JON BONSO AND ADRIAN FORMARAN

AWS CERTIFIED SOLUTIONS ARCHITECT PROFESSIONAL

Tutorials Dojo
Study Guide and Cheat Sheets

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, and cost-effective AWS cloud architectures. 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, and design new solutions to meet the 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. This eBook contains the essential concepts, exam domains, exam tips, sample questions, cheat sheets, and other relevant information about the AWS Certified Solutions Architect – Professional exam. It begins with the presentation of the exam structure, giving 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 AWS exam guide to structure the contents of this guide, where each section discusses a particular exam domain. Various AWS concepts, related AWS services, and technical implementations are covered to provide you an idea of what to expect on the actual exam.

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 hands-on sessions and practice exams to further expand your knowledge and improve your test taking skills.*

AWS CERTIFIED SOLUTIONS ARCHITECT PROFESSIONAL EXAM OVERVIEW

Exam Details

The AWS Certified Solutions Architect - Professional certification exam validates various skills which are necessary to become a full-fledged AWS Solutions Architect. The exam will check your capability in implementing and managing continuous delivery systems and methodologies on AWS. Automating security controls, validating compliance and optimizing the governance processes are also included in the test.

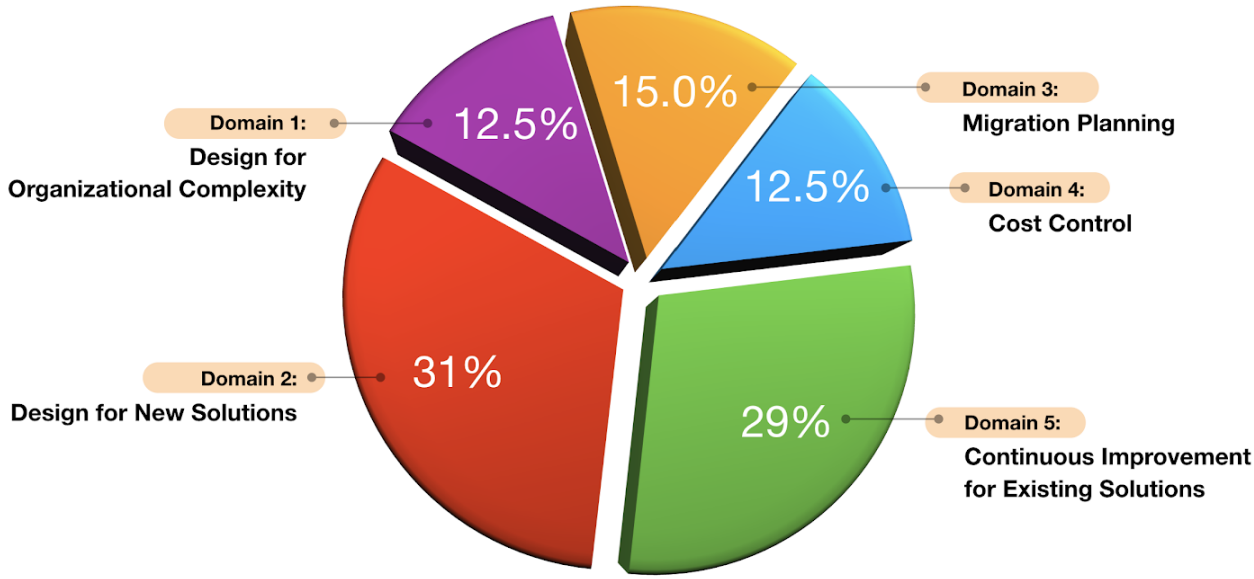
It is composed of scenario-based questions that can either be in multiple-choice or multiple response formats. The first question type has one correct answer and three incorrect responses, while the latter has two or more correct responses out of five or more options. You can take the exam from a local testing center or online from the comforts of your home.

Exam Code:	SAP-C01
Release Date:	February 2019
Prerequisites:	None
No. of Questions:	75
Score Range:	100 - 1000
Cost:	300 USD (Practice exam: 40 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.

Don't be confused if you see in your Pearson Vue booking that the duration is 190 minutes since they included an additional 10 minutes for reading the Non-Disclosure Agreement (NDA) at the start of the exam and the survey at the end of it. If you booked via PSI, the exam duration time that you will see is 180 minutes.

Exam Domains

The AWS Certified Solutions Architect Professional (SAP-C01) exam has 5 different domains, each with corresponding weight and topic coverage. The exam domains are as follows: Design for Organizational Complexity (12.5%), Design for New Solutions (31%), Migration Planning (15%), Cost Control (12.5%), and lastly, Continuous Improvement for Existing Solutions (29%):



Domain 1: Design for Organizational Complexity

1.1 Determine cross-account authentication and access strategy for complex organizations (for example, an organization with varying compliance requirements, multiple business units, and varying scalability requirements)
1.2 Determine how to design networks for complex organizations (for example, an organization with varying compliance requirements, multiple business units, and varying scalability requirements)
1.3 Determine how to design a multi-account AWS environment for complex organizations (for example, an organization with varying compliance requirements, multiple business units, and varying scalability requirements)

Domain 2: Design for New Solutions

2.1 Determine security requirements and controls when designing and implementing a solution
2.2 Determine a solution design and implementation strategy to meet reliability requirements

2.3 Determine a solution design to ensure business continuity
2.4 Determine a solution design to meet performance objectives
2.5 Determine a deployment strategy to meet business requirements when designing and implementing a solution

Domain 3: Migration Planning

3.1 Select existing workloads and processes for potential migration to the cloud
3.2 Select migration tools and/or services for new and migrated solutions based on detailed AWS knowledge
3.3 Determine a new cloud architecture for an existing solution
3.4 Determine a strategy for migrating existing on-premises workloads to the cloud

Domain 4: Cost Control

4.1 Select a cost-effective pricing model for a solution
4.2 Determine which controls to design and implement that will ensure cost optimization
4.3 Identify opportunities to reduce cost in an existing solution

Domain 5: Continuous Improvement for Existing Solutions

5.1 Troubleshoot solution architectures
5.2 Determine a strategy to improve an existing solution for operational excellence
5.3 Determine a strategy to improve the reliability of an existing solution
5.4 Determine a strategy to improve the performance of an existing solution
5.5 Determine a strategy to improve the security of an existing solution
5.6 Determine how to improve the deployment of an existing solution

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. Right after you completed the actual exam, you'll immediately see a pass or fail notification on the testing screen. A *"Congratulations! You have successfully passed..."* message will be shown if you passed the exam.

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

Within 5 business days of completing your exam, your AWS Certification Account will have a record of your complete exam results. The score report contains a table of your performance at each section/domain, which indicates whether you met the competency level required for these domains or not. AWS 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.

- **Free Practice Exam** - To help you prepare for your next exam, AWS provides another voucher that you can use to take any official AWS practice exam for free. You can access your voucher code from the "Benefits" section of your AWS Certification Account.

- **AWS Certified Store** - All AWS certified professionals will be given access to exclusive AWS Certified merchandise. You can get your store access from the "Benefits" section of your AWS Certification Account.

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

- **Eligibility to join AWS IQ -** With the AWS IQ program, you can monetize your AWS skills online by providing hands-on assistance to customers around the globe. AWS IQ will help you stay sharp and be well-versed on various AWS technologies. You can work at the comforts of your home and decide when or where you want to work. Interested individuals must be based in the US, have an Associate, Professional, or Specialty AWS Certification and be over 18 of age.

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

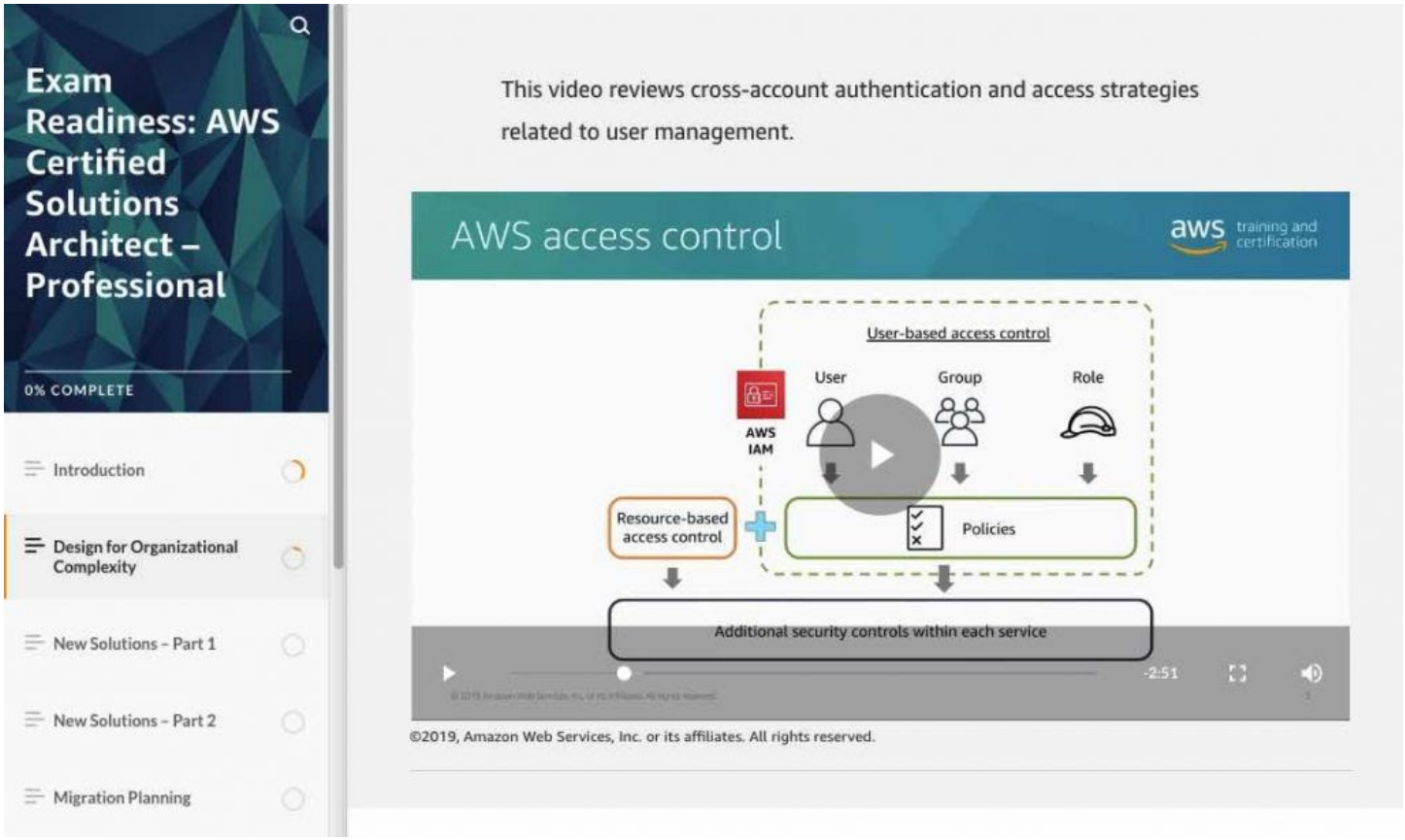
Few years ago, before you could take the AWS Certified Solutions Architect Professional exam (or SA Pro for short), you would first have to pass the associate level exam of this track. This is to ensure that you have sufficient knowledge and understanding on architecting in AWS, before tackling the more 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 if you have not done so yet. They contain very important materials such as review materials that will be crucial for passing the exam.

Study Materials

The FREE AWS Exam Readiness course, official AWS sample questions, Whitepapers, FAQs, AWS Documentation, Re:Invent videos, forums, labs, AWS cheat sheets, AWS practice exams, and personal 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. 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 guide, you should also study the following:

1. Securing Data at Rest with Encryption
2. Web Application Hosting in the AWS Cloud
3. Migrating AWS Resources to a New Region
4. Practicing Continuous Integration and Continuous Delivery on AWS Accelerating Software Delivery with DevOps
5. Microservices on AWS
6. AWS Security Best Practices
7. AWS Well-Architected Framework
8. Architecting for the Cloud AWS Best Practices
9. Amazon Web Services: Overview of Security Processes
10. Using Amazon Web Services for Disaster Recovery
11. AWS Architecture Center architecture whitepapers

The instructor-led classroom called "Advanced Architecting on AWS" should also provide additional information on how to implement the concepts and best practices that you have learned from whitepapers and other forms of documentation. Be sure to check it out.

Your AWS exam could also include a lot of migration scenarios. Visit this AWS documentation to learn about the different ways of performing cloud migration.

AWS Services to Focus On

Generally, as a soon-to-be AWS Certified SA Pro, you should 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**
    a. Know how to create organizational units (OUs), service control policies (SCPs), and any additional parameters in AWS Organizations.
    b. 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.
    c. Differentiate SCP from IAM policies.
    d. 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.
    e. 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 Server Migration Services**
    a. Study the different ways to migrate on-premises servers to the AWS Cloud.
    b. Also study how you can perform the migration in a secure and reliable manner.
    c. You should be aware of what types of objects AWS SMS can migrate for you i.e. VMs, and what is the output of the migration process.
3. **AWS Database Migration Service + Schema Conversion Tool**
    a. 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.
    b. Read over what schemas can be converted by SCT.
4. **AWS Serverless Application Model**
    a. The AWS SAM has a syntax of its own. Study the syntax and how AWS SAM is used to deploy serverless applications through code.
    b. Know the relationship between SAM and CloudFormation. **Hint:** You can use these two together.
5. **AWS EC2 Systems Manager**
    a. 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 Cloudwatch Events.

b. It is also important to know how you can troubleshoot EC2 issues using Systems Manager.

c. 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:
   a. CodeCommit
   b. CodeBuild
   c. CodeDeploy
   d. CodePipeline
7. **AWS Service Catalog**
   a. 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.
   b. You can enforce tagging on services using service catalog. This way, users can only launch resources that have the tags you defined.
   c. 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)**
   a. You should have a deep understanding of this service. Questions commonly include Direct Connect Gateway, public and private VIFs, and LAGs.
   b. 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.
   c. 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.
   d. 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:
   a. You can use CloudFormation to enforce tagging by requiring users to only use resources that CloudFormation launched.
   b. CloudFormation can be used for managing resources across different AWS accounts in an Organization using StackSets.
   c. 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)**
    a. 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.
    b. Study about transit gateway and how it can be used together with Direct Connect.
    c. Remember longest prefix routing.
    d. 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**
    a. Differentiate task role from task execution role.
    b. Compare using ECS compute instances from the Fargate serverless model.
    c. 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)**
    a. Differentiate the Internet protocols used by each type of ELB for listeners and target groups: HTTP, HTTPS, TCP, UDP, TLS.
    b. Know how you can configure load balancers to forward client IP to target instances.
    c. 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**
    a. Study the different deployment options for Elastic Beanstalk.
    b. Know the steps in performing a blue/green deployment.
    c. Know how you can use traffic splitting deployment to perform canary testing
    d. Compare Elastic Beanstalk's deployment options to CodeDeploy.
14. **WAF and Shield**
    a. Know at what network layer WAF and Shield operate in
    b. 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**
    a. 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.
    b. 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**
    a. Know your caching options especially when it comes to databases.

   b. If there is a feature that is readily integrated with the database, it would be better to use that integrated feature instead for less overhead.
18. **Snowball Edge vs Direct Connect vs S3 Acceleration** - 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 Solutions Architects 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.

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.

Sample Practice Test Questions:

Question 1

The AWS resources in your production account is shared among various business units of the company. A single 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?

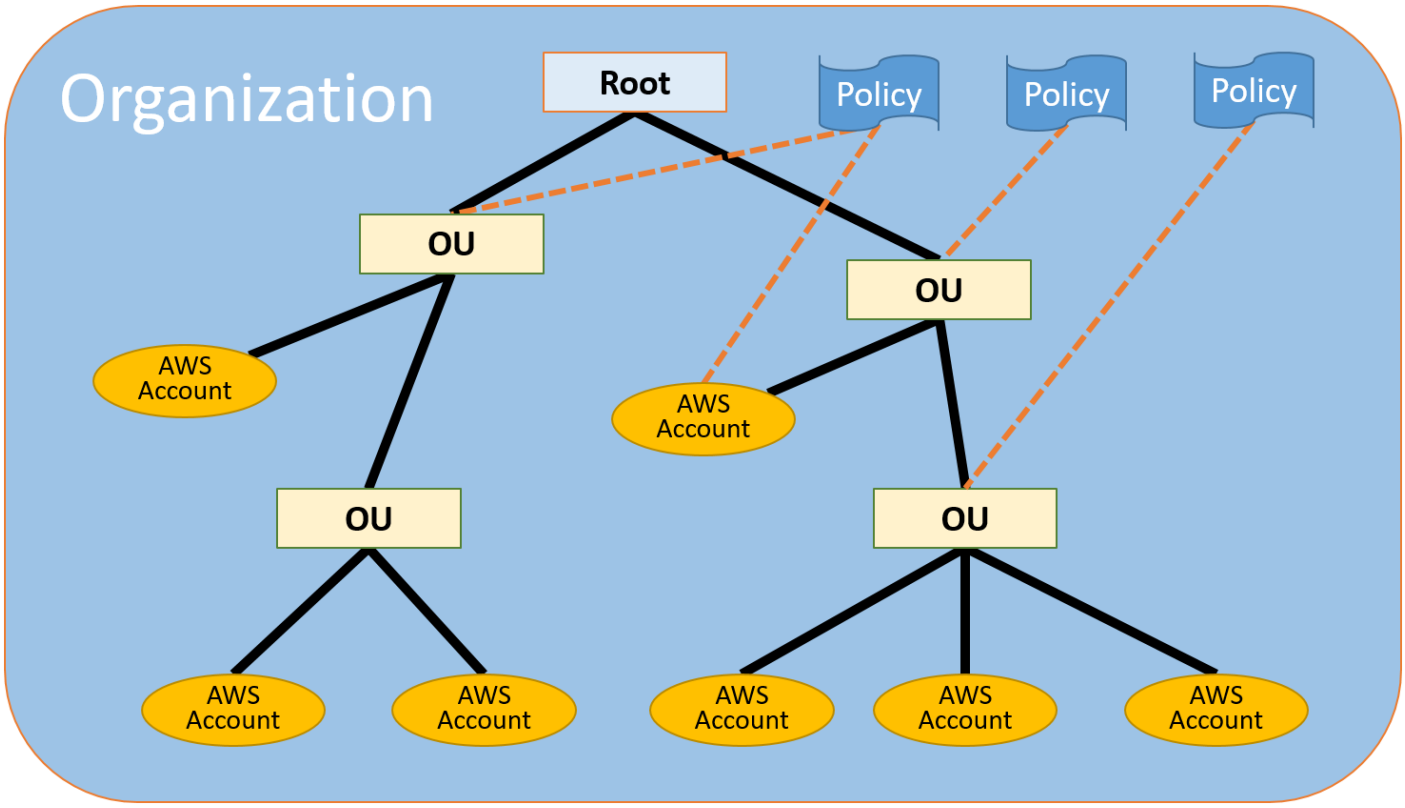
1. Use AWS Organizations to centrally manage all of your accounts. Group your accounts, which belong to a specific business unit, to individual Organization Unit (OU). Create an IAM Role in the production account for each business unit which has a policy that allows access to the EC2 instances including 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 belong 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 account of the OU.
4. Use AWS Organizations to centrally manage all of your accounts. Group your accounts, which belong 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.

Option 1 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.

Options 2 and 4 are incorrect because an SCP policy simply specifies the services and actions that users and roles can use in the accounts. SCPs are similar to IAM permission policies except that they don't grant any permissions.

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-cheat-sheet-aws-organizations/

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

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

Question 2

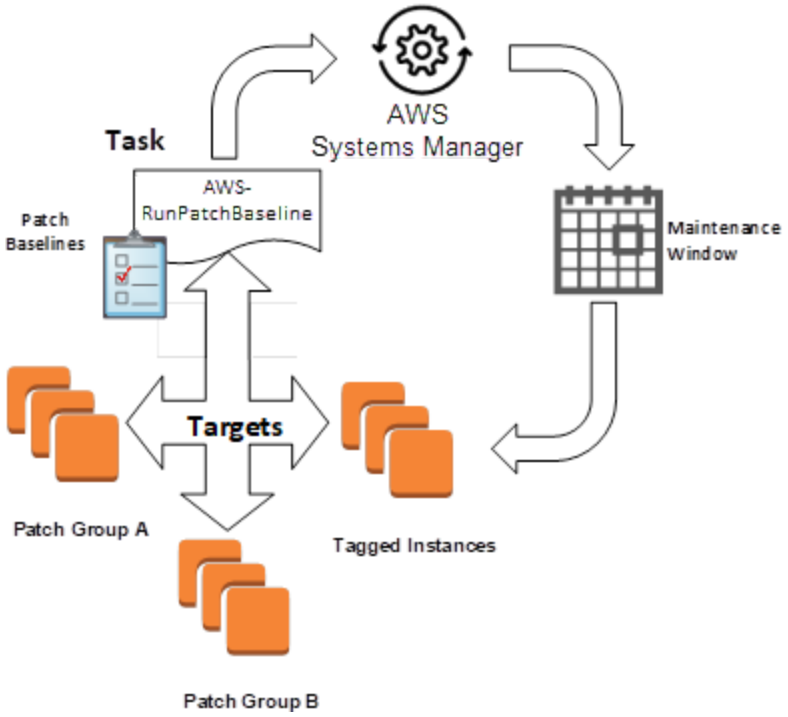
A known security vulnerability was discovered in the outdated Operating System of your company's EC2 fleet. As the Systems Administrator, you are responsible for mitigating the vulnerability as soon as possible to safeguard your systems from various cyber security attacks. In addition, you are also required to record all of the changes to patch and association compliance statuses.

What is the most efficient way to solve this issue?

1. Configure the EC2 fleet to automatically install the security OS patch every week on the provided maintenance window.
2. Use AWS Systems Manager and AWS Config to manage, record, and deploy the security patches for the OS for the entire fleet of EC2 instances.
3. Set up Amazon QuickSight and Kibana to apply, monitor, and visualize the patch statuses of all EC2 instances.
4. Use AWS Systems Manager and Amazon ES to manage, record, and deploy the security patches for the OS for the entire fleet of EC2 instances.

Correct Answer: 2

AWS Systems Manager Patch Manager automates the process of patching managed instances with security-related updates. For Linux-based instances, you can also install patches for non-security updates. You can patch fleets of Amazon EC2 instances or your on-premises servers and virtual machines (VMs) by operating system type. This includes supported versions of Windows, Ubuntu Server, Red Hat Enterprise Linux (RHEL), SUSE Linux Enterprise Server (SLES), Amazon Linux, and Amazon Linux 2. You can scan instances to see only a report of missing patches, or you can scan and automatically install all missing patches.



Since you are also required to record all of the changes to patch and association compliance statuses, you can use AWS Config to meet this requirement. AWS Config is a service that enables you to assess, audit, and evaluate the configurations of your AWS resources. Config continuously monitors and records your AWS resource configurations and allows you to automate the evaluation of recorded configurations against desired configurations.

Option 1 is incorrect because the EC2 Spot Fleet does not have a built-in function to automatically install the security OS patch every week on the provided maintenance window.

Option 3 is incorrect because QuickSight and Kibana are primarily used for data visualization and not for patch management. You can use Amazon Elasticsearch (ES) with Kibana but this service is not suitable for this scenario.

Option 4 is incorrect because the Amazon Elasticsearch Service (Amazon ES) is just an AWS-managed service that makes it easy to deploy, operate, and scale Elasticsearch clusters in the AWS Cloud. Elasticsearch is a popular open-source search and analytics engine for use cases such as log analytics, real-time application monitoring, and clickstream analysis. This service is not helpful in this scenario since the task is to manage the security patches of your EC2 instances.

References:
https://docs.aws.amazon.com/systems-manager/latest/userguide/systems-manager-patch.html
https://aws.amazon.com/config/

Check out this AWS Systems Manager Cheat Sheet:
https://tutorialsdojo.com/aws-cheat-sheet-aws-systems-manager/

Domain 1: Design 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 with AWS become more complex. You need to be able to segregate these groups according to purpose and simplify each of these groups' 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 catastrophe for the organization.

Around 12.5% of questions in the actual exam revolve around these topics.

- Determine cross-account authentication and access strategy for complex organizations (for example, an organization with varying compliance requirements, multiple business units, and varying scalability requirements)
- Determine how to design networks for complex organizations (for example, an organization with varying compliance requirements, multiple business units, and varying scalability requirements)
- Determine how to design a multi-account AWS environment for complex organizations (for example, an organization with varying compliance requirements, multiple business units, and varying scalability requirements)

In this chapter, we will cover the related topics for Complex Organizational Designs 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.

