up to 25 items to one or more tables.
    - Calculating the Required Read and Write Capacity Unit for Your DynamoDB table: https://tutorialsdojo.com/calculating-the-required-read-and-write-capacity-unit-for-your-dynamodb-table/

DynamoDB Auto Scaling

- When you use the AWS Management Console to create a new table, DynamoDB auto scaling is enabled for that table by default.

- Uses the AWS Application Auto Scaling service to dynamically adjust provisioned throughput capacity on your behalf, in response to actual traffic patterns.
- You create a *scaling policy* for a table or a global secondary index. The scaling policy specifies whether you want to scale read capacity or write capacity (or both), and the minimum and maximum provisioned capacity unit settings for the table or index. The scaling policy also contains a *target utilization*, which is the percentage of consumed provisioned throughput at a point in time.
- DynamoDB auto scaling doesn't prevent you from manually modifying provisioned throughput settings.
- If you enable DynamoDB auto scaling for a table that has one or more global secondary indexes, AWS highly recommends that you also apply auto scaling uniformly to those indexes.

Tagging

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

DynamoDB Items

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

Time To Live (TTL)

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

DynamoDB Queries

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

DynamoDB Scans

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

On-Demand Backup and Restore

- You can use IAM to restrict DynamoDB backup and restore actions for some resources.

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

DynamoDB Transactions

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

Global Tables

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

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

Security

- Encryption
    - Encrypts your data at rest using an AWS Key Management Service (AWS KMS) managed encryption key for DynamoDB.
    - Encryption at rest can be enabled only when you are creating a new DynamoDB table.
    - After encryption at rest is enabled, it can't be disabled.
    - Uses AES-256 encryption.
    - The following are encrypted:
        - DynamoDB base tables
        - Local secondary indexes
        - Global secondary indexes
    - Authentication and Access Control
        - Access to DynamoDB requires credentials.
        - Aside from valid credentials, you also need to have permissions to create or access DynamoDB resources.
        - Types of Identities
            - **AWS account root user**
            - **IAM user**
            - **IAM role**

- You can create indexes and streams only in the context of an existing DynamoDB table, referred to as *subresources*.
- Resources and subresources have unique Amazon Resource Names (**ARNs**) associated with them.
- A *permissions policy* describes who has access to what.
  - Identity-based Policies
    - Attach a permissions policy to a user or a group in your account
    - Attach a permissions policy to a role (grant cross-account permissions)
  - Policy Elements
    - Resource - use an ARN to identify the resource that the policy applies to.
    - Action - use action keywords to identify resource operations that you want to allow or deny.
    - Effect - specify the effect, either allow or deny, when the user requests the specific action.
    - Principal - the user that the policy is attached to is the implicit principal.
- Web Identity Federation - Customers can sign in to an identity provider and then obtain temporary security credentials from AWS Security Token Service (AWS STS).

Monitoring

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

DynamoDB Accelerator (DAX)

- DAX is a fully managed, highly available, in-memory cache for DynamoDB.
- **DynamoDB Accelerator (DAX)** delivers microsecond response times for accessing eventually consistent data.
- It requires only minimal functional changes to use DAX with an existing application since it is API-compatible with DynamoDB.
- For read-heavy or bursty workloads, DAX provides increased throughput and potential cost savings by reducing the need to overprovision read capacity units.

- DAX lets you scale on-demand.
- DAX is fully managed. You no longer need to do hardware or software provisioning, setup and configuration, software patching, operating a reliable, distributed cache cluster, or replicating data over multiple instances as you scale.
- DAX is not recommended if you need strongly consistent reads
- DAX is useful for read-intensive workloads, but not write-intensive ones.
- DAX supports server-side encryption but not TLS.
- Use Cases
    - Applications that require the fastest possible response time for reads.
    - Applications that read a small number of items more frequently than others. For example, limited-time on-sale items in an ecommerce store.
    - Applications that are read-intensive, but are also cost-sensitive. Offload read activity to a DAX cluster and reduce the number of read capacity units that you need to purchase for your DynamoDB tables.
    - Applications that require repeated reads against a large set of data. This will avoid eating up all your DynamoDB resources which are needed by other applications.
- To achieve high availability for your application, provision your DAX cluster with at least three nodes, then place the nodes in multiple Availability Zones within a Region.
- There are two options available for scaling a DAX cluster:
    - **Horizontal scaling**, where you add read replicas to the cluster. A single DAX cluster supports up to 10 read replicas, and you can add or remove replicas while the cluster is running.
    - **Vertical scaling**, where you select different node types. Larger nodes enable the cluster to store more data in memory, reducing cache misses and improving overall application performance. You can't modify the node types on a running DAX cluster. Instead, you must create a new cluster with the desired node type.

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.

- Design and Use Partition Keys Effectively
    - DynamoDB provides some flexibility in your per-partition throughput provisioning by providing **burst capacity**.

    - To better accommodate uneven access patterns, **DynamoDB adaptive capacity** enables your application to continue reading and writing to 'hot' partitions without being throttled, by automatically increasing throughput capacity for partitions that receive more traffic.
    - Amazon DynamoDB now applies adaptive capacity in real time in response to changing application traffic patterns, which helps you maintain uninterrupted performance indefinitely, even for imbalanced workloads. In addition, instant adaptive capacity helps you provision read and write throughput more efficiently instead of overprovisioning to accommodate uneven data access patterns. Instant adaptive capacity is on by default at no additional cost for all DynamoDB tables and global secondary indexes.
    - The optimal usage of a table's provisioned throughput depends not only on the workload patterns of individual items, but also on the partition-key design. In general, you will use your provisioned throughput more efficiently as the ratio of partition key values accessed to the total number of partition key values increases.
    - Structure the primary key elements to avoid one heavily requested partition key value that slows overall performance.
    - Distribute loads more evenly across a partition key space by adding a random number to the end of the partition key values. Then you randomize the writes across the larger space.
    - A randomizing strategy can greatly improve write throughput, but it's difficult to read a specific item because you don't know which suffix value was used when writing the item. Instead of using a random number to distribute the items among partitions, use a number that you can calculate based upon something that you want to query on.
    - Distribute write activity efficiently during data upload by using the sort key to load items from each partition key value, keeping more DynamoDB servers busy simultaneously and improving your throughput performance.
- Use Sort Keys to Organize Data
    - Well-designed sort keys gather related information together in one place where it can be queried efficiently.
    - Composite sort keys let you define hierarchical (one-to-many) relationships in your data that you can query at any level of the hierarchy.
- Use indexes efficiently by keeping the number of indexes to a minimum and avoid indexing tables that experience heavy write activity.
- Choose Projections Carefully.
- Optimize Frequent Queries to Avoid Fetches.
- Be Aware of Item-Collection Size Limits When Creating Local Secondary Indexes.
- For Querying and Scanning Data
    - Performance considerations for scans
    - Avoiding sudden spikes in read activity
    - Taking advantage of parallel scans

Pricing

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

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

AWS Fargate

- A serverless compute engine for containers that works with both Amazon Elastic Container Service (ECS) and Amazon Elastic Kubernetes Service (EKS).
- With Fargate, no manual provisioning, patching, cluster capacity management, or any infrastructure management required.
- **Use Case**
    - Launching containers without having to provision or manage EC2 instances.
    - If you want a managed service for container cluster management.
- **Configurations**
    - Amazon ECS task definitions for Fargate require that you specify CPU and memory at the task level (task definition).
    - Amazon ECS task definitions for Fargate support the limits parameter to define the resource limits to set for a container.
    - Amazon ECS task definitions for Fargate support the awslogs, splunk, firelens, and fluentd log drivers for the log configuration.
    - When provisioned, each Fargate task receives the following storage:
        - 10 GB of Docker layer storage
        - An additional 4 GB for volume mounts.
    - Task storage is ephemeral.
    - If you have a service with running tasks and want to update their platform version, you can update your service, specify a new platform version, and choose Force new deployment. Your tasks are redeployed with the **latest** platform version.
    - If your service is scaled up without updating the platform version, those tasks receive the platform version that was specified on the service's current deployment.
- **Network**

    - Amazon ECS task definitions for Fargate require that the network mode is set to awsvpc. The awsvpc network mode provides each task with its own elastic network interface.
- **Compliance**
    - PCI DSS Level 1, ISO 9001, ISO 27001, ISO 27017, ISO 27018, SOC 1, SOC 2, SOC 3, and HIPAA
    - AWS Fargate is not yet available in AWS GovCloud.
- **Pricing**
    - You pay for the amount of vCPU and memory resources consumed by your containerized applications.

References:
https://aws.amazon.com/fargate/
https://aws.amazon.com/fargate/faqs/
https://docs.aws.amazon.com/AmazonECS/latest/developerguide/AWS_Fargate.html

AWS WAF

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

Features

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

Conditions, Rules, and Web ACLs

- You define your conditions, combine your conditions into rules, and combine the rules into a web ACL.
- **Conditions** define the basic characteristics that you want WAF to watch for in web requests.
- You combine conditions into **rules** to precisely target the requests that you want to allow, block, or count. WAF provides two types of rules:
    - **Regular rules** - use only conditions to target specific requests.
    - **Rate-based rules** - are similar to regular rules, with a rate limit. Rate-based rules count the requests that arrive from a specified IP address every five minutes. The rule can trigger an action if the number of requests exceed the rate limit.
- **WAF Managed Rules** are an easy way to deploy pre-configured rules to protect your applications common threats like application vulnerabilities. All Managed Rules are automatically updated by AWS Marketplace security Sellers.
- After you combine your conditions into rules, you combine the rules into a **web ACL**. This is where you define an action for each rule—allow, block, or count—and a default action, which determines whether to allow or block a request that doesn't match all the conditions in any of the rules in the web ACL.

Pricing

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

References:
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/

AWS Shield

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

Shield Tiers and Features

- **Standard**
    - All AWS customers benefit from the automatic protections of Shield Standard.
    - Shield Standard provides always-on network flow monitoring which inspects incoming traffic to AWS and detect malicious traffic in real-time.
    - Uses several techniques like deterministic packet filtering, and priority based traffic shaping to automatically mitigate attacks without impact to your applications.

