JON BONSO AND CARLO ACEBEDO

AWS CERTIFIED DEVELOPER ASSOCIATE DVA-C02 EXAM

Tutorials Dojo Study Guide

TABLE OF CONTENTS

INTRODUCTION

The trend of emerging technologies that have a large impact on industries is a common theme in today's headlines. More and more companies are adopting newer technologies that will allow them to grow and increase returns. The Amazon Web Services (AWS) Cloud is one example of it. There are more than a million active users of Amazon Web Services. These users are a mix of small businesses, independent contractors, large enterprises, developers, students, and many more. There is no doubt that AWS already has a community of users, and the number of newcomers is increasing rapidly each day.

With AWS, you don't have to purchase and manage your infrastructure. It is a considerable cost saving for your company, and it is a sigh of relief as well for your sysadmin team. Although AWS offers a huge collection of services and products that require little configuration, they also offer traditional ones like VMs and storage devices that work similarly to their physical counterparts. This means that transitioning from an on-premises type of environment to an AWS virtualized environment should be straightforward.

We think that developers benefit the most from the cloud. You can access your applications from anywhere if you have a browser and an Internet connection (and maybe an SSH/RDP client too). You can spin up machines in a matter of minutes from a catalog of OS and versions that suit your needs. You also have control over your storage devices, which you can also provision and de-provision easily. Aside from traditional VMs, AWS also has other options for you, such as container instances, batch instances, and serverless models that can easily be integrated with APIs. You also do not need to worry about databases. You can host them on your own in VMs with your own licenses or use AWS-provided licenses, or you can even use a managed database, so you do not need to worry about infrastructure. If these are too much for a simple app that you just want to test, AWS has a platform-as-a-service solution wherein you can simply deploy your code, and the service handles the infrastructure for you. In essence, AWS has already provided the tools that its users need to quickly take advantage of the cloud.

With all of the benefits presented in using AWS, the true value of being an AWS Certified Developer has your knowledge and skills in developing in AWS recognized by the industry. With this recognition, you'll gain more opportunities for career growth, financial growth, and personal growth as well. Certified individuals can negotiate for higher salaries, aim for promotions, or land higher job positions. Becoming an AWS Certified Developer Associate is also a great way to start on your journey in DevOps, which is one of the most in-demand and well-compensated roles in the IT industry today. This certificate is a must-have in your portfolio if you wish to progress your career in AWS.

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

AWS CERTIFIED DEVELOPER ASSOCIATE EXAM OVERVIEW

Amazon Web Services (AWS) began the Global Certification Program with the primary purpose of validating the technical skills and knowledge for building secure and reliable cloud-based applications using the AWS platform. By successfully passing the AWS exam, individuals can prove their AWS expertise to their current and future employers.

The AWS Certified Solutions Architect - Associate exam was the first AWS certification that was launched in 2013, followed by two other role-based certifications: Systems Operations (SysOps) Administrator and Developer Associate later that year. AWS is always releasing new development services and features on its cloud platform; thus, there is a continuous stream of updates on its exam content. Some changes are minor ones, while others are major updates that overhaul the entire certification test.

The very first version of the AWS Certified Developer - Associate exam (DVA-C00) and was released in January 2014. Four years later, in June 2018, AWS released the updated version of this test with an exam code of DVA-C01. AWS unveiled the third iteration of this test with **DVA-C02** as its latest exam code. You might notice that for every major exam update, the last digit of the exam code is incremented, so expect that the next one will be called DVA-C03 after several years. It's quite important to know the relevant details of the AWS exam version to ensure that you are using up-to-date review materials.

Exam Details

The AWS Certified Developer - Associate examination is intended for individuals who perform a development role and have one or more years of hands-on experience developing and maintaining an AWS-based application.

Exam Code:	DVA-C02
No. of Questions:	65
Score Range:	100/1000
Cost:	150 USD
Passing Score:	720/1000
Time Limit:	2 hours 10 minutes (130 minutes)
Format:	Multiple choice/multiple answers
Delivery Method:	Testing center or online proctored exam

Exam Domains

The AWS Certified Developer - Associate exam has four (4) different domains, each with corresponding weight and topic coverage. The exam domains are Development with AWS Services (32%), Security (26%), Deployment (24%), and Troubleshooting and Optimization (18%)

Domain 1: Development with AWS Services (32%)

1.1 Develop code for applications hosted on AWS
1.2 Develop code for AWS Lambda
1.3 Use data stores in application development

Domain 2: Security (26%)

