Amazon Q Business

Introduction to Amazon Q Business

What does Amazon Q Business do?

Amazon Q Business is a generative AI-powered assistant that can answer questions, generate content, create summaries, and complete tasks, all based on the information in your enterprise. Amazon Q Business is delivered using a built-in web experience or through APIs. This helps business users leverage the power of generative AI without any overhead.

Amazon Q Business can connect to your company data, information, and systems with more than 40 built-in connectors. It has built-in plug-ins for systems such as Salesforce, Jira, ServiceNow, and Zendesk to help complete tasks like creation of tickets, directly within your enterprise systems.

What problems does Amazon Q Business solve?

Amazon Q Business helps solve problems around building and using generative AI-powered digital assistants.

1. **User experience:** provides a built-in web experience that can be deployed for users to interact with the application. Additionally, can be embedded into existing enterprise applications such as Slack and Microsoft Teams to have a integrated user experience and conversation.
2. **Time to value:** allow quickly create a generative AI-powered digital assistant without any coding. It provides a user-friendly console, where an administrator can create an application with simple configurations. Has built-in web experience, generative AI capability, data integrations to enterprise data sources, plug-ins for enterprise applications, and APIs.
3. **Infrastructure overhead:** is a fully managed service that removes all infrastructure overhead from application creation, deployment, or management.
4. **User access controls:** retrieves and uses the existing access controls for users within integrated enterprise applications and data sources. This allows the users to view the data with their existing authorization.
5. **Data source integrations:** provides 40+ built-in integrations to popular enterprise data sources like Amazon S3, Salesforce, Oracle, and so on. It can connect to both cloud-based and on-premise data sources.
6. **Guardrails:** provides straightforward configurations for administrative controls and guardrails. For example, you can apply restrictions such as blocking specific words or topics.

What are the benefits of Amazon Q Business?

- Delivers quick, accurate, and relevant answers to your business questions: quickly connects to your enterprise systems so you can have tailored conversations, solve problems, generate content, and take actions relevant to your business. It generates answers and insights according to the material and knowledge that you provide, backed by references and citations to source documents.
- Connects to over 40 popular enterprise applications and document repositories: has over 40 built-in connectors to popular enterprise applications and document repositories, including Amazon Simple Storage Service (Amazon S3), Salesforce, Google Drive, Microsoft

365, ServiceNow, Gmail, Slack, Atlassian, and Zendesk. This helps with faster integrations to your enterprise systems, providing a tailored response to user queries. The connectors include both cloud-based systems and on-premise systems.

- Respects existing access controls based on user permissions: is built to be secure and private. It can understand and respect your existing identities, roles, and permissions within enterprise data sources. If a user doesn't have permission to access certain data without Amazon Q Business, they can't access it using Amazon Q Business either. This reduces security overhead for administrators while providing relevant responses to individual user queries.
- Helps administrators easily apply guardrails to customize and control responses: provides administrative controls, such as the ability to block entire topics and filter both questions and finalized answers using keywords. This helps ensure that it responds in a way that is consistent with a company's guidelines. You can also choose to limit the response to the knowledge available in the connected data sources or allow Amazon Q Business to use its world knowledge to deliver a response.

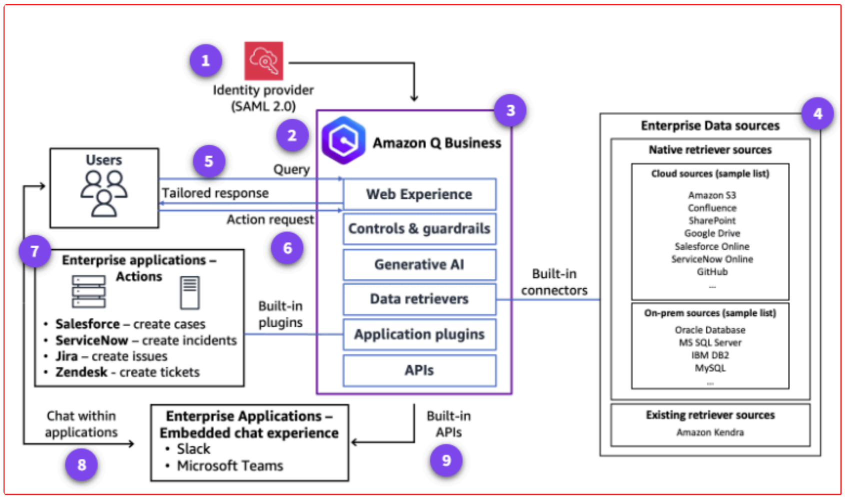
Quiz:

1. Which of the following problems does Amazon Q Business solve?
   Amazon Q Business addresses the need for faster integrations with enterprise systems and data repositories, alignment with existing user access controls for enterprise data, and reduces the time spent on coding to create a generative AI-powered digital assistant.
2. Which of the following options describes capabilities of Amazon Q Business?
   The key concepts for Amazon Q Business include built-in data integrations to enterprise data sources, plugins for enterprise applications, and fully managed RAG capability.

Architecture and Use Cases

How does Amazon Q Business generative AI-powered assistant work?

The following architecture logically illustrates how you can utilize Amazon Q Business to get query responses, take actions, or have conversations within existing applications.



> 1. **Users:** users are authenticated and authorized using a SAML 2.0 supported identity provider.
> 2. **User query:** users can provide a natural language query or prompt to the Amazon Q business web experience or chat application.
> 3. **Amazon Q Business:** has a built-in web experience for conversation. It has administered controls and guardrails. It uses generative AI capabilities, built-in data retrievers, application plug-ins, and APIs to deliver a tailored response.
> 4. **Enterprise data sources:** has 40+ built-in connectors for data retrieval from enterprise data sources, which includes a list of native data retrievers and existing data retrievers.
> 5. **Tailored response:** a tailored response is provided back to the user or consumer. This response includes specific data from the enterprise data sources based on users' existing access controls within those enterprise data sources.
> 6. **User action request:** users can directly create enterprise application actions like creating tickets, cases, incidents, and issues from the Amazon Q Business web experience interface without separately logging into enterprise applications.
> 7. **Enterprise applications for actions:** actions are created in enterprise applications like Salesforce, ServiceNow, Jira, and Zendesk. These are integrated with Amazon Q Business using built-in plugins.
> 8. **Chat within applications:** users can chat with Amazon Q Business application directly within existing enterprise applications without having to use a separate user interface.
> 9. **Embedded chat experience:** Amazon Q Business application can be directly embedded within enterprise applications like Slack and Microsoft Teams using built-in APIs.

What are the basic technical concepts of Amazon Q Business?