    - When you use Shield Standard with CloudFront and Route 53, you receive comprehensive availability protection against all known infrastructure attacks.
    - You can also view all the events detected and mitigated by AWS Shield in your account.
- **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, Business Support+, or Unified Operations levels of AWS Premium Support.
    - It automatically provides additional mitigation capacity to protect against larger DDoS attacks. The DDoS Response Team also applies manual mitigations for more complex and sophisticated DDoS attacks.
    - It gives you complete visibility into DDoS attacks with near real-time notification via CloudWatch and detailed diagnostics on the "AWS WAF and AWS Shield" Management Console.
    - Shield Advanced comes with "DDoS cost protection", a safeguard from scaling charges as a result of a DDoS attack that cause usage spikes on your AWS services. It does so by providing service credits for charges due to usage spikes.
    - It is available globally on all CloudFront and Route 53 edge locations.
    - With Shield Advanced you will be able to see the history of all incidents in the trailing 13 months.

Pricing

- **Shield Standard** provides protection at no additional charge.
- **Shield Advanced**, however, is a paid service. It requires a 1-year subscription commitment and charges a monthly fee, plus a usage fee based on data transfer out from CloudFront, ELB, EC2, and AWS Global Accelerator.

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

AWS Developer Services

AWS has a lot of tools and services that can help you accelerate your application development – enabling you to create feature-rich apps in no time! These services can help you expedite the development of your web and mobile apps; easily add a user authentication capability or social logins to your applications; launch production-ready REST and GraphQL APIs in a matter of days instead of weeks or months, and deploy, manage, and connect to a data store easily.

We will cover the related development services in AWS for the AWS Certified Solutions Architect exam in this section.

AWS Amplify

AWS Amplify is one of the many development services in AWS that helps you build extensible, full-stack web and mobile apps faster. It allows you to easily start your application development and automatically scale your resources with less management overhead. In other words, AWS Amplify deploys your application in a serverless architecture without any manual intervention on your part. Plus, it also gives you the option to integrate Machine Learning capabilities into your applications!

Put simply, AWS Amplify is a set of purpose-built tools and features. It consists of the Amplify Studio, Amplify Libraries, Amplify CLI, Amplify Hosting, and many more. Let's quickly cover its different modules one by one.

Amplify Studio is a visual development environment that simplifies the development of your web and mobile apps. You can easily build your frontend UI with its set of ready-to-use UI components, which you can directly connect to your app backend. Since your UI components are already pre-built, the number of your boilerplate code will be reduced, allowing you more time to focus on implementing the application business logic.

You can define your data models, implement user authentication, and add file storage using the Amplify Studio without any backend expertise. In addition, you can import Figma prototypes built by your application designers into the Amplify Studio for more seamless collaboration.

The AWS Amplify Libraries are open-source JavaScript libraries in the Amplify Framework. These are specifically designed to aid you in building AWS-powered mobile and web apps. The Amplify JavaScript libraries are supported in React, React Native, Angular, Ionic, Vue, and different web and mobile frameworks.

The Amplify Command Line Interface is a CLI toolchain that you can use to configure and maintain your app backend straight from your local desktop. The Amplify CLI has an interactive workflow and intuitive use cases that you can leverage, such as authentication, storage, and API. It allows you to test your features locally and deploy your app to multiple environments. This tool provides infrastructure-as-code templates which are automatically loaded to CloudFormation or AWS SAM. With the Amplify CLI, you can have a much more effective team collaboration and easy integration with Amplify's CI/CD workflow.

It also has a feature called Amplify Hosting, which allows you to host secure, reliable, fast web apps or websites via the AWS content delivery network. Under the hood, it deploys your applications to Amazon CloudFront's content delivery network, which comes with hundreds of points of presence globally. Amplify Hosting also allows you to set up your own custom domain for your applications and add custom alarms that send notifications if a certain metric exceeds a threshold that you specify.

These are some of the tools and features available in AWS Amplify that you can leverage with. Aside from Amplify Studio, Amplify Libraries, Amplify CLI, and Amplify Hosting, there are many other features and tools in AWS Amplify that you can try out, and AWS is adding more features soon to fast-track your software development experience!

AWS Device Farm

AWS Device Farm is an app testing service for testing and interacting with your Android, iOS, and web apps on real, physical phones and tablets. Instead of doing the manual testing yourself, you can just use this service so you can do a wide range of tests without having to provision and manage any testing infrastructure.

AWS Device Farm supports both native and hybrid Android, iOS, and progressive web apps. Cross-platform frameworks are also supported, such as PhoneGap, Titanium, Xamarin, Unity, and others.

This service is also capable of running Selenium tests on different desktop browsers and browser versions that are hosted in the AWS Cloud. AWS Device Farm can even execute your test suite on your own local machine and interact with browsers hosted on AWS Device Farm through the Selenium API.

This accelerates your mobile app development workflow by executing tests on multiple devices without the administrative overhead of maintaining a test environment. With AWS Device Farm, you can do integration tests that check and detect any customer issues before they are even released in production.

Amazon Managed Grafana

Amazon Managed Grafana is a fully managed service for Grafana. Grafana is an open-source analytics platform that is commonly used to query, visualize, observe, and make use of your system data that are gathered from multiple sources. When you hear the phrase "Amazon Managed", that means AWS is managing the underlying infrastructure required to run an open-source program or a particular tool. For Amazon Managed Grafana, AWS is the one that provisions and manages the required resources to run your Grafana dashboards, along with its other dependencies.

Let's take a good look at what Grafana really is. What you see here is an actual Grafana platform. It has tables, graphs, time series data, maps, statistics, and a range of other useful metrics. On its sidebar menu, you'll see the Dashboards, Explorer, Alert, Grafana Machine Learning, Kubernetes Monitoring, Synthetic Monitoring, and so much more. Grafana can collect metrics from different data sources so you can have a single place for all your data. In this way, all the metrics from your databases, servers, storage systems, APIs, custom data, logs, and other sources can be aggregated in one location. This is also the case in Amazon Managed Grafana.

In AWS, Amazon Managed Grafana can collect system metrics from multiple data sources in your observability stack, such as Amazon Managed Service for Prometheus, Amazon CloudWatch, and Amazon OpenSearch Service. System alerts can also be automated by using different notification services in AWS. You can also integrate this with third-party vendors like Datadog, Splunk et cetera. In addition, you can set up your own self-managed data source like InfluxDB and integrate it with your Grafana workspace. You also don't have to worry about the infrastructure required to run your Grafana visualizations and dashboards since the necessary resources are all provisioned and managed by AWS itself.

Amazon Managed Grafana is somewhat similar to Amazon CloudWatch Dashboard since both of them have a dashboard where you can see your system metrics. However, Grafana has a much better user interface and can integrate easier with a range of other systems and data providers.

Amazon Managed Service for Prometheus

Amazon Managed Service for Prometheus is a fully managed service for the open-source monitoring tool called Prometheus. This is commonly used for monitoring modern cloud-native applications and Kubernetes clusters. Prometheus can enable you to securely ingest, store, and query metrics from different container environments.

In Amazon Managed Service for Prometheus, the resources required to run the open-source Prometheus tool are all provisioned and managed by the AWS team themselves. Scaling the underlying architecture is also handled by Amazon. You have the option to collect system metrics from your container clusters running in AWS, running on-premises, or even both.

Amazon Managed Service for Prometheus allows you to use the open-source Prometheus query language or PromQL. This query language helps you to monitor the performance of your containerized workloads that are running on the AWS Cloud or on-site. It can also scale automatically as your workloads increase or shrink and uses AWS security services to enable fast and secure access to data.

Amazon Managed Service for Prometheus has a feature called Alert Manager that handles all alerts that are sent from your workspace. These alerts are grouped and routed to downstream receivers that you specify. Duplicate alerts and messages can also be deduplicated by Alert Manager. Under the hood, it uses the Amazon Simple Notification Service as a receiver and routes messages to different SNS topics in the same account.

In this service, you create a workspace for your Prometheus metrics. A workspace isolates access control to a particular application suite for ingestion, storage, and querying of your metrics. The container metrics that are available in your Prometheus workspace can also be collected by the Amazon Managed Grafana service.

AWS Machine Learning Services

Machine learning is a branch of artificial intelligence that is all about imitating how humans learn. Just like humans, a machine can learn by watching, hearing, reading, and analyzing the things in its environment. This machine can be in the form of a simple computer program, an API, or an entire enterprise system. Once a machine consumes enough information, it can provide a range of helpful information and useful results that can save you time, effort, and even money.

Some people may find this topic to be quite difficult to understand, especially with all the complex math and algorithms it entails. To make it simple, let's start with a couple of real-world machine learning applications that you might be using already.

Most of us have a subscription to online streaming services like Netflix, Amazon Prime Video, Disney Plus, or HBO. In these streaming services, you might notice that they can recommend a particular movie or TV series based on shows that you have watched in the past. The recommendation engine will analyze your viewing history and predict the genre or type of shows you like. So if you are watching a lot of horror movies, you will be presented with different horror shows since this is the genre that you are always watching.

Let's look at another use case. Do you know that if you take a photo on your iPhone, it will automatically detect any text in the image? In this way, you can easily copy text and not manually type them on your phone. You can even try it right now! Take a picture of something that has a label, a word, or a number on it. Open that picture, then double-tap any text present in the image, and you will be able to copy them right away. This iPhone feature is called Live Text, which is a form of Optical Character Recognition that uses machine learning to transcribe text from images in your Photo Library.

Here's another example: if you're living in the US, you might have shopped at Amazon Go which is a convenience store with no checkout! Amazon Go uses computer vision and machine learning to automatically track all the products you grabbed from the store and bill you online without having to wait in line to check out your goods! This computer vision technology is also what powers self-driving cars, facial recognition systems, and many others. There is no doubt that we are already using Machine Learning in almost every moment of our lives.

In this section, we will learn about the various Machine Learning services that are available in Amazon Web Services. The primary machine learning platform in AWS is called Amazon SageMaker AI, which is followed by a lot of other ML services. Some services in AWS have machine learning features as well, like Amazon Aurora

Machine Learning, Redshift ML, Deep Learning AMIs, and so much more. Take note that we won't cover them here as we will only focus on the dedicated ML services that are relevant to your upcoming AWS exam.