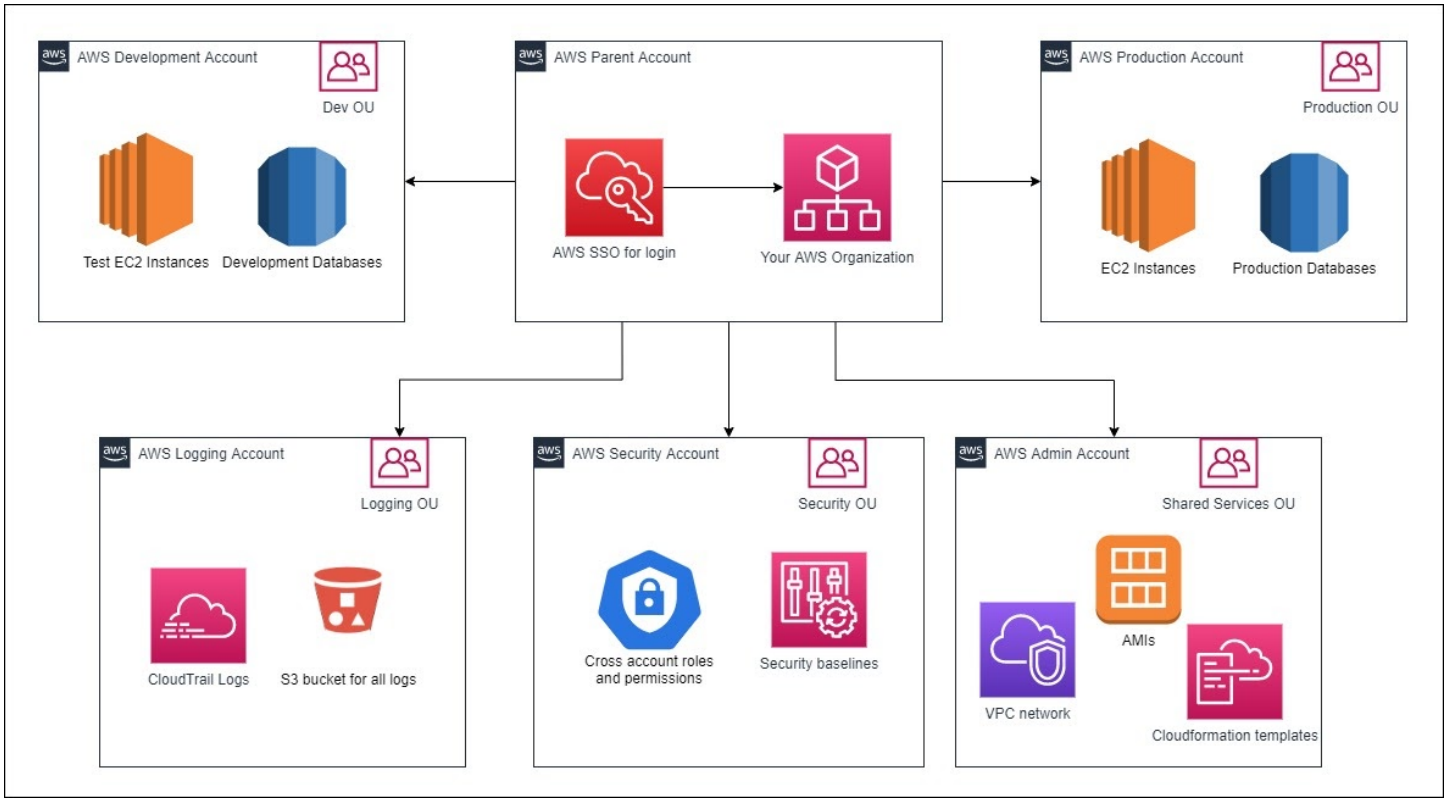


Figure: Example Setup of Landing Zone

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.

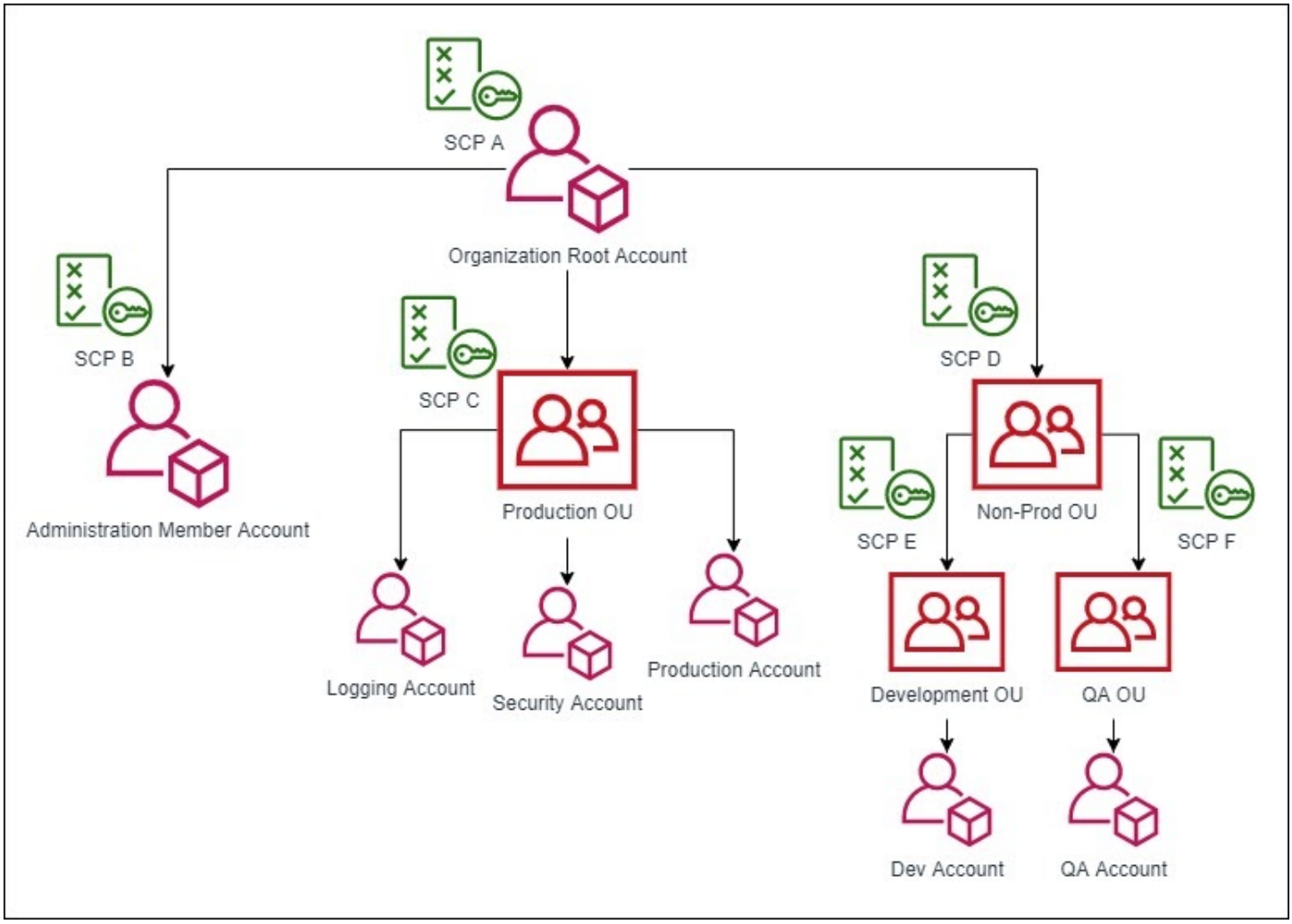


Figure: Example Structure of an AWS Organization with Multiple Accounts, OUs and SCPs

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.

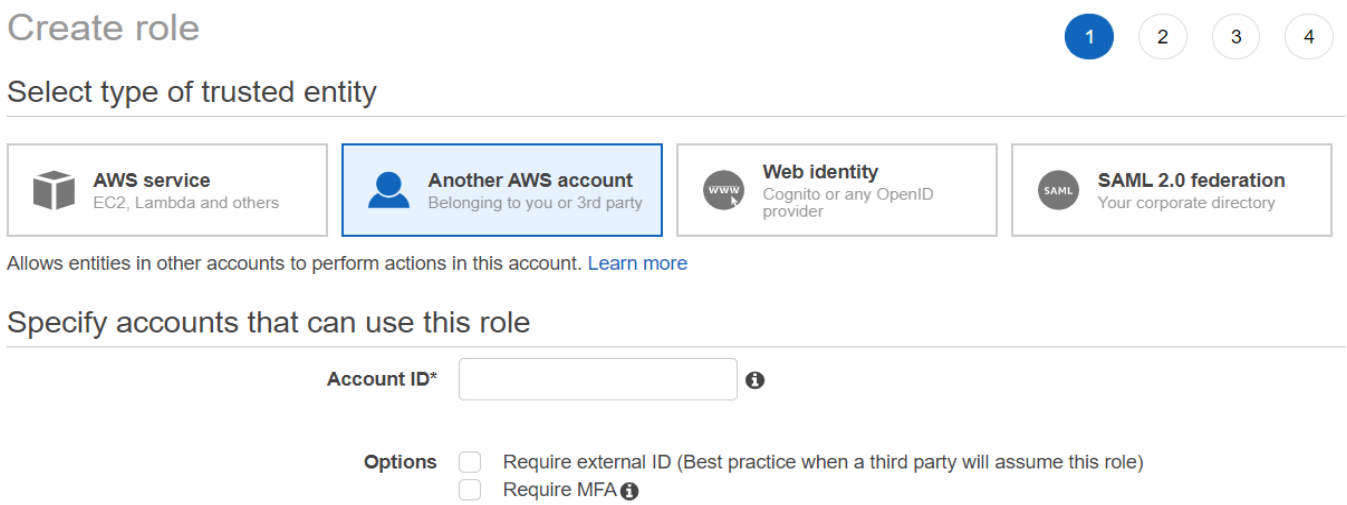


Figure: Create a cross-account role in IAM

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.

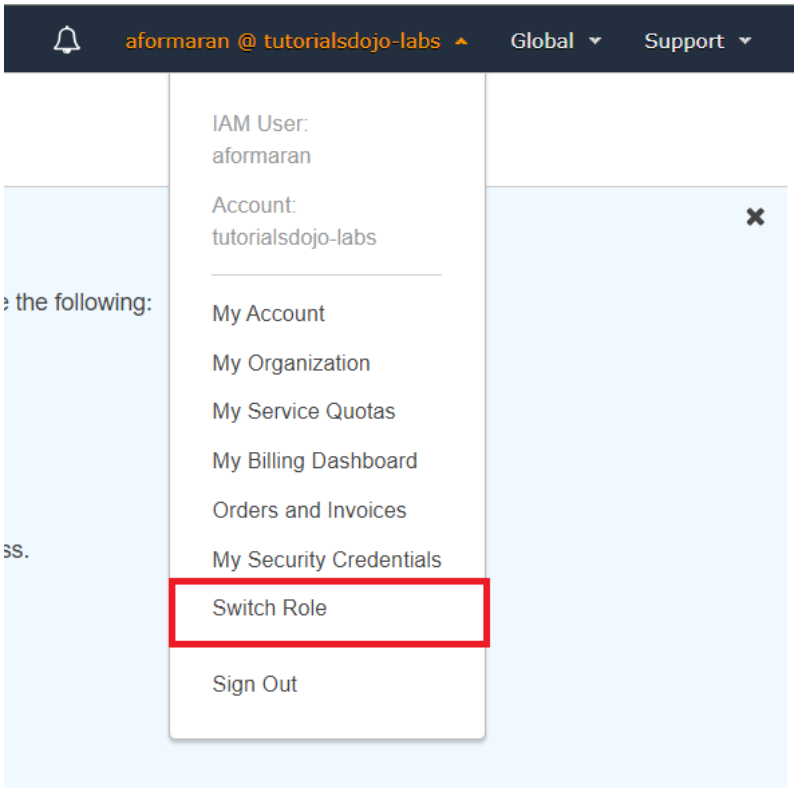


Figure: How to switch role in the AWS Console

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.

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 Single Sign-on (SSO) 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://d0.awsstatic.com/aws-answers/AWS_Multi_Account_Security_Strategy.pdf
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.

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/

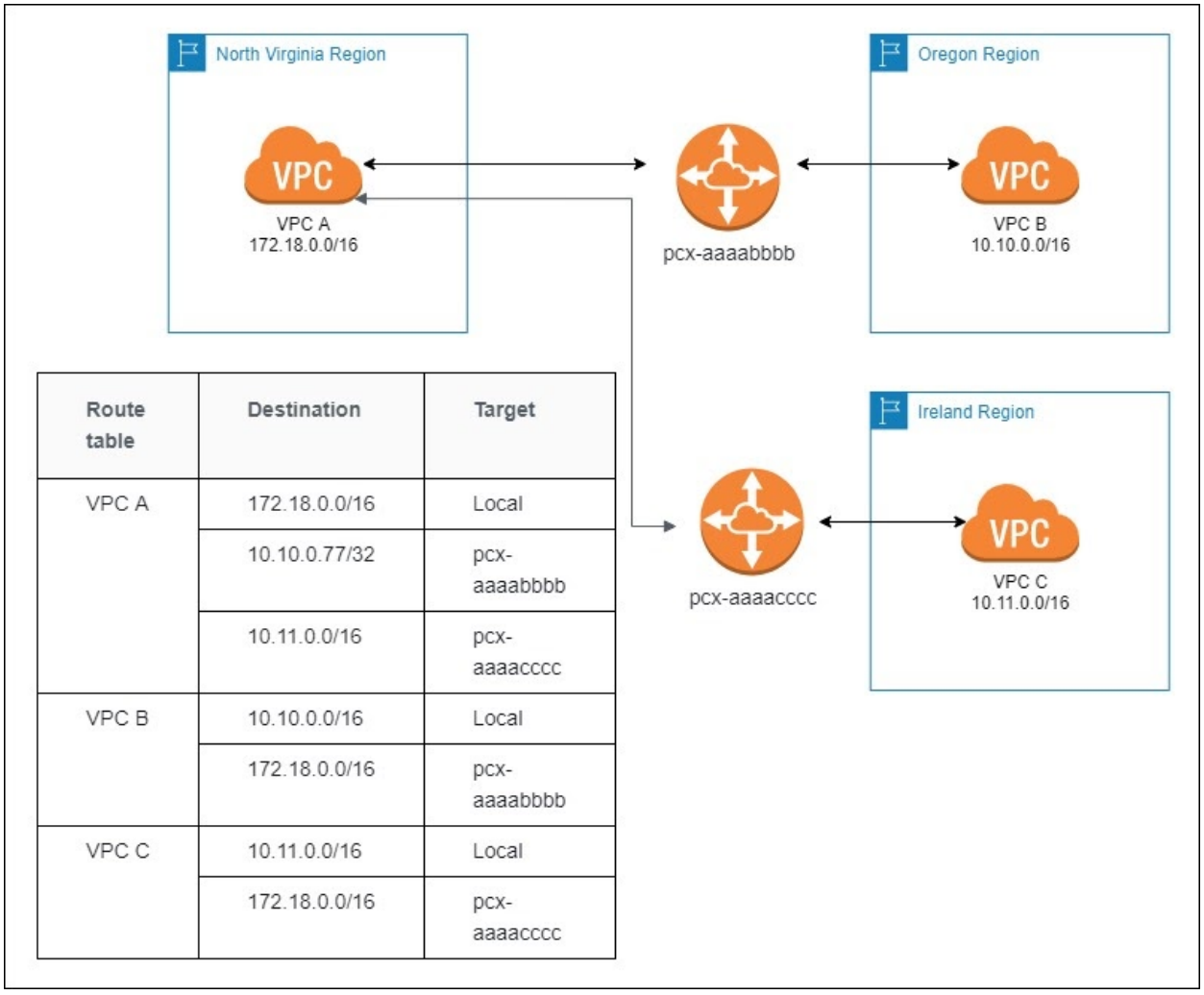
Multi-Account Network Configuration

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.

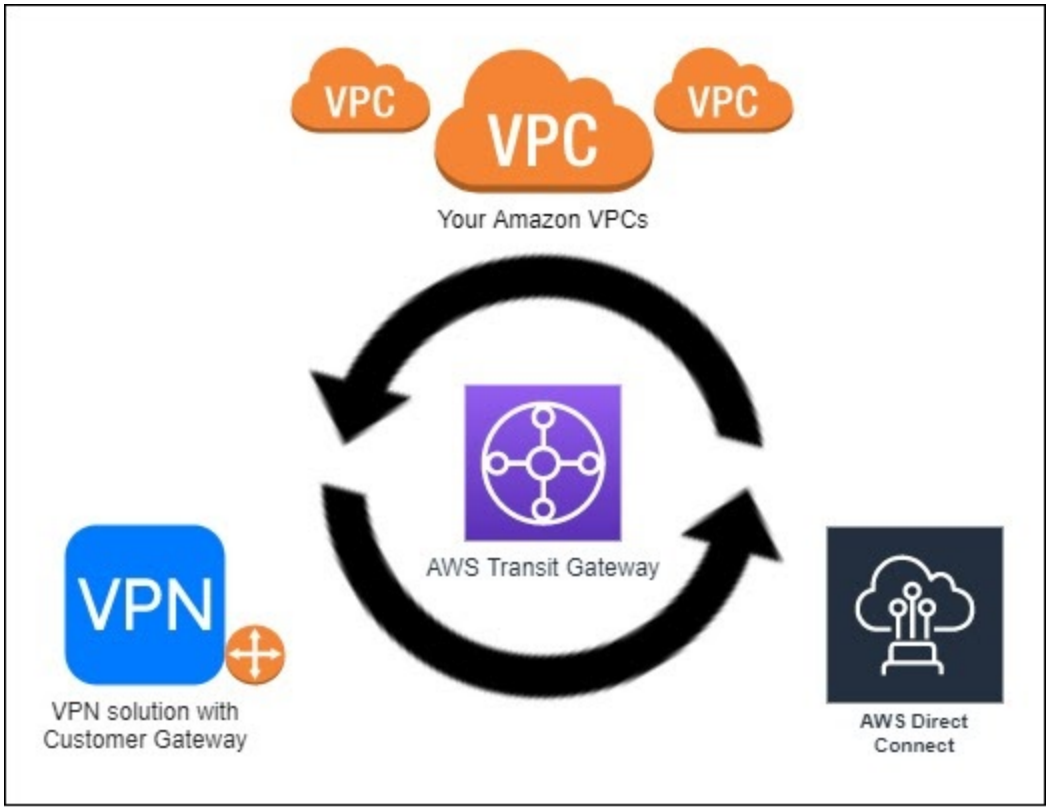


Figure: AWS Transit Gateway as a Hub-and-Spoke Solution

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.

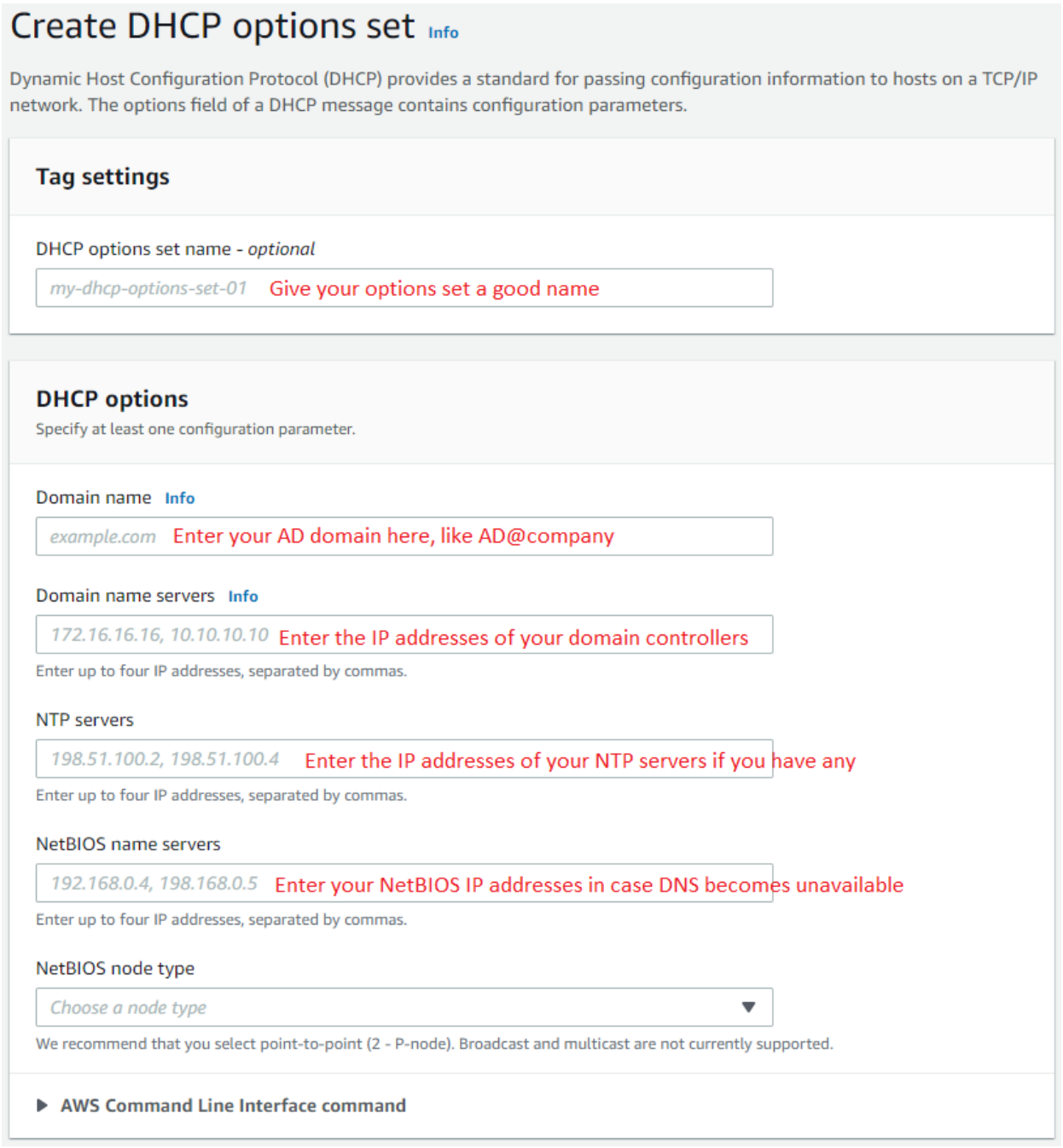
References:
https://d1.awsstatic.com/whitepapers/building-a-scalable-and-secure-multi-vpc-aws-network-infrastructure.pdf
https://d1.awsstatic.com/whitepapers/aws-amazon-vpc-connectivity-options.pdf
https://docs.aws.amazon.com/vpc/latest/peering/what-is-vpc-peering.html
https://docs.aws.amazon.com/directconnect/latest/UserGuide/direct-connect-gateways.html

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.

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. Majority of your work as a Solutions Architect Professional lies in building concrete solutions that follow industry best practices for your customers. You need to be very knowledgeable with the different AWS services and other well-known tools and practices used in your industry. Having an in-depth understanding of what you're working with will allow you to design a strategy that works best for your customers' business objectives.

31% of the questions in the actual exam revolve around these topics.

- Determine security requirements and controls when designing and implementing a solution
- Determine a solution design and implementation strategy to meet reliability requirements
- Determine a solution design to ensure business continuity
- Determine a solution design to meet performance objectives
- Determine a deployment strategy to meet business requirements when designing and implementing a solution

In this chapter, we will cover the related topics for designing solutions centered around security, performance, 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.