- IAM Identity Center: To create and use an application in Amazon Q Business, you need to configure and connect IAM Identity Center. IAM Identity Center is not meant to replace your existing identity provider, rather it serves as an overlay to your identity provider of choice. In case already have an IAM Identity Center configured, you may use that or create a new one for your account to connect to the application. Using IAM Identity Center, you can manage access to the application by assigning existing users or creating new users and groups from your identity center directory. Once the IAM Identity Center is connected to the application, assigned users can use the Amazon Q Business built-in web experience.
- Retrieval Augmented Generation (RAG): Generative AI foundation models are usually trained offline, making the model agnostic to any data that is created after the model was trained. Additionally, foundation models are trained on general domain data, making them less effective for domain-specific response or tasks. RAG is a technique used to retrieve data from outside a foundation model and augment the prompts by adding the relevant retrieved data as context. This helps generative AI based digital assistants to provide tailored response, by augmenting enterprise data sources to a foundation model. Is provided a fully managed RAG approach where administrators and users won't have to manage the underlying data augmentations, prompt engineering, and vector embeddings. Administrators can configure Amazon Q Business to respond strictly from enterprise documents or allow it to use external knowledge to respond to queries when the answer is not available in enterprise documents.
- Enterprise data access control: supports access control for your data so that users have access to the right content based on their permissions. Can integrate the Amazon Q web experience with the external SAML 2.0-supported identity provider (such as Okta, Microsoft Entra ID, and Ping Identity) to manage user authentication and authorization.
- Data integration and updates: has multiple pre-built connectors that can connect to enterprise data sources so that you can implement your generative AI solution with minimal configuration. Connectors offer modes for full synchronization or incremental data synchronization.
- Plugins: provides built-in plugins to interact with popular third-party applications, such as Jira, ServiceNow, Salesforce, and Zendesk. Administrators can enable these plugins to extend the capabilities of their Amazon Q application.

What are typical use cases for Amazon Q Business?

Typical use cases	Definition	Examples
		You can ask Amazon Q Business to write a compelling blog post

Accelerated Content Creation	Using Amazon Q Business, you can accelerate content creation for various business functions like marketing, sales, research, HR, legal, and so on.	and three social media headlines announcing the product described in this uploaded document. Amazon Q Business will generate content based on your enterprise's knowledge and the uploaded file.
Streamlined Enterprise Search Experience	Using Amazon Q Business, you can integrate your enterprise systems and knowledge repositories using pre-built connectors. This provides a streamlined enterprise search experience where users can get relevant response with specific source references.	You can ask Amazon Q Business, "where can I find the latest brand guidelines for logo usage?" Amazon Q Business will find what you need without having to jump between multiple systems. Because Amazon Q Business understands follow-up questions, you can continue asking questions, such as, "where can I find different color sets of our logo?" Amazon Q Business will understand the context of the previous question, surfacing the location of the relevant information.
Generates Summaries	Using Amazon Q Business, you can quickly generate a summary of enterprise documents within repositories, content within systems, and uploaded files. This improves productivity and information consumption.	You can ask Amazon Q Business to create a nicely formatted summary of the customer feedback on the new pricing structure in Slack. Amazon Q Business will generate a summary to help you understand the essence of the feedback, scaling knowledge, and speeding comprehension.
Extracts Key Insights for Business Decisions	Using Amazon Q Business, you can generate and extract insights from multiple documents through natural language queries. You can compare documents, ask insight questions relevant to the documents, and more.	You can ask Amazon Q Business to analyze the two latest customer satisfaction reports from Q1 and Q2 and identify the main differences between them. Amazon Q Business will sift through the reports and provide insights so that you can make faster and more informed decisions.

What else should I keep in mind about Amazon Q Business?

- Use of customer data: customer data is not used to train, improve, or enhance the machine learning models used by Amazon Q business. Additionally, the following data uses AWS Key Management Service (AWS KMS) encryption for protection with service-managed AWS KMS keys.:
  - Indexed customer data

    - Conversation history stored by the service
    - Feedback data (thumbs up or thumbs down)
- Chat history: automatically saves your conversation history for one month. Administrators have the ability to delete conversation history from the application.
- Supported file types: supports many common document types and formats, such as .PDF, .CSV, .DOCX, .HTML, .JSON, and .PPT.
- Choice of large language model (LLM): doesn't provide the ability to choose the underlying LLM. However, AWS uses various foundation models from Amazon Bedrock within Amazon Q Business.
- Custom connectors: allows to use custom connectors. You can add custom connectors and then use the Amazon Q SDK to implement them.

How do I create a data source for Amazon Q Business?

1. To begin, log in to the AWS Management Console and select the appropriate AWS Region from the Region menu.
2. Next, create an S3 bucket to store the files you will use in this demonstration. In the search bar, enter S3 and select S3 from the results. Now, on the S3 landing page, choose **Create bucket**.
3. Then, in the General configuration panel of the Create bucket page, provide a bucket name that is globally unique and lowercase. Your bucket name will be different from the name shown in this demo.
4. Now, scroll to the end of the page. Leave the rest of the configuration options set to their default values. Choose **Create bucket**.
5. Then, a banner is displayed after the bucket is created. The bucket is also listed in the General purpose buckets panel. Select the bucket name link.
6. Now, you can upload the files needed for this demonstration. You will find a .zip file following this video that contains the files you will need. Extract the files to your system. Please note that you should not upload the file named Amazon-Q4_2023_Transcript.pdf in this step. It will be used in a later demonstration. To add the files to your bucket, choose **Upload** or drag the files to this page from your system.
7. Next, your files are listed on the **Upload** page.
8. Now, scroll to the end of the page. Choose **Upload**.
9. Finally, after the files have been successfully uploaded to your bucket, a success banner is displayed.

How Do I Set Up an AWS IAM Identity Center and Create Users?

In this lesson, you will learn how to enable and use the AWS IAM Identity Center to create and manage users for Amazon Q Business.

How do I manage users for Amazon Q Business?

1. You will create an IAM Identity Center instance to manage users for Amazon Q Business. To create an application in Amazon Q Business, you need to configure and use IAM Identity Center to manage users and access for the application user interface. (Note that IAM Identity Center is not meant to replace your existing identity provider. It serves as an overlay to your identity provider of choice). Also, IAM Identity Center and user setup can be done either on the IAM Identity Center service console or within the Amazon Q Business console. In this demo, you will use the IAM Identity Center console to show how to enable the instance and add users.
2. To begin, log in to the AWS Management Console and select the appropriate AWS Region from the Region menu.
3. In the search bar, enter IAM Identity Center and select **IAM Identity Center** from the Services list.
4. Now, in the IAM Identity Center landing page, you can enable IAM Identity Center. Choose **Enable**.
5. You have the option to enable IAM Identity Center either at an organization level or at the account level. For this demo, choose **Enable in only this AWS account**. Then, choose **Continue**. In an enterprise-level setup, the recommendation is to use IAM Identity Center at an organization level. Organization level setup may require necessary administrative privileges.
6. Next, choose **Enable**.
7. After you create the IAM Identity Center instance, it will show on the dashboard. Choose **Users** from the navigation menu on the left.
8. Next, choose **Add user**.