It is easy to get overwhelmed by the sheer number of services in AWS, so we will divide this lecture into several sections. The AWS Machine Learning services can be classified into these use cases: Computer Vision, Language AI, Automated Data Extraction and Analysis, Customer Experience Improvement, Business Metrics, and DevOps.

For Computer Vision, we have:

- Amazon Rekognition
- Amazon Lookout for Vision
- AWS Panorama

For Automated data extraction and analysis, AWS has:

- Amazon Textract
- Amazon Augmented AI
- Amazon Comprehend

For Language AI, you have:

- Amazon Lex
- Amazon Transcribe
- Amazon Polly

For Customer Experience improvement, you can use the following services:

- Amazon Kendra
- Amazon Personalize
- Amazon Translate

For Business metrics, you can use:

- Amazon Fraud Detector

And lastly, for DevOps, we have:

- Amazon DevOps Guru
- Amazon CodeGuru Reviewer
- Amazon CodeGuru Profiler
- Amazon CodeWhisperer

Let's discuss each of these services one by one.

Amazon SageMaker AI

First off, let's talk about Amazon SageMaker AI. Think of this as a full-fledged machine learning platform in AWS with tons of services, features, and components. Amazon SageMaker AI is not just a simple ML service but a fully managed cloud platform with lots of modules that you can use. With this, you can build, train, and deploy ML models for any use case with fully managed infrastructure, tools, and workflows. SageMaker removes the manual tasks from each step of the ML process to make it easier for you to develop high-quality models. This platform has so many modules that you choose from, namely: Amazon SageMaker AI Canvas, SageMaker Studio Lab, SageMaker Data Wrangler, SageMaker Autopilot, SageMaker JumpStart, SageMaker Clarify, and so much more!

Amazon Rekognition

Amazon Rekognition provides pre-trained and customizable computer vision capabilities to extract information and insights from your images and videos. Just as its name implies, it can recognize certain objects, faces, texts, scenes, labels, and other attributes from your media files or streaming videos. This service is perfect for facial recognition, where it can detect the face of a particular person or a well-known celebrity. It can also determine if someone is wearing a piece of Personal Protective Equipment like a mask, a helmet, or gloves. If you upload an image of you holding a guitar while sitting on your sofa, Amazon Rekognition can detect your face, your guitar, and the sofa where you sit. It also has a feature called Amazon Rekognition Custom Labels. With this, you can easily build a machine learning model to classify custom components or products from your dataset.

Amazon Lookout for Vision

Amazon Lookout is a suite of services that is comprised of Amazon Lookout for Equipment, and Amazon Lookout for Vision. The last one uses computer vision to detect defects on industrial products at scale. Amazon Lookout for Vision is primarily used in factories and manufacturing lines to quickly and accurately identify defects in each product. The dataset can be in a form of product images that are stored in an Amazon S3 bucket. You can provide a couple of baseline images to Amazon Lookout for Vision containing defect-free products, and this service will be able to automatically build a model for you within a few hours. From there, it can automatically detect anomalies in your product like dents, cracks, and scratches.

Amazon Textract

The name of this service is a combination of the words ”text” and “extract”. This will give you a hint that it is used to extract texts from scanned documents, notes, and images. Amazon Textract is a service that uses optical character recognition to automatically extract text from scanned files like PDFs, Word documents, hand-written notes, receipts, passports, IDs, and many others. What makes this service great is its capability to generate the results into a table form or a CSV file. It also has a query feature that allows you to extract a particular field using natural language questions. So if you upload your driver's license to Amazon Textract, you can submit a query like “What's the first name?” or “What's the driver's ID?“ and you'll get the value you asked for. You can also batch upload your documents to S3 and automate the text analysis process.

Amazon Augmented AI

Amazon Augmented AI or A2I is a service that provides human review workflows for common machine learning use cases. A human review literally means that a human being will review a certain output that your machine learning model generated before it can proceed to the next step of the workflow. This service augments your AI to ensure the accuracy of prediction results and helps provide continuous improvements to your machine learning model. You can directly integrate your workflows from Amazon Rekognition or Amazon Textract to Amazon Augmented AI. For example, you can do a human review of the key-value pairs that are extracted by Amazon Textract or implement image moderation by doing a human review of unsafe content, such as explicit adult or violent content from Amazon Rekognition. It is also possible to run a human review with a custom machine learning workflow of your choice.

Amazon Comprehend

Amazon Comprehend is a natural language processing service in AWS that can find insights and relationships in a text. It performs text analytics that can automatically extract key phrases, sentiment, language, syntax, topics, and even Personally Identifiable Information or PII from unstructured data. In essence, Amazon Comprehend is a service that comprehends or understands the information written in your text documents. This is different from Amazon Textract since Amazon Comprehend cannot read text from scanned documents. You need to have raw text data first in order to use Amazon Comprehend.

Amazon Lex

Amazon Lex is a machine learning service that allows you to develop chatbots. You can build Voice-based or Text-based chatbots with Amazon Lex easily. This is helpful if your company needs a self-service bot or a virtual agent for your conversational Interactive Voice Response (IVR) system, corporate website, or other customer-facing application. Amazon Lex can significantly reduce the costs of companies in maintaining its contact center.

Amazon Transcribe

Amazon Transcribe is simply a speech-to-text transcription service. The word transcribe means to make a written record of a speech, a phone call, or any spoken language, and this is exactly what Amazon Transcribe does. It is also helpful in contact centers as it can generate call transcripts and provide conversation insights to help improve customer experience and agent productivity. Amazon Transcribe also offers real-time transcription – where you can just talk to its endpoint, and it will immediately generate transcripts of your speech.

Amazon Polly

The other service relating to Language AI is called Amazon Polly. Essentially, Amazon Polly is the exact opposite of the Amazon Transcribe service. Instead of turning speech into text, it converts text into speech! If you input a text into Amazon Polly, it will generate a lifelike speech in different voices that you specify. So, for example, you typed: “Beautiful Philippine Islands” in the Amazon Polly console. You can hear the phrase: “Beautiful Philippine Islands“ in a male voice, a female voice, a kid's voice, or in any voice that you prefer. You can also customize the pronunciation of specific words and phrases by uploading your own lexicon files. A lexicon is simply a vocabulary of a particular language, and this is usually used if you have a non-English text that you want to turn into speech.

Amazon Kendra

Amazon Kendra is an intelligent search service in AWS. It is not just a typical search service that simply returns a match to your query from a single data source. Amazon Kendra can search from multiple data sources that can be structured or unstructured, then intelligently analyze the content before it sends a result. This service supports natural language processing, so you can ask questions using a language that you use in your everyday life. For instance, you can ask Amazon Kendra, “Who is the founder of the EdTech startup: Tutorials Dojo?” and it will search all of the documents in your S3 bucket, Amazon FSx file systems, Amazon RDS databases, Github repository, Jira, Slack, Sharepoint and other data sources for the answer. Again, it can search for information from a wide range of sources and not just from a traditional SQL database, then uses machine learning to provide context to your search results.

Amazon Personalize

Amazon Personalize is a service that provides personalized recommendations to your customers based on their past activity and behavior. It's just like the recommendation feature in Amazon Prime or Netflix, where new movies are automatically recommended for you based on your viewing history. If you watched a lot of Sci-Fi movies on their online streaming platform, they will automatically recommend more Sci-Fi shows on your profile. This is definitely a customer experience improvement since personalizing the user's content tends to convert more because they align with what the customers actually do and buy.

Amazon Translate

Amazon Translate is a real-time translation service in AWS. It works pretty much like Google Translate, where you input text in one language, and the service will translate it to a language that you choose. You can also create your own custom terminology. This allows you to customize the output of Amazon Translate based on a company-specific and domain-specific vocabulary. For example, I can set the acronym TD as “Tutorials Dojo” in English. The Amazon Translate service can accept input with my custom vocabulary and include it in the translation. In this case, I can enter the Tagalog phrase “Magandang Umaga TD” and Amazon Translate will return “Good morning Tutorials Dojo” as an output. The Filipino phrase “magandang umaga” means “good morning” in English, while “TD” is the custom term for “Tutorials Dojo” which we configured in Amazon Translate. You can also enable the Formality option, which controls whether the translation output uses a formal tone or not. The translation can also mask profane words or phrases, which is a very useful feature for customer-facing applications.

Amazon Fraud Detector

Amazon Fraud Detector is yet another machine learning service that can automate fraud detection, just as its name suggests. It can identify potential fraudulent activity, fake reviews and spam account creation in near-real-time. For instance, your website recently got a visitor whose IP address has a history of malicious activity such as spamming, hacking attempts, and DDoS attacks. Users with exactly the same IP address are posting spam on your website repeatedly. For this situation, you can use Amazon Fraud Detector to block any visitor who uses an offending IP address, an email domain, or any other attribute that you define.

Amazon DevOps Guru

Amazon DevOps Guru detects abnormal behavior in your application or AWS resources that might cause unexpected downtimes or operational issues in the near future. It can monitor applications and AWS resources within your own account or on all accounts across your AWS Organization. It uses machine learning to identify operational defects long before they impact you and your customers. Amazon DevOps Guru can analyze your RDS databases and automatically determine an unusually high DB load that is more than three times or 5 times its normal value. It can also detect issues in your serverless stack like an extremely high number of invocations in your Lambda function that is beyond the currently provisioned concurrency or an overprovisioned write capacity on your DynamoDB tables.

Amazon CodeGuru

Amazon CodeGuru is a suite of development services in AWS. It contains different tools and features such as Amazon CodeGuru Reviewer, Amazon CodeGuru Profiler, BugBust, and many more. The primary function of Amazon CodeGuru Reviewer is to provide intelligent recommendations for improving your application performance, efficiency, and code quality. It can scan your code and detect a plethora of code defects like bad exception handling, insecure CORS policy, path traversal, hardcoded credentials, and many more. You can also integrate this with your CI/CD workflow so you can run code reviews and recommendations to improve your codebase. The other module for this service is called the Amazon CodeGuru Profiler. A profiler is basically a component that collects your CPU data and analyzes the runtime performance data from your live applications. This is helpful in identifying expensive lines of codes that inefficiently use the CPU, which causes CPU bottlenecks.