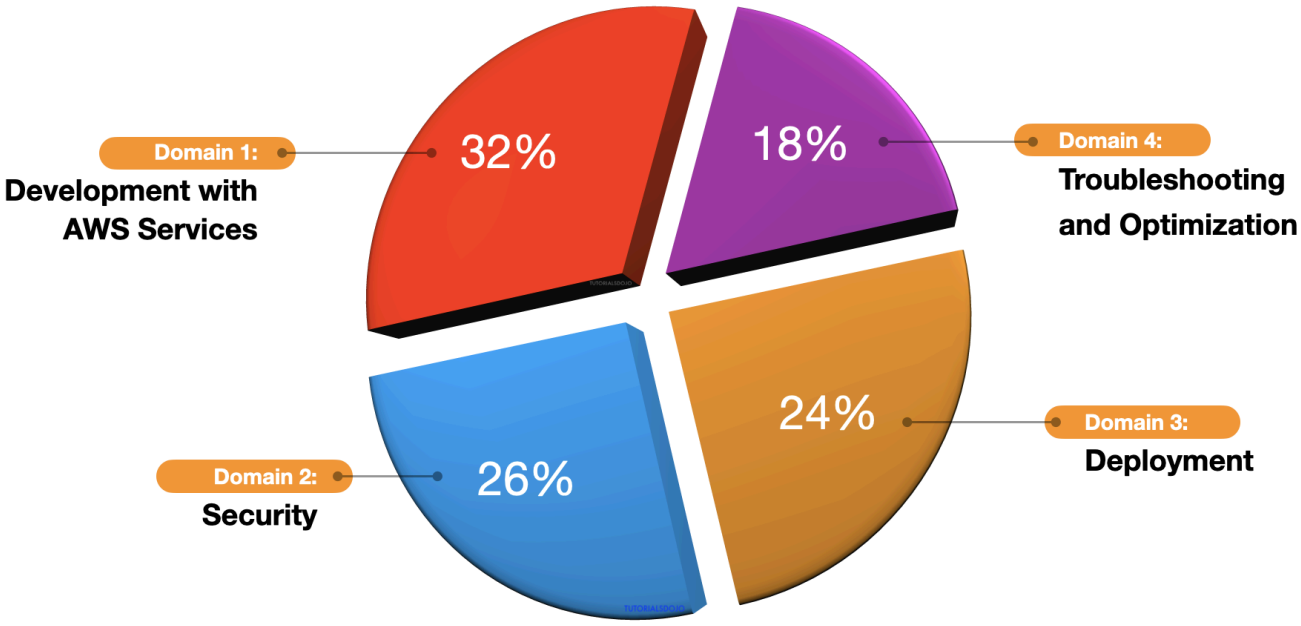
2.1 Implement authentication and/or authorization for applications and AWS services
2.2 Implement encryption by using AWS services
2.3 Manage sensitive data in application code.

Domain 3: Deployment (24%)

3.1 Prepare application artifacts to be deployed to AWS.
3.2 Test applications in development environments
3.3 Automate deployment testing
3.4 Deploy code by using AWS CI/CD services.

Domain 4: Troubleshooting and Optimization (18%)

4.1 Assist in a root cause analysis
4.2 Instrument code for observability.
4.3 Optimize applications by using AWS services and features.



Exam Scoring

You can get a score from 100 to 1,000 with a minimum passing score of **720** when you take the AWS Certified Developer Associate exam. AWS uses a scaled scoring model to associate scores across multiple exam types that may have different levels of difficulty. Your complete score report will be sent to you by email 1 - 5 business days after your exam. However, as soon as you finish your exam, you'll immediately see a pass or fail notification on the testing screen.

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

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

Exam Benefits

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

- **Exam Discount** - You'll get a 50% discount voucher to apply for your recertification or any other exam you plan to pursue. To access your discount voucher code, go to the "Benefits" section of your AWS Certification Account, and apply the voucher when you register for your next exam.

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

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

AWS CERTIFIED DEVELOPER ASSOCIATE EXAM - STUDY GUIDE AND TIPS

The AWS Certified Developer Associate certification is for those who are interested in handling cloud-based applications and services. Typically, applications developed in AWS are sold as products in the AWS Marketplace. This allows other customers to use the customized, cloud-compatible application for their own business needs. Because of this, AWS developers should be proficient in using the AWS CLI, APIs, and SDKs for application development.