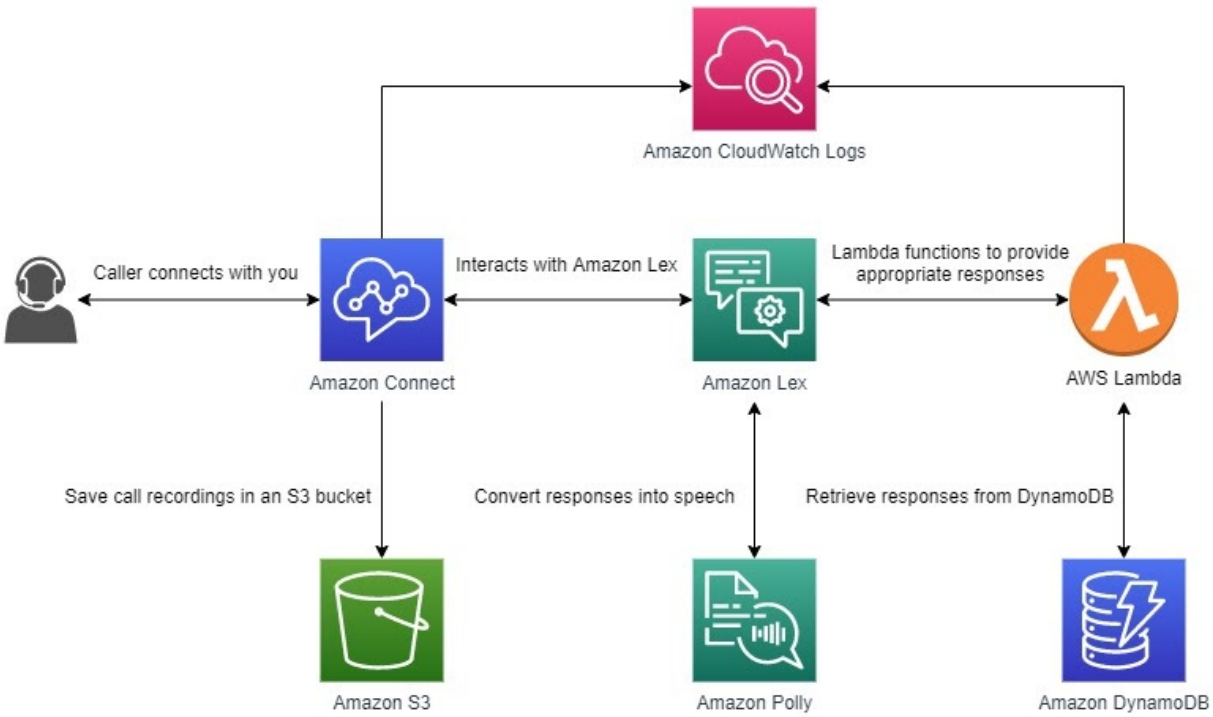
If you only need to share access to your applications and not a whole desktop environment then Amazon Appstream is the service to use. You can stream your applications to any computer without having to provision and operate hardware. All you need is a HTML5-capable browser. Appstream allows you to restrict 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.



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.

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/

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.

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.

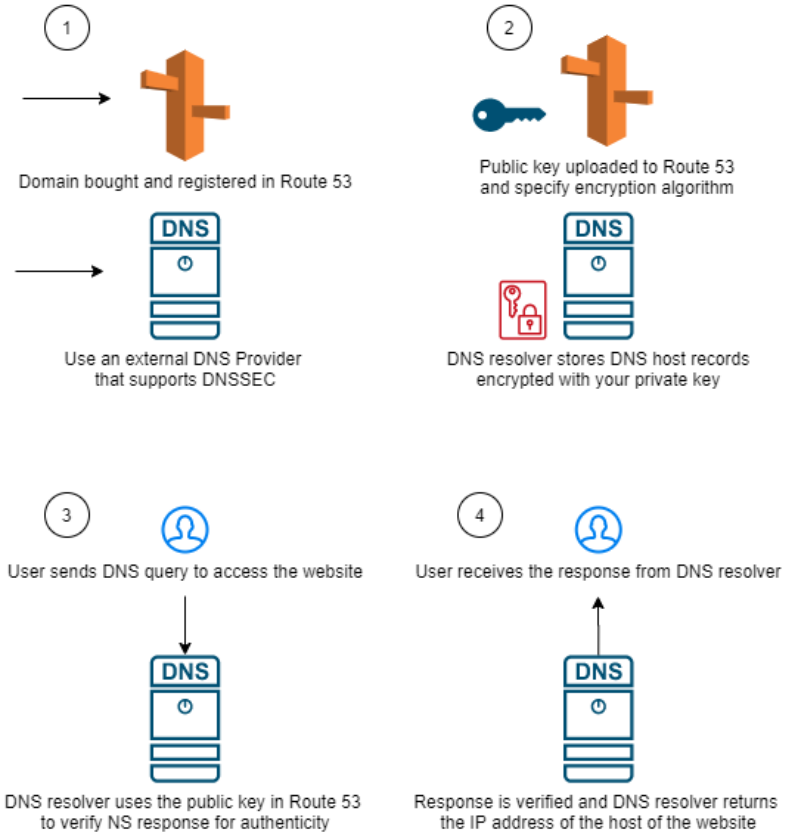
Other options you might encounter include using load balancers and auto scaling groups, or enforcing rules in security groups and network ACLs. Although these services can help prolong your website's availability and keep a few attacker IP addresses at bay, they do not protect your web servers completely 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. However, Route 53 does not support DNSSEC for DNS service, regardless of whether the domain is registered with Route 53. If you want to configure DNSSEC for a domain that is registered with Route 53, you must either use another DNS service provider or set up your own DNS server.



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

Configuration Management in AWS with AWS OpsWorks Stacks

Perhaps you are managing multiple machines and VMs that it becomes impractical to configure them one by one. You would then have to use configuration management tools such as Chef, Puppet, and Ansible to automate your tasks. In AWS, you would be using AWS OpsWorks. Of course if you are already familiar with Chef or Puppet then you would use AWS OpsWorks for Chef or AWS OpsWorks for Puppet; but if you are new to the configuration management platform then you should also take a look at AWS OpsWorks Stacks. AWS OpsWorks Stacks supports Chef recipes and Bash/PowerShell scripts, but does not use a Chef server. Instead, it uses an embedded chef-solo client that is installed on Amazon EC2 instances.

AWS OpsWorks lets you manage applications and servers on AWS and on-premises. It supports both Linux (Amazon Linux and Ubuntu) and Windows OS (2012 R2) on AWS, but only supports Linux machines that can install the OpsWorks Stacks agent for on-premises servers. To get started, you configure the stack and the different layers it will contain. A layer represents a set of Amazon EC2 instances that serve a particular purpose, such as serving applications or hosting a database server. For all stacks, AWS OpsWorks Stacks includes “service layers”, which are the following services:

- Amazon RDS
- Elastic Load Balancing
- Amazon ECS

Layers give you complete control over which packages are installed, how they are configured, how applications are deployed, and more using Chef recipes. Each layer also includes a set of lifecycle events—**Setup, Configure, Deploy, Undeploy, and Shutdown**—which automatically run a specified set of recipes at the appropriate time on each instance.

After an EC2 instance boots up, AWS OpsWorks Stacks installs the agent that handles communication between the instance and the service, and executes your Chef recipes in response to lifecycle events. AWS OpsWorks Stacks supports **instance autohealing**, so if an agent stops communicating with the service, the instance is automatically restarted/replaced. If you need to update your OpsWorks Stacks instance settings, such as instance size, the instance must first be stopped.

Since AWS OpsWorks is generally a DevOps tool, it also provides multiple deployment methods when you need to update your stacks, layers, instances, and applications.

Using a Rolling Deployment

- A rolling deployment updates an application on a stack's online application server instances in multiple phases. With each phase, you update a subset of the online instances and verify that the update is successful before starting the next phase. If you encounter problems, the instances that are still running the old app version can continue to handle incoming traffic until you resolve the issues.

Using Separate Stacks

- You use different stacks or “environments” to manage your applications (e.g. production, dev, staging). This arrangement allows you to do development and testing on stacks that are not publicly accessible. When you are ready to deploy an update, switch user traffic from the stack that hosts the current application version to the stack that hosts the updated version.

Using a Blue-Green Deployment Strategy

- Similar to using separate stacks, you have a production (blue) stack, which hosts the current application, and a staging (green) stack, which hosts the updated application. When you are ready to deploy the updated app to production, you switch user traffic from the blue stack to the green stack, which becomes the new production stack. If your users encounter issues on the new stack, you can quickly repoint the traffic back to the old one. You then retire the old blue stack once everything is verified to be working smoothly.

References:
https://aws.amazon.com/opsworks/stacks/

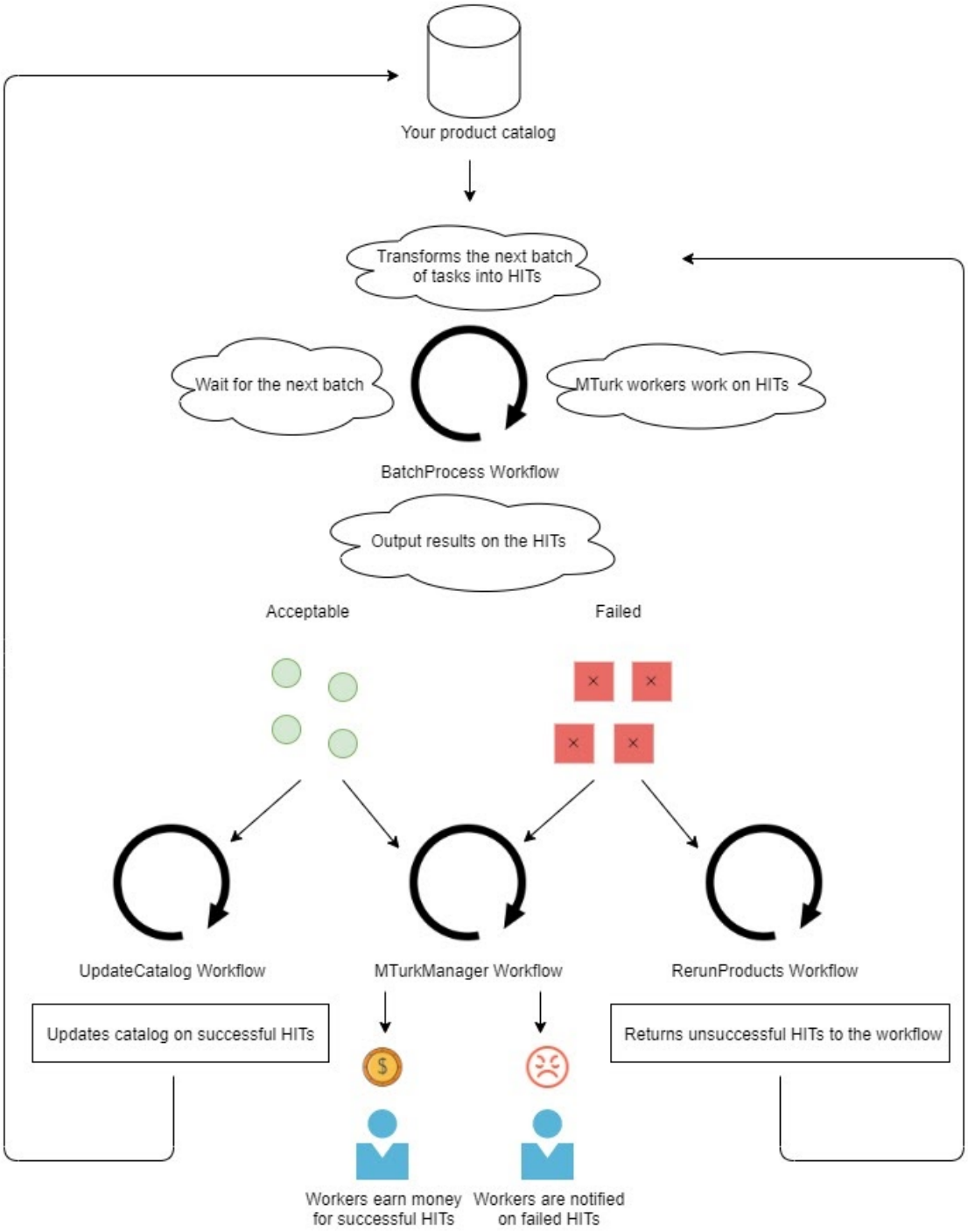
Processing Large Product Catalogs using Amazon Mechanical Turk and Amazon SWF

If this is the first time you've heard of Amazon Mechanical Turk then that's alright. This is not a service that a Solutions Architect often encounters. Amazon Mechanical Turk (or MTurk) is an offering from AWS that is essentially a crowdsourcing marketplace. If you have jobs that cannot be easily automated by machines, you can submit your jobs to a distributed workforce who can perform these tasks virtually. Examples of these jobs include conducting data validation and research on tasks like survey participation, content moderation, and more.

You, as an MTurk Requester, creates HITs (Human Intelligence Tasks) on the marketplace which will be made available to MTurk workers. You can create a whole bunch of HITs which becomes your own product catalog. As this catalog grows larger, the amount of work needed to validate and process the results increases too. To maximize your output, you can design a workflow in Amazon Simple Workflow (Amazon SWF) to automate these steps and break down your catalog into batches of products. Amazon SWF is a fully-managed state tracker and task coordinator in the Cloud. It helps you build, run, and scale background jobs that have parallel or sequential steps. When HITs get completed, the results are then assessed and MTurk workers are compensated for acceptable results. Failed HIT results are re-batched and reprocessed, while the acceptable HIT results are used to update the catalog.

Before we discuss how to automate your product catalog with MTurk, we must first discuss what an SWF workflow and SWF activity worker is. **Workflows** coordinate and manage the execution of activities that can be run asynchronously across multiple computing devices and that can feature both sequential and parallel processing. They provide you a visual representation of how your tasks will be accomplished on success and on failure of the operation. An **activity worker** is a process or thread that performs the activity tasks that are part of your workflow. The activity task represents one of the tasks that you identified in your application. You can have a fleet of multiple activity workers performing activity tasks of the same activity type. When an activity worker has completed its activity task, it reports to Amazon SWF on the status of the task, and it includes any relevant results that were generated. Amazon SWF updates the workflow execution history to indicate that the task is completed and then proceeds to the following step of your workflow.

Now that we have defined the actors in this process, we can start designing our intended workflow. To get started, a *BatchProcess* workflow is generated to handle the processing for a single batch of products. It has workers that extract the data, transform it, and send it through Amazon Mechanical Turk. The *BatchProcess* workflow outputs the acceptable HITs and the failed ones. This is used as the input for three other workflows: MTurkManager, UpdateCatalog, and RerunProducts. The *MTurkManager* workflow makes payments for acceptable HITs, responds to the human workers who produced failed HITs, and updates its own database for tracking results quality. The *UpdateCatalog* workflow updates the master catalog based on acceptable HITs. The *RerunProducts* workflow waits until there is a large enough batch of products with failed HITs, which it then sends back to the *BatchProcess* workflow. The entire end-to-end catalog processing is performed by a *CleanupCatalog* workflow that initiates child executions of the above workflows.



References:
https://www.mturk.com/
https://aws.amazon.com/swf/

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.

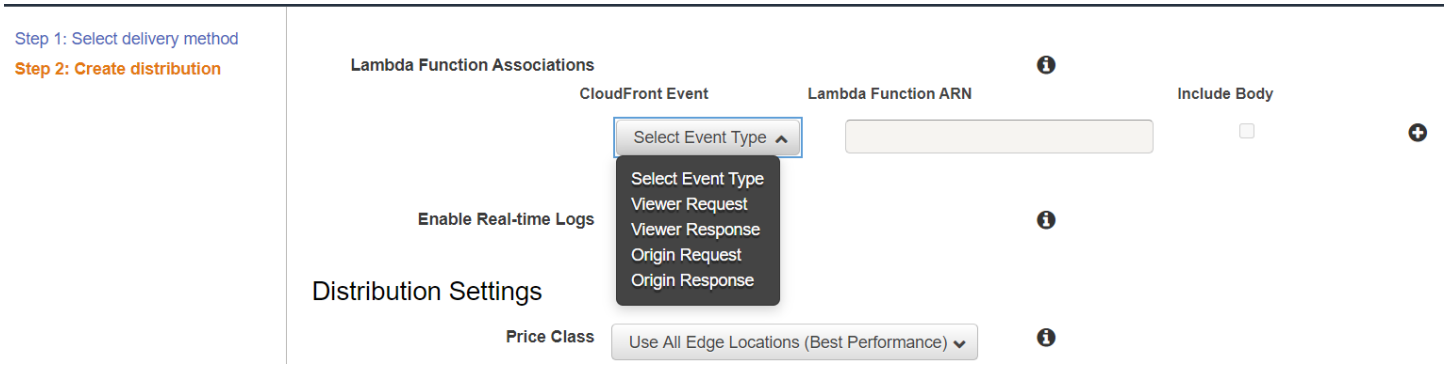


Figure: Configuring your CloudFront to add a Lambda function

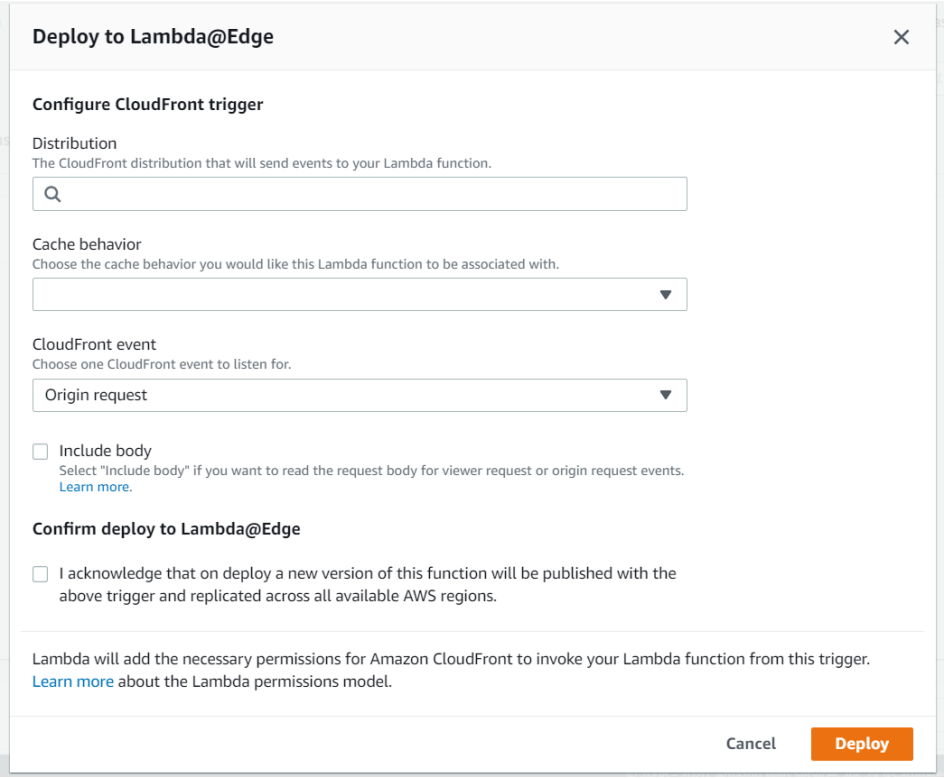


Figure: Adding a trigger on your Lambda function and specifying the CloudFront distribution

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 ES

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 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.	Logstash ingests data from multiple sources, then the data is transformed through a series of filters which you design, and finally outputs your data into a "stash".	Logs and documents sent to Elasticsearch can be visualized in Kibana for graphical views and 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 Elasticsearch Service (ES) if you can deploy your own ELK stack? Amazon Elasticsearch 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 ES 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 ES integrates with other AWS services such as Amazon Kinesis Data Firehose, Amazon CloudWatch Logs, and AWS IoT, giving you the flexibility to select the data ingestion tool that meets your use case requirements.

Solutions Architect Professional Exam Notes:

Whenever you encounter Elasticsearch in AWS, always consider the options that discuss Amazon Elasticsearch 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 ES domain first and configure your new cluster. After that, you can utilize AWS Database Migration Service (AWS DMS) to migrate data to Amazon ES 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 ES cluster. Once done, you can repoint your log ingestion to use your new cluster. 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 QuickSight

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 QuickSight for interactive data visualization of your query results. Before you try to read files from S3 buckets, make sure that you grant Amazon QuickSight 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 Quicksight 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 QuickSight 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://aws.amazon.com/athena/
https://aws.amazon.com/quicksight/

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: Migration Planning

Overview

The third exam domain of the AWS Certified Solutions Architect Professional certification is all about migration in AWS. As a Solutions Architect, it falls on your shoulders to help customers migrate properly onto the Cloud. This is a daunting task as there are many intricacies involved, not just with technologies, but also with infrastructure design, personnel, and budget. Companies would often perform a lift and shift onto AWS which is not the way it should be done. They are missing out on the advantages provided by the cloud (which is the reason why you are moving away from on-premises in the first place!). Fortunately, AWS has already provided multiple tools and approaches that you can adapt for your migration workloads and accelerate the whole process.

15% of questions in the actual exam revolve around these topics.

- Select existing workloads and processes for potential migration to the cloud
- Select migration tools and/or services for new and migrated solutions based on detailed AWS knowledge
- Determine a new cloud architecture for an existing solution
- Determine a strategy for migrating existing on-premises workloads to the cloud

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 six (6) migration strategies, which are also known as the 6 R's, that we can perform in AWS. Each one has its own advantages and disadvantages.

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.

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.

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.

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.

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.

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
Rehost	Small effort and cost	Small opportunity
Repurchase	Average effort and cost	Small opportunity
Replatform	Above average effort and cost	Average opportunity
Refactor	High effort and cost	High opportunity

References:
https://d1.awsstatic.com/whitepapers/Migration/aws-migration-whitepaper.pdf
https://d1.awsstatic.com/Migration/migrating-to-aws-ebook.pdf
https://aws.amazon.com/blogs/enterprise-strategy/6-strategies-for-migrating-applications-to-the-cloud/

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

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** 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.

AWS Snow Family, which includes Snowball Edge, Snowmobile, and the fairly new Snowcone, are hardware devices that you can provision to move data into and out of Amazon S3. If you need **quick large scale data transfers ranging from terabytes to exabytes**, these are often your best option for data migration. Snowball Edge and Snowcone can also be used for **local storage and compute workloads**.

Amazon Kinesis Data Firehose is the easiest way to load **streaming data** into AWS. It can capture and automatically load streaming data into **Amazon S3 and Amazon Redshift**, enabling **near real-time analytics**. It is also fully managed. Additionally, it can batch, compress, and encrypt the data before loading it to AWS.

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 Server Migration

The primary service you will use for this purpose is the AWS Server Migration Service. **AWS Server Migration Service** (AWS SMS) is an agentless service that automates the migration of your on-premises **VMware vSphere, Microsoft Hyper-V/SCVMM, and Azure virtual machines** to the AWS Cloud.

Solutions Architect Professional Exam Notes:

Should I use AWS Server Migration Server or EC2 VM Import/Export for my server migration purposes?

AWS Server Migration Service is a significant enhancement of EC2 VM Import/Export, so you should always prefer AWS SMS over VM Import/Export. Do take note however that VM Import/Exports supports a few different virtualization formats from AWS SMS, such as VMware ESX or Workstation, Microsoft Hyper-V, and Citrix Xen. It only supports Windows and Linux using these virtualization platforms as well. Keep an eye out for the specified platform on your exam questions.

1) To start with, you first install the *Server Migration Connector* in your on-premises virtualization environment.
2) You can use the AWS console or the CLI to initiate the migration. If you have not yet imported a catalog, choose **Servers**, **Import server catalog**.
3) Select a server to replicate and choose **Create replication job**.
4) On the **Configure server-specific settings** page, in the License type column, select the license type for AMIs to be created from the replication job. Linux servers can only use Bring Your Own License (BYOL). Windows servers can use either an AWS-provided license or BYOL.
5) Enter **Configure replication job settings** such as:
    a) Replication job type — Specify the replication interval (every 1-24 hours) or choose One-time migration.
    b) Start replication run — Choose **Immediately** to start a replication run immediately or **At a later date and time** to start replication at the specified date and time, up to 30 days in the future. The date and time are specified using the local time of your browser.
    c) IAM service role — Choose **Allow automation role creation** to have AWS SMS create a service-linked role on your behalf or **Use my own role** to specify an existing IAM role.
    d) You can also configure automatic AMI deletion, AMI encryption, SNS notifications, and more which we will not be discussing in-depth.

AWS SMS supports the automated migration of multi-server application stacks from your on-premises data center to Amazon EC2. Application migration replicates all of the servers in an application as AMIs and generates an AWS CloudFormation template to launch them in a coordinated fashion.

References:
https://docs.aws.amazon.com/whitepapers/latest/aws-overview/migration-services.html
https://aws.amazon.com/server-migration-service/
Application Migrations Using AWS Server Migration Service (SMS) - AWS Online Tech Talks
https://aws.amazon.com/ec2/vm-import/

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/

Domain 4: Cost Control

Overview

The fourth exam domain of the AWS Certified Solutions Architect Professional exam is about cost controls. The primary reason why most companies move their operations to the cloud is to significantly reduce infrastructure costs. Ideally, when you transition from a physical server setup to the cloud, you'll experience lower overall expenses. This is because one of the advantages of the cloud is economies of scale. Now take note of the word **ideally**. Achieving a lower expenditure in AWS requires a lot of planning, designing, optimizing, understanding of the pricing models, and continuous monitoring from you as a Solutions Architect professional. This whole process of cost savings is not something that can be done instantaneously. The only way to become cost efficient in AWS is to collect data and analyze how you can properly right-size and optimize your resources based on the data.