Amazon CodeWhisperer

Amazon CodeWhisperer is a coding tool that automatically generates code and functions in real-time. This tool is similar to Github CoPilot, which is an extension that you usually install in your visual studio IDE. The lines of codes are generated right from your IDE editor based on the comments that you write. For example, you can simply write a comment that outlines a specific task in plain English, such as “Upload a file to an Amazon S3 bucket with server-side encryption”. Amazon CodeWhisperer will take your comment as input and generate an entire function in the programming language that you define, which can upload a file to your S3 bucket with the required encryption and many more

AWS Deployment Services

There are different ways to build, test, and deploy your applications to your development or production environments. Back in the day, we used to just copy a ZIP file, a WAR file, or the binary files in the webapps directory of our web servers. Then, we have to manually configure and restart our Apache, NGINX, and IBM WebSphere servers to reflect the changes. This process is called a manual deployment and is prone to a lot of human errors. There are a lot of moving parts that you have to do by hand, which may take several hours to complete. Worse, provisioning the required virtual web server or a dedicated database cluster will take you weeks or even months to accomplish due to a lack of automation. Good thing that nowadays, you can do your deployment in a blink of an eye through the various deployment tools available at your disposal.

AWS offers a number of services that provide deployment and management capabilities for one or more aspects of your application lifecycle. These services enable organizations to build and deliver applications faster than their traditional CI/CD workflow. You and your team don't have to spend a lot of time manually provisioning, configuring, updating, monitoring, or securing your AWS resources anymore. These laborious tasks can be programmed into code and easily automated with the different deployment services available in AWS. Some of these services use the concept of "Infrastructure as Code" or IaC. An IaC is a process of managing and provisioning your servers, databases, CDNs, and other resources through machine-readable definition files. Simply put, you only need to provide a text-based definition file that will automatically provision the required resources for your application in just one click of a button.

The advent of cloud computing enables companies to deploy all their workloads entirely in the cloud. They also have the option to run a hybrid cloud architecture in which they utilize both their physical on-premises resources and cloud services in AWS at the same time. There's even an option now to do a multi-cloud deployment where you deploy your infrastructure to AWS, Azure, Google Cloud, and other public cloud providers simultaneously. You can run both your multitier applications and Kubernetes or Docker container cluster almost anywhere – whether it be on the cloud, on-premises, on multiple public clouds, or a combination of all three.

Let's discover the different deployment services in this lesson. These services have the capability to provision, configure, deploy, scale, and monitor your cloud architecture without any manual intervention on your part. They are:

- AWS CloudFormation
- AWS Elastic Beanstalk
- AWS CodeDeploy
- Amazon ECS Anywhere
- Amazon EKS Anywhere
- AWS Systems Manager
- AWS Proton

AWS CloudFormation

AWS CloudFormation is a service that enables you to provision and manage your AWS resources using a custom code template. You can create a custom template in YAML or JSON format that defines the AWS resources that you require, like Amazon EC2 instances, Amazon FSx filesystems, Amazon Aurora databases, CloudFront distributions, or any other resource, and the AWS CloudFormation service can deploy all of these automatically for you. AWS CloudFormation also comes with a graphic tool called the AWS Infrastructure Composer. This is a drag-n-drop online tool for creating, viewing, and modifying your AWS CloudFormation templates.

CloudFormation is the primary Infrastructure as Code service in AWS. It works like any other IaC tools in the market, like Terraform, Ansible, Chef, and Puppet. The key difference, however, is the additional features that it provides, which are fully compatible with the AWS Cloud.

A CloudFormation template deploys your cloud infrastructure resources in a group called a "stack". This stack can represent your entire cloud architecture or just its subset. If you have a large multi-tier architecture, you can create multiple templates to represent the different tiers of your enterprise application suite. You can create a CloudFormation template for your presentation layer stack, another template for your application layer stack, and one more for your data layer stack.

You can bundle your multiple stacks together into something called a nested stack. In CloudFormation, you can have a root stack with a hierarchy of nested stacks under it. This will effectively make the modules of your infrastructure code to be loosely coupled with each other, which makes the management of each individual stack much easier. Your application layer stack will only contain EC2, EKS, ECS, and other compute resources, while the data layer stack will have RDS, Aurora, DocumentDB, Amazon Neptune, Amazon Timestream, or any database services. Any change in one nested stack won't adversely affect the other stack.

Aside from provisioning your resources, Amazon CloudFormation allows you to change, modify, or scale your services that are already deployed in your AWS account. It even has a dry-run mode to check your upcoming changes before they are deployed in your cloud environment. This feature is called a "Change Set".

Essentially, a change set allows you to see how your changes might impact your running resources before finally implementing them. Say you want to change the name of your Amazon Aurora Serverless database. You can create a change set in CloudFormation that will create a new Aurora database and delete the old one. Of course, you will lose the data in the old database unless you've taken a DB snapshot for backup. The change set will show you the upcoming change so you can plan accordingly before you update your stack.

A CloudFormation stack is usually mapped in a single AWS account only, so if you are running your applications in two or more AWS accounts, you can use StackSets. A StackSet extends the capability of CloudFormation stacks by enabling you to create, update, or delete stacks across multiple accounts and AWS

Regions with a single operation. You can select an administrator account in your AWS Organization and then choose a particular CloudFormation template for your StackSet. This template will be the basis for provisioning the stacks into your selected target accounts.

The stack, change set, StackSet, and the graphical designer tool are the base features of the AWS CloudFormation service. There are different services in AWS that extend the capabilities in CloudFormation, namely the AWS Cloud Development Kit and AWS Serverless Application Model. The AWS Cloud Development Kit, or AWS CDK for short, is an open-source software development kit for Amazon Web Services. You can use this to programmatically model your AWS infrastructure using TypeScript, Python, Java, .NET, or any other programming languages that you prefer.

AWS Serverless Application Model (AWS SAM)

The AWS Serverless Application Model service, or AWS SAM, is an open-source framework that simplifies the development of your serverless applications on AWS. This is commonly used if your cloud stack is using AWS Lambda, Amazon DynamoDB, API Gateway, and other serverless services. AWS SAM can also have a SAM template that is essentially just an extension of the AWS CloudFormation template. A SAM template has some additional components that make it easier for you to work with serverless services in AWS.

Your apps can be stored in the AWS Serverless Application Repository. This repository service makes it easy for developers and companies to deploy, manage, and share their serverless applications in AWS and to the greater public. You can easily publish your serverless apps and share them with the community at large or privately within your organization. Each and every application in the AWS Serverless Application Repository is packaged with an AWS Serverless Application Model (SAM) template that defines the different AWS resources which the application uses.

AWS Elastic Beanstalk

AWS Elastic Beanstalk is a managed platform that allows you to upload your application code in AWS and provision the required cloud environment easily. You only need to upload your application package that is written in Java, .NET, PHP, Node.js, Python, Ruby, Go, or Docker, and then Elastic Beanstalk will deploy the necessary resources to run your application. You can either run a Web Server environment or a Worker environment. A Web Server environment runs a static website, a web app, or a web API that serves HTTP requests, while a worker environment, on the other hand, runs a worker application that processes long-running workloads on demand. The latter also performs tasks on a schedule that you define and can be integrated with the Amazon SQS queue. The AWS Elastic Beanstalk service also uses a configuration file like CloudFormation, to automatically deploy and configure your applications. These configuration files in Elastic Beanstalk are stored in the .ebextensions folder.

AWS CodeDeploy

AWS CodeDeploy is a fully managed deployment service that automates your application deployments in AWS. You can deploy your applications to Amazon EC2 instances, Amazon ECS clusters, AWS Lambda functions, and other computing services in AWS. You can even use this to deploy your application to the servers located on your on-premises network. This service is different from AWS CloudFormation, AWS SAM, and Elastic Beanstalk since you can only deploy your applications to existing compute resources. AWS CodeDeploy does not create AWS resources on your behalf and is intended for application deployments only.

Amazon ECS Deployment Options

Amazon ECS is a fully managed container orchestration service that supports Docker containers. This orchestration service allows you to easily run containerized applications on a managed cluster that you can control. Basically, container orchestration is an automation tool that reduces the operational effort needed to run your containerized workloads and services. Amazon ECS can automatically orchestrate or control the manual tasks of provisioning, deploying, scaling, networking, and many other tasks, so you don't have to.

Amazon ECS Anywhere is a feature of Amazon ECS that enables you to run and manage container workloads on your own on-premises infrastructure. You'll still have the same cluster management, workload scheduling, monitoring, and support features in Amazon ECS Anywhere, whether you are running your workload on-premises or in the cloud. This ubiquitous service can help you meet your compliance requirements and scale your hybrid architecture without undermining the previous investments you already have in your on-premises hardware.

When you create an Amazon ECS cluster, you can choose whether the compute resources will be deployed or launched in your own VPC, in AWS Fargate, or externally via Amazon ECS Anywhere. A cluster launched in a VPC uses Amazon EC2 instances that are orchestrated and controlled by Amazon ECS. You can use the Amazon CloudWatch Container Insights to monitor your container workloads.

If the launch type is AWS Fargate, then your cluster will be serverless, and the computing resources will be fully managed by AWS. Using AWS Fargate can significantly reduce the operating costs of running your containers. If you decide to launch your cluster externally, then your cluster will be run on your own server located on-site through the help of the Amazon ECS Anywhere service.

Amazon EKS Deployment Options

Amazon Elastic Kubernetes Service or Amazon EKS is a managed service that you can use to run Kubernetes on AWS. It's like Amazon ECS, but instead of Docker containers, this service is used for running Kubernetes clusters. Amazon EKS automates the installation, operation, and maintenance of your own Kubernetes control plane, pods, and nodes.

You can deploy your Kubernetes cluster in various ways in AWS and can include additional networking add-ons to improve your containerized architecture. A Kubernetes container can be deployed to an Amazon EKS cluster in your AWS account, to Amazon EKS on AWS Outposts, to Amazon EKS Anywhere, and through the Amazon EKS Distro. The first option allows you to launch a Kubernetes cluster using managed or self-managed Amazon EC2 nodes that you can customize and control. You can also choose to deploy your Kubernetes pods on AWS Fargate to make the cluster serverless and extremely cost-effective.