9. Specify and enter user details like **user name**, **email address**, **confirm email address**, **first name**, **last name**, and **display name**. Leave all other settings as the default and scroll down.
10. Then, choose **Next**.
11. Adding a group is optional. Choose **Next**.
12. Now, review the details and choose **Add user**.
13. A banner will display, showing that you have successfully added the user.
14. An automated email is sent to the new user's email address. The email will have a button to access the invitation link. Choose **Accept invitation** within the email.
15. A new web browser window is opened where the user will need to add a new password and confirm the password. After you enter a password, choose **Set new password**.
16. Then, enter the username and choose **Next**.
17. Next, enter the password and choose **Sign in**.
18. For this multi-factor authentication (MFA) step, please use a separate mobile device with an authenticator application like Google Authenticator mobile app. On this page, choose **Show QR code**. Scan the QR code using your mobile device within the authenticator app. Add the **authentication code** displayed on the authenticator app for the user, **demo_user**. Choose **Assign MFA**.
19. After the MFA is set up, it will show that the authenticator application is registered for the specified user. Choose **Done**.
20. Next, you can test the newly added user. Add the username and choose **Next**.
21. Then, add the password and choose **Sign in**.
22. Now, add the MFA code from the authenticator app for the user and choose **Sign in**.
23. The user is taken to the AWS access portal where any application assigned to the user will be shown. Currently, no application is assigned to the user, so the screen is blank.

How do I Create an Amazon Q Business Application?

In this demo, you will learn how to create and customize an Amazon Q Business application, integrate a data source, and assign users.

Setting Up an Amazon Q Business Application

1. To begin, verify that your Region is selected from the Region menu in the AWS Management Console. Then, enter **Amazon Q Business** in the search bar. Select **Amazon Q Business** from the results panel.
2. Now, from the Amazon Q Business landing page, choose **Get Started**.
3. In the Amazon Q Business console, you can create, deploy, and manage generative AI digital assistant applications. You can also create a quick application to experiment before you create or deploy an application. Choose **Create application**.
4. Step 1 is to create the application. In the **Application details** pane, enter an **Application name**. Leave other settings as default. The IAM Identity Center instance that was created earlier as a part of this course is automatically assigned to this application. Choose **Create**.
5. Step 2 is to select a data retriever. You can either use native data retrievers or use existing retrievers. Select **Use native retriever** and choose **Next**.
6. Step 3 is to connect data sources for the application. Amazon Q Business has over 40 built-in connectors to various enterprise systems, like Amazon S3, SharePoint, Google Drive, and more. It has connectors for both cloud based systems and on-premises systems. Choose the **plus sign** by **Amazon S3**.
7. In the connect data sources Amazon S3 page, add a **data source name**. In the IAM role dropdown list, select **Create a new service role**, which is the recommended option. Then, choose **Browse S3** to select the S3 bucket that you created in a previous demonstration. Scroll down.
8. Under **Sync run schedule**, select **Run on demand** from the **Frequency** menu. Then, choose **Add data source**. It can take a few seconds or minutes to add the data source. Then, scroll down. After you successfully add the data source, choose **Next**.
9. Step 4 is to add groups and users to the application. Choose **Add groups and users**. In the **add or assign users and groups** pop-up window, choose **Assign existing users and groups** and choose **Next.**

10. Then, choose **Get started**. In the search bar, search for the username that you created. In this demo, the username is **demo_user**. Choose the user from the dropdown list.
11. Then, choose **Assign**. In the **Add groups and users** page, verify the user is added by choosing the **Users** tab.
12. After the user assignment and the appropriate subscription level is verified, choose **Create application**.
13. Now, the Amazon Q Business application is created and the web experience is deployed. You can verify the web experience status has deployed successfully under the **Web experience status** column. You can also verify the Web Experience URL, which will be used by the assigned users to access the application interface. Choose the demo application name link. In the demo application page, choose the radio button by your data source. Then, choose **Sync now**.
14. After the sync process starts, you will see a notification on the top of the page indicating that the sync started successfully. The sync process can take from a few minutes to a few hours. Sync speeds are limited by factors such as remote repository throughput and throttling, network bandwidth, and the size of the documents in the source system. The sync process continues in the background and doesn't need to be actively monitored, and the user session doesn't need to be active.
15. After the sync process is complete, you will see the last sync status is completed and the last sync time listed. As an admin, you have options to set Admin controls and guardrails for the application. Choose **Admin controls and guardrails** from the menu on the left. In the this page, you can set global controls. You can restrict the application response to only the connected enterprise data sources. You can also augment the response with generative AI large language models (LLMs) within Amazon Q Business to provide a generic response.
16. Now, choose **Edit**. Select the check boxes next to the two options **Allow end users to send queries directly to the LLM** and **Allow Amazon Q to fall back to the LLM knowledge**. Then, choose **Save**.
17. Now, choose **Applications** from the menu on the left.
18. Then, choose the radio button by the demo application you created. Choose **Customize web experience**. As an admin, you can preview the web experience for the application and also customize the title, subtitle, and welcome message. Choose **Save**. As an admin, you won't be able to have conversations with the application on this page. To have a conversation, a user needs to access the web experience URL with their respective credentials.
19. Now, you can access the web experience URL either from this preview page or from the application list. On this page, choose **View web experience**. After you select the web experience URL, you will be redirected to a separate web experience page where the assigned user needs to be authenticated before using the application. You have now successfully created an Amazon Q Business application.

How Do I Chat with an Amazon Q Business Application?

In this lesson, you will learn how to use the Amazon Q Business built-in web experience on a web browser to have a conversation with the Amazon Q Business application.

Chatting with Amazon Q Business

1. After the admin user creates the application in Amazon Q Business, you can choose the web experience URL to have a conversation as a business user. The Web experience URL is shown in the Applications list. Use the built-in web experience URL in a web browser. The assigned business user needs to follow the authentication process. This user was set up in IAM Identity Center and assigned to the Amazon Q Business application. Enter the user name and choose **Next**.
2. Then, enter the password and choose **Sign in**.
3. Then, enter the multi-factor authentication code from the Authenticator app that you set up during the user creation process in IAM Identity Center. Then, choose **Sign in**.
4. This is the personalized web experience page of the Amazon Q Business application, which was created and configured in a previous demonstration. You can start having conversations using the **Enter a prompt** section. You can also see and manage all the chat history under **Conversations** on the upper left panel of the page. For this demonstration, you will use the publicly available Amazon 10K financial annual report documents in a S3 bucket as a data source for this application.
5. You can ask questions, such as "What was Amazon's revenue in 2021." Amazon Q Business will respond with the correct answer and also show the data source references within the answer and in the sources dropdown list. In this case, the response was completed

using one of the data source files you uploaded to your S3 bucket. You have the option of providing feedback to the application by choosing the thumbs up and thumbs down icons. You can also choose the copy icon to copy the response for further use.

6. When you ask a question, like "What was Amazon's revenue in 2015," the application doesn't find any answer within the data source files. Because you allowed the application to fall back to the generative AI large language models (LLMs) of Amazon Q Business, it was able to provide an answer from its own knowledge. Please keep in mind that the answers from the LLMs could be generic and sometimes inaccurate.
7. You can ask the application to write a summary blog of the CEO shareholder letter within the 2022 annual report. You can write a specific prompt to direct the application based on your specification. The application will provide a response along with source references.
8. As a user, you can upload files to the prompt section and have a specific conversation related to the file. You can use the file in the zipped folder provided in this course called **Amazon-Q4_2023_Transcript.pdf**. This is a publicly available earnings call transcript from Q4 2023. Choose the paper clip icon to upload the file. Then, navigate to the file within your computer's directory and choose the file.
9. After you upload the file to the conversation, you can verify that the file successfully uploaded within the prompt section. Now, you are ready to have a specific conversation around this file. In the prompt section, enter "Write a 200 word summary of the attached transcript file." Then, choose the arrow icon to submit the prompt.
10. Amazon Q Business will create a summary from the uploaded file.