Around 12.5% of questions in the actual exam revolve around these topics.

- Select a cost-effective pricing model for a solution
- Determine which controls to design and implement that will ensure cost optimization
- Identify opportunities to reduce cost in an existing solution

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.

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.

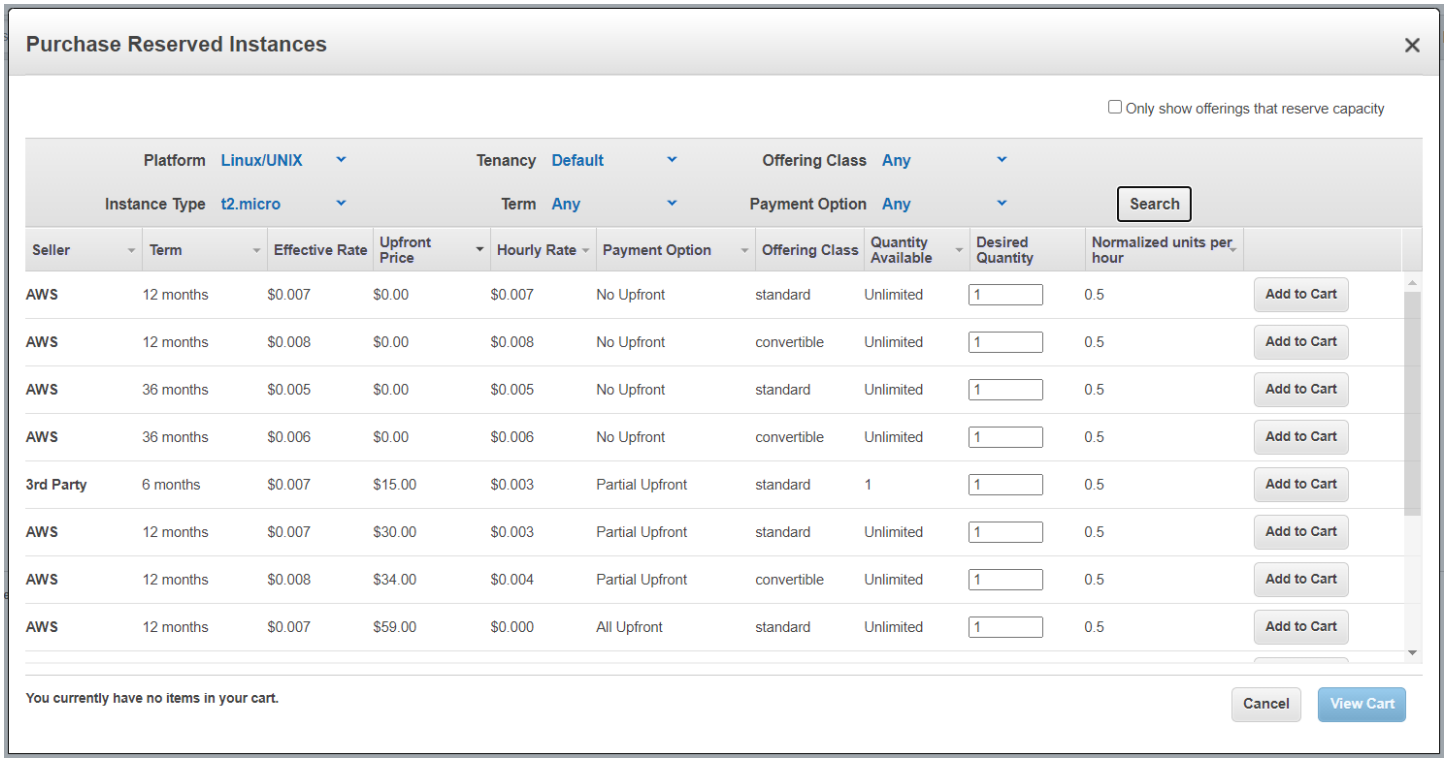
References:
https://d1.awsstatic.com/whitepapers/aws_pricing_overview.pdf
https://d1.awsstatic.com/whitepapers/cost-optimization-reservation-models.pdf
https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/instance-purchasing-options.html
https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/useconsolidatedbilling-discounts.html

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.

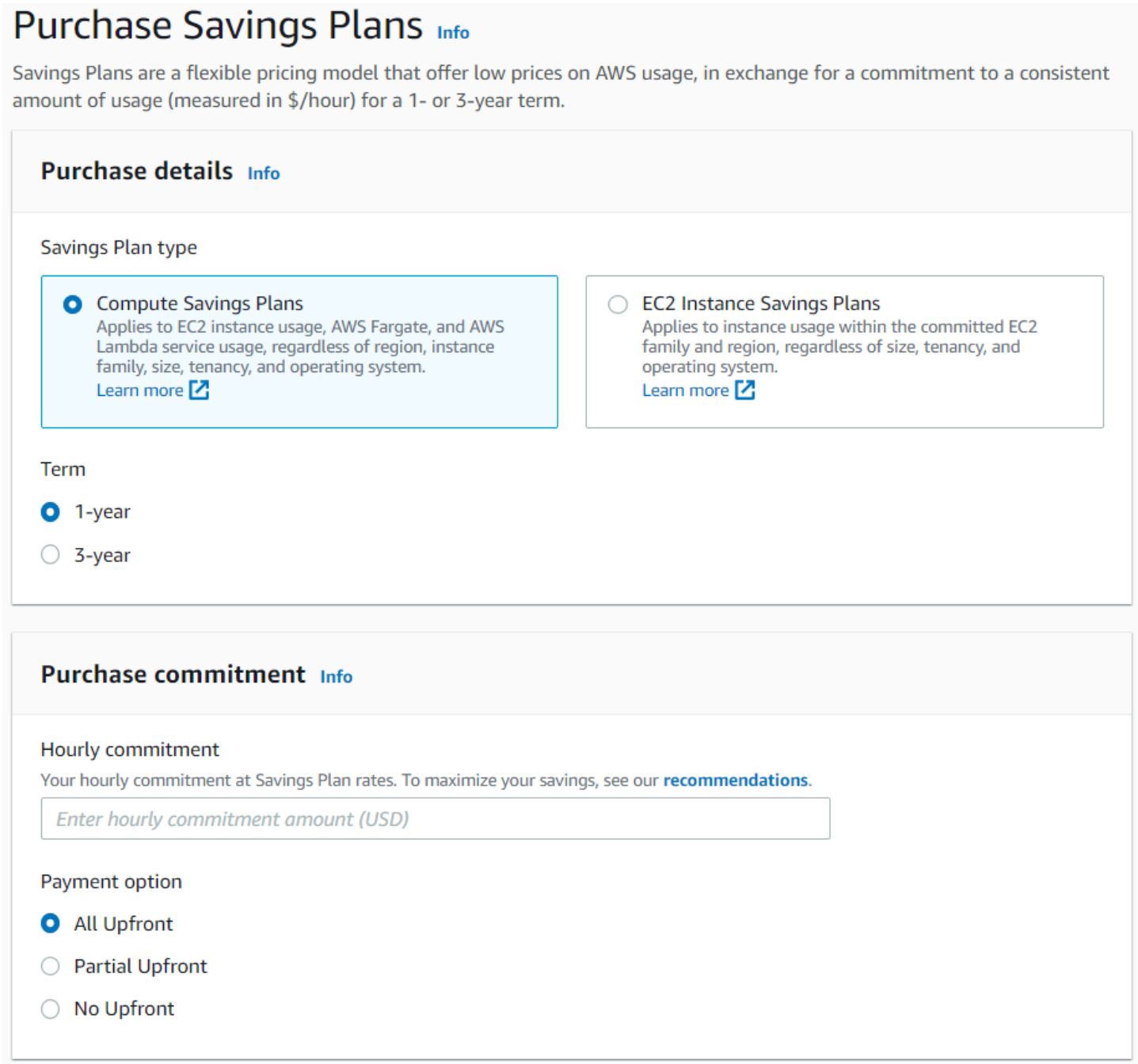


Figure: AWS Compute Savings Plans

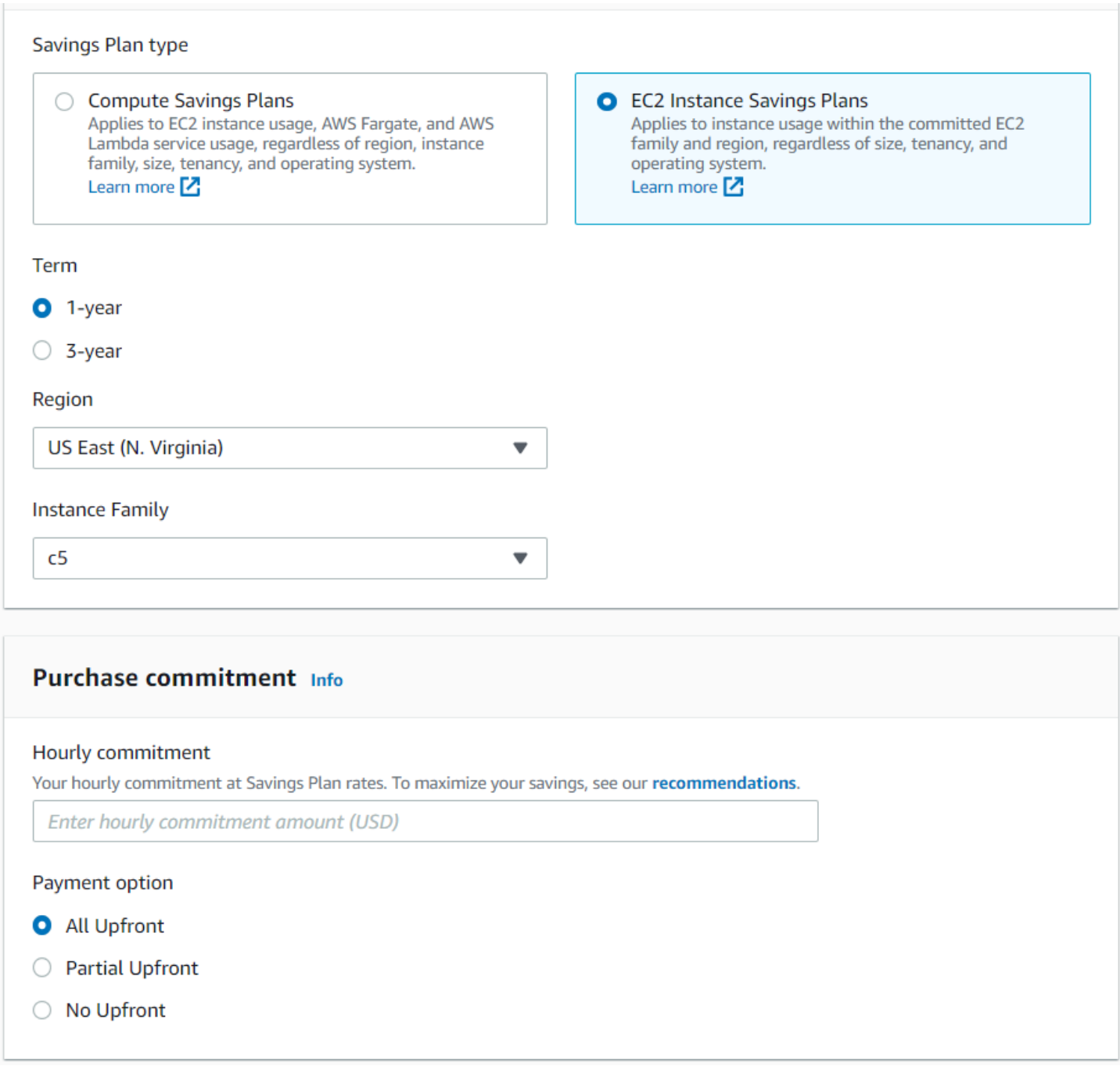


Figure: AWS EC2 Instance Family Savings Plans

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/

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://d1.awsstatic.com/whitepapers/total-cost-of-operation-benefits-using-aws.pdf

Domain 5: Continuous Improvement for Existing Solutions

Overview

The fifth and last domain of the AWS Certified Solutions Architect Professional exam focuses on continuously improving your solutions by updating yourself with newer technologies and practices. Each month, AWS releases a set of updates on their services which are most of the time new features, but can sometimes be new services themselves. By keeping yourself up-to-date with the newest technologies and best practices available, you can help your customers improve their own architectures and operations.

29% of questions in the actual exam revolve around these topics.

- Troubleshoot solution architectures
- Determine a strategy to improve an existing solution for operational excellence
- Determine a strategy to improve the reliability of an existing solution
- Determine a strategy to improve the performance of an existing solution
- Determine a strategy to improve the security of an existing solution
- Determine how to improve the deployment of an existing solution

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 all patch our 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.

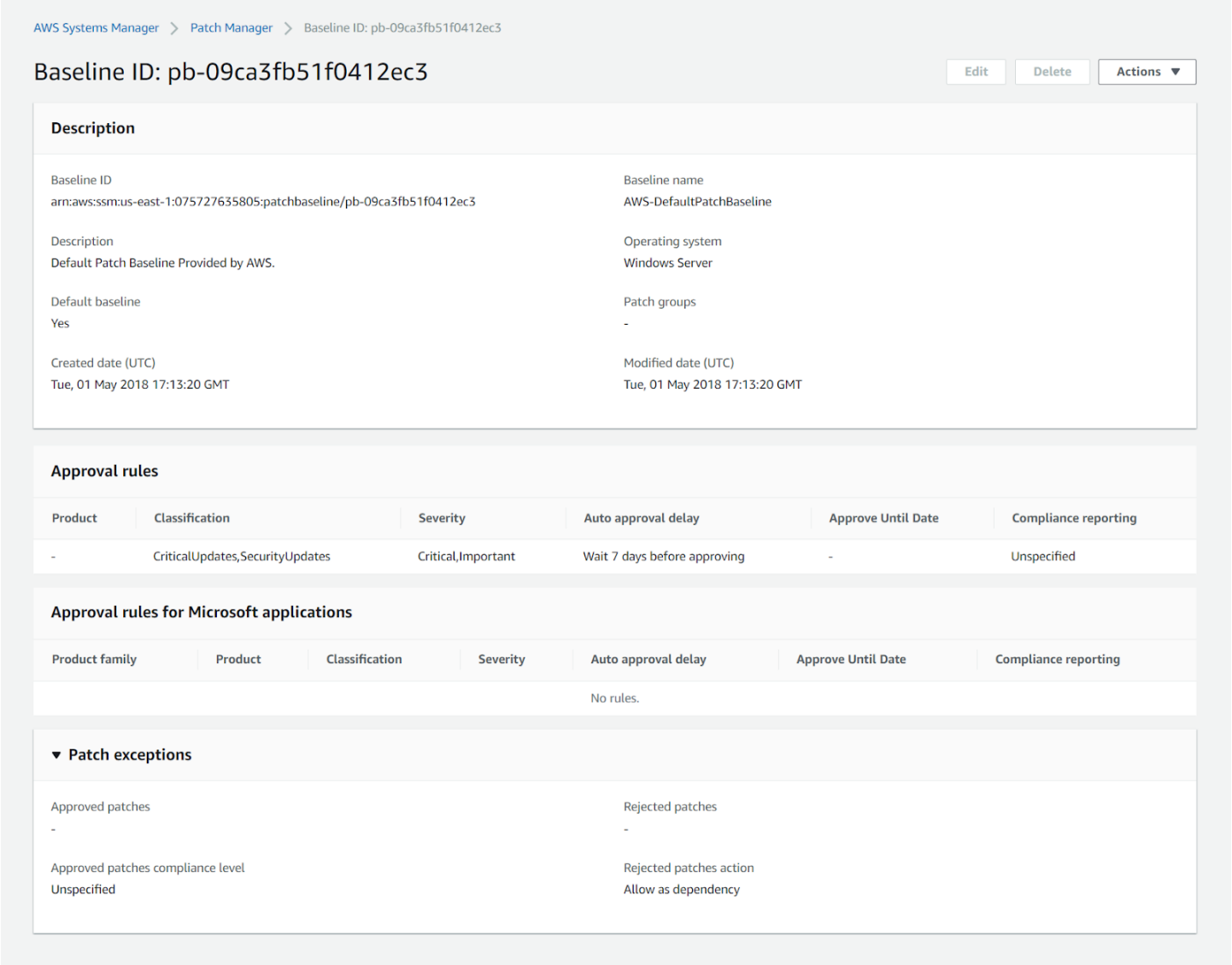
Note: Since you are letting AWS Systems Manager handle your instances, you will need to assign the appropriate IAM role to your instances that would grant permissions for AWS Systems Manager to perform patching and other related tasks. You also need to make sure that your instances have SSM Agent installed 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.

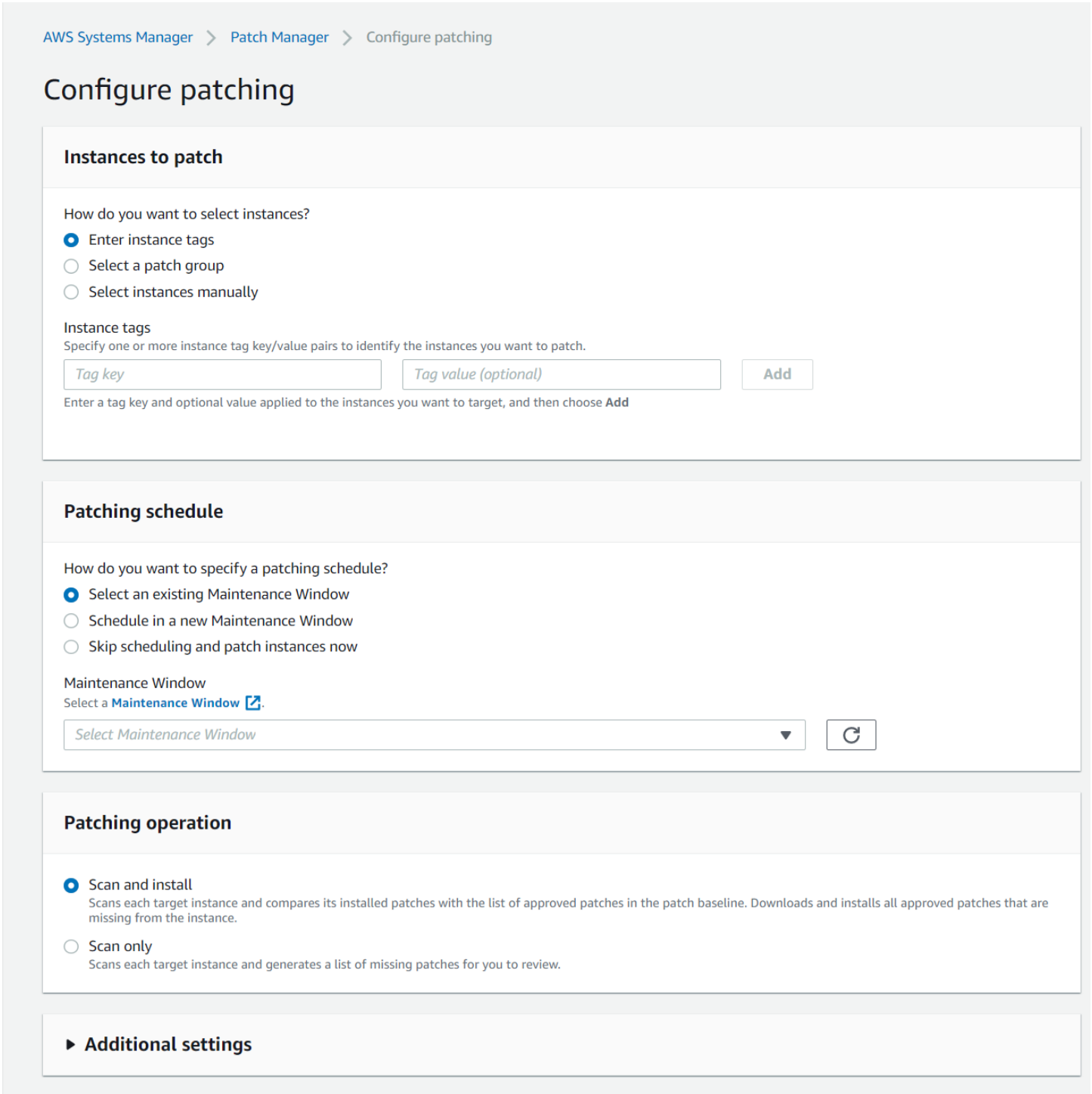


Maintenance Windows, on the other hand, lets you define a **schedule** for when to perform potentially disruptive actions on your instances such as patching an operating system, updating drivers, or installing software or patches. Each maintenance window has a schedule, a maximum duration, a set of registered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

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 Cloudwatch Events 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 CodeCommit, 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.

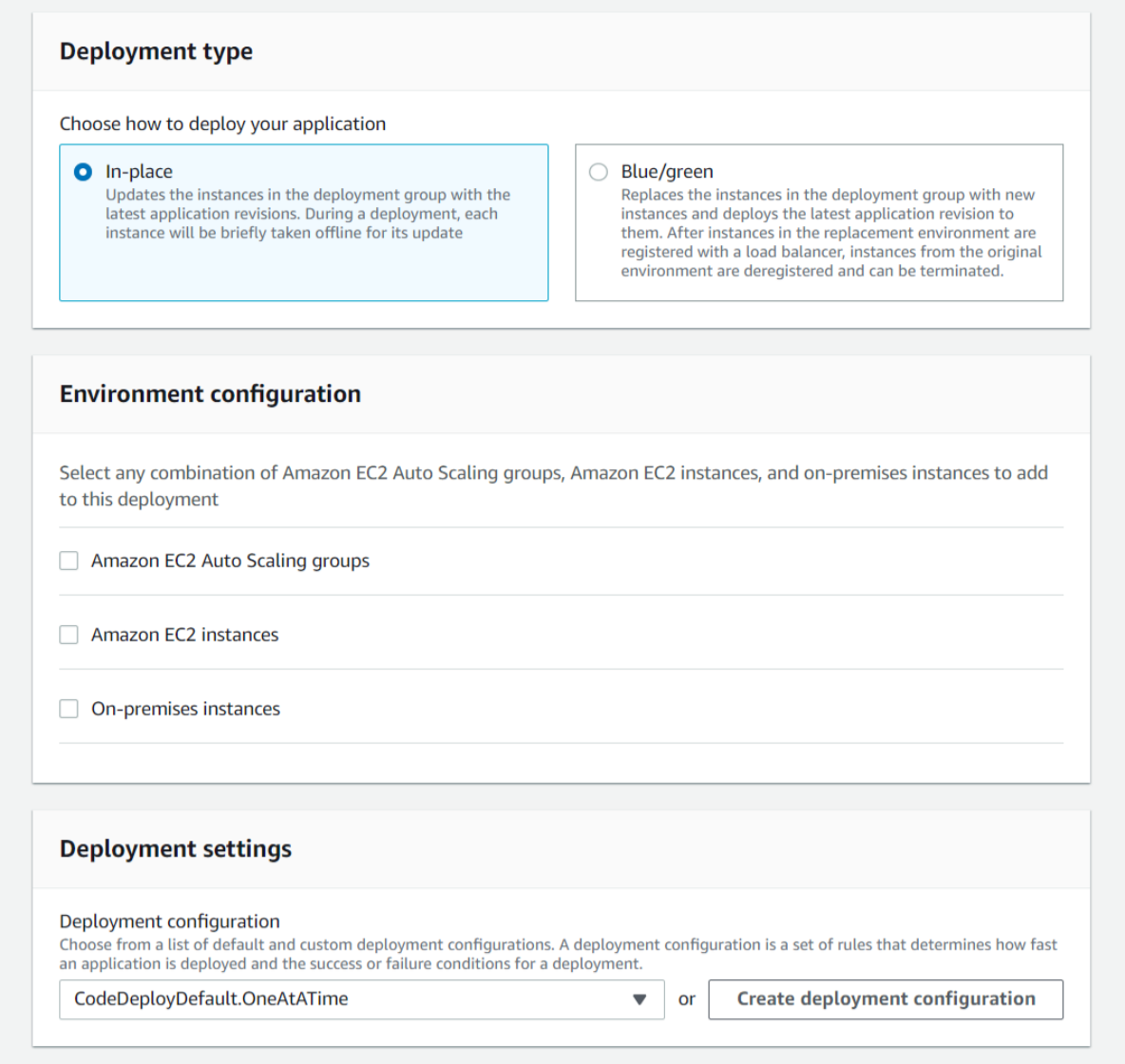
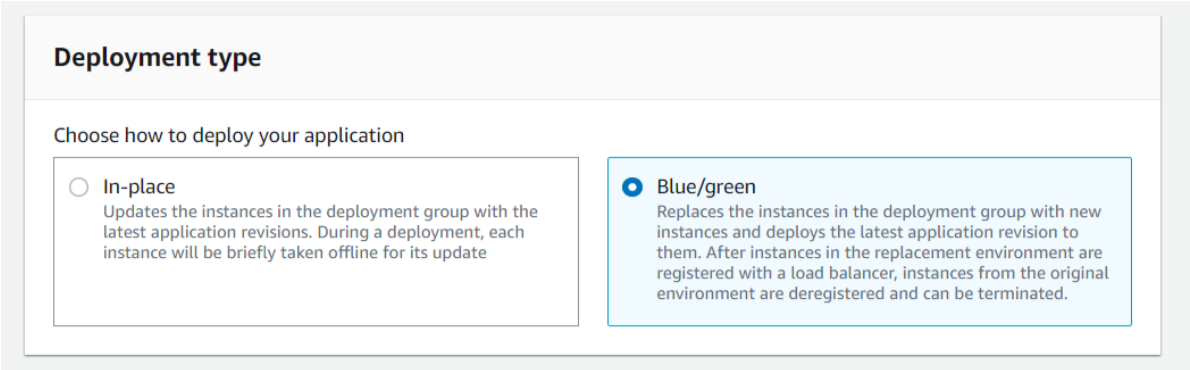