The AWS Certified Developer Associate exam (or AWS CDA for short) will test your ability to:

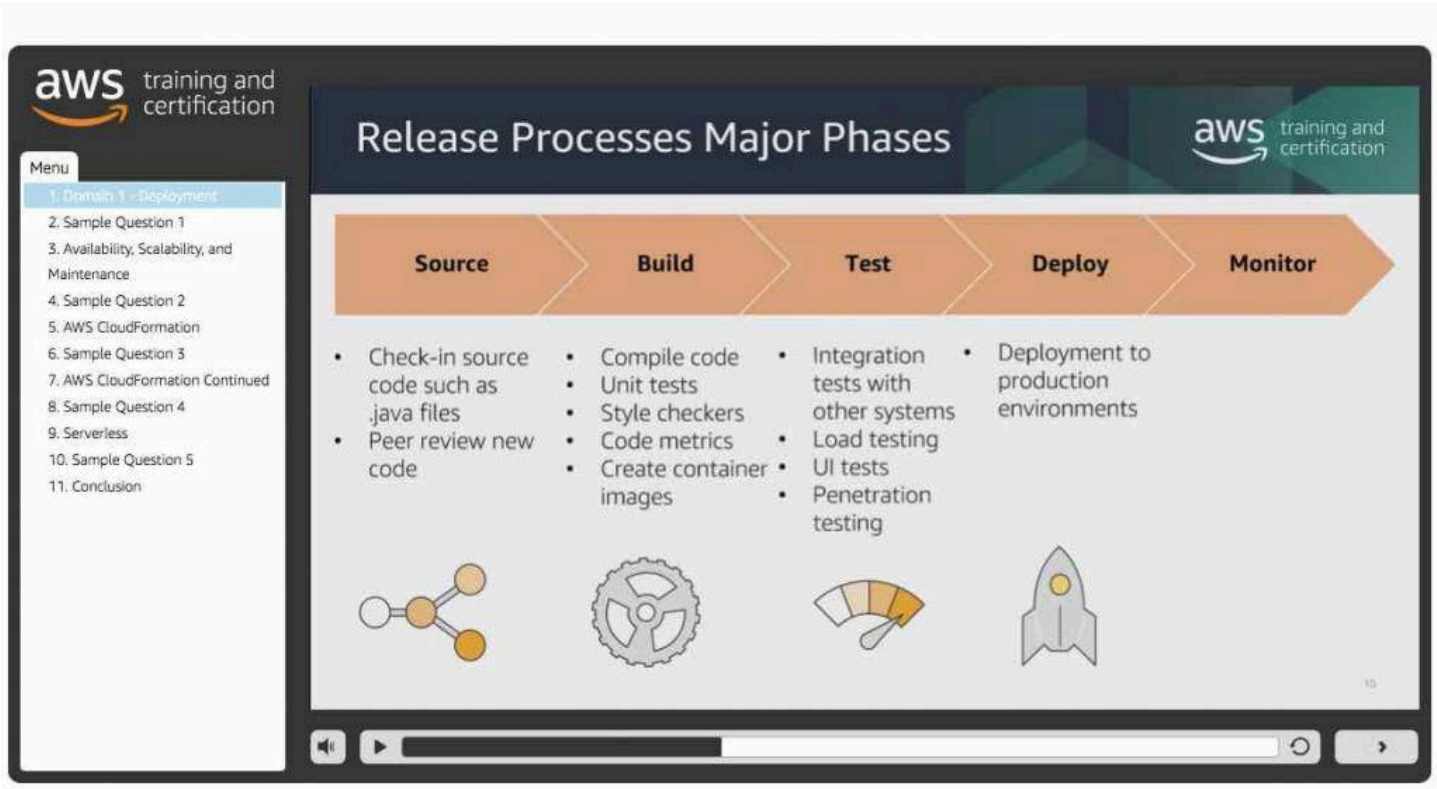
- Demonstrate an understanding of core AWS services, uses, and basic AWS architecture best practices.
- Demonstrate proficiency in developing, deploying, and debugging cloud-based applications using AWS.

Having prior experience in programming and scripting for both standard, containerized, and/or serverless applications will greatly make your review easier. Additionally, we recommend having an AWS account available for you to play around with to better visualize parts in your review that involve code. For more details regarding your exam, you can check out this AWS exam study path.

Study Materials

If you are not well-versed in the fundamentals of AWS, we suggest that you visit our AWS Certified Cloud Practitioner review guide to get started. AWS also offers a free virtual course called AWS Cloud Practitioner Essentials that you can take in their training portal. Knowing the basic concepts and services of AWS will make your review more coherent and understandable for you.

The primary study materials you'll be using for your review are the: FREE AWS Exam Readiness video course, official AWS sample questions, AWS whitepapers, FAQs, AWS cheat sheets, AWS practice exams and AWS video course with included hands-on labs.



For whitepapers, they include the following:

1. Implementing Microservices on AWS – This paper introduces the ways you can implement a microservice system on different AWS Compute platforms. You should study how these systems are built and the reasoning behind the chosen services for that system.

2. Running Containerized Microservices on AWS – This paper talks about the best practices in deploying a containerized microservice system in AWS. Focus on the example scenarios where the best practices are applied, how they are applied, and using which services to do so.