The second type is Amazon EKS on AWS Outposts which is a deployment option that uses a physical AWS Outpost rack on your on-premises network to run your Kubernetes workloads. The data plane is also located on-premises, so you can have more control compared with running it exclusively in AWS.

Using Amazon EKS Anywhere is another way to deploy your containers on-premises. It works like Amazon ECS Anywhere, which allows you to run your Kubernetes cluster entirely on your own. This means that the hardware, app deployment location, control plane, and data plane are all controlled on your own physical network. This gives you extensive control over all the components of your containerized application suite while maintaining official support from AWS.

The other deployment option that you can choose is Amazon EKS Distro. The word "distro" simply refers to the distribution of the same open-source Kubernetes software deployed by Amazon EKS in the AWS cloud. Amazon EKS Distro follows the same Kubernetes version release cycle as Amazon EKS and is provided to you as an open-source project that you can deploy on your own computer or on-site environment. It's similar to the Amazon EKS Anywhere option, except that it does not include support services offered by AWS.

AWS Proton

AWS Proton is a service that automates container and serverless deployments in AWS. It empowers your platform teams and developers to have consistent development standards and best practices. This service is very useful if you have a large number of developers in your organization. AWS Proton enables your developers to deploy container and serverless applications using pre-approved stacks that your platform team manages. It balances control and flexibility in your organization by allowing developers to innovate within the set guardrails that you implement.

It also offers a self-service portal for your developers, which contains AWS Proton templates that they can use and deploy. A Proton template contains all the information required to deploy your custom environments and services. You can create an AWS Proton Component as well, which provides flexibility to your service templates. These components in AWS Proton provide platform teams with a way to extend core infrastructure patterns and define guardrails for your developers.

AWS Audit Manager

The AWS Audit Manager is a service that you can use to continuously audit your AWS usage. It simplifies the process of doing risk and compliance assessment by utilizing a pre-built standard framework that is based on common compliance standards and AWS best practices. This service uses the data gathered from AWS Security Hub, AWS Config, AWS License Manager, and other AWS services to generate audit reports. The AWS Audit Manager can import license usage limits and the daily usage data from AWS License Manager to generate assessment reports accordingly.

Keep in mind that this is different from AWS Artifact, which only offers evidence to prove that the AWS Cloud infrastructure has met the compliance requirements. AWS Artifact does not check the AWS resources that you are actually using or even the way you use them. On the other hand, the AWS Audit Manager gathers evidence to prove that your usage of AWS services complies with the industry standards or regulations.

Amazon Inspector

Amazon Inspector is an automated security assessment service that helps improve the security and compliance of applications deployed on AWS. Amazon Inspector automatically assesses applications for vulnerabilities or deviations from best practices. After performing an assessment, Amazon Inspector produces a detailed list of security findings prioritized by level of severity. It provides an automated security assessment report that will identify unintended network access to your Amazon EC2 instances and vulnerabilities in those instances. These findings can be reviewed directly or as part of detailed assessment reports which are available via the Amazon Inspector console or API.

Amazon Detective

Amazon Detective makes it easy to analyze, investigate, and quickly identify the root cause of potential security issues or suspicious activities. Amazon Detective automatically collects log data from your AWS resources. What it basically does is to collect logs from AWS CloudTrail, Amazon VPC Flow Logs, Amazon GuardDuty findings, and other AWS services then use machine learning to analyze and conduct security investigations.

AWS Security Hub

AWS Security Hub is a service that provides a centralized and comprehensive view of the security posture of your cloud infrastructure across multiple AWS accounts. This service helps you to comply with specific security standards and best practices that you organization require. Basically, it works like a hub that collects security alerts and findings from multiple AWS services, such as Amazon GuardDuty, Amazon Inspector,

Amazon Macie, AWS Identity and Access Management (IAM) Access Analyzer, AWS Firewall Manager and other sources. You can the AWS Security Hub to comply with the Payment Card Industry Data Security or PCI DSS Standard, Center for Internet Security or CIS Benchmarks, and many other security standards.

AWS Network Firewall

AWS Network Firewall is a managed network firewall service for your Amazon Virtual Private Clouds. This network security service is a managed network firewall that comes with intrusion prevention and detection capabilities. The AWS Network Firewall service allows you to filter traffic within the perimeter of your Amazon VPCs.

This service is commonly used in various network security use cases such as inspecting VPC-to-VPC traffic, filtering outbound traffic, securing both AWS Direct Connect connection and VPN traffic as well as filtering the Internet traffic. AWS Network Firewall also offers fine-grained network security controls for interconnected VPCs via the AWS Transit Gateway.

You can also use this to filter your outbound traffic to prevent unwanted data loss, block malware, and satisfy your strict network security compliance requirement. A single AWS Network Firewall can be configured with thousands of rules that can filter out network traffic routed to known bad IP addresses or suspicious domain names. It can also protect the AWS Direct Connect or VPN traffic that originates from client devices and your on-premises environments. The AWS Network Firewall can ensure that only authorized sources of traffic are granted access to your mission-critical VPC resources. It is also capable of performing the same activities as your Intrusion Detection Systems and Intrusion Prevention Systems or IDP/IPS. This is achieved by inspecting all inbound Internet traffic using features such as ACL rules, stateful inspection, protocol detection, intrusion prevention et cetera.

The AWS Network Firewall has 3 basic components, namely the Firewall, the Firewall Policy, and the Rule Group.

A firewall is a resource that you create in AWS Network Firewall. This firewall is connected to the Amazon VPC of your choice, where a network filter will be implemented based on the behavior defined in your firewall policy. Your firewall can have one firewall policy only, which contains rule groups that you define.

You can deploy the firewall in multiple Availability Zones and to one subnet per zone. Each subnet that is associated with your firewall must have at least one available IP address. Afterward, you must update your VPC route tables to send incoming and outgoing traffic through the firewall endpoints.

The second component is the firewall policy. This is a policy that defines the behavior of your firewall using a collection of stateless and stateful rule groups. A firewall policy can be associated with one or more firewalls.

However, a firewall can only have one firewall policy. Changing a firewall policy affects all other firewalls that reference it.

The third component in AWS Network Firewall is called a rule group. This is basically a collection of stateless or stateful rules that define how to inspect and handle the network traffic in your VPC. A rules configuration includes 5-tuple network values and domain name filtering. The 5-tuple format includes the source IP address, source port, destination IP address, destination port, and protocol.

AWS Network Firewall has two types of rules which could be stateless or stateful. The first one is stateless in the sense that it does not have any context of the packet's traffic flow. A stateless rule just checks the packet itself. On the contrary, a stateful rule knows the context or the state of the packet, including its direction flow and other information that is not provided by the packet itself. Each rule in your stateful rule group has an associated order for its evaluation sequence. This concept is similar to network access control lists and security groups. A network ACL is stateless, while a security group is stateful.

You can also choose how you want the AWS Network Firewall to handle the packets that match your rule criteria. You can either pass or drop a packet that was filtered by the network firewall. A packet can also be forwarded to another stateful rule group for re-evaluation. Setting up custom actions is possible as well. A custom action can be configured to publish data to CloudWatch metrics for future network analysis.

Comparison of AWS Services and Features

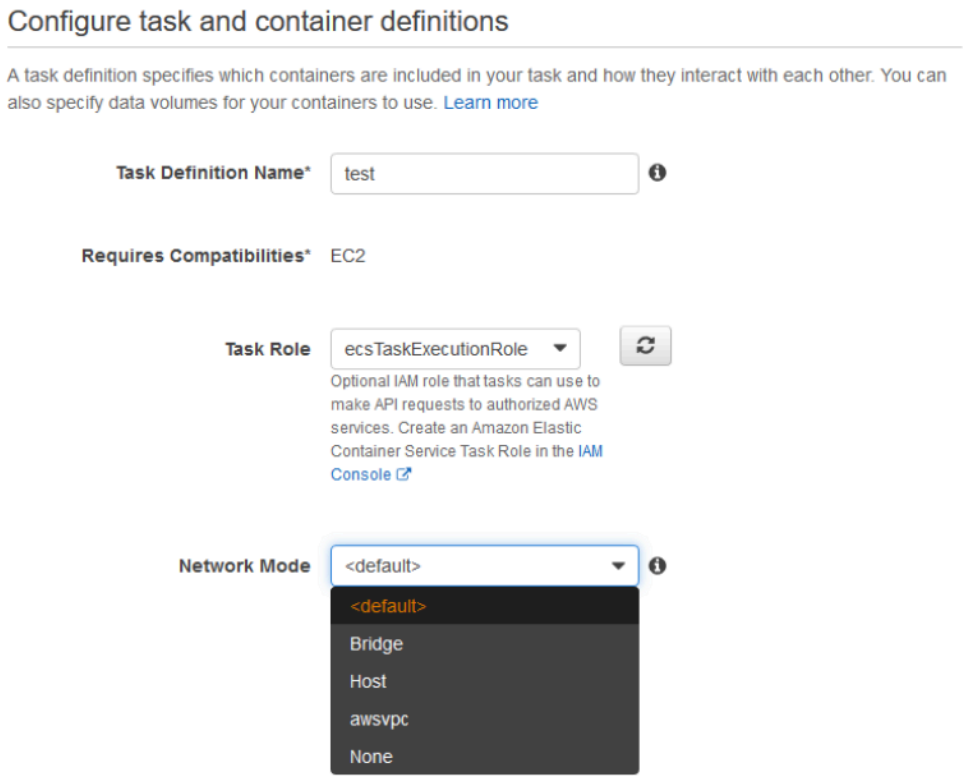
ECS Network Mode Comparison

Amazon Elastic Container Service (ECS) allows you to run Docker-based containers on the cloud. Amazon ECS has two launch types for operation: EC2 and Fargate. The EC2 launch type provides EC2 instances as hosts for your Docker containers. For the Fargate launch type, AWS manages the underlying hosts so you can focus on managing your containers instead. The details and configuration on how you want to run your containers are defined on the ECS Task Definition which includes options on networking mode.

In this post, we'll talk about the different networking modes supported by Amazon ECS and determine which mode to use for your given requirements.