Figure: In Place Deployment in AWS CodeDeploy



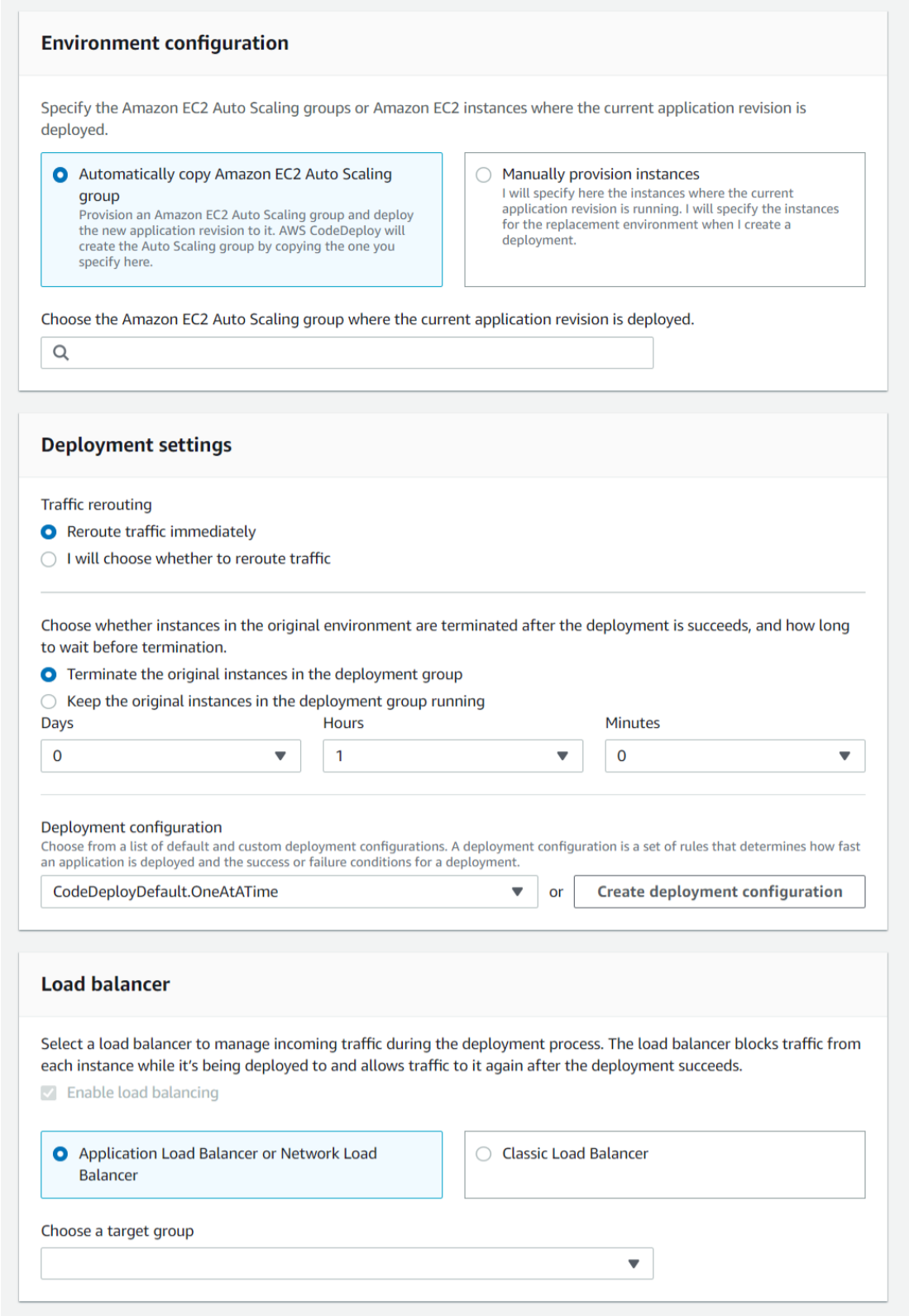


Figure: Blue Green Deployment in AWS CodeDeploy

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)

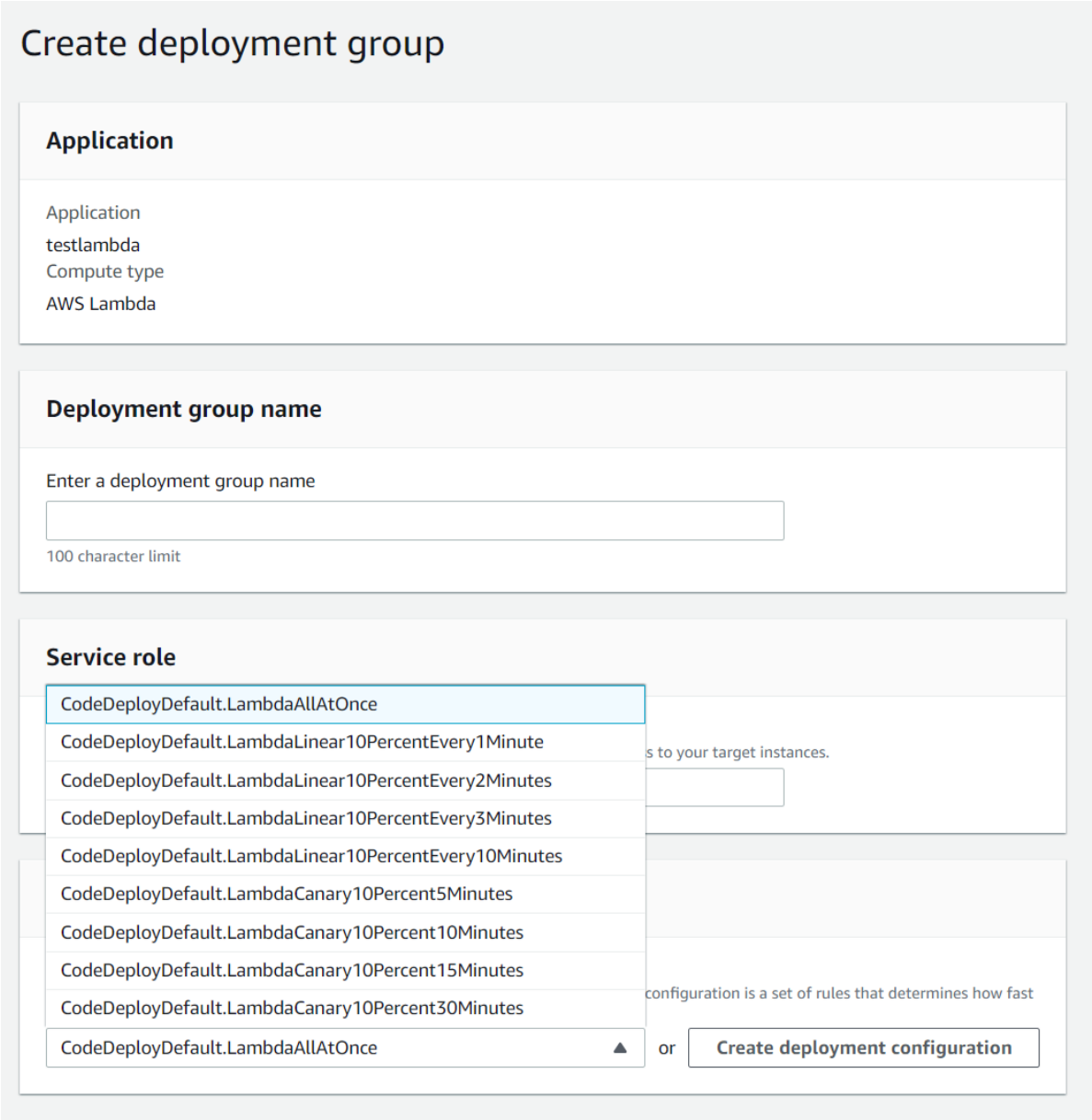


Figure: Lambda Deployment in AWS CodeDeploy

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

2) **AWS CodeCommit**

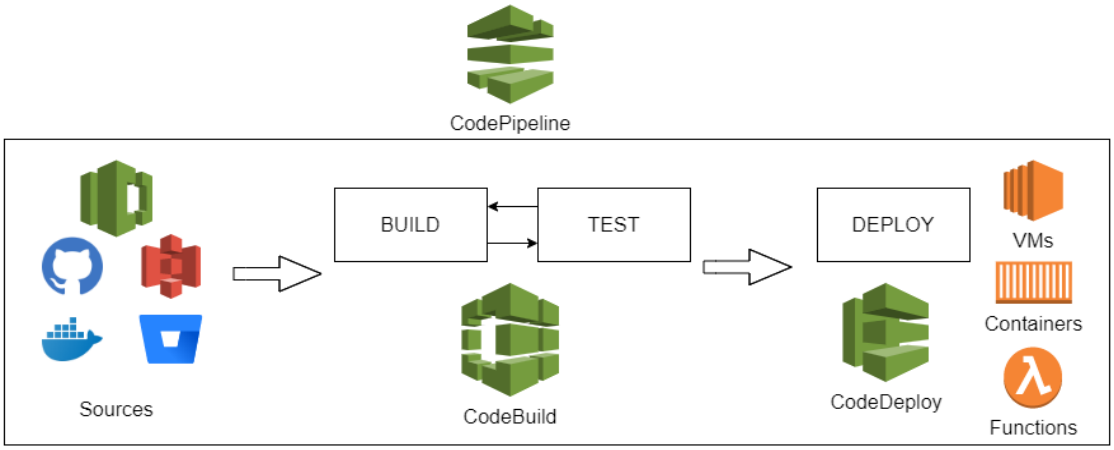
AWS CodeCommit is basically Git in AWS. It is a source control service that works in similar fashion with any Git-based repository. You can use AWS CodeCommit to store anything from code to binaries. The important thing is that AWS CodeCommit can be integrated with the following AWS services:

- AWS CodeBuild - You can store the source code to be built and the build specification in a CodeCommit repository. You can use CodeBuild directly with CodeCommit, or you can incorporate both CodeBuild and CodeCommit in a continuous delivery pipeline with CodePipeline.
- AWS CodePipeline - You can configure CodePipeline to use a CodeCommit repository as a source action in a pipeline, and automate building, testing, and deploying your changes.
- AWS Elastic Beanstalk - You can use the Elastic Beanstalk CLI to deploy your application directly from a CodeCommit repository.
- Amazon SNS / AWS Lambda - You can configure triggers for CodeCommit repositories that send Amazon SNS notifications in response to repository events, or invoke Lambda functions in response to repository events. Example would be validating CodeCommit pull requests

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

3) **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

4) **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/

5) **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/

6) **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 SSO and AWS IAM. Use AWS SSO 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 Single Sign-On (AWS SSO)

AWS SSO 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 SSO, but you need to enable all features in your organizations first. AWS SSO 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 SSO, 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.

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

To federate with AWS SSO:

1) Navigate to the AWS SSO console. Select the directory that stores the identities of your users and groups.
    - AWS SSO provides you a directory by default that you can use to manage users and groups in AWS SSO.
    - You can also connect to a Microsoft AD directory by clicking through a list of Managed Microsoft AD and AD Connector instances that AWS SSO discovers in your account automatically
2) Grant users SSO access to AWS accounts in your organization by selecting the AWS accounts from a list populated by AWS SSO, and then selecting users or groups from your directory and the permissions you want to grant them.
    - You use permission sets to define the level of access that users and groups have to an AWS account. Permission sets are stored in AWS SSO and provisioned to the AWS account as IAM roles.
    - You can assign more than one permission set to a user. Users who have multiple permission sets must choose one when they sign in to the user portal.
3) Grant users access to your business cloud applications if you have any.

4) Give your users the AWS SSO sign-in web address that was generated when you configured the directory so that they can sign in to AWS SSO and access the accounts and business applications.

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.

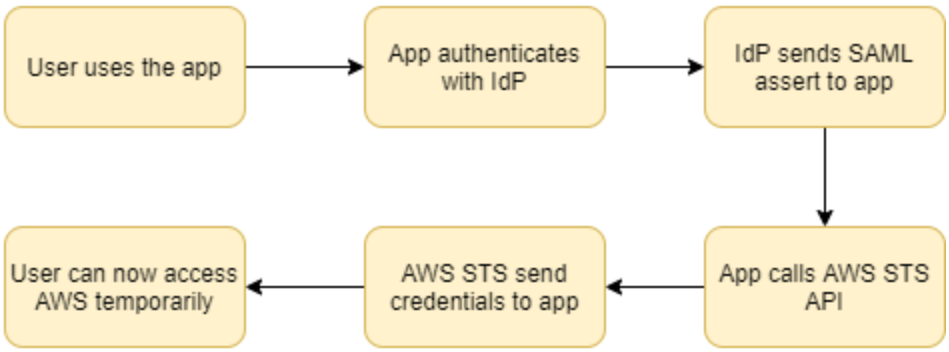


Figure: AWS IAM SAML federation

References:

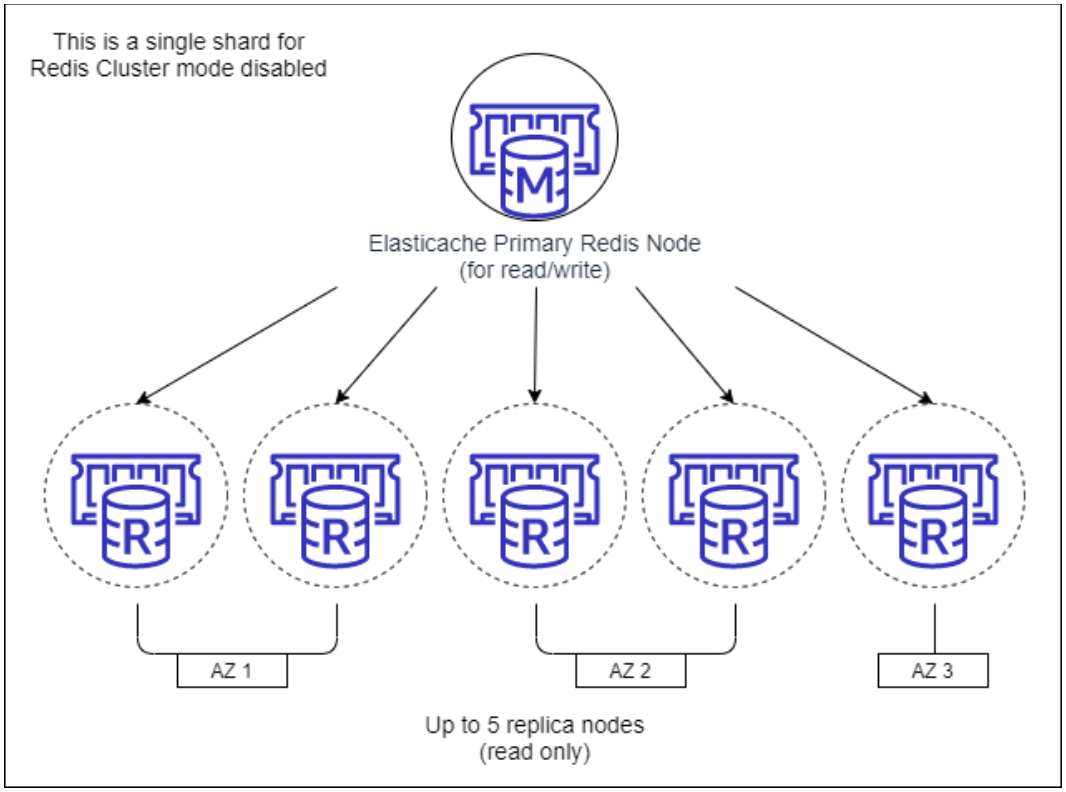
https://aws.amazon.com/identity/federation/
https://docs.aws.amazon.com/singlesignon/latest/userguide/what-is.html
https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles_providers.html

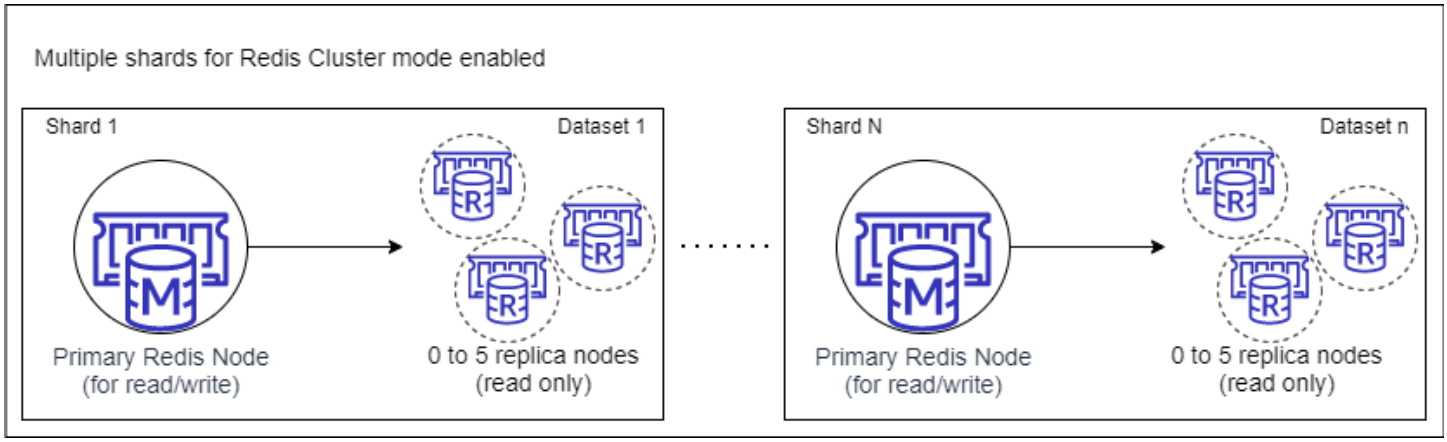
Setting Up a Fault Tolerant Cache Layer with Amazon Elasticache

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.

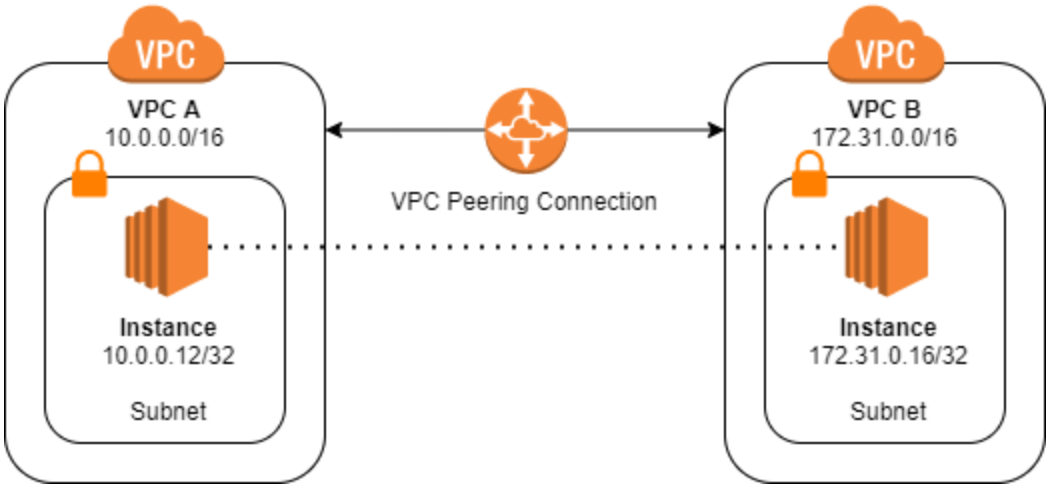
These types of weighted multi-record answers can be achieved by using a combination of records and weighted alias records. You can group multiple endpoints into distinct record sets with each containing a subset of the IP addresses. You can then create a weighted alias record that points to each group 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.