3. Optimizing Enterprise Economics with Serverless Architectures – Read upon the use cases of serverless in different platforms. Understand when it is best to use serverless vs. maintaining your own servers. Also, familiarize yourself with the AWS services that are under the serverless toolkit.

4. Serverless Architectures with AWS Lambda – Learn about Serverless and Lambda as much as you can. Concepts, configurations, code, and architectures are all important and are most likely

to come up in the exam. Creating a Lambda function of your own will help you remember features faster.

5. Practicing Continuous Integration and Continuous Delivery on AWS Accelerating Software Delivery with DevOps – If you are a developer aiming for the DevOps track, then this whitepaper is packed with practices for you to learn. CI/CD involves many stages that allow you to deploy your applications faster. Therefore, you should study the different deployment methods and understand how each of them works. Also, familiarize yourself with the implementation of CI/CD in AWS. We recommend performing a lab of this in your AWS account.

6. Blue/Green Deployments on AWS – Blue/Green Deployments is a popular deployment method that you should learn as an AWS Developer. Study how blue/green deployments are implemented and know the set of AWS services used in application deployment. It is also crucial that you understand the scenarios where blue/green deployments are beneficial and where they are not. Do NOT mix up your blue environment from your green environment.

7. AWS Security Best Practices – Understand the security best practices and their purpose in your environment. Some services offer more than one form of security feature, such as multiple key management schemes for encryption. It is important that you can determine which form is most suitable for the given scenarios in your exam.

8. AWS Well-Architected Framework – This whitepaper is one of the most important papers that you should study for the exam. It discusses the different pillars that make up a well-architected cloud environment. Expect the scenarios in your exam to be heavily based upon these pillars. Each pillar will have a corresponding whitepaper of its own that discusses the respective pillar in more detail.

AWS Services to Focus On

AWS offers extensive documentation and well-written FAQs for all of its services. These two will be your primary source of information when studying AWS. You need to be well-versed in many AWS products and services since you will almost always be using them in your work. I recommend checking out Tutorials Dojo's AWS Cheat Sheets, which provide a summarized but highly informative set of notes and tips for your review on these services.

Services to study for:

1. Amazon EC2 / ELB / Auto Scaling – Be comfortable with integrating EC2 to ELBs and Auto Scaling. Study the commonly used AWS CLI commands, APIs and SDK code under these

services. Focus as well on security, maintaining high availability, and enabling network connectivity from your ELB to your EC2 instances.AWS Elastic Beanstalk – Know when Elastic Beanstalk is more appropriate to use than other computing or infrastructure-as-a-code solutions like CloudFormation. Experiment with the service yourself in your AWS account, and understand how you can deploy and maintain your own application in Beanstalk.

2. Amazon ECS – Study how you can manage your own cluster using ECS. Also, figure out how ECS can be integrated into a CI/CD pipeline. Be sure to read the FAQs thoroughly since the exam includes multiple questions about containers.

3. AWS Lambda – The best way to learn Lambda is to create a function yourself. Also, remember that Lambda allows custom runtimes that a customer can provide himself. Figure out what services can be integrated with Lambda and how Lambda functions can capture and manipulate incoming events. Lastly, study the Serverless Application Model (SAM).

4. Amazon RDS / Amazon Aurora – Understand how RDS integrates with your application through EC2, ECS, Elastic Beanstalk, and more. Compare RDS to DynamoDB and Elasticache and determine when RDS is best used. Also, know when it is better to use Amazon Aurora than Amazon RDS and when RDS is more useful than hosting your own database inside an EC2 instance.

5. Amazon DynamoDB – You should have a complete understanding of the DynamoDB service as this is very crucial in your exam. Read the DynamoDB documentation since it is more detailed and informative than the FAQ. As a developer, you should also know how to provision your own DynamoDB table, and you should be capable of tweaking its settings to meet application requirements.

6. Amazon Elasticache – Elasticache is a caching service you'll often encounter in the exam. Compare and contrast Redis from Memcached. Determine when Elasticache is more suitable than DynamoDB or RDS.

7. Amazon S3 – S3 is usually your go-to storage for objects. Study how you can secure your objects through KMS encryption, ACLs, and bucket policies. Know how S3 stores your objects to keep them highly durable and available. Also, learn about lifecycle policies. Compare S3 to EBS and EFS to know when S3 is more preferred than the other two.

8. Amazon EFS – EFS is used to set up file systems for multiple EC2 instances. Compare and contrast S3 to EFS and EBS. Study file encryption and EFS performance optimization as well.

9. Amazon Kinesis– There are usually tricky questions on Kinesis, so you should read its documentation too. Focus on Kinesis Data Streams and explore the other Kinesis services.

Familiarize yourself with Kinesis APIs, Kinesis Sharding, and integration with storage services such as S3 or compute services like AWS Lambda.