ECS Network Modes

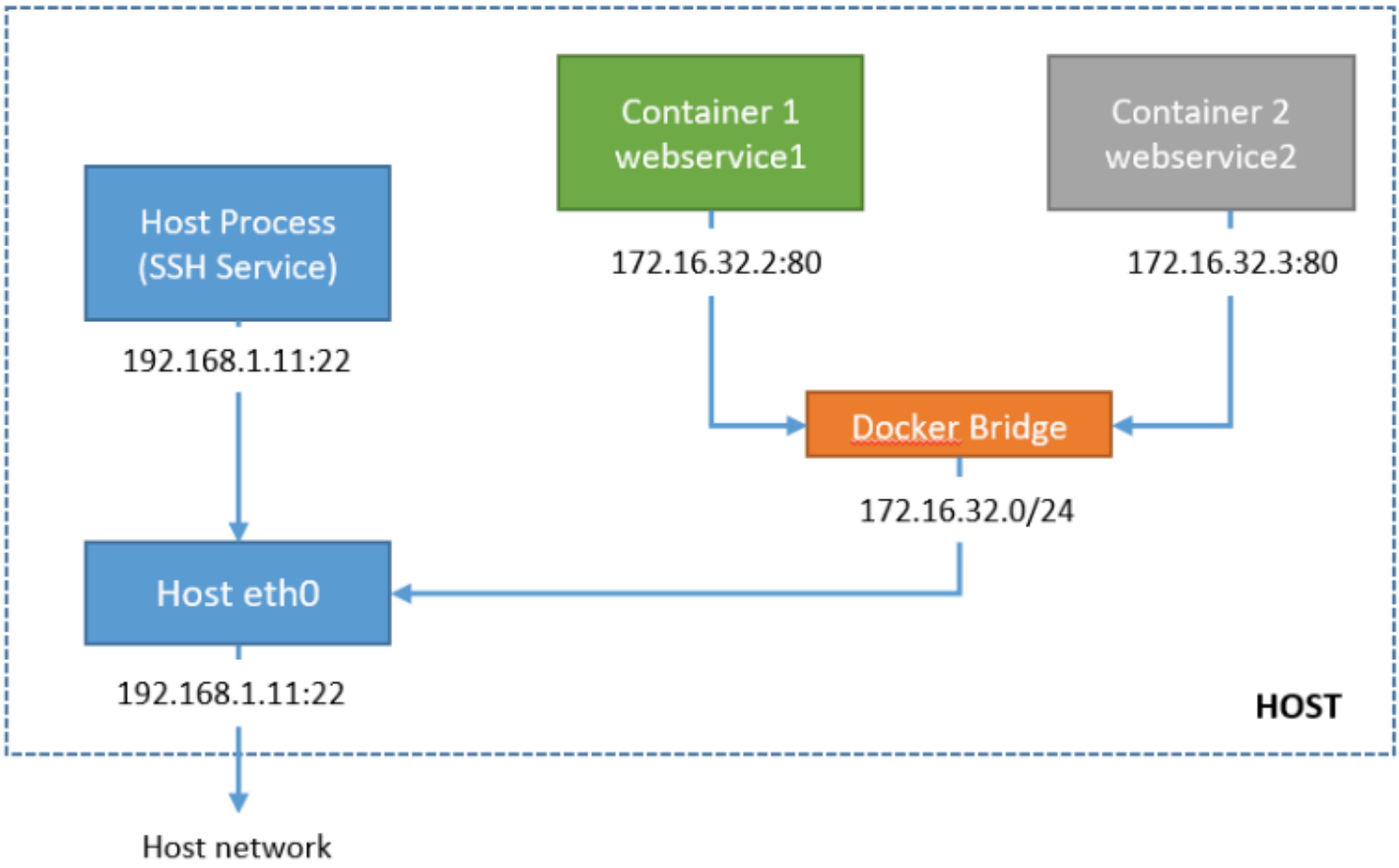
Amazon Elastic Container Service supports four networking modes: **Bridge**, **Host**, **awsvpc**, and **None**. This selection will be set as the Docker networking mode used by the containers on your ECS tasks.



Bridge network mode - Default

When you select the **<default>** network mode, you are selecting the **Bridge** network mode. This is the default mode for Linux containers. For Windows Docker containers, the **<default>** network mode is **NAT**. You must select **<default>** if you are going to register task definitions with Windows containers.

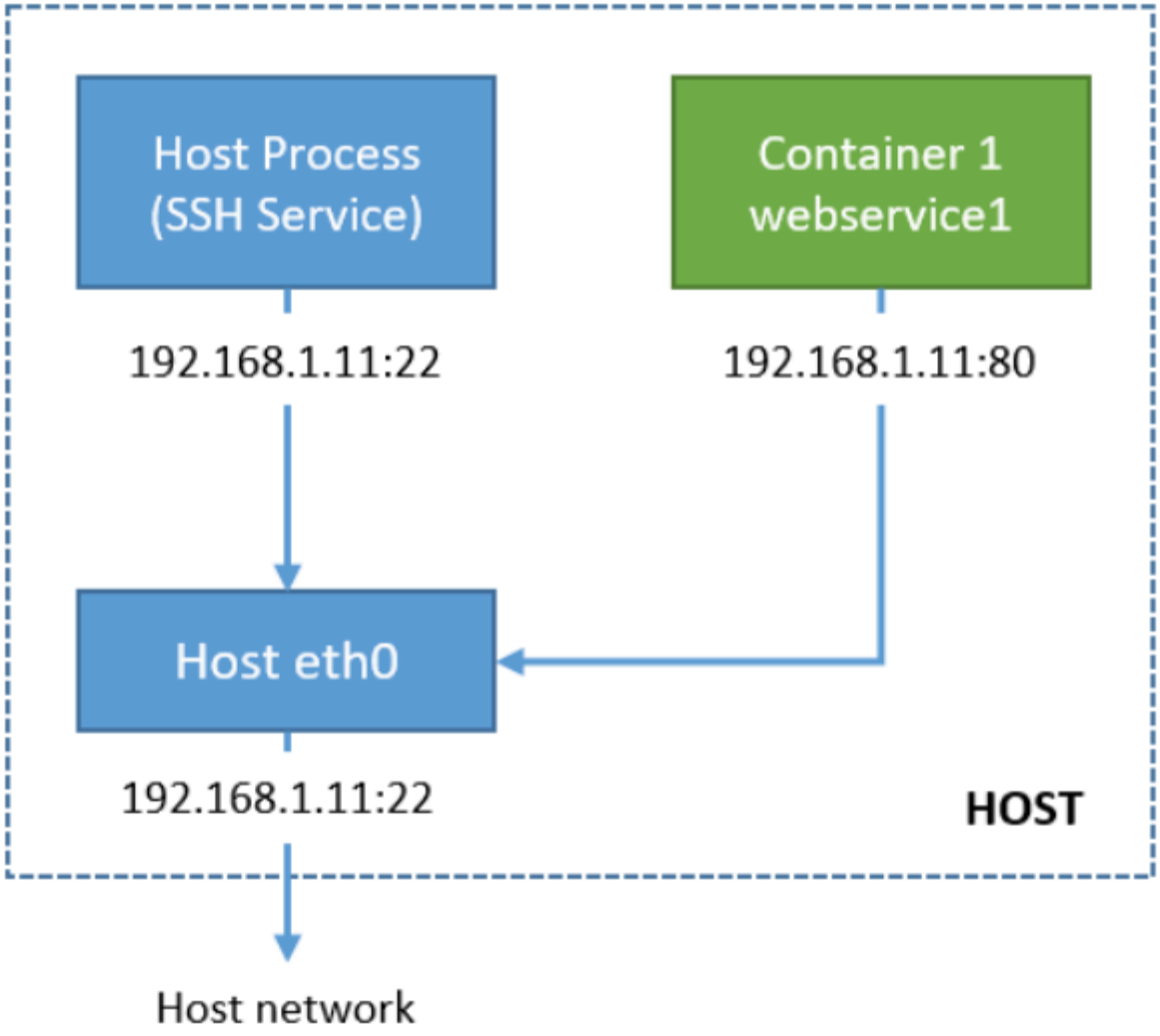
Bridge network mode utilizes Docker's built-in virtual network which runs inside each container. A bridge network is an internal network namespace in the host that allows all containers connected on the same bridge network to communicate. It provides isolation from other containers not connected to that bridge network. The Docker driver handles this isolation on the host machine so that containers on different bridge networks cannot communicate with each other.



This mode can take advantage of dynamic host port mappings as it allows you to run the same port (ex: port 80) on each container, and then map each container port to a different port on the host. However, this mode does not provide the best networking performance because the bridge network is virtualized and Docker software handles the traffic translations on traffic going in and out of the host.