How Do I Clean Up Resources?

In this lesson, you will learn how to delete the resources and application you created using AWS management console.

Cleaning Up Resources

1. To begin, return to the Amazon Q Business page for Applications. After the application is no longer needed, select the radio button next to your application. Then, from the Actions menu, choose **Delete**.
2. When you choose Delete, you will be asked for confirmation before the application is deleted permanently. Enter **Delete** in the text box, and then choose **Delete**.
3. Then, after you have deleted the application, you can delete the S3 bucket you created. Return to the S3 landing page. From the menu on the left, choose **Buckets**. Select the radio button next to the bucket you created for this demonstration. Before you can delete the bucket, it must be empty. Choose **Empty**.
4. On the **Empty bucket** page, enter **permanently delete** in the text box to permanently delete all objects in your bucket. Then, choose **Empty**. You will see a banner notification that you successfully emptied the bucket. Choose **Exit**.
5. Now, you will select the radio button next to your bucket. Choose **Delete**. On the **Delete bucket** page, enter the name of the bucket in the text box.
6. Then, choose **Delete bucket**. You will see a banner notification that you successfully deleted the bucket.

Amazon Q Developer

Introduction to Amazon Q Developer

What does Amazon Q Developer do?

Amazon Q Developer is a generative AI powered assistant. It helps you understand, build, extend, and operate Amazon Web Services (AWS) applications throughout the software development lifecycle. It uses natural language processing to answer questions on AWS architecture, best practices, documentation, and support, providing contextually relevant and actionable answers.

When used in IDE, Amazon Q developer provides software development assistance. It helps you with code explanation, generation, debugging, optimization, feature development, and code transformation.

Amazon Q Developer is accessible through multiple channels, such as the AWS Management Console, where it can analyze and troubleshoot issues across services. In addition, it is available from:

- The AWS website
- AWS documentation pages
- The AWS mobile app
- IDEs with the Amazon Q extension
- The AWS Chatbot for Microsoft Teams and Slack
- Amazon CodeCatalyst

What problems does Amazon Q Developer solve?

Reduces the time spent on manual tasks, such as coding, testing, upgrading, troubleshooting, and optimizing your code. It is available wherever you need it: in your IDE, AWS console, terminal, or Slack. Amazon Q Developer is your developer companion.

To recognize problems that Amazon Q Developer can solve, review the following information.

1. **Increase productivity:** streamlines repetitive tasks and accelerates the development workflow. It provides software development assistance, including code explanation, code generation, and code improvements such as debugging and optimization.
2. **Onboard developers new to AWS:** helps new developers understand AWS services and how to select the right services for their needs. It provides guidance on how to start their journey in AWS and recommendations based on the AWS Well-Architected Framework.
3. **Develop code features:** you can develop code features and projects in your programming language of choice. You explain the feature you want to develop. Then, Amazon Q Developer uses the context of your current project to generate a detailed implementation plan that includes the code for the changes you described.
4. **Transform code:** within IDEs, Amazon Q Developer can update the language version of your code files (currently, Amazon Q Code Transformation supports updating Java 8 and Java 11 code to Java 17).

5. **Troubleshoot issues:** you can often run into errors or issues when deploying and operating applications. Amazon Q Developer helps troubleshoot and resolve common deployment, configuration, and runtime issues through natural language conversations. It uses knowledge from AWS documentation.
6. **Autonomous agents:** Agents take a lot of work out of complex, multistep tasks. The agent for software development helps with implementing features, documenting code, and bootstrapping new projects, all from a single prompt.

What are the benefits of Amazon Q Developer?

Amazon Q Developer accelerates the software development lifecycle by boosting employee's productivity leading to lower cost to build an application.

More about the benefits of Amazon Q Developer:

- Accelerates the entire software development process: more than 70% of developer time is spent on undifferentiated activities slowing down creativity and innovation. These activities include writing boilerplate code, developing and running unit tests, and translating code from one language to another.
- Boosts employee productivity and accelerates new feature development: Amazon Q Developer expedites the development of new features. It can boost productivity by generating new software code suggestions for application development tasks.
- Saves time and reduces costs across organization: includes security scans that you can run in the IDE. These scans help you find and correct potential vulnerabilities earlier in the application lifecycle, thereby lowering the cost, time, and risk of application development.

Quiz:

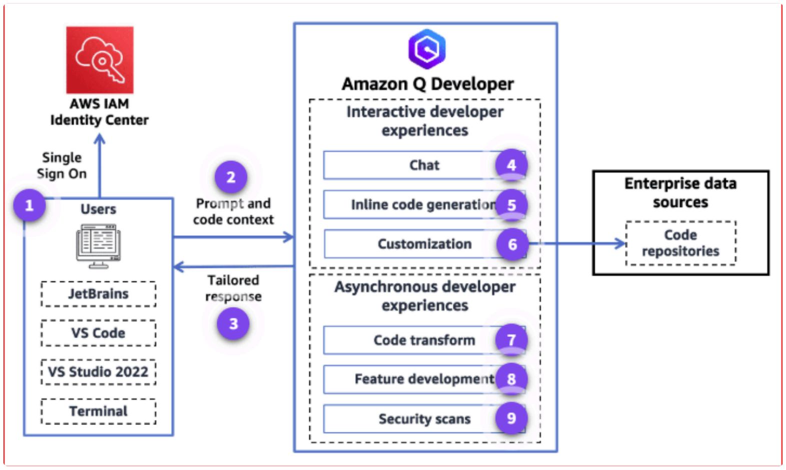
1. What are the benefits of using Amazon Q Developer?

   Some of the primary benefits of Amazon Q Developer are the ability to boost employee productivity, reduce costs, and accelerate the software development process.

Architecture and Use Cases

How is Amazon Q Developer used to architect a cloud solution?

Amazon Q Developer helps you explore new AWS capabilities, review your resources, analyze your bill, and architect solutions. It's an expert in AWS well-architected patterns, documentation, solutions implementations, and more. The following architecture logically illustrates how you can use Amazon Q Developer. (note that in the diagram, use of the AWS IAM Identity Center is only available when using the Amazon Q Developer Pro tier. All of the other features are available regardless of tier).



1. **Users:** AWS IAM Identity Center provides authentication and authorization for users.
2. **Prompts and code context:** users can provide a natural language query or prompt to the Amazon Q Developer chat application.

3. **Tailored response:** a tailored response is provided back to the user. This response includes specific data from the enterprise data sources based on the user's existing access controls within those enterprise data sources. In addition, AWS documentation is used to generate a response when you use the Diagnose with Amazon Q feature in the AWS Management Console.
4. **Chat:** provides the ability to ask questions about AWS, including questions about support, architecture, best practices, and documentation. You can also ask questions about software development and understanding and updating code.
5. **Inline code generation:** provides code suggestions based on your current and previous inputs. It generates line-by-line recommendations or an entire function block in one recommendation.
6. **Customization:** help you get more relevant code recommendations in the IDE and help with these recommendations by making the service aware of your selected internal libraries, APIs, best practices, and architectural patterns, significantly accelerating development.
7. **Code transform:** gives developers the ability to upgrade the program language version of the code files directly.
8. **Feature development:** with feature development, developers can define, collaborate, and solve software engineering tasks.
9. **Security scans:** identify security vulnerabilities and suggest how to improve your code.