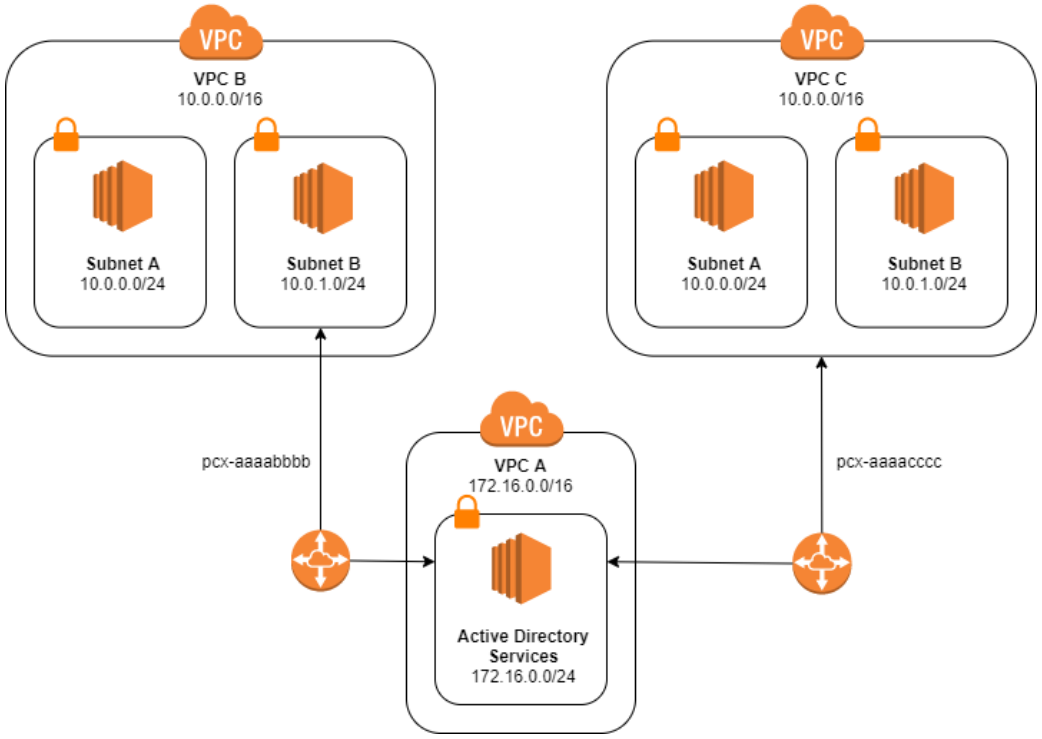
One example of VPC Peering is the integration of third-party services to your AWS account. Say you want to utilize a cloud database service from an external provider, like MongoDB Atlas which provides AWS, GCP, and Azure-backed clusters. In order for your EC2 instances to communicate with your external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

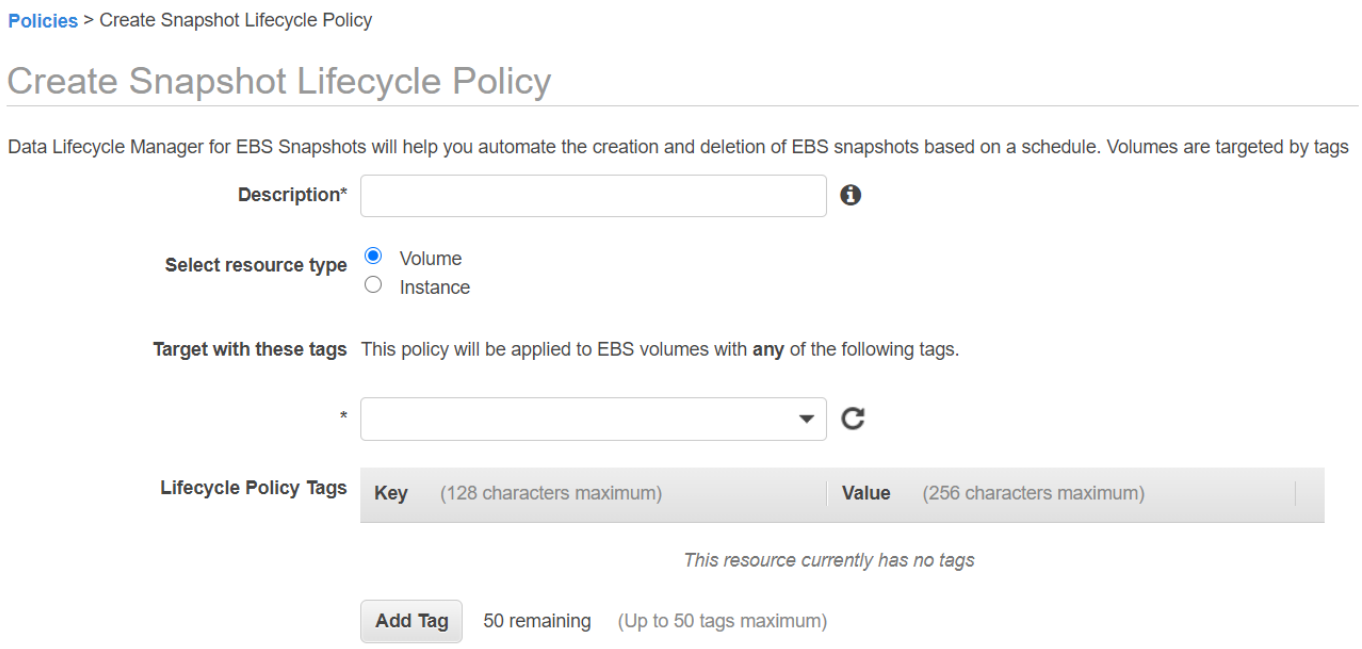
Reference:
https://docs.aws.amazon.com/vpc/latest/peering/peering-configurations-partial-access.html#one-to-two-vpcs-lpm

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

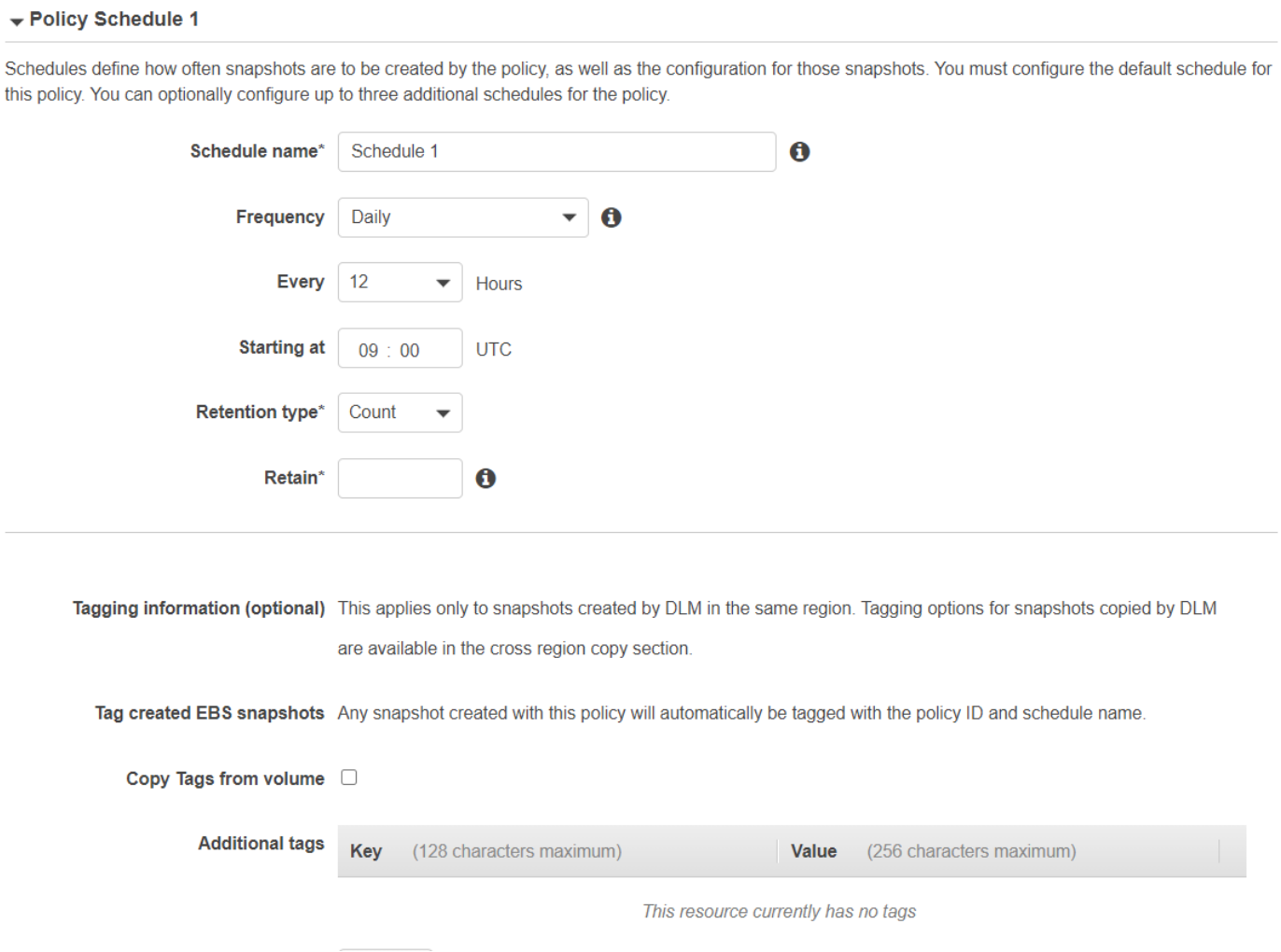
Amazon Data Lifecycle Manager brings a ton of convenience to customers who take regular EBS snapshots. You won't need to script your own Lambda function/Cloudwatch Event anymore just to backup your important files. Amazon DLM features a scheduler which you can configure to create a regular schedule for 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 Kinesis Data Firehose stream, 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 Kinesis 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 configuration (or a launch template) to determine how it will provision additional EC2 instances for you. Remember that you can only associate one launch configuration at a time and you are also not allowed to modify an existing launch configuration's settings. If you wish to update your autoscaling group's launch configuration settings, you must create a new launch configuration from scratch or copy and modify the existing launch config, then you modify the ASG to use the newly created launch configuration.

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 configuration 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

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.

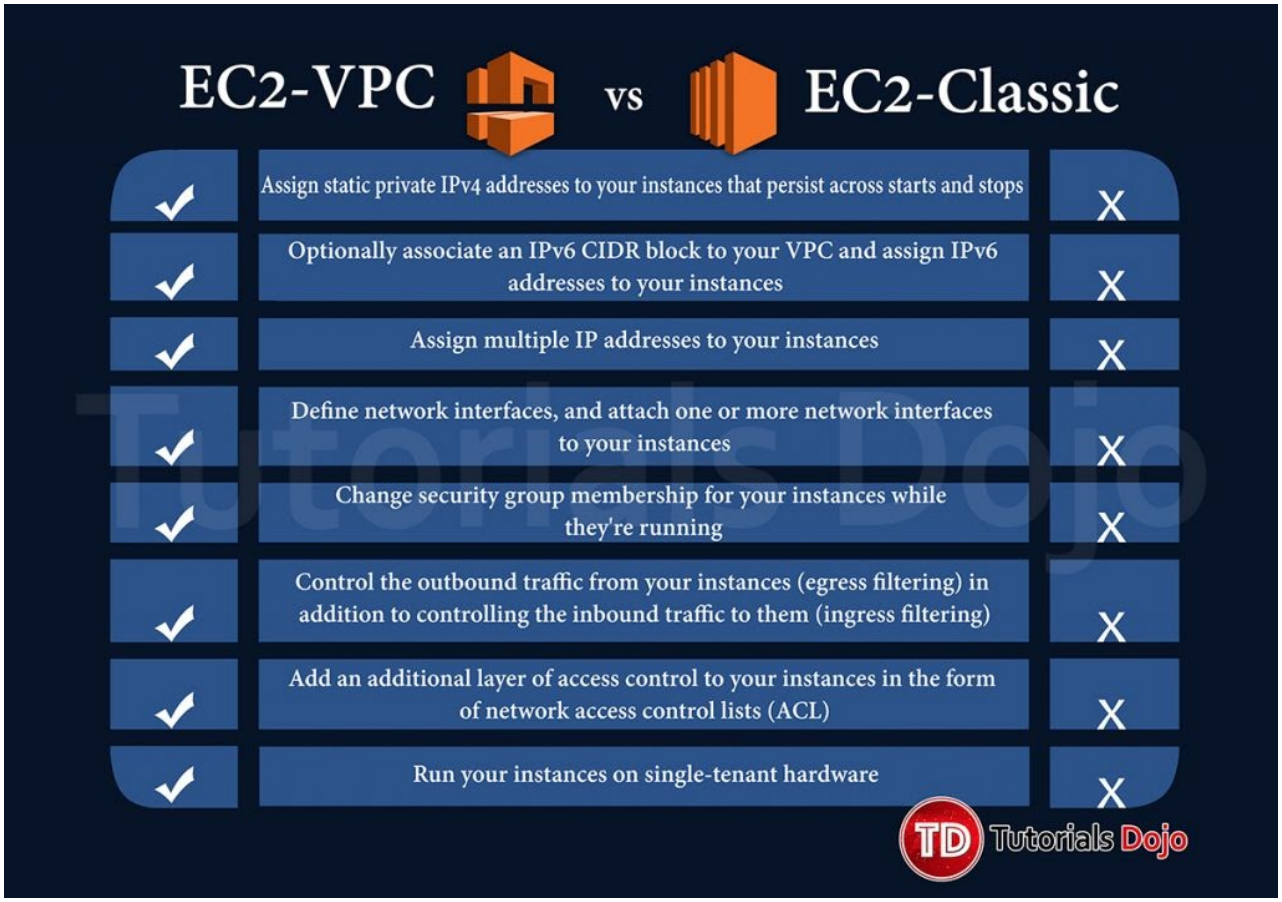
Amazon VPC

- 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.

EC2-VPC vs EC2-Classic



Default vs Non-Default VPC

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

Tutorials Dojo

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.

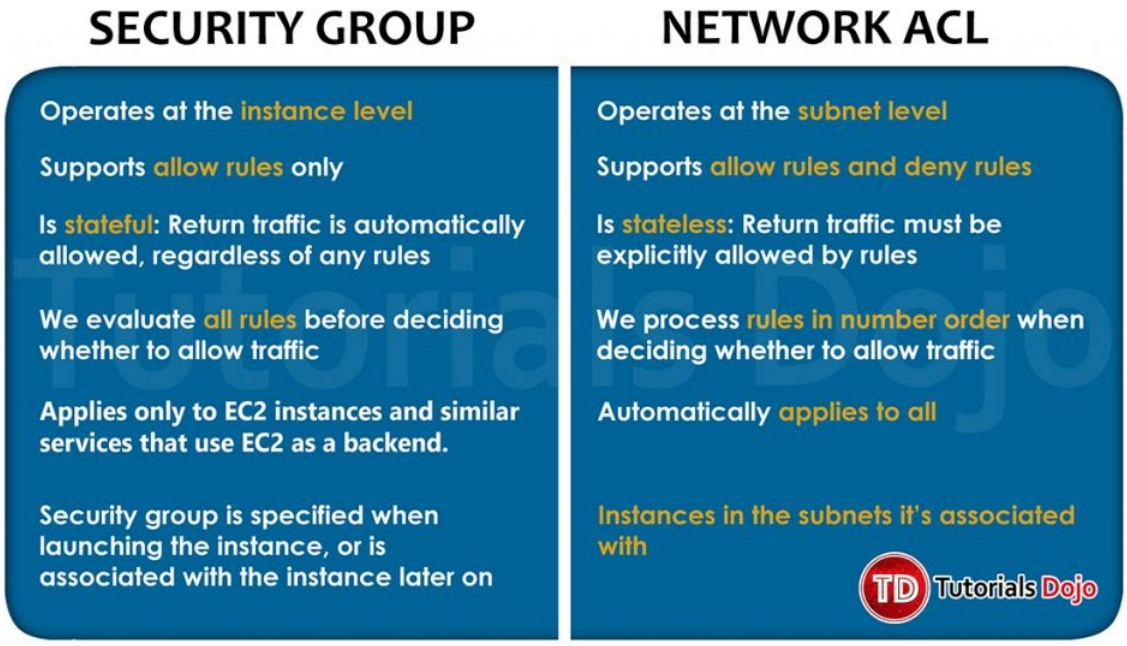
Subnet Routing

- 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.
- 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.



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

Tutorials Dojo

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

- 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 **Origin Access Identity** feature, you can restrict access to an S3 bucket to only be accessible from CloudFront.
  - **Field-Level Encryption** is a feature of CloudFront that allows you to securely upload user-submitted data such as credit card numbers to your origin servers.
- **Pricing**
  - Charge for storage in an S3 bucket.
  - Charge for serving objects from edge locations.
  - Charge for submitting data to your origin.

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

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.
- 1Gbps and 10Gbps ports are available.
- Supports hosted connection capacities of 1, 2, 5 and 10 Gbps. 1, 2, 5 and 10 Gbps hosted connections will provide customers with higher capacities that were previously only available via dedicated connections.
- Amazon Direct Connect also supports AWS Transit Gateway, aside from configuring Site-to-Site VPN connections. With this feature, customers can connect thousands of Amazon VPCs in multiple AWS Regions to their on-premises networks using 1/2/5/10 Gbps AWS Direct Connect connections.
- **Beneficial Use Cases**
    - When transferring large data sets.
    - When developing and using applications that use real-time data feeds.
    - When building hybrid environments that satisfy regulatory requirements requiring the use of private connectivity.
- **Setting Up Methods**

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

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

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

    - Prerequisites
        - Connection: The Direct Connect connection or link aggregation group for which you are creating the virtual interface.
        - Virtual interface name: A name for the virtual interface.
        - Virtual interface owner
        - (Private virtual interface only) Connection to
        - VLAN: A unique virtual local area network tag that's not already in use on your connection.
        - Address family: Whether the BGP peering session will be over IPv4 or IPv6.
        - Peer IP addresses: A virtual interface can support a BGP peering session for IPv4, IPv6, or one of each (dual-stack). You cannot create multiple BGP sessions for the same IP addressing family on the same virtual interface
        - BGP information: A public or private Border Gateway Protocol Autonomous System Number for your side of the BGP session, and an MD5 BGP authentication key.
        - (Public virtual interface only) Prefixes you want to advertise: Public IPv4 routes or IPv6 routes to advertise over BGP. You must advertise at least one prefix using BGP.
    - 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 **master account** is the AWS account you use to create your organization. You cannot change which account in your organization is the master account.
    - From the master account, you can create other accounts in your organization, invite and manage invitations for other accounts to join your organization, and remove accounts from your organization.
    - You can also attach policies to entities such as administrative roots, organizational units (OUs), or accounts within your organization.
    - The master account has the role of a payer account and is responsible for paying all charges accrued by the accounts in its organization.
- A **member account** is an AWS account, other than the master account, that is part of an organization. A member account can belong to only one organization at a time. The master account has the responsibilities of a payer account and is responsible for paying all charges that are accrued by the member accounts.

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

Pricing

- This service is free.

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

AWS CloudFormation

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

Features

- CloudFormation allows you to model your entire infrastructure in a text file called a **template**. You can use JSON or YAML to describe what AWS resources you want to create and configure. If you want to design visually, you can use *AWS CloudFormation Designer*.
- CloudFormation automates the provisioning and updating of your infrastructure in a safe and controlled manner. You can use **Rollback Triggers** to specify the CloudWatch alarm that CloudFormation should monitor during the stack creation and update process. If any of the alarms are breached, CloudFormation rolls back the entire stack operation to a 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.
- **CloudFormation Designer** is a graphic tool for creating, viewing, and modifying CloudFormation templates. You can diagram your template resources using a drag-and-drop interface, and then edit their details using the integrated JSON and YAML editor.
- Custom resources enable you to write custom provisioning logic in templates that CloudFormation runs anytime you create, update (if you changed the custom resource), or delete stacks.
- Template macros enable you to perform custom processing on templates, from simple actions like find-and-replace operations to extensive transformations of entire templates.

StackSets

- 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.
- 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 now use AWS Systems Manager Patch Manager to select and apply Microsoft application patches automatically across your Amazon EC2 or on-premises instances.
- **Maintenance Window**
    - Set up recurring schedules for managed instances to execute administrative tasks like installing patches and updates without interrupting business-critical operations.
    - Supports running four types of tasks:
        - Systems Manager Run Command commands
        - Systems Manager Automation workflows
        - AWS Lambda functions
        - AWS Step Functions tasks
- **Systems Manager Document (SSM)**
    - Defines the actions that Systems Manager performs.
    - Types of SSM Documents

Type	Use with	Details
Command document	Run Command, State Manager	Run Command uses command documents to execute commands. State Manager uses command documents to apply a configuration. These actions can be run on one or more targets at any point during the lifecycle of an instance.
Policy document	State Manager	Policy documents enforce a policy on your targets. If the policy document is removed, the policy action no longer happens.
Automation document	Automation	Use automation documents when performing common maintenance and deployment tasks such as creating or updating an AMI.

Package document	Distributor	In Distributor, a package is represented by a Systems Manager document. A package document includes attached ZIP archive files that contain software or assets to install on managed instances. Creating a package in Distributor creates the package document.

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

State Manager

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

Parameter Store

- Provides secure, hierarchical storage for configuration data and secrets management.
- You can store values as plain text or encrypted data 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 CloudWatch Events, OpsCenter helps you reduce the mean time to resolution (MTTR) of incidents, alarms, and operational tasks.

Monitoring

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

- Log System Manager API calls with CloudTrail.

Security

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

Pricing

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

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 CloudWatch Events to detect and react to changes in the status of AWS Config events.
- Use AWS CloudTrail to capture API calls to Config.

Security

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

Compliances

- ISO
- PCI DSS
- HIPAA

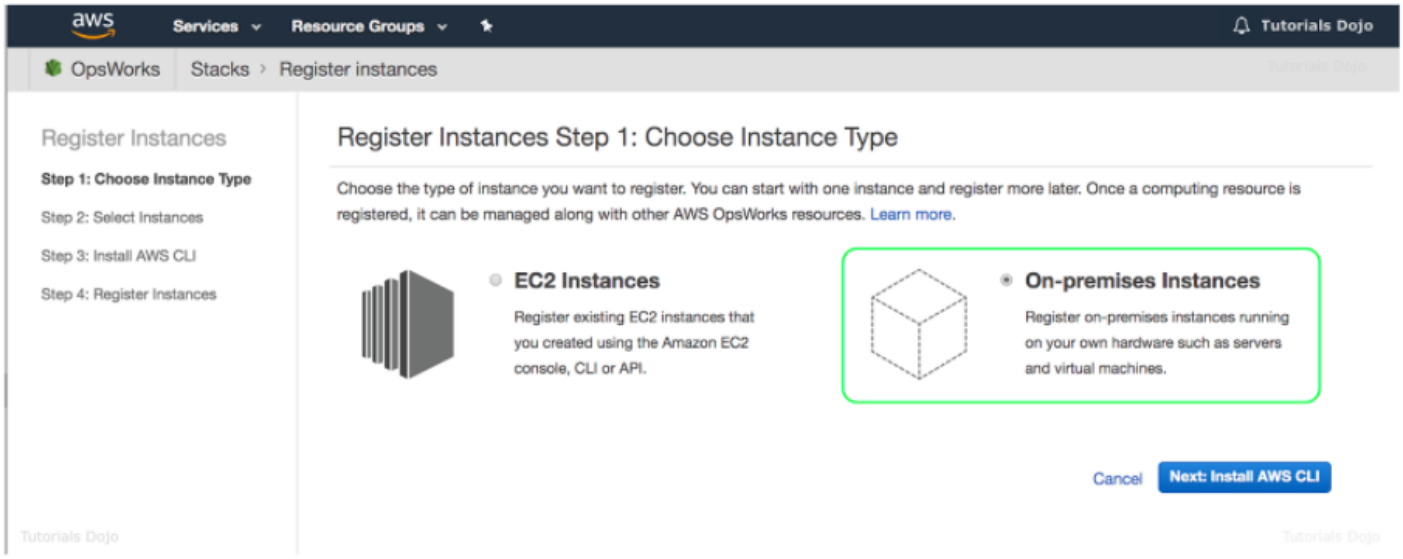
Pricing

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

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/

AWS OpsWorks

- A configuration management service that helps you configure and operate applications in a cloud enterprise by using **Puppet** or **Chef**.
- AWS OpsWorks Stacks and AWS OpsWorks for Chef Automate (1 and 2) let you use Chef cookbooks and solutions for configuration management, while OpsWorks for Puppet Enterprise lets you configure a Puppet Enterprise master server in AWS.
- With AWS OpsWorks, you can automate how nodes are configured, deployed, and managed, whether they are Amazon EC2 instances or on-premises devices:



OpsWorks for Puppet Enterprise

- Provides a fully-managed Puppet master, a suite of automation tools that enable you to inspect, deliver, operate, and future-proof your applications, and access to a user interface that lets you view information about your nodes and Puppet activities.
- Does not support all regions.
- Uses puppet-agent software.
- **Features**
    - AWS manages the Puppet master server running on an EC2 instance. You retain control over the underlying resources running your Puppet master.
    - You can choose the weekly maintenance window during which OpsWorks for Puppet Enterprise will automatically install updates.
    - Monitors the health of your Puppet master during update windows and automatically rolls back changes if issues are detected.

    - You can configure automatic backups for your Puppet master and store them in an S3 bucket in your account.
    - You can register new nodes to your Puppet master by inserting a user-data script, provided in the *OpsWorks for Puppet Enterprise StarterKit*, into your Auto Scaling groups.
    - Puppet uses SSL and a certification approval process when communicating to ensure that the Puppet master responds only to requests made by trusted users.
- Deleting a server also deletes its events, logs, and any modules that were stored on the server. Supporting resources are also deleted, along with all automated backups.
- **Pricing**
    - You are charged based on the number of nodes (servers running the Puppet agent) connected to your Puppet master and the time those nodes are running on an hourly rate, and you also pay for the underlying EC2 instance running your Puppet master.

OpsWorks for Chef Automate

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

OpsWorks Stacks

- Provides a simple and flexible way to create and manage stacks and applications.
- **Stacks** are group of AWS resources that constitute an full-stack application. By default, you can create up to 40 Stacks, and each stack can hold up to 40 layers, 40 instances, and 40 apps.
- You can create stacks that help you manage cloud resources in specialized groups called **layers**. A layer represents a set of EC2 instances that serve a particular purpose, such as serving applications or hosting a database server. Layers depend on Chef recipes to handle tasks such as installing packages on instances, deploying apps, and running scripts.
- OpsWorks Stacks does NOT require or create Chef servers.
- **Features**
    - You can deploy EC2 instances from template configurations, including EBS volume creation.
    - You can configure the software on your instances on-demand or automatically based on lifecycle events, from bootstrapping the base OS image into a working server to modifying running services to reflect changes.
    - OpsWorks Stacks can auto heal your stack. If an instance fails in your stack, OpsWorks Stacks can replace it with a new one.
    - You can adapt the number of running instances to match your load, with time-based or load-based auto scaling.
    - You can use OpsWorks Stacks to configure and manage both Linux and Windows EC2 instances.
    - You can use AWS OpsWorks Stacks to deploy, manage, and scale your application on any Linux server such as EC2 instances or servers running in your own data center.
- **Instance Types**
    - **24/7 instances** are started manually and run until you stop them.
    - **Time-based instances** are run by OpsWorks Stacks on a specified daily and weekly schedule. They allow your stack to automatically adjust the number of instances to accommodate predictable usage patterns.
    - **Load-based instances** are automatically started and stopped by OpsWorks Stacks, based on specified load metrics, such as CPU utilization. They allow your stack to automatically adjust the number of instances to accommodate variations in incoming traffic.
        - Load-based instances are available only for Linux-based stacks.
- **Lifecycle Events**
    - You can run recipes manually, but OpsWorks Stacks also lets you automate the process by supporting a set of five lifecycle events:
        - **Setup** occurs on a new instance after it successfully boots.
        - **Configure** occurs on all of the stack's instances when an instance enters or leaves the online state.
        - **Deploy** occurs when you deploy an app.
        - **Undeploy** occurs when you delete an app.
        - **Shutdown** occurs when you stop an instance.