Host network mode

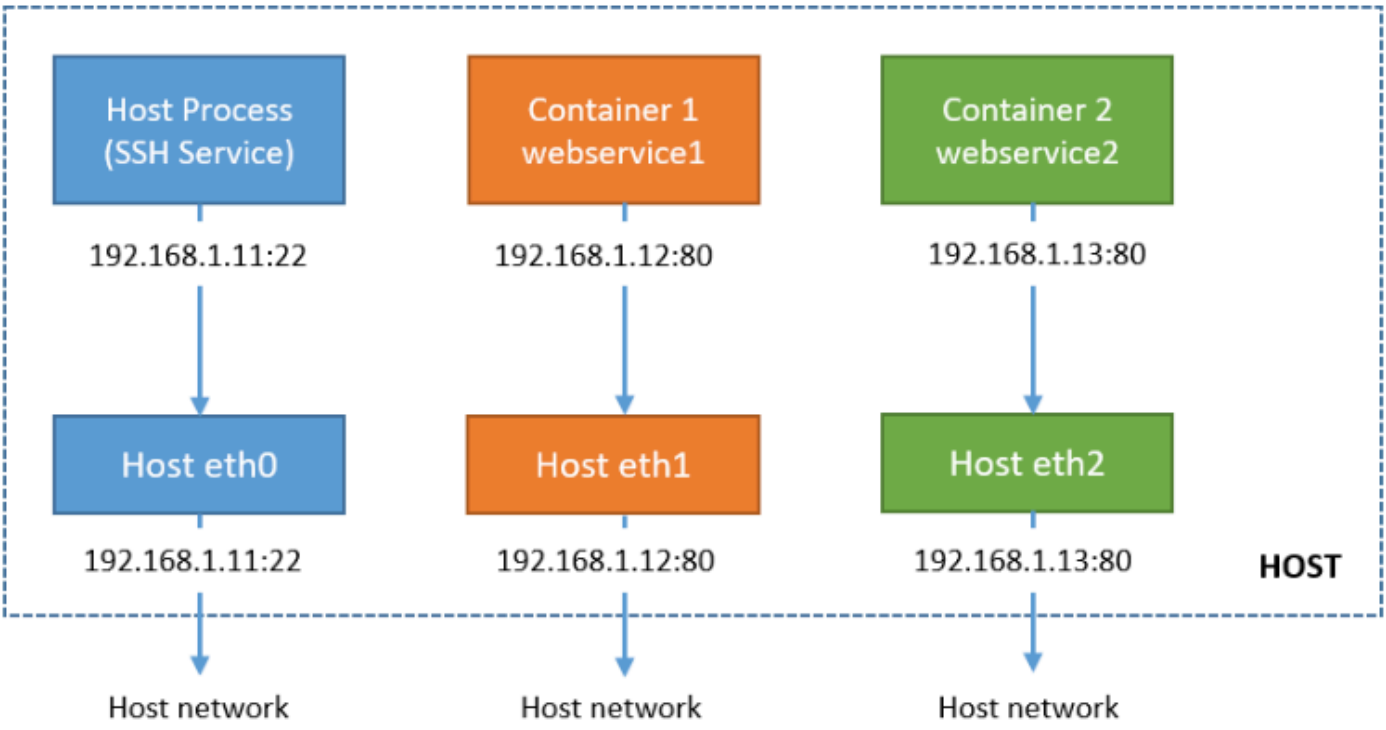
Host network mode bypasses the Docker's built-in virtual network and maps container ports directly to your EC2 instance's network interface. This mode shares the same network namespace of the host EC2 instance so your containers share the same IP with your host IP address. This also means that you can't have multiple containers on the host using the same port. A port used by one container on the host cannot be used by another container as this will cause conflict.



This mode offers faster performance than the bridge network mode since it uses the EC2 network stack instead of the virtual Docker network.

awsvpc mode

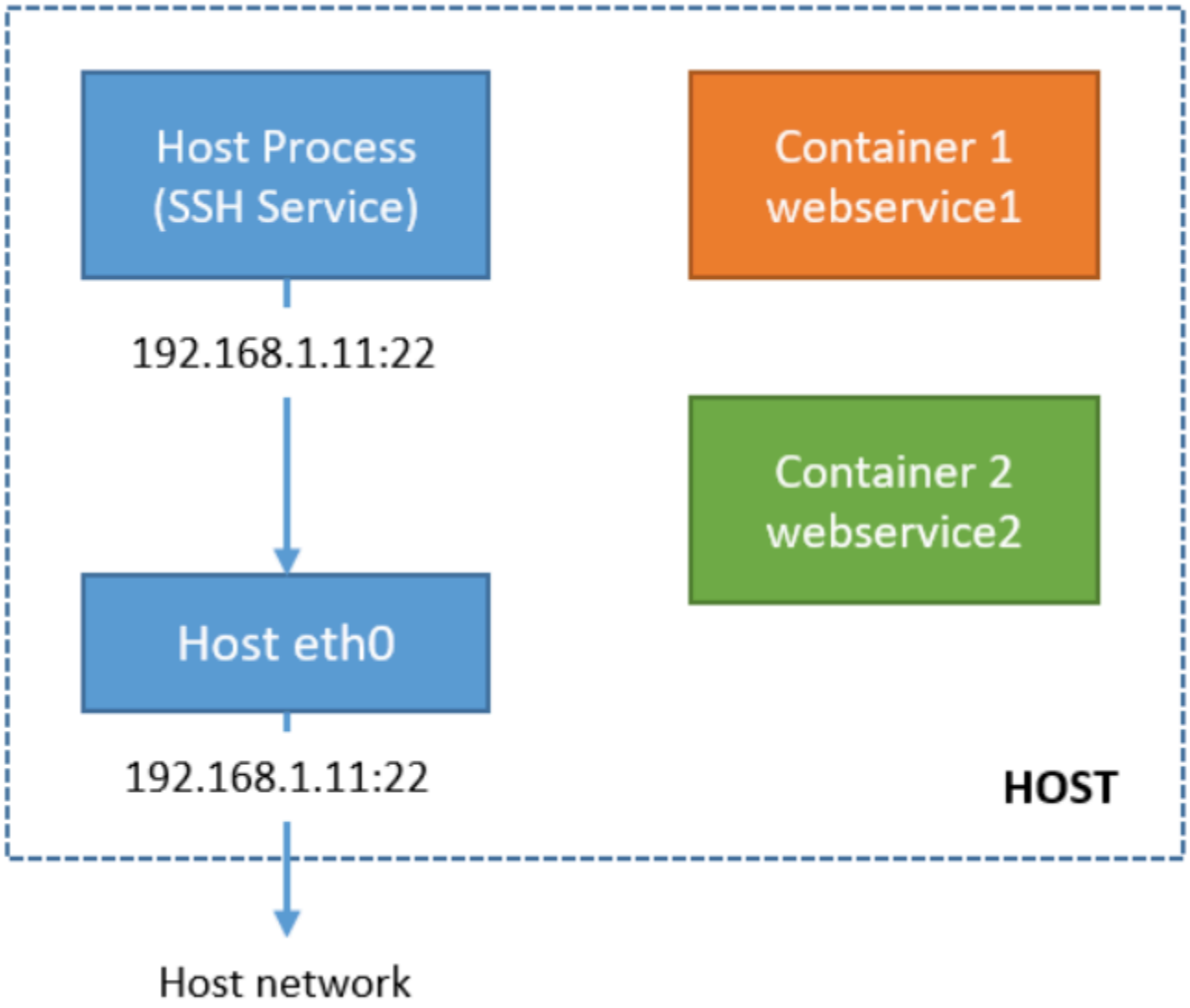
The **awsvpc** mode provides an elastic network interface for each task definition. If you have one container per task definition, each container will have its own elastic network interface and will get its own IP address from your VPC subnet IP address pool. This offers faster performance than the bridge network since it uses the EC2 network stack, too. This essentially makes each task act like their own EC2 instance within the VPC with their own ENI, even though the tasks actually reside on an EC2 host.



Awsvpc mode is recommended if your cluster will contain several tasks and containers as each can communicate with their own network interface. This is the only supported mode by the ECS Fargate service. Since you don't manage any EC2 hosts on ECS Fargate, you can only use awsvpc network mode so that each task gets its own network interface and IP address.

None network mode

This mode completely disables the networking stack inside the ECS task. The loopback network interface is the only one present inside each container since the loopback interface is essential for Linux operations. You can't specify port mappings on this mode as the containers do not have external connectivity.



You can use this mode if you don't want your containers to access the host network, or if you want to use a custom network driver other than the built-in driver from Docker. You can only access the container from inside the EC2 host with the Docker command.

References:
https://docs.aws.amazon.com/AmazonECS/latest/developerguide/task_definition_parameters.html#network_mode
https://docs.aws.amazon.com/AmazonECS/latest/developerguide/task-networking.html
https://docs.aws.amazon.com/AmazonECS/latest/userguide/fargate-task-networking.html

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	Yes	Yes	Yes
Logging	Yes	Yes	Yes
Zonal Failover	Yes	Yes	Yes
Connection Draining (deregistration delay)	Yes	Yes	Yes
Load Balancing to multiple ports on the same instance	Yes	Yes	Yes
IP addresses as targets	Yes	Yes (TCP, TLS)	Yes
Load Balancer deletion protection	Yes		
Configuration idle connection timeout	Yes		
Cross-zone load balancing	Yes	Yes	Yes
Sticky sessions	Yes	Yes	Yes
Static IP		Yes	
Elastic IP address		Yes	
Preserve Source IP address	Yes	Yes	Yes
Resource-based IAM permissions	Yes	Yes	Yes
Slow start	Yes		
Web sockets	Yes	Yes	Yes

PrivateLink Support		Yes (TCP, TLS)	Yes (GWLBE)
Source IP address CIDR-based routing	Yes		
Layer 7			
Path-based routing	Yes		
Host-based routing	Yes		
Native HTTP/2	Yes		
Redirects	Yes		
Fixed response	Yes		
Lambda Functions as targets	Yes		
HTTP header-based routing	Yes		
HTTP method-based routing	Yes		
Query string parameter-based routing	Yes		
Security			
SSL offloading	Yes	Yes	
Server Name Indication (SNI)	Yes	Yes	
Back-end server encryption	Yes	Yes	
User authentication	Yes		
Session Resumption	Yes	Yes	
Terminates flow/ proxy behavior	Yes	Yes	Yes

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.

S3 Pre-Signed URLs vs CloudFront Signed URLs vs Origin Access Control

S3 Pre-signed URLs	CloudFront Signed URLs	Origin Access Control (OAC)
All S3 buckets and objects, by default, are **private**. Only the object owner has permission to access these objects. Pre-signed URLs use the owner's security credentials to grant others time-limited permission to download or upload objects.	You can control user access to your private content in two ways • Restrict access to files in CloudFront edge caches • Restrict access to files in your Amazon S3 bucket (unless you've configured it as a website endpoint).	You can configure an S3 bucket as the origin of a CloudFront distribution. OAC prevents users from viewing your S3 files by simply using the direct URL for the file. Instead, they would need to access it through a CloudFront URL.
When creating a pre-signed URL, you (as the owner) need to provide the following: • Your security credentials • An S3 bucket name • An object key • Specify the HTTP method (GET to download the object or PUT to upload an object) • Expiration date and time of the URL.	You can configure CloudFront to require that users access your files using either **signed URLs** or **signed cookies**. You then develop your application either to create and distribute signed URLs to authenticated users or to send Set-Cookie headers that set signed cookies on the viewers for authenticated users. When you create signed URLs or signed cookies to control access to your files, you can specify the following restrictions: • An expiration date and time for the URL • (Optional) The date and time the URL becomes valid • (Optional) The IP address or range of addresses of the computers that can be used to access your content You can use signed URLs or signed cookies for any CloudFront distribution, regardless of whether the origin is an Amazon S3 bucket or an HTTP server.	To require that users access your content through CloudFront URLs, you perform the following tasks: • Create a special CloudFront user called an **origin access control**. • Give the origin access control permission to read the files in your bucket. • Remove permission for anyone else to use Amazon S3 URLs to read the files (through bucket policies or ACLs). You cannot set OAC if your S3 bucket is configured as a website endpoint.