What are the basic technical concepts of Amazon Q Developer?

Amazon Q Developer tailors its response based on the environment.

	Definition	Examples
Using in AWS Management Console	Amazon Q Developer helps you understand, prepare, and analyze data from data sources (Amazon Simple Storage Service (Amazon S3), Amazon DynamoDB, Amazon Relational Database Service (Amazon RDS), and Amazon Redshift.) by generating extract, transform, and load (ETL) scripts with AWS Glue. Amazon Q Developer can generate the sample ETL script to load the data from an existing S3 bucket to get started quickly with your thought.	You have signed in to the AWS Management Console and would like to know about the service quotas for your account. You can ask Amazon Q Developer the question in the console chat: How do I check my service quotas? Amazon Q Developer will respond to you with a step-by-step guide to view the console and also give you an option to validate through the AWS Command Line Interface (AWS CLI).
Access control	Amazon Q Developer supports granular access controls. Through IAM Identity Center, it also supports integration with identity providers such as Okta, Microsoft Entra ID, and Ping Identity.	This integration allows for seamless authentication and authorization within the development environment.
IDE plugin	The Amazon Q Developer IDE plugin significantly enhances your IDE by establishing built-in connections to essential tools and services.	This integration streamlines your workflow and ensures seamless interactions within the IDE, boosting efficiency and convenience throughout your development process.

Terminal plugin	Evolutionizes command line operations by integrating directly into your terminal. It supports IDE-style completions for hundreds of popular command line interfaces such as git, npm, docker, and AWS.	It gives you the ability to input natural language instructions such as Copy all files in my current directory to Amazon S3. These instructions translate into instantly executable shell code snippets.
Customization	Customization capability provides tailored suggestions as the service works with your private code repositories.	This customization allows Amazon Q Developer to offer accurate recommendations based on your organization's unique libraries and coding standards.
Chat	You can chat with Amazon Q in Slack or Microsoft Teams by asking natural language questions in English.	You can ask questions regarding the AWS resources in your account, choosing between AWS services, best practices, and other topics. Amazon Q will respond with step-by-step instructions or summaries of information found in AWS documentation. The answer includes links to the source of the information.

What are typical use cases for Amazon Q Developer?

Amazon Q Developer plays a crucial role in enhancing the efficiency and effectiveness of the software development lifecycle (SDLC). The five phases are plan, create, test and secure, operate, and maintain and modernize.

- **Plan:** navigating through vast amounts of technical documentation and examples can be daunting and time-consuming. Amazon Q Developer simplifies this task by providing targeted, business-specific guidance and code explanations through conversational coding, aiding in the planning phase of application development.
- **Create:** as an inline coding assistant in both IDEs and the AWS CLI, Amazon Q Developer accelerates the development of new features and the maintenance of existing infrastructure. Its conversational coding capabilities help developers swiftly implement and iterate on their ideas.
- **Test and secure:** ensuring that code changes are robust and secure is crucial. Amazon Q Developer enhances this phase by assisting with generating unit tests and identifying security vulnerabilities. It also offers remediation advice through its integrated security scanning features, helping developers enforce best security practices effortlessly.
- **Operate:** for AWS developers, Amazon Q Developer is indispensable, offering expert troubleshooting and optimization across various AWS services. These services include Amazon S3, AWS Lambda, Amazon Elastic Compute Cloud (Amazon EC2), and Amazon Elastic Kubernetes Service (Amazon EKS). Integrated directly within your IDE, terminal, or console, Amazon Q Developer helps ensure smooth operations and efficient problem resolution.
- **Maintain and modernize:** the Amazon Q Code Transformation feature of Amazon Q Developer streamlines the task of modernizing code and updating dependencies to newer programming languages. This feature reduces complexity and minimizes the risk of errors during the modernization process which helps you keep your applications up to date and efficient.

What else should I keep in mind about Amazon Q Developer?

Amazon Q Developer uses conversational coding where you guide the service to generate desired outputs. This section will go through the best practices of conversational coding:

Conversational coding

Writing clear and specific prompts is crucial when using Amazon Q Developer. The prompts should include as much detail as possible, be stated clearly, specify the intention of the prompt, and provide context. When you ask Amazon Q Developer to act on your code, it uses the current file open in your IDE, the programming language, and the file path for context. If Amazon Q Developer includes code in its response, you can copy or insert it directly into your file by choosing Insert code.

The following guidelines will help you formulate great prompts:

- Be specific and clear in prompts to get your desired responses. State the task directly and provide details.
    - Clarity involves accuracy and specificity, leaving little room for misinterpretation.
    - Intent refers to the purpose or goal behind the prompt.
    - Context includes the surrounding information or environment related to the prompt.
- Use relevant examples to provide additional context and guide better outputs.
- Use an iterative approach. Refine and rephrase prompts based on responses.
- Break down complex queries into smaller, manageable parts.

Quiz:

1. Which option is a possible use case for Amazon Q Developer?

    One possible use case for Amazon Q Developer is to generate unit tests and identify security vulnerabilities.Which of the following options describes capabilities of Amazon Q Business?

How Do I Set Up a Development Environment to Use with Amazon Q Developer?

In this lesson, you will learn how to install an configure the Amazon Q extension in Visual Studio Code and establish an AWS Builder ID.

How do I install the Amazon Q extension in Visual Studio Code?

Setting Up a Development Environment to Use with Amazon Q Developer

1. To follow the demonstrations in this course, you will use an integrated development environment, or IDE, named Visual Studio Code, or VS Code. (Visual Studio Code - Code Editing. Redefined). When you have VS Code running, choose the Extensions icon in the Activity bar of your VS Code IDE.
2. Now, enter Amazon Q in the search bar. From the results panel, choose Amazon Q to learn more about the extension from the displayed tab.
3. Next, choose Install from the search result or from the Amazon Q extension summary tab.
4. Then, the Amazon Q icon is displayed in the Activity bar after installation has completed. Choose the Amazon Q icon.
5. Next, the Amazon Q: Login panel is displayed. Choose Use For Free as your sign-in option. Choose Continue.
6. Now, a confirmation code is displayed in a pop-up dialog. Choose Proceed To Browser.
7. Then, in the next pop-up window verifying that you want to open an external site, choose Open. If you choose Configure Trusted Domains now or in the future, you can add Cloud Computing Services - Amazon Web Services (AWS) as a trusted domain so that you will not get this pop-up again.
8. Next, a browser tab will open that displays the Authorization requested window. The code that was displayed in VS Code should have already populated. Choose Confirm and continue.
9. Now, the Create AWS Builder ID panel is displayed in your browser. If you already have a Builder ID, choose the sign in option. Otherwise, enter your email address, and choose Next.
10. Next, the Your name field is added to the panel. Enter your name, and choose Next.
11. Then, AWS will send a confirmation code to the email address you submitted. In the Verification code text box, enter the code. Choose Verify.
12. Next, enter and confirm a password for your AWS Builder ID. Choose Create AWS Builder ID.