10. Amazon API Gateway – API gateway is usually used together with AWS Lambda as part of the serverless application model. Understand API Gateway's structure, such as resources, stages, and methods. Learn how you can combine API Gateway with other AWS services, such as Lambda or CloudFront. Determine how you can secure your APIs so that only a select number of people can execute them.

11. Amazon Cognito – Cognito is used for mobile and web authentication. You usually encounter Cognito questions in the exam along with Lambda, API Gateway, and DynamoDB. This usually involves some mobile application requiring an easy sign-up/sign-in feature from AWS. It is highly suggested that you try using Cognito to better understand its features.

12. Amazon SQS – Study the purpose of different SQS queues, timeouts, and how your messages are handled inside queues. Messages in an SQS queue are not deleted when polled, so be sure to read on that as well. There are different polling mechanisms in SQS, so you should compare and contrast each.

13. Amazon CloudWatch – CloudWatch is a primary monitoring tool for all your AWS services. Verify that you know what metrics can be found under CloudWatch monitoring and what metrics require a CloudWatch agent installed. Also, study CloudWatch Logs, CloudWatch Alarms, and Billing monitoring. Differentiate the kinds of logs stored in CloudWatch vs. logs stored in CloudTrail.

14. AWS IAM – IAM is the security center of your cloud. Therefore, you should familiarize yourself with the different IAM features. Study how IAM policies are written and what each section in the policy means. Understand the usage of IAM user roles and service roles. You should have read up on the best practices whitepaper in securing your AWS account through IAM.

15. AWS KMS – KMS contains keys that you use to encrypt EBS, S3, and other services. Know what these services are. Learn the different types of KMS keys and in which situations each type of key is used.

16. AWS CodeBuild / AWS CodeDeploy / AWS CodePipeline – These are your tools for implementing CI/CD in AWS. Study how you can build applications in CodeBuild (`buildspec`), and how you'll prepare configuration files (`appspec`) for CodeDeploy. I suggest that you build a simple pipeline of your own in CodePipeline to see how you should manage your code deployments. It is also important to learn how you can roll back to your previous application version after a failed deployment. The whitepapers above should have explained in-place deployments and blue/green deployments and how to perform automation.

17. AWS CloudFormation – Study the structure of CloudFormation scripts and how you can use them to build your infrastructure. Be comfortable with both JSON and YAML formats. Read a bit about StackSets. List down the services that use CloudFormation in the backend for provisioning AWS resources, such as AWS SAM, and processes, such as in CI/CD.

Aside from the concepts and services, you should study the AWS CLI, the different commonly used APIs (for services such as EC2, EBS, or Lambda), and the AWS SDKs. Read up on the AWS Serverless Application Model (AWS SAM) and AWS Application Migration Service, as well as these that may come up in the exam. It will also be very helpful to have experience interacting with AWS APIs and SDKs and troubleshooting any errors that you encounter while using them.

Common Exam Scenarios

Scenario	Solution
AWS Lambda	
An application running in a local server is converted to a Lambda function. When the function was tested, an *Unable to import module* error showed.	Install the missing modules in your application's folder and package them into a ZIP file before uploading to AWS Lambda.
A Developer is writing a Lambda function that will be used to send a request to an API in different environments (Prod, Dev, Test). The function needs to automatically invoke the correct API call based on the environment.	Use Environment Variables
A Lambda function needs temporary storage to store files while executing.	Store the files in the `/tmp` directory
Lambda function is writing data into an RDS database. The function needs to reuse the database connection to reduce execution time.	Use execution context by placing the database connection logic outside of the event handler.
A Developer needs to increase the CPU available to a Lambda function to process data more efficiently.	Increase the allocated memory of the function.
Amazon API Gateway	
A Developer has an application that uses a RESTful API hosted in API Gateway. The API requests are failing with a `"No 'Access-Control-Allow-Origin'`	Enable CORS in the API Gateway Console.

`header is present on the requested resource`" error message.	
A website integrated with API Gateway requires user requests to reach the backend server **without intervention** from the API Gateway. Which integration type should be used?	`HTTP_PROXY`
A serverless application is composed of AWS Lambda, DynamoDB, and API Gateway. Users are complaining about getting HTTP 504 errors.	The API requests are reaching the maximum integration timeout for API Gateway (29 seconds).
How to invalidate API Gateway cache?	1. Send a request with a `Cache-Control: max-age` header. 2. Enable the `Require Authorization` option on your API cache settings.
A developer needs to deploy different API versions in API Gateway	Use stage variables
Amazon DynamoDB	
A Developer needs a cost-effective solution to delete session data in a DynamoDB table.	Expire session data with DynamoDB TTL
New changes to a DynamoDB table should be recorded in another DynamoDB table.	Use DynamoDB Streams
Reduce the DynamoDB database response time.	Use DynamoDB Accelerator (DAX)
Choosing the best partition key for the DynamDB table.	Use the partition key with the highest cardinality (e.g. student ID, employee ID)
An application uses a DynamoDB database with Global Secondary Index. DynamoDB requests are returning a `ProvisionedThroughputExceededException` error. Why is this happening?	The write capacity of the GSI is less than the base table.
CloudFormation and AWS SAM	
What section must be added to a CloudFormation template to include resources defined by AWS SAM?	Transform
A developer needs a reliable framework for building serverless applications in AWS	AWS SAM