- **Monitoring**
  - OpsWorks Stacks sends all of your resource metrics to CloudWatch.
  - Logs are available for each action performed on your instances.
  - CloudTrail logs all API calls made to OpsWorks.
- **Security**
  - Grant IAM users access to specific stacks, making management of multi-user environments easier.
  - You can also set user-specific permissions for actions on each stack, allowing you to decide who can deploy new application versions or create new resources.
  - Each EC2 instance has one or more associated *security groups* that govern the instance's network traffic. A security group has one or more rules, each of which specifies a particular category of allowed traffic.
- **Pricing**
  - You pay for AWS resources created using OpsWorks Stacks in the same manner as if you created them manually.

References:
https://aws.amazon.com/opsworks/chefautomate/faqs
https://aws.amazon.com/opsworks/puppetenterprise/faqs
https://aws.amazon.com/opsworks/stacks/faqs

Amazon CloudWatch

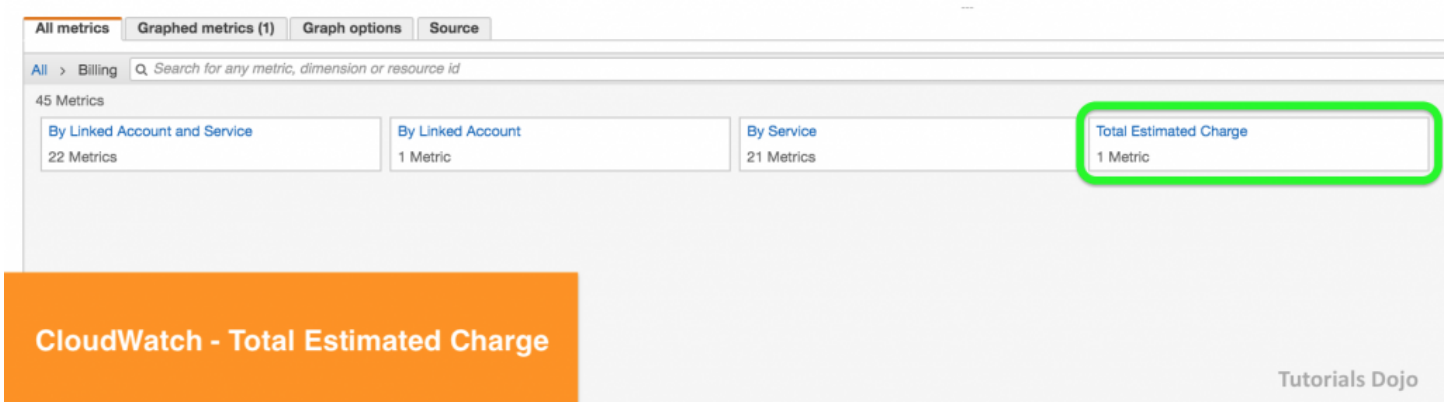
- Monitoring tool for your AWS resources and applications.
- Display metrics and create alarms that watch the metrics and send notifications or automatically make changes to the resources you are monitoring when a threshold is breached.
- CloudWatch is basically a metrics repository. An AWS service, such as Amazon EC2, puts metrics into the repository and you retrieve statistics based on those metrics. If you put your own custom metrics into the repository, you can retrieve statistics on these metrics as well.
- CloudWatch does not aggregate data across regions. Therefore, metrics are completely separate between regions.
- **CloudWatch Concepts**
    - **Namespaces** - a container for CloudWatch metrics.
        - There is no default namespace.
        - The AWS namespaces use the following naming convention: AWS/*service*.
    - **Metrics** - represents a time-ordered set of data points that are published to CloudWatch.
        - Exists only in the region in which they are created.
        - Cannot be deleted, but they automatically expire after 15 months if no new data is published to them.
        - As new data points come in, data older than 15 months is dropped.
        - Each metric data point must be marked with a *timestamp*. The timestamp can be up to two weeks in the past and up to two hours into the future. If you do not provide a timestamp, CloudWatch creates a timestamp for you based on the time the data point was received.
        - By default, several services provide free metrics for resources. You can also enable **detailed monitoring**, or publish your own application metrics.
        - **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*.

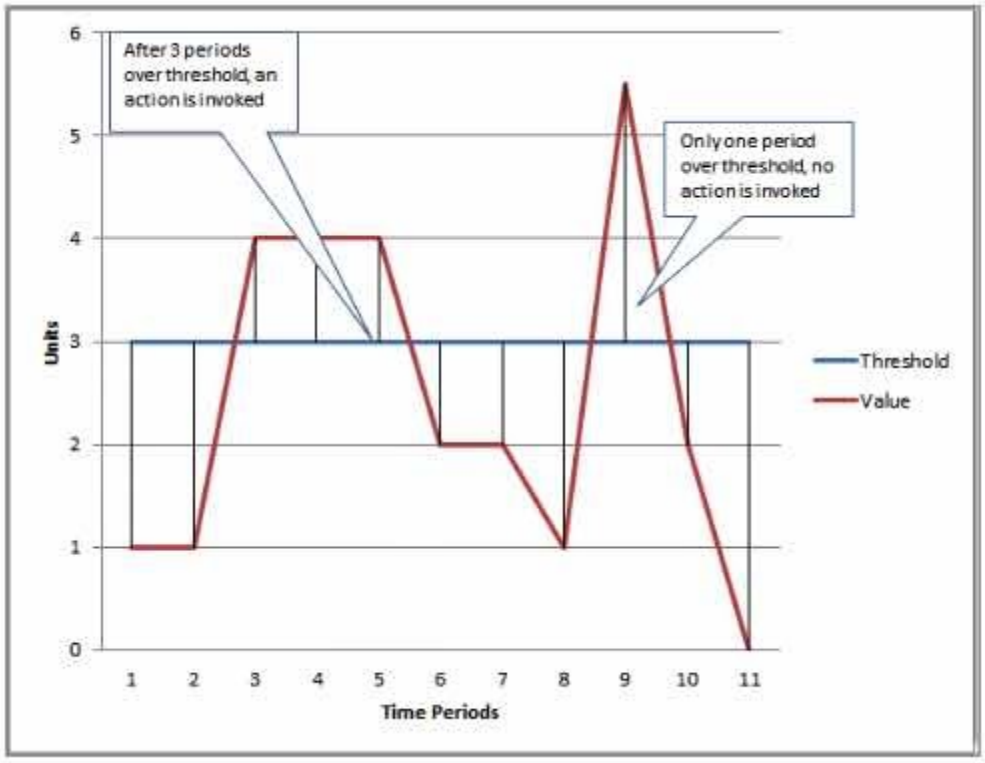
Statistic	Description
Minimum	The lowest value observed during the specified period. You can use this value to determine low volumes of activity for your application.
Maximum	The highest value observed during the specified period. You can use this value to determine high volumes of activity for your 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.

CloudWatch Events

- Deliver near real-time stream of system events that describe changes in AWS resources.
- Events respond to these operational changes and take corrective action as necessary, by sending messages to respond to the environment, activating functions, making changes, and capturing state information.

- Concepts
  - **Events** - indicates a change in your AWS environment.
  - **Targets** - processes events.
  - **Rules** - matches incoming events and routes them to targets for processing.

CloudWatch Logs

- Features
  - Monitor logs from EC2 instances in real-time
  - Monitor CloudTrail logged events
  - By default, logs are kept indefinitely and never expire
  - Archive log data
  - Log Route 53 DNS queries
- **CloudWatch Logs Insights** enables you to interactively search and analyze your log data in CloudWatch Logs using queries.
- **CloudWatch Vended logs** are logs that are natively published by AWS services on behalf of the customer. **VPC Flow logs** is the first Vended log type that will benefit from this tiered model.
- 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.

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
- 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 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).

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 1 replica. Optional. On by default.	Required
Backup/Restore	Yes	Yes

Tutorials Dojo

- You can vertically scale up or scale down your sharded Redis Cluster on demand. Amazon ElastiCache resizes your cluster by changing the node type, while the cluster continues to stay online and serve incoming requests.
- You can set up automatic snapshots or initiate manual backups, and then seed new ElastiCache for Redis clusters. You can also export your snapshots to an S3 bucket of your choice for disaster recovery, analysis or cross-region backup and restore.
- Endpoints
  - **Single Node Redis** (**cluster mode disabled**) Endpoints - used to connect to the cluster for both reads and writes.
  - **Multi-Node Redis** (**cluster mode disabled**) Endpoints - use the primary endpoint for all writes to the cluster. The read endpoint points to your read replicas.
  - **Redis** (**cluster mode enabled**) Endpoints - has a single configuration endpoint. By connecting to the configuration endpoint, your application is able to discover the primary and read endpoints for each shard in the cluster.

- Parameter Groups
  - **Cache parameter group** is a named collection of engine-specific parameters that you can apply to a cluster.
  - Parameters are used to control memory usage, eviction policies, item sizes, and more.
- Redis Security
  - ElastiCache for Redis node access is restricted to applications running on whitelisted EC2 instances. You can control access of your cluster by using subnet groups or security groups. By default, network access to your clusters is turned off.
  - By default, all new ElastiCache for Redis clusters are launched in a VPC environment. Use subnet groups to grant cluster access from Amazon EC2 instances running on specific subnets.
  - ElastiCache for Redis supports TLS and in-place encryption for nodes running specified versions of the ElastiCache for Redis engine.
  - You can use your own customer managed customer master keys (CMKs) in AWS Key Management Service to encrypt data at rest in ElastiCache for Redis.
- Redis Backups
  - A point-in-time copy of a Redis cluster.
  - Backups consist of all the data in a cluster plus some metadata.
- 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	Yes
Node type upgrading	No	Yes	No
Engine upgrading	Yes		
Cluster replication (create multiple copies of a primary cluster)	Supported	Supported	Unsupported
Multi-AZ for automatic failover	Required	Optional	Unsupported
Transactions (execute a group of commands as an isolated and atomic operation)	Supported	Supported	Unsupported
Pub/Sub capability	Yes	Yes	No
Backup and restore (keep your data on disk with a point in time snapshot)	Supported	Supported	Unsupported
Lua Scripting (execute transactional Lua scripts)	Supported	Supported	Unsupported
Use Case	• You need to partition your data across two to 90 node groups (clustered mode only). • You need geospatial indexing (clustered mode or non-clustered mode). • You don't need to support multiple databases • Plus features of non-clustered mode	• You need complex data types, such as strings, hashes, lists, sets, sorted sets, and bitmaps. • You need to sort or rank in-memory datasets. • You need persistence of your key store. • You need to replicate your data from the primary to one or more read replicas for read intensive applications. • You need automatic failover if your primary node fails. • You need pub/sub capabilities. • You need backup and restore capabilities. • You need to support multiple databases.	• You need the simplest model possible. • You need to run large nodes with multiple cores or threads. • You need the ability to scale out and in, adding and removing nodes as demand on your system increases and decreases. • You need to cache objects, such as a database. • Needs Auto Discovery to simplify the way an application connects to a cluster.

Pricing

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

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.
    - **Amazon CloudWatch Events** – Match events and route them to one or more target functions or streams to make changes, capture state information, and take corrective action.
    - **AWS CloudTrail Log Monitoring** – Share log files between accounts, monitor CloudTrail log files in real time by sending them to CloudWatch Logs, write log processing applications in Java, and validate that your log files have not changed after delivery by CloudTrail.
- Using the information collected by CloudTrail, you can determine the request that was made to DynamoDB, the IP address from which the request was made, who made the request, when it was made, and additional details.

DynamoDB Accelerator (DAX)

- 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 ulimits 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.
- **Advanced**
  - Shield Advanced provides enhanced detection, inspecting network flows and also monitoring application layer traffic to your Elastic IP address, Elastic Load Balancing, CloudFront, or Route 53 resources.
  - It handles the majority of DDoS protection and mitigation responsibilities for **layer 3**, **layer 4**, and **layer 7** attacks.
  - You have 24x7 access to the AWS DDoS Response Team. To contact the DDoS Response Team, customers will need the Enterprise or Business Support levels of AWS Premium Support.
  - 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/

Amazon Mechanical Turk

- A forum where **Requesters** post work as **Human Intelligence Tasks** (HITs). Workers complete HITs in exchange for a reward. Essentially crowdsourcing.
- You write, test, and publish your HIT using the Mechanical Turk developer sandbox, Amazon Mechanical Turk APIs, and AWS SDKs.
- Benefits
    - Optimize efficiency since MTurk is well-suited to take on simple and repetitive tasks in your workflows which need to be handled manually.
    - Increase flexibility since MTurk lets you gain access to a global, on-demand, 24x7 workforce without the difficulty associated with dynamically scaling.
    - Reduce cost by hiring and managing a temporary workforce. MTurk provides a pay-per-task model.
- Concepts
    - A **Requester** is a company, organization, or person that creates and submits tasks (HITs) to Amazon Mechanical Turk for Workers to perform.
    - A **Human Intelligence Task** (HIT) represents a single, self-contained task that a Requester submits to Amazon Mechanical Turk for Workers to perform.
        - Each HIT has a lifetime, specified by the Requester, that determines how long the HIT is available to Workers.
        - A HIT also has an assignment duration, which is the amount of time a Worker has to complete a HIT after accepting it.
    - A **Worker** is a person who performs the tasks specified by a Requester in a HIT.
        - The Requester specifies how many Workers can work on a task.
        - Amazon Mechanical Turk guarantees that a Worker can work on each task only one time.
    - **Developers** create the Mechanical Turk applications that Requesters and Workers use.
        - Requesters can create and advertise work using the Mechanical Turk command line interface or the Requester User Interface and thereby not need developers
    - An **Assignment** specifies how many people can submit completed work for your HIT. When a Worker accepts a HIT, MTurk creates an assignment to track the work to completion. The assignment belongs exclusively to the Worker and guarantees that the Worker can submit results and be eligible for a reward until the time the HIT or assignment expires.
    - A **reward** is the money a Requester pays to Workers for the satisfactory work they do on HITs.
    - A **Qualification** is an attribute assigned by you to a Worker. It includes a name and a number value. A HIT can include Qualification requirements that a Worker must meet before they are allowed to accept the HIT.
        - A Qualification type may include a *Qualification test*. A Qualification test is a set of questions, similar to a HIT, that the Worker must answer to request the Qualification.

References:
https://aws.amazon.com/premiumsupport/knowledge-center/mechanical-turk-use-cases/
https://www.mturk.com/
https://docs.aws.amazon.com/AWSMechTurk/latest/AWSMechanicalTurkRequester/Welcome.html

Comparison of AWS Services and Features

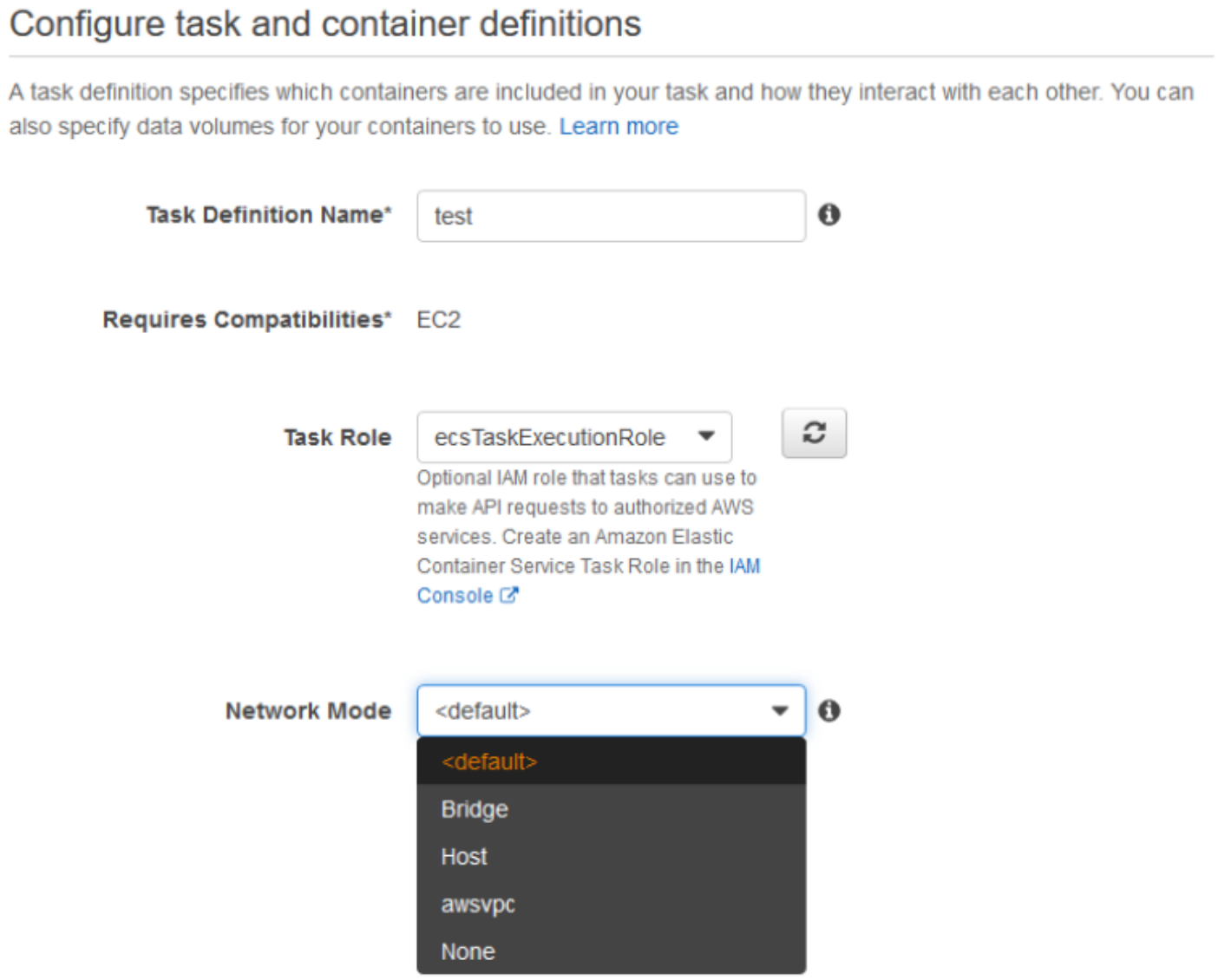
ECS Network Mode Comparison

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.

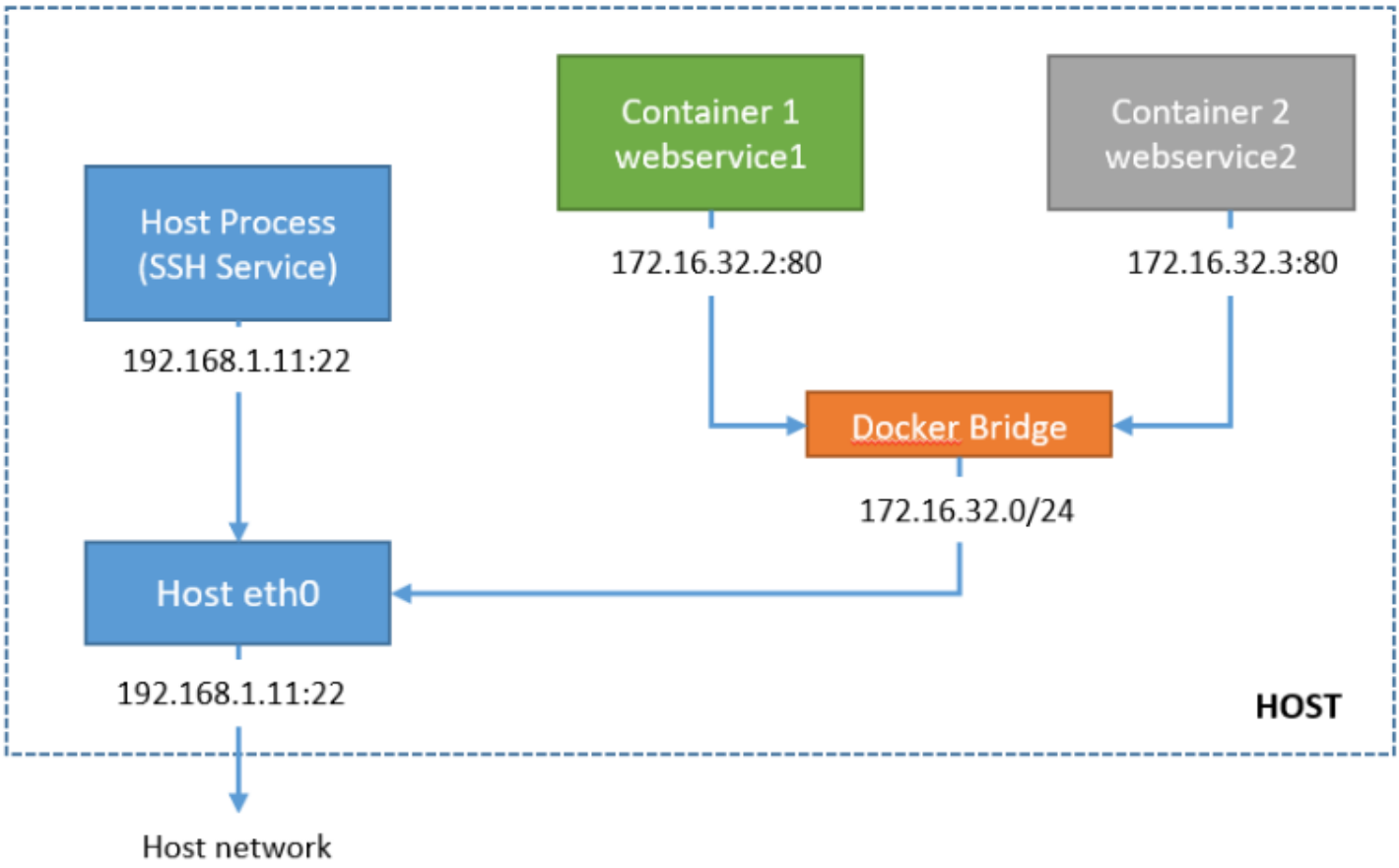


Bridge network mode - Default

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

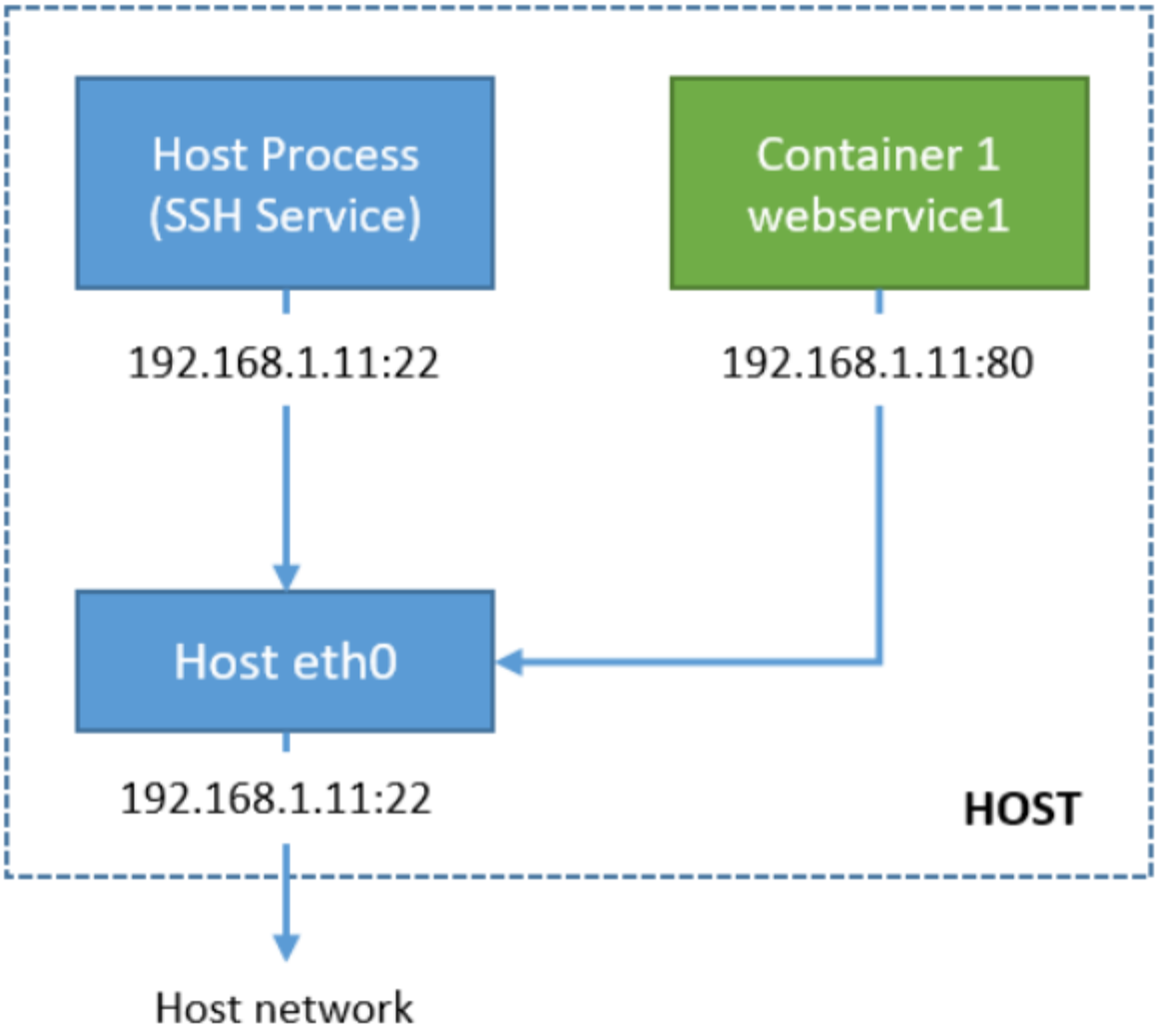
Docker driver handles this isolation on the host machine so that containers on different bridge networks cannot communicate with each other.