13. Now, you are asked to allow the AWS IDE Extensions for VSCode to access your data. Choose Allow access.
14. Next, a confirmation displays indicating that you approved the request. You can close your browser or the tab.
15. Now, return to VS Code. It is connected to Amazon Q Developer and configured to use your AWS Builder ID.

How Do I Interact with Amazon Q Developer in Visual Studio Code?

In this lesson, you will learn how to interact with Amazon Q Developer in Visual Studio Code and how to use the Amazon Q Developer chat interface.

Interacting with Amazon Q Developer in Visual Studio Code

1. To begin, verify that the Amazon Q Chat tab is open in Visual Studio Code, or VS Code. If it is not, from the Activity bar, choose the Amazon Q icon. If you would like to expand the Chat window, hover over the line separating the Chat panel from the editor window. Choose and drag the line to increase the width of the Chat panel.
2. Now, you will ask for information about the AWS Cloud Development Kit (CDK). In the Chat input, enter the following question: What is the CDK? Press **Enter** or choose the **paper airplane** icon to submit your question.
3. Then, Amazon Q generates an explanation of the CDK, including key features, and provides a link to the source used to create the response.
4. Any time you want to begin a new conversation, enter/clear in the prompt input box. Then, choose the **paper airplane** icon or press **Enter**.
5. In the next section of this demonstration, you will ask Amazon Q questions regarding the Python code that is provided in the course. To open the VS Code Explorer, from the Activity bar, choose the **Explorer** icon.
6. Next, you will open a folder containing the files you extracted from the provided .zip file. The .zip file is named *Ewallet-application-demo.zip*, and you can download it from this course. Choose **Open Folder**.
7. Now, navigate to the folder where you extracted the files. Choose **Open**.
8. Then, choose the file named *dynamodb_wallet_repository.py*. The file is displayed in the VS Code editor.
9. Now, scroll to the end of the file. The last method is named find, and it begins with the string def find. Select all of the text beginning with def to the end of the file. To open the context menu for the code, right-click the highlighted code. From the context menu, choose **Send to Amazon Q**, and then choose **Send to prompt**. The highlighted code is sent to the Chat panel where you can enter questions about the code.
10. Then, enter questions or make requests about the code you sent to the prompt. As an example, you can ask for an explanation of the code. A brief summary of the purpose of the code is provided, followed by a step-by-step breakdown of what it does.
11. Now, ask if there are any security flaws in the code. Amazon Q evaluates the code and responds with several potential issues and a mitigation strategy for each issue.
12. Next, ask Amazon Q to modify the code to address the first area of concern, which is a lack of input validation. It responds with a modification that validates the format of the ID entered.
13. Then, in the Chat panel after the modified code, Amazon Q provides a summary of the changes that were made.
14. Now, if you want to incorporate the modifications into your code, choose the find method and delete it.
15. Then, in the Chat panel, choose Insert at cursor to paste the code into your file.
16. Now, because the find method was evaluated without the context of the rest of the code, it is pasted without the necessary padding. To fix this issue, move the import re line to the beginning section of the file. To fix the pasted code, select the entire section of new code and press the **Tab** key.
17. Then, with the formatting fixed, you can save the file.
18. Now, for the next demonstration, you will ask Amazon Q to help optimize code in another file. Open the *withdraw.py* file, and highlight the *validate_payload* method.
19. Next, open the context menu for the highlighted code. Choose **Send to Amazon Q**, and then choose **Optimize**.
20. Then, Amazon Q provides a list of optimizations that can be applied to the code. The recommendations that Amazon Q provides are code examples that can be directly implemented. You might want to expand the Chat window if you changed it earlier.

21. Now, one of the optimization recommendations for this example use case is to use a dictionary instead of hardcoding the validation rules. This modification can improve the maintainability of the code as additional requirements around validations are being discussed. Using a dictionary provides a more straightforward method for adding rules in the future.
22. Next, if you decide to implement the recommendation provided by Amazon Q Developer to optimize the code by using dictionaries, Amazon Q can insert the code into the file. In the Chat panel after the modified code, choose **Copy**. Then, highlight the *validate_payload* method in the *withdraw.py* file, and paste the modified code in its place.
23. Finally, the line that reads *"import string"* must be moved to the beginning of this file. Delete **"import string"** from the file. Then, scroll to the beginning of the file, and paste it where the other import statements are located. Now, save your modified file.

How Do I Transform Code from Java 8 to Java 17 by Using Amazon Q Developer?

In this lesson, you will learn how to transform Java 8 code into Java 17 by using Amazon Q Developer.

The lesson demonstrates a smooth transition from Java 8 to Java 17 by using the Amazon Q Developer Agent for code transformation. As such, it represents a best-case scenario. It's essential to recognize that, in your own usage, partial successes might require troubleshooting and manual interventions on your part.

Prerequisites

To complete the next two demonstrations on your own system, it is necessary that you do the following:

1. Due to the processing and total data processed in this demonstration, an Amazon Q Developer Pro Tier subscription is required. Information regarding the Pro Tier is available on the Amazon Q Developer pricing page(opens in a new tab).
2. To transform an application from Java 8 to Java 17, you must have Java 8 and Apache Maven installed on your system. If you have a different version of Java installed, you will need to uninstall it and install Java 8.
3. For the second demonstration, you will need to replace Java 8 with Java 17 to complete the application build.

Transform Code From Java 8 to Java 17 by Using Amazon Q Developer

1. Following this lesson is a link to a .zip file containing a Maven project written in Java 8. Download the file, and extract the contents to a folder on your system. To open the folder in Visual Studio Code, or VS Code, choose **Open Folder** or **Open** (depending on your operating system) from the Welcome page. If you prefer, choose **Open Folder** from the File menu, or use the keyboard shortcut.
2. Now, expand each folder in the project until *DefaultGreeting.java* is displayed. Choose the file to open it.
3. Next, choose the Amazon Q Developer extension, and a Chat tab will open. Enter /t and then /transform is presented as an option. Choose /transform or finish entering the word, and then choose the paper airplane icon.
4. Then, a new tab is displayed requesting a confirmation of the project to transform, the current source code version, and the target code version. Choose **Confirm** to begin the Amazon Q Code Transformation.
5. Next, if Amazon Q Developer requires any additional information, such as the path to your Java Development Kit, or JDK, 8 installation, it will be requested in the chat. When all the information has been provided, the transformation of the code begins with an analysis of the entire project.
6. Now, a TRANSFORMATION HUB tab is displayed. If it does not display on your system, choose the ellipsis icon in the terminal panel and choose **Transformation Hub**. On this tab, you can monitor the progress of the transformation. In the background, Amazon Q Developer will analyze each file of the code, update the dependencies, and create the code in Java 17. Subsequently, it will initiate the building process in a virtual environment.
7. Then, as the transformation progresses, a transformation plan is displayed that details each of the planned steps for your review. When the the transformation is complete, choose **Download Proposed Changes** to copy the files to your system.
8. Next, a banner is displayed confirming that the files have been downloaded. In the PROPOSED CHANGES panel, the project's files are listed and a capital letter appears next to each file name. A capital A indicates that the file was added to the project, and a capital M indicates that the original file was modified.
9. Now, choose a file that has been modified, and its contents will be displayed. Two columns of line numbers are displayed. The first column is the line number in the original file. The second column is the line number in the modified file. This information can help you

precisely identify the proposed changes. Modified lines are prepended with a minus or plus sign. The minus sign indicates that a given line is the original code, and a plus sign indicates that the line has been modified. After you have reviewed each file, choose **Accept** to approve all of the proposed changes.