A CloudFormation stack creation process failed unexpectedly.	CloudFormation will roll back by deleting resources that it has already created.
A CloudFormation template will be used across multiple AWS accounts	Use CloudFormation StackSets
Deployment and Security	
It is required that incoming traffic is shifted in two increments. 10% of the traffic must be shifted in the first increment, and the remaining 90% should be deployed after some minutes.	Canary
You need to authenticate users of a website using social media identity profiles.	Amazon Cognito Identity Pools
A company has two accounts. The developers from Account A need to access resources on Account B.	Use cross-account access role
Multiple developers need to make incremental code updates to a single project and then deploy the new changes.	Use GitHub/GitLab/Bitbucket as the code repository and directly deploy the new package using AWS CodeDeploy.
A development team is using CodePipeline to automate their deployment process. The code changes must be reviewed by a person before releasing to production	Add a manual approval action stage
Relevant API/CLI commands	
A Developer needs to decode an encoded authorization failure message.	Use the `aws sts decode-authorization-message` command.
How can a Developer verify permission to call a CLI command without actually making a request?	Use the `--dry-run` parameter along with the CLI command.
A Developer needs to deploy a CloudFormation template from a local computer.	Use the `aws cloudformation package` and `aws cloudformation deploy` command
A Developer has to ensure that no applications can fetch a message from an SQS queue that's being processed or has already been processed.	Increase the `VisibilityTimeout` value using the `ChangeMessageVisibility` API and delete the message using the `DeleteMessage` API.
A Developer has created an IAM Role for an application that uploads files to an S3 bucket. Which API call should the Developer use to allow the application to make upload requests?	Use the `AssumeRole` API