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

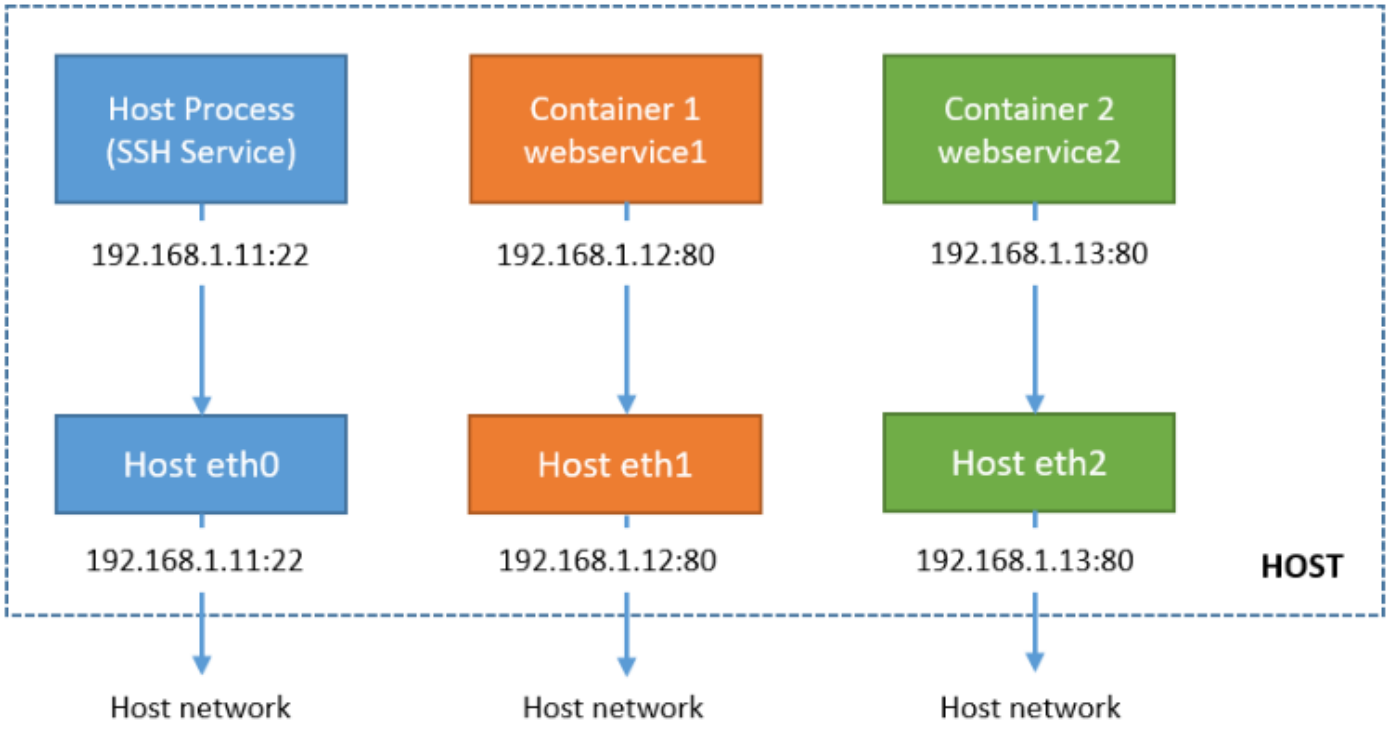
Host network mode bypasses the Docker's built-in virtual network and maps container ports directly to your EC2 instance's network interface. This mode shares the same network namespace of the host EC2 instance so your containers share the same IP with your host IP address. This also means that you can't 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

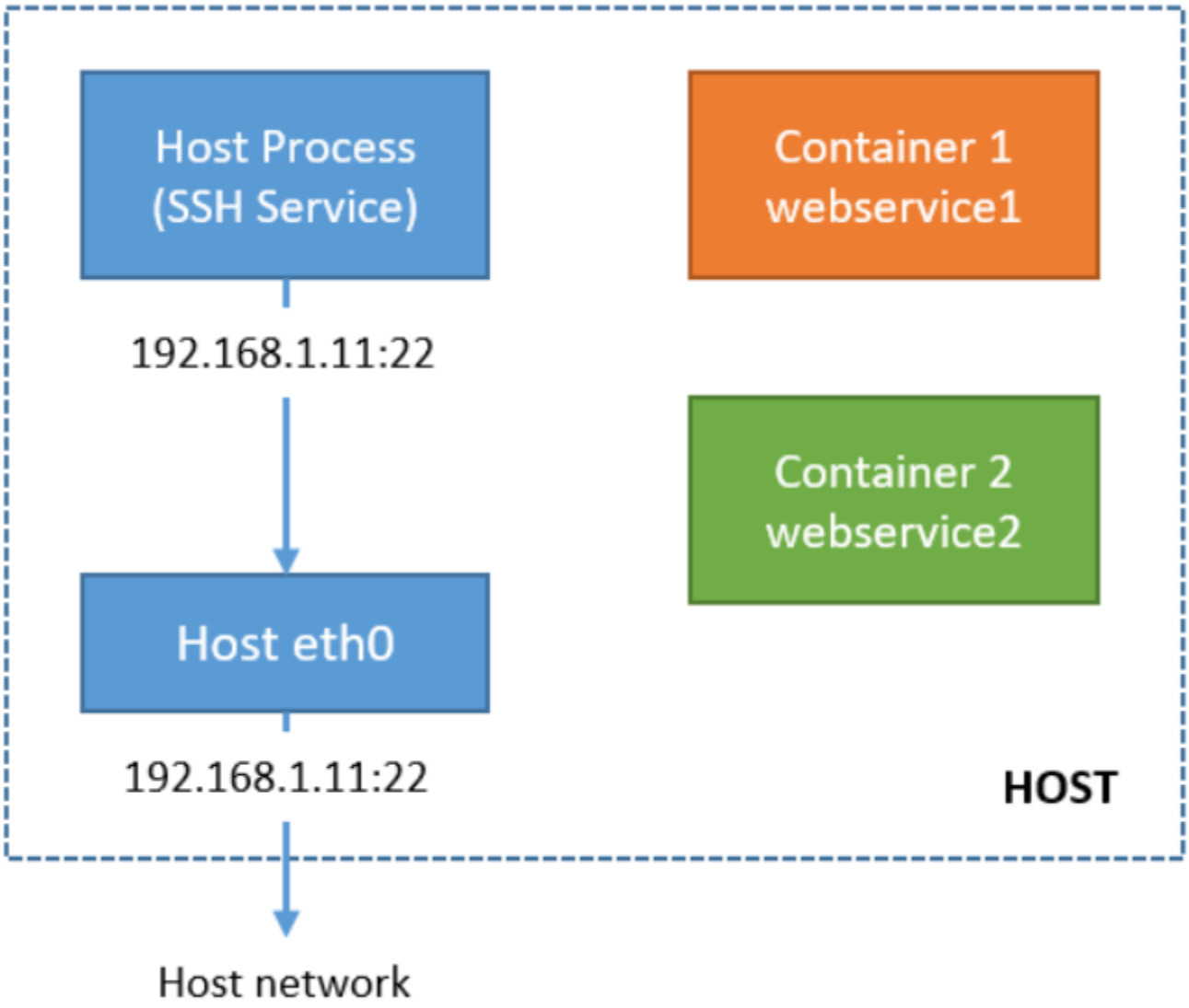
The **awsvpc** mode provides an elastic network interface for each task definition. If you have one container per task definition, each container will have its own elastic network interface and will get its own IP address from your VPC subnet IP address pool. This offers faster performance than the bridge 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.

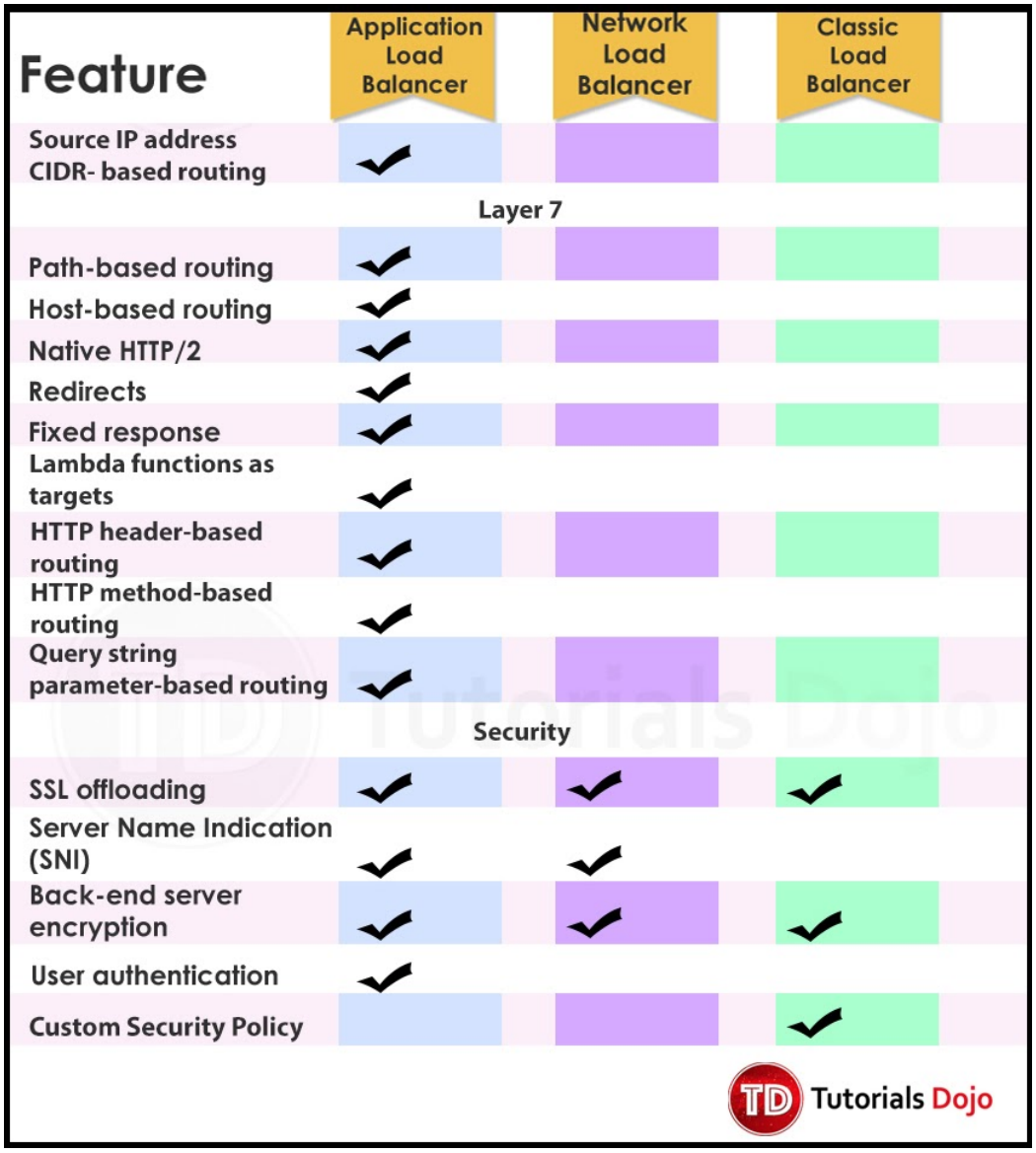


You can use this mode if you don't want your containers to access the host network, or if you want to use a custom network driver other than the built-in driver from Docker. You can only access the container from inside the EC2 host with the Docker command.

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 Classic Load Balancer

Feature	Application Load Balancer	Network Load Balancer	Classic Load Balancer
Protocols	HTTP HTTPS	TCP, UDP, TLS	TCP, SSL/TLS, HTTP, HTTPS
Platforms	VPC	VPC	EC2-Classic, VPC
Healthchecks	✓	✓	✓
Cloudwatch Metrics	✓	✓	✓
Logging	✓	✓	✓
Zonal Failover	✓	✓	✓
Connection Draining (deregistration delay)	✓		✓
Load Balancing to multiple ports on the same instance	✓	✓	
IP addresses as targets	✓	✓ (TCP, TLS)	
Load balancer deletion protection	✓	✓	
Configurable idle connection timeout	✓		✓
Cross-zone load balancing	✓	✓	✓
Sticky sessions	✓	✓	✓
Static IP		✓	
Elastic IP address		✓	
Preserve Source IP address		✓	
Resource-based IAM permissions	✓	✓	✓
Tag-based IAM permissions	✓	✓	
Slow start	✓		
Web sockets	✓	✓	
PrivateLink Support		✓ (TCP, TLS)	



UNIQUE FEATURES

Application Load Balancer

- You can set Lambda functions as load balancing targets
- Only ALB supports the following content- basaed routing method:
    - Path based routing
    - Host-based routing
    - HTTP header based-routing
    - HTTP method based-routing
    - Query string parameter based-routing
    - Source IP address CIDR based-routing

- Natively supports HTTP/2 IPv6
- Support for multiple SSL certificates on the ALB using
- Server Name Indication (SNI)
- Allows tag based- IAM permission policies
- Can be configured for slow start (linearly increase the number of requests sent to target)
- Supports round-robin load balancing
- You can offload the authentication functionality from your apps into ALB
- Can redirect an incoming request from one URL to a other URL INCLUDING HTTP to HTTPS
- You can set HTTP or custom responses for incoming requests to the ALB, offloading this task from your application

Network Load Balancer

- High throughput/low latency ELB
- Can be assigned a static IP address
- Can be assigned an elastic IP address
- Preserves source IP address of non-HTTP applications on EC2 instances
- Offer multi-protocol listeners, allowing you to run applications such as DNS that rely on both TCP and UDP protocols on the same port behind a Network Load Balancer.
- TLS Termination

Classic Load Balancer

- You can create custom security policies detailing which ciphers and protocols are supported by the ELB
- Supports both IPv4 and IPv6 for EC2-classic network

Common features between the three load balancers

- Has instance health check features
- Has built-in CloudWatch monitoring
- Logging features
- Support zonal failover
- Support connection draining when deregistering targets/instances
- Support cross-zone load balancing (evenly distributes traffic across registered instances in enabled AZs)
- Support SSL offloading/termination
- Backend server encryption
- Resource-based IAM permission policies

S3 Pre-Signed URLS vs CloudFront Signed URLS vs Origin Access Identity

S3 Pre-signed URLs	CloudFront Signed URLs	Origin Access Identity (OAI)
All S3 buckets and objects by default are **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. OAI prevents users from viewing your S3 files by simply using the direct URL for the file. Instead, they would need to access it through a CloudFront URL.
When creating a pre-signed URL, you (as the owner) need to provide the following: • 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 identity**. • Give the origin access identity 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 OAI if your S3 bucket is configured as a website endpoint.

S3 Transfer Acceleration vs Direct Connect vs VPN vs Snowball Edge vs Snowmobile

S3 Transfer Acceleration (TA)

- Amazon S3 Transfer Acceleration makes public Internet transfers to S3 faster, as it leverages Amazon CloudFront's globally distributed AWS Edge Locations.
- There is no guarantee that you will experience increased transfer speeds. If S3 Transfer Acceleration is not likely to be faster than a regular S3 transfer of the same object to the same destination AWS Region, AWS will not charge for the use of S3 TA for that transfer.
- This is not the best transfer service to use if transfer disruption is not tolerable.
- S3 TA provides the same security benefits as regular transfers to Amazon S3. This service also supports multi-part upload.
- ***S3 TA vs AWS Snow****
    - The AWS Snow* Migration Services are ideal for moving large batches of data at once. In general, if it will take more than a week to transfer over the Internet, or there are recurring transfer jobs and there is more than 25Mbps of available bandwidth, S3 Transfer Acceleration is a good option.
    - Another option is to use AWS Snowball Edge or Snowmobile to perform initial heavy lift moves and then transfer incremental ongoing changes with S3 Transfer Acceleration.
- ***S3 TA vs Direct Connect***
    - AWS Direct Connect is a good choice for customers who have a private networking requirement or who have access to AWS Direct Connect exchanges. S3 Transfer Acceleration is best for submitting data from distributed client locations over the public Internet, or where variable network conditions make throughput poor.
- ***S3 TA vs VPN***
    - You typically use (IPsec) VPN if you want your resources contained in a private network. VPN tools such as OpenVPN allow you to 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**.

Snowball Edge

- Snowball Edge is a **petabyte-scale data transport** solution that uses secure appliances to transfer large amounts of data into and out of AWS.
- Benefits of Snowball Edge include:
    - lower network costs,
    - Shorter transfer times,
    - and security using 256-bit encryption keys you manage through AWS Key Management Service (KMS)..
- Similar to Direct Connect, AWS Snowball Edge is **physical hardware**. It includes a 10GBaseT network connection. You can order a Snowball Edge Compute Optimized device which provides 42 TB usable HDD capacity for S3 compatible object storage or EBS-compatible block volumes, as well as 7.68 TB of usable NVMe SSD capacity for EBS-compatible block volumes, or a Snowball Edge Storage Optimized

device which provides 80 TB of usable HDD capacity for EBS-compatible block volumes and Amazon S3-compatible object storage, and 1 TB of SATA SSD for block volumes.
- Data transported via Snowball Edge are stored in Amazon S3 once the device arrives at AWS centers.
- AWS Snowball Edge is not only for shipping data into AWS, but also out of AWS.
- AWS Snowball Edge can be used as a quick order for additional temporary petabyte storage.
- For security purposes, data transfers must be completed **within 90 days of a Snowball's preparation**.
- When the transfer is complete and the device is ready to be returned, the E Ink shipping label will automatically update to indicate the correct AWS facility to ship to, and you can track the job status by using Amazon Simple Notification Service (SNS), text messages, or directly in the console.
- Snowball Edge is the best choice if you need to more securely and quickly transfer terabytes to many petabytes of data to AWS. Snowball Edge can also be the right choice if you don't want to make expensive upgrades to your network infrastructure, if you frequently experience large backlogs of data, if you're located in a physically isolated environment, or if you're in an area where high-bandwidth Internet connections are not available or cost-prohibitive.
- If you will be transferring data to AWS on an ongoing basis, it is better to use AWS Direct Connect.
- If multiple users located in different locations are interacting with S3 continuously, it is better to use S3 TA.
- You **cannot** export data directly from S3 Glacier. It should be first restored to S3.

Snowmobile

- Snowmobile is Snowball Edge with larger storage capacity. Snowmobile is literally a mobile truck.
- Snowmobile is an **Exabyte-scale data transfer** service.
- You can transfer up to **100PB** per Snowmobile.
- Snowmobile uses multiple layers of security to help protect your data including dedicated security personnel, GPS tracking, alarm monitoring, 24/7 video surveillance, and an optional escort security vehicle while in transit. All data is encrypted with 256-bit encryption keys you manage through the AWS Key Management Service (KMS).
- After the data transfer is complete, the Snowmobile will be returned to your designated AWS region where your data will be uploaded into the AWS storage services such as S3 or Glacier.
- Snowball Edge vs Snowmobile
    - To migrate large datasets of 10PB or more in a single location, you should use Snowmobile. For datasets less than 10PB or distributed in multiple locations, you should use Snowball.
    - If you have a high speed backbone with hundreds of Gb/s of spare throughput, then you can use Snowmobile to migrate the large datasets all at once. If you have limited bandwidth on your backbone, you should consider using multiple Snowballs to migrate the data incrementally.
    - Snowmobile **does not** support data export. Use Snowball/Snowball Edge for this cause.
- When the data import has been processed and verified, AWS performs a software erasure based on NIST guidelines.

Backup and Restore vs Pilot Light vs Warm Standby vs Multi-site

Backup Restore

- This DR plan provides the slowest system restoration after a DR event.
- 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.
- 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
  - Transfer using transport hardware such as AWS Snowball and Snowmobile
- 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.
- 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.
  - Gateway-VTL of AWS Storage Gateway serves as a replacement for traditional magnetic tape backup.
- You can quickly create local volumes or Amazon EBS volumes from snapshots in S3. 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
- Backup and restore is used in combination with other DR plans since it is crucial to always have a working backup of your system.

Pilot Light

- 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 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.
- Pilot light is an example of active/passive failover configuration.
  Infrastructure elements for the pilot light itself typically include your database servers, which would be configured for data mirroring replication.
- Restoring the rest of the system includes utilizing EBS snapshots and EC2 AMIs that you should be regularly generating.
- Pilot light tends to be more costly than backup and restore since you leave a few core AWS resources running all the time.
- 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.
- Consider redundancy especially at your data layer (enable multi-AZ, cluster sharding, etc).
- 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

- This DR plan is faster in system restoration than performing Pilot Light after a DR event, but is not as fast as having a Multi-site System.
- Warm standby describes a DR scenario in which a scaled-down version of a fully functional environment is always running in the cloud.
- Since it is not only your core elements that are running all the time, warm standby is usually more costly than pilot light.
- Warm standby is another example of active/passive failover configuration.
- Servers can be left running in a minimum number of EC2 instances on the smallest sizes possible. Once failover occurs, quickly resize them and add scaling capabilities. It is best to place these i nstances behind a load balancer as well.
- For the data layer, the practice is similar to pilot light where a standby resource is present and changing data is constantly being replicated to the other.
- In the case of failure of the production system, the standby environment will be scaled up for production load , and DNS records will be changed to route all traffic to AWS.
- If your data is constantly changing and failover occurs, you would have to reverse replicate your data in the DR site back to the primary site, so that any data updates received while the primary site was down can be replicated back, without the loss of data.

Multi-site

- This DR plan is the fastest in system restoration during a DR event.
- Multi-site is a one-to-one copy of your infrastructure that is located and running in another region or AZ, known as an active-active configuration.
- Because of this, multi-site is the most expensive among all DR plans.
- Multi-site gives you the best RTO and RPO as no downtime is expected and little to no data loss should be experienced.
  In addition to recovery point options, there are various replication methods, such as synchronous and asynchronous methods.
- You can use a DNS service that supports weighted routing, such as Amazon Route 53, to route production traffic to different sites that deliver the same application or service.
- During failover, you can quickly increase compute capacity by using AWS Auto Scaling or by resizing your instances to a larger size.
- Multiple services in AWS such as RDS offer a multi-AZ feature which allows you to provision resources in a different location for a more fault-tolerant setup.
- If your data is constantly changing and failover occurs, you would have to reverse replicate your data in the DR site back to the primary site, so that any data updates received while the primary site was down can be replicated back, without the loss of data.

Tutorials Dojo

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-C01) 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-C01 exam, allocate some time to check your readiness by taking our **AWS practice test course** in the Tutorials Dojo Portal. This will help you identify the topics that you need to improve on and help reinforce the concepts that you need to fully understand in order to pass 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, Adrian Formaran, and the Tutorials Dojo Team

ABOUT THE AUTHORS

Jon Bonso (9x 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 9x 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.

Adrian Formaran (3x AWS Certified)

As a Computer Scientist and a proud university scholar, Adrian has a passion for learning cutting edge technologies, such as blockchain, cloud services, and information security, and is passionate about teaching these to others as well. He currently has 3 AWS certifications under his belt, including the AWS Certified Solutions Architect Professional. He also has a deep love for mathematics, sciences, and philosophy. A gamer at heart.