10. Then, a success message is displayed, and you are ready to install the updated code on your system.
11. Finally, return to the terminal and enter mvn clean install. After a minute or so, a BUILD SUCCESS message is displayed. The transformation of the project from Java 8 to Java 17 completed successfully. After verifying the changes, you'll be ready to commit them to your repository and proceed with deployment. This example illustrates how you can enhance productivity by efficiently completing major transformation tasks and focusing on business challenges.

How Do I Ask Amazon Q Developer to Implement a New Feature to a Project?

In this lesson, you will add a new feature to the project transformed in the previous demonstration by using an Amazon Q Developer.

Amazon Q Developer Agents take a lot of work out of complex, multistep tasks. The agent for software development helps with implementing features, documenting code, and bootstrapping new projects, all from a single prompt.

Asking Amazon Q Developer to Implement a New Feature to a Project–

1. This lesson builds on the project that was transformed from Java 8 to Java 17 in the previous demonstration.
2. To begin, if the project you transformed in the previous demonstration is no longer open, choose **Open** or **Open Folder**. Then, navigate to the *AMAZON-Q-DEMO* folder on your system.
3. Now, from the Activity bar, choose the **Amazon Q Developer** extension. A Chat tab will open. Enter /d and choose the /dev feature of Amazon Q Developer.
4. Then, enter the following request after /dev: Create a RESTful API endpoint to expose the functionality through a web service. To submit the request, choose the paper airplane icon.
5. Next, Amazon Q Developer will analyze the entire project, analyze each file, and create a plan for you to implement the requested task. It might take a few minutes depending on the size of the project. When the analysis is finished, a step-by-step plan is displayed in the Chat panel that details the changes to your code. Scroll through the steps to review them.
6. Then, when you reach the end of the plan, you have the option to Generate code. This is not the final step, and you will have an opportunity to review the changes before updating your code base. Choose **Generate code** to prepare the updates.
7. Next, Amazon Q Developer will begin generating the code and modifying the files with the proposed changes that will add the feature to your application.
8. Now, the full tree structure of your code base is displayed. Any files that will be modified or added are displayed at the end of the folder path in which they reside. Choose *pom.xml* to review the proposed modifications.
9. Then, the file is displayed in the code panel. As you observed in the transformation demonstration, two columns of line numbers show what was in the original file and the proposed change. In addition, a minus sign preceding the line indicates that it is the original line, and a plus sign indicates the proposed modification. The text in the tab indicates that two versions of *pom.xml* are being compared.
10. Next, choose **GreetingResource.java**, and it is displayed in the code panel. This file did not exist before the modification plan was created. All of the lines following the first one are shown as additions because the file is new. In contrast to the previous comparison, the tab for this one displays empty when being compared to *GreetingResourse.java*.
11. Now, scroll to the end of the Chat contents. Choose **Insert code**.
12. Then, a message is displayed to indicate that the code update has been completed. The implementation of the requested REST API feature is finished. Choose **Close session**.
13. Now, you will validate that the new changes did not negatively affect the Maven build. Choose the **TERMINAL** menu in VS Code, and choose **New Terminal**. Enter mvn clean install in the terminal to test the new changes. BUILD SUCCESS is displayed at the end of the clean install process. Your code is ready to commit and deploy to your environment for testing.

Amazon Bedrock

Índice

Description:

This course is designed for application developers interested in building generative artificial intelligence (generative AI) applications using either the Amazon Bedrock APIs or AWS-LangChain integration. In this course, you will explore the architecture patterns and implementations to support generative AI use cases such as generating and summarizing text, retrieval augmented generation (RAG), and question answering.

You learn to build RAG application using Amazon Bedrock Knowledge Bases, and AI Assistants that use knowledge bases and user-developed tools to answer questions using Amazon Bedrock Agents. You'll also learn to implement safeguards customized to your application requirements and responsible AI policies using Amazon Bedrock Guardrails.

Course objectives

In this course, you will learn to:

- Identify the components of a generative AI application and the options to customize a foundation model (FM)
- Describe Amazon Bedrock foundation models, inference parameters, and key Amazon Bedrock APIs
- Describe the architecture patterns that can be used to build generative AI applications
- Identify Amazon Web Services (AWS) offerings that help with monitoring, securing, and governing your Amazon Bedrock applications

- Describe LangChain components such as prompt templates, chains, retrievers, and agents
- Apply LangChain components to Amazon Bedrock models to build and test use cases such as text and code generation, summarization, RAG, and question answering
- Use Amazon Bedrock Knowledge Bases to implement RAG applications using best practices
- Use Amazon Bedrock Agents with Amazon Bedrock Knowledge Bases and Amazon Bedrock Guardrails for agent applications

Intended audience

This course is intended for:

- Generative AI application developers

Prerequisites

We recommend that attendees of this course have:

- Intermediate to expert-level proficiency with Python programming language
- *AWS Technical Essentials* (Fundamental)
- *AWS Lambda Foundations* (Fundamental)
- *Amazon Bedrock Getting Started* (Fundamental)
- *Foundations of Prompt Engineering* (Intermediate)

Course outline

Module 1: Introduction to Amazon Bedrock

Building Generative AI Applications on Amazon Bedrock

Applications and Use Cases

The rise of generative artificial intelligence (generative AI) has revolutionized the work of machine learning (ML) engineers, developers, and data scientists. They can use it to work on innovative and impactful projects by automating repetitive implementation tasks.

Data scientists can operate at a higher, more strategic level, designing innovative generative AI solutions to address real business problems that directly impact end users. They can focus on architecting solutions, such as AI assistants, supply chain optimizers, and personalized recommendation systems.

Amazon Bedrock offers several natural language processing (NLP) capabilities that can assist data scientists in their work.