S3 Transfer Acceleration vs Direct Connect vs VPN

S3 Transfer Acceleration (TA)

- Amazon S3 Transfer Acceleration makes public Internet transfers to S3 faster, as it leverages Amazon CloudFront's globally distributed AWS Edge Locations.
- There is no guarantee that you will experience increased transfer speeds. If S3 Transfer Acceleration is not likely to be faster than a regular S3 transfer of the same object to the same destination AWS Region, AWS will not charge for the use of S3 TA for that transfer.
- This is not the best transfer service to use if transfer disruption is not tolerable.
- S3 TA provides the same security benefits as regular transfers to Amazon S3. This service also supports multi-part upload.
- ***S3 TA vs Direct Connect***
    - AWS Direct Connect is a good choice for customers who have a private networking requirement or who have access to AWS Direct Connect exchanges. S3 Transfer Acceleration is best for submitting data from distributed client locations over the public Internet, or where variable network conditions make throughput poor.
- ***S3 TA vs VPN***
    - You typically use (IPsec) VPN if you want your resources contained in a private network. VPN tools such as OpenVPN allow you to set up stricter access controls if you have a private S3 bucket. You can complement this further with the increased speeds from S3 TA.

AWS Direct Connect

- Using AWS Direct Connect, data that would have previously been transported over the Internet can now be delivered through a **private physical network connection** between AWS and your datacenter or corporate network. Customers' traffic will remain in AWS global network backbone, after it enters AWS global network backbone.
- Benefits of Direct Connect vs internet-based connections
    - reduced costs
    - increased bandwidth
    - a more consistent network experience
- Each AWS Direct Connect connection can be configured with one or more **virtual interfaces**. Virtual interfaces may be configured to access AWS services such as Amazon EC2 and Amazon S3 using public IP space, or resources in a VPC using private IP space.
- You can run IPv4 and IPv6 on the same virtual interface.
- Direct Connect does not support multicast.
- A Direct Connect connection is **not redundant**. Therefore, a second line needs to be established if redundancy is required. Enable *Bidirectional Forwarding Detection* (BFD) when configuring your connections to ensure fast detection and failover.
- AWS Direct Connect offers SLA.
- Direct Connect vs IPsec VPN
    - A VPC VPN Connection utilizes IPSec to establish **encrypted network connectivity** between your intranet and Amazon VPC **over the Internet.** VPN Connections can be configured in minutes and are a good solution if you have an immediate need, have low to modest bandwidth

requirements, and can tolerate the inherent variability in Internet-based connectivity. AWS Direct Connect **does not involve the Internet**; instead, it uses **dedicated, private network connections** between your intranet and Amazon VPC.

- You can combine one or more Direct Connect dedicated network connections with the Amazon VPC VPN. This combination provides an IPsec-encrypted private connection that also includes the benefits of Direct Connect.

AWS VPN

- AWS VPN is comprised of two services:
  - AWS Site-to-Site VPN enables you to securely connect your on-premises network or branch office site to your Amazon VPC.
  - AWS Client VPN enables you to securely connect users to AWS or on-premises networks.
- Data transferred between your VPC and datacenter routes over an encrypted VPN connection to help maintain the confidentiality and integrity of data in transit.
- If data that passes through Direct Connect moves in a dedicated private network line, AWS VPN instead encrypts the data before passing it through the Internet.
- VPN connection throughput can depend on multiple factors, such as the capability of your customer gateway, the capacity of your connection, average packet size, the protocol being used, TCP vs. UDP, and the network latency between your customer gateway and the virtual private gateway.
- All the VPN sessions are **full-tunnel VPN**. (cannot split tunnel)
- AWS Site-to-Site VPN enables you to create **failover** and CloudHub solutions **with AWS Direct Connect**.
- AWS Client VPN is designed to connect devices to your applications. It allows you to choose from an **OpenVPN-based client**.

Backup and Restore vs Pilot Light vs Warm Standby vs Multi-site

Backup and Restore	Pilot Light
• This DR plan provides the slowest system restoration after a DR event.	• The pilot light method gives you a quicker recovery time than the backup-and-restore method because the core pieces of the system are already running and are continually kept up to date, but is not as fast as Warm Standby.
• You take frequent snapshots of your data such as those in Amazon EBS Volumes and Amazon RDS databases, and you store them in a durable and secure storage location such as Amazon S3.	• You can maintain a pilot light by configuring and running the **most critical core elements** of your system in AWS. When the time comes for recovery, you can rapidly provision a full-scale production environment around the critical core.

• There are many ways for you to move data in and out of S3
  • Transfer over the network via S3 Transfer Acceleration
  • Transfer over a dedicated network line using AWS Direct Connect
	• Pilot light is an example of **active/passive failover configuration**.

• With S3 Glacier, you get to reduce a large portion of your costs compared to using S3 Standard, since Glacier is meant for long term archival storage which is perfect for backups.	• Infrastructure elements for the pilot light itself typically include your database servers, which would be configured for data mirroring replication.
• AWS Storage Gateway enables snapshots of your on-premises data volumes to be transparently copied into S3 for backup. • Storage-cached volumes allow you to store your primary data in S3, but keep your frequently accessed data local for low-latency access.	• Restoring the rest of the system includes utilizing EBS snapshots and EC2 AMIs that you should be regularly generating.
• Gateway-VTL of AWS Storage Gateway serves as a replacement for traditional magnetic tape backup.	• Pilot light tends to be more costly than backup and restore since you leave a few core AWS resources running all the time.
• You can quickly create local volumes or Amazon EBS volumes from snapshots in S3.	• From a networking point of view, you have two main options for provisioning web servers: • Use Elastic IP addresses, which can be pre-allocated and pre-identified, and associate them with your instances.

	• Use Elastic Load Balancing (ELB) to distribute traffic to multiple instances. You would then update your DNS records to point at your EC2 instance or point to your load balancer using a CNAME.

• You can create AMIs out of your EC2 instances which preserve the following:
  • A template for the root volume for the instance (for example, an operating system, an application server, and applications)
  • Launch permissions that control which AWS accounts can use the AMI to launch instances
  • A block device mapping that specifies the volumes to attach to the instance when it's launched
	• Consider redundancy especially at your data layer (enable multi-AZ, cluster sharding, etc).

• Backup and restore is used in combination with other DR plans since it is crucial to always have a working backup of your system.	• If your data is constantly changing and failover occurs, you would have to reverse replicate your data in the DR site back to the primary site, so that any data updates received while the primary site was down can be replicated back, without the loss of data.
Warm Standby	**Multi-site**
• This DR plan is faster in system restoration than performing Pilot Light after a DR event, but is not as fast as having a Multi-site System.	• This DR plan is the fastest in system restoration during a DR event.
• Warm standby describes a DR scenario in which a scaled-down version of a fully functional environment is always running in the cloud.	• Multi-site is a one-to-one copy of your infrastructure that is located and running in another region or AZ, known as an **active-active configuration**.
• Since it is not only your core elements that are running all the time, warm standby is usually more costly than pilot light.	• Because of this, multi-site is the most expensive among all DR plans.
• Warm standby is another example of **active/passive failover configuration**.	• Multi-site gives you the best RTO and RPO as no downtime is expected and little to no data loss should be experienced.

• Servers can be left running in a minimum number of EC2 instances on the smallest sizes possible. Once failover occurs, quickly resize them and add scaling capabilities. It is best to place these instances behind a load balancer as well.	• In addition to recovery point options, there are various replication methods, such as synchronous and asynchronous methods.
• For the data layer, the practice is similar to pilot light where a standby resource is present and changing data is constantly being replicated to the other.	• You can use a DNS service that supports weighted routing, such as Amazon Route 53, to route production traffic to different sites that deliver the same application or service.
• In the case of failure of the production system, the standby environment will be scaled up for production load , and DNS records will be changed to route all traffic to AWS.	• During failover, you can quickly increase compute capacity by using AWS Auto Scaling or by resizing your instances to a larger size.
• If your data is constantly changing and failover occurs, you would have to reverse replicate your data in the DR site back to the primary site, so that any data updates received while the primary site was down can be replicated back, without the loss of data.	• Multiple services in AWS such as RDS offer a multi-AZ feature which allows you to provision resources in a different location for a more fault-tolerant setup.
	• If your data is constantly changing and failover occurs, you would have to reverse replicate your data in the DR site back to the primary site, so that any data updates received while the primary site was down can be replicated back, without the loss of data.

FINAL REMARKS AND TIPS

That's a wrap! Thank you once again for choosing our Study Guide and Cheat Sheets for the AWS Certified Solutions Architect Professional (SAP-C02) exam. The Tutorials Dojo team spent considerable time and effort to produce this content to help you pass the AWS exam.

We also recommend that before taking the actual SAP-C02 exam, allocate some time to check your readiness by taking our **AWS practice tests** in the Tutorials Dojo Portal. This will help you identify the topics that you need to improve on and help reinforce the concepts that you need to fully understand in order to pass this certification exam. It also has different training modes that you can choose from such as Timed mode, Review mode, Section-Based tests, and Final test plus bonus flashcards. In addition, you can read the technical discussions in our forums or post your queries if you have one. If you have any issues, concerns or constructive feedback on our eBook, feel free to contact us at support@tutorialsdojo.com.

On behalf of the Tutorials Dojo team, we wish you all the best on your upcoming AWS Certified Solutions Architect Professional exam. May it help advance your career, as well as increase your earning potential.

With the right strategy, hard work, and unrelenting persistence, you can definitely make your dreams a reality! You can make it!

Sincerely,
Jon Bonso and the Tutorials Dojo Team

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.