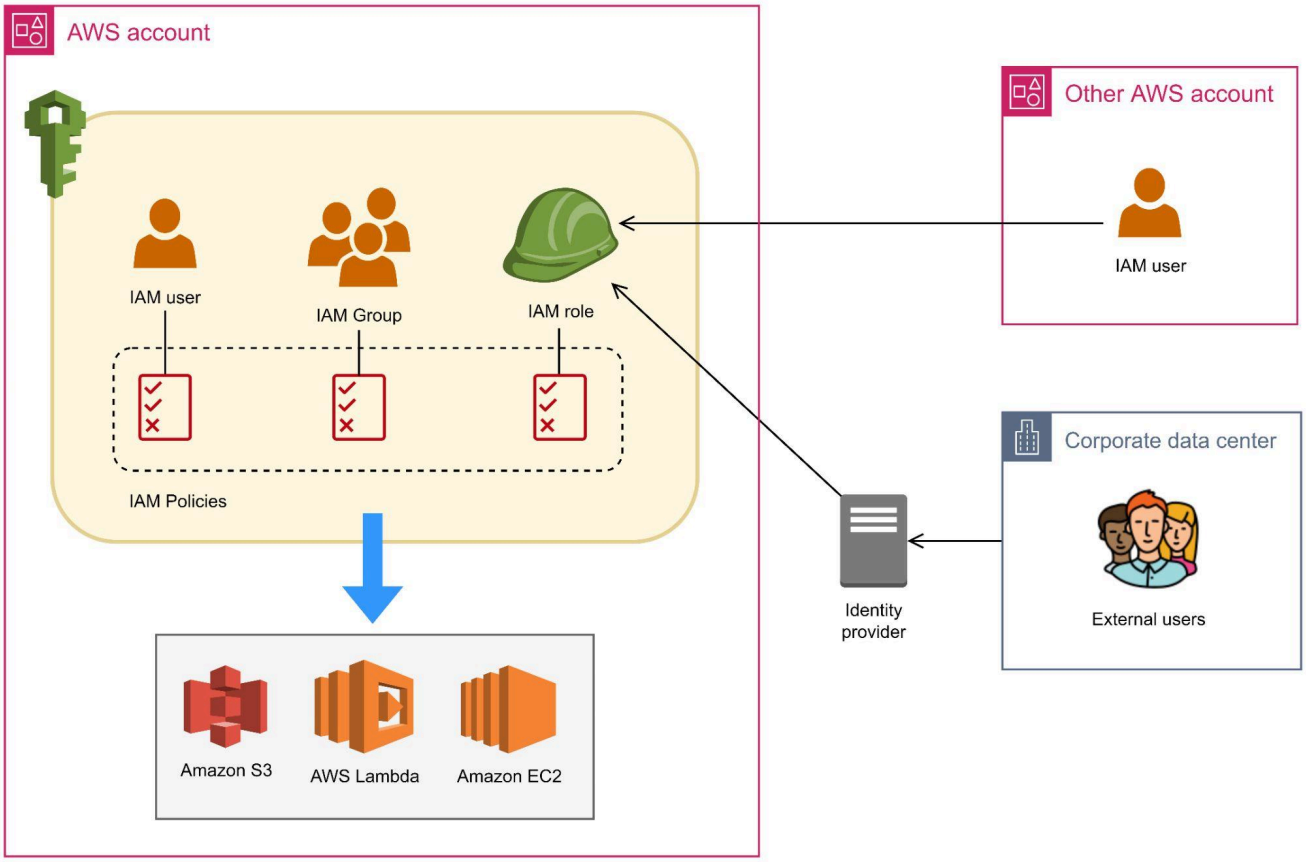
AWS Deep Dives

AWS Identity Access Management (IAM)

IAM is the primary tool for controlling and managing access to an AWS account. It sits at the core of AWS security; everything you do with AWS, whether it's creating a Lambda function, uploading a file to an S3 bucket, or something mundane as viewing EC2 instances on the Console, is governed by IAM. It allows you to specify who, which AWS resources, as well as what actions they can and cannot do. These are also known as *authentication* and *authorization*.

IAM Identities

IAM identities handle the authentication aspect of your AWS account. It pertains to any user, application, or group that belongs to your organization.

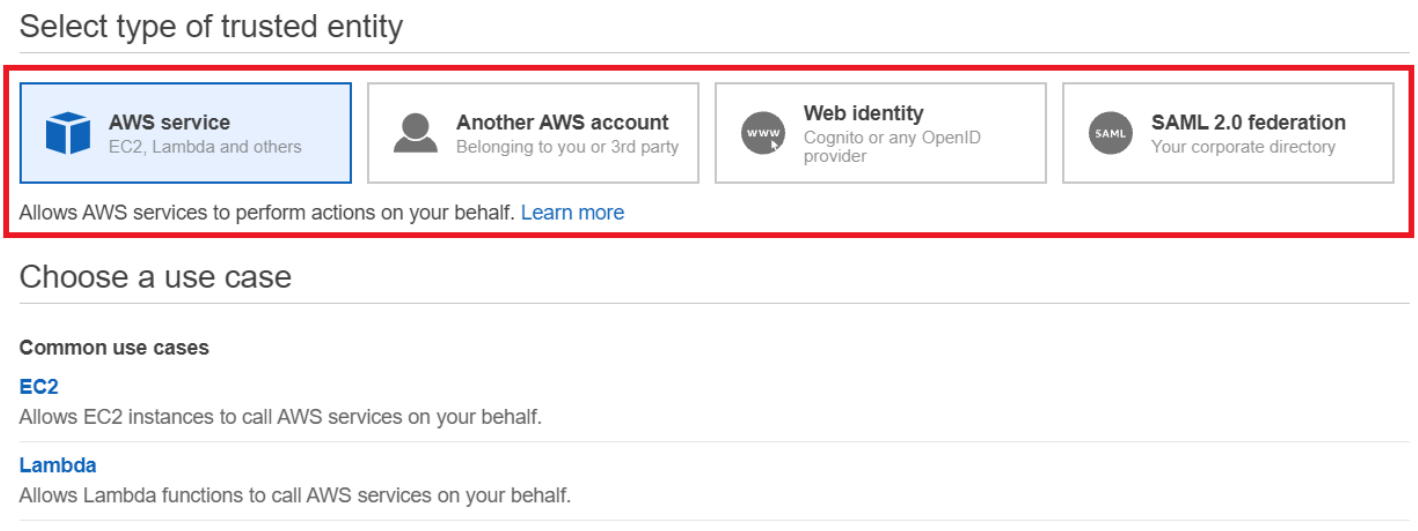


An IAM identity can be an **IAM User**, **IAM role**, or **IAM Group**.

An **IAM User** represents the user or application who interacts with AWS services. Take note that an IAM user is different from the root user. Although full admin permissions can be given to an IAM user, there are certain actions that only a root user can carry out, such as deleting an AWS account. IAM users are given long-term credentials for accessing AWS services. These credentials can be in the form of a username and password (for console access) or an access key and secret key (for programmatic access). The former is primarily used when logging into the AWS Console, whereas the latter is used when interacting with AWS Services via CLI/API commands.

An **IAM role** isn't intended to be associated with a unique user, rather, it is meant to be assumed by certain AWS Services or a group of external users with common business roles. Unlike IAM users, roles use time-limited security credentials for accessing AWS. When creating a role, you choose a trusted entity. The trusted entity determines who is authorized to assume the IAM role.

An IAM role can be assumed by AWS Services to perform actions on your behalf, an IAM user within your account or from an external one, and users federated by identity providers that AWS trusts. Examples of these identity providers are Microsoft Active Directory, Facebook, Google, Amazon, or any IdP that is compatible with OpenID Connect (OIDC) and SAML 2.0.



IAM Group is simply a collection of IAM users. The policies attached to an IAM Group are inherited by the IAM users under it. It's possible to assign an IAM user to multiple groups, but groups cannot be nested (a group within a group). IAM group offers an easy way of managing users in your account. For example, if you have a team of developers that needs access to AWS Lambda, instead of attaching the necessary permission for them individually, you can put the developers together in an IAM Group called 'Developers' and associate the

Lambda permissions to that group. If a new member joins the team, simply add him/her to the 'Developers' IAM group.

AWS IAM Identity Center

AWS IAM Identity Center (formerly known as AWS Single Sign-On) is the recommended service for managing workforce access to multiple AWS accounts and business applications in one place. While IAM is designed to manage identities and permissions within a single AWS account, IAM Identity Center sits on top of IAM to provide centralized identity and access management across an entire AWS organization. Instead of creating IAM users in every AWS account, you create user identities once in Identity Center and grant them federated access to whichever accounts and applications they need.

IAM Identity Center is tightly integrated with AWS Organizations. Once enabled, you can assign users and groups to AWS accounts within your organization through permission sets. A permission set is a collection of IAM policies that defines what a user is allowed to do once they sign in to an assigned account. Behind the scenes, IAM Identity Center provisions IAM roles in the target accounts based on the permission sets, which means users automatically receive short-lived, temporary credentials when they assume access — no long-term access keys to manage.

Users can be sourced from three identity providers:

- Identity Center directory - the built-in identity store provided by IAM Identity Center, ideal for organizations that don't already have an external identity provider.
- Active Directory - either AWS Managed Microsoft AD or an on-premises Active Directory connected via AWS Directory Service.
- External identity provider - any SAML 2.0-compliant identity provider such as Okta, Microsoft Entra ID (formerly Azure AD), Ping Identity, or Google Workspace. SCIM 2.0 is supported for automatic user and group provisioning.

Once users sign in to the IAM Identity Center access portal, they see a list of all AWS accounts and business applications they've been granted access to. From there, they can launch the AWS Management Console, retrieve temporary credentials for the AWS CLI, or open SAML-integrated applications — all using a single set of credentials.

References:

https://docs.aws.amazon.com/singlesignon/latest/userguide/what-is.html
https://aws.amazon.com/iam/identity-center/

IAM Policy

An IAM identity cannot perform AWS actions without an IAM Policy attached to it unless the resource being accessed allows the IAM Identity to do so. An IAM Policy is what authorizes an IAM user or role to control and access your AWS resources. There are three types of IAM Policies to choose from:

- AWS Managed Policies - these are built-in policies that have been constructed to conform to common use cases and job roles. For example, let's say you're working as a system administrator, and you have to give your new database administrator the necessary permissions to do his/her job. Rather than figuring out the right permissions, you may simply use the `DatabaseAdministrator` managed policy, which grants complete access to AWS services necessary to set up and configure AWS database services. AWS Managed Policies cannot be deleted or modified.

- Customer-managed Policy - this refers to the policies that you manually create in your account.

- Inline Policy - a policy that is embedded in an IAM Identity. Unlike AWS Managed Policies and Customer-managed Policies, an inline policy does not have its own ARN; thus, it can't be referenced by other IAM identities. It is scoped to a specific IAM user or role.

References:
https://docs.aws.amazon.com/IAM/latest/UserGuide/id_users.html
https://docs.aws.amazon.com/IAM/latest/UserGuide/access_policies.html
https://docs.aws.amazon.com/IAM/latest/UserGuide/id_groups.html

IAM Policy Structure

An IAM Policy is expressed as a JSON document. It is made up of two components: policy-wide information and one or more statements. The policy-wide information is an optional element that is placed on top of the document. It's usually just a Version element pertaining to the AWS policy language version being used. This element usually has a static value of `2012-10-17`, referring to the release date of the current AWS policy access language.

The Statement block is where you add the permissions you need for accessing various AWS services. A policy can have single or multiple statements where each statement is enclosed within a bracket.

The following is an example of an IAM Policy.