- **Text summarization:** using Amazon Bedrock foundation models (FMs) helps data scientists quickly understand key information in large amounts of text for efficient data exploration and cleaning. Summaries help explain model behaviors, speed up report writing, and improve text data analysis.
- **Text generation:** from language models helps data scientists by augmenting training data, generating code, explaining models, and drafting content. AI assistants act as natural language interfaces to query data and models interactively. This course teaches architectures to generate better and highly relevant summaries. These techniques include Amazon Bedrock, LangChain, and Retrieval Augmented Generation (RAG) with persistent embeddings for contextual awareness and key information retention.
- **Question answering systems:** automate tedious data tasks, like documentation reading. They provide insights by answering analytical questions, generate code snippets, and summarize documents. RAG AI assistants can query knowledge bases interactively and generate contextual answers on demand. This course teaches how to build question answering systems and RAG AI assistants.
- **Agents:** understands natural language user requests, break down complex tasks into API calls and data lookups, maintain conversation context, and take actions to fulfill requests. The service orchestrates prompt engineering with company-specific or domain-specific information and provides natural language responses. Amazon Bedrock Agents handles infrastructure, monitoring, encryption, permissions, and invocation management without custom code. Amazon Bedrock Agents integrates with Amazon Bedrock Guardrails to prevent unwanted behavior from model responses or user messages. This course explains how you can use Amazon

Bedrock Agents to synthesize and manage generative AI workflows. It also explains how to use Amazon Bedrock Agents to accelerate generative AI application development.

Quiz:

1. Amazon Bedrock offers several natural language processing (NLP) capabilities that can assist data scientists in their work. Which of the following NLP applications can be implemented with Amazon Bedrock?
   a. Amazon Bedrock offers several NLP capabilities that can assist data scientists in their work. These capabilities include **text summarization**, **text generation**, and **question answering systems**.
2. Using Amazon Bedrock, data scientists can design innovative generative artificial intelligence (generative AI) solutions to address real business problems that directly impact end users. What are some examples of generative AI solutions?
   a. By using generative AI, data scientists can focus on architecting innovative solutions, such as AI assistants, supply chain optimizers, and personalized recommendation systems. They can focus on architecting solutions such as **AI assistants**, **supply chain optimizers**, and **personalized recommendation systems**.

Module 2: Foundation Models

Objectives	Topics
In this module, you will learn how to do the following: • Identify the foundation models available with Amazon Bedrock. • Describe how to control the inference parameters to tune desired output. • Describe how to use APIs to invoke foundation models or create customization jobs. • Identify monitoring and logging capabilities and tools for governance and audit requirements.	The module is organized into the following topics: • Introduction to Amazon Bedrock Foundation Models • Using Amazon Bedrock FMs for Inference • Amazon Bedrock Methods • Data Protection and Auditability

Amazon Bedrock offers a wide choice of high-performing foundation models (FMs) from leading artificial intelligence (AI) startups and Amazon. Each of these FMs cater to different generative artificial intelligence (generative AI) use cases, such as summarization, language translation, coding, and image generation.

Company	Foundation model	Description
Amazon	**Amazon Titan**	Family of models built by Amazon that are pretrained on large datasets, which makes them powerful, general-purpose models.
AI2I Labs	**Jurassic-2** **Jumba**	Multilingual large language models (LLMs) for text generation in Spanish, French, German, Portuguese, Italian, and Dutch.
Anthropic	**Claude 3**	Claude 3 models offer increasingly powerful performance, allowing users to select the optimal balance of intelligence, speed, and cost for their specific application.
Cohere	**Command and Embed**	Text generation model for business applications and embeddings model for

		search, clustering, or classification in more than 100 languages.
Meta	**Llama**	Llama models for are particularly useful for applications such as chat interfaces, virtual assistants, and language translation.
Mistral AI	**Mistral/Mixtral**	Models for Synthetic Text Generation, Code Generation, RAG, or Agents
Stability AI	**Stable Diffusion**	Text-to-image model for generation of unique, realistic, high-quality images, art, logos, and designs. Now available: Stable Diffusion XL (SDXL) 1.0

Inference parameters

When interacting with an FM, you can configure the inference parameters to customize the FM's response. Generally, you should only adjust one parameter at a time, and the results can vary depending on the FM. The following parameters can be used to modify the output from the LLMs (not all parameters are available with all LLMs).

Randomness and diversity

Foundation models typically support the following parameters to control randomness and diversity in the response.

- **Temperature:** controls randomness in word choice. Lower values lead to more predictable responses. The following table lists minimum, maximum, and default values for the temperature parameter.

Parameter	JSON Field Format	Minimum	Maximum	Default
Temperature	temperature	0	1	0

- **Top K:** limits word choices to the K most probable options. Lower values reduce unusual responses.
- **Top P:** cuts off low probability word choices based on cumulative probability. It tightens overall response distribution. The following table lists minimum, maximum, and default values for the Top P parameter.

Parameter	JSON Field Format	Minimum	Maximum	Default
Top P	topP	0	1	1

Length

Foundation models typically support the following parameters to control the length of the generated response.

- **Response length:** sets minimum and maximum token counts. It sets a hard limit on response size. The following table lists minimum, maximum, and default values for the response length parameter. Maximum response length is dependent on the specific FM.

Parameter	JSON Field Format	Minimum	Maximum	Default
Response length	maxTokenCount	0	8,000	512

- **Length penalty:** encourages more concise responses by penalizing longer ones. It sets a soft limit on size.

- **Stop sequences:** include specific character combinations that signal the model to stop generating tokens when encountered. It is used for the early termination of responses.

Working with Amazon Bedrock FMs

Some inference parameters are common across most models, such as temperature, Top P, Top K, and response length. You will dive deep into unique model-specific parameters and I/O configuration you can tune to achieve the desired output based on the use case.

Amazon Titan foundation models

Amazon Titan models are Amazon foundation models. Amazon offers the Amazon Titan Text model and the Amazon Titan Embeddings model through Amazon Bedrock.

Amazon Titan models support the following unique inference parameters in addition to temperature, Top P, and response length, which are common parameters across multiple models.

Stop sequences: with stop sequences **(stopSequences)**, you can specify character sequences to indicate where the model should stop. Use the pipe symbol **(|)** to separate different sequences (maximum 20 characters).

Amazon Titan Text: is a generative LLM for tasks such as summarization, text generation, classification, open-ended question answering, and information extraction. The text generation model is trained on many different programming languages and Rich Text Format (RTF), like tables, JSON, comma-separated values (CSV), and others. The following example shows an input configuration used to invoke a response from Amazon Titan Text using Amazon Bedrock. You can pass input configuration parameters along with an input prompt to the model.

- **Input**

```
{
    "inputText": "<prompt>",
    "textGenerationConfig" : {
        "maxTokenCount": 512,
        "stopSequences": [],
        "temperature": 0.1,
        "topP": 0.9
    }
}
```

The following example shows the output from Amazon Titan Text for the input supplied in the previous code block. The model returns the output along with parameters, such as the number of input and output tokens generated, along with the reason the response finished being generated.

- **Output**

```
{
    "inputTextTokenCount": 613,
    "results": [{
        "tokenCount": 219,
        "outputText": "<output>",
"completionReason" : "string"
    }]
}
```

Amazon Titan Text Embeddings: translates text inputs (words and phrases) into numerical representations (embeddings). Applications of this model include personalization and search. Comparing embeddings produces more relevant and contextual responses than word matching. The following example demonstrates how you can create embeddings vectors from prompts using the Amazon Titan Embeddings model V2.

- **Input**

```
{
    body = json.dumps({"inputText": <prompt>,
"dimensions": <256   512   1024>,
"normalize": True   False,
})
    model_id = 'amazon.titan-embed-text-v2:0'
    accept = 'application/json'
    content_type = 'application/json'

    response = bedrock_runtime.invoke_model(
         body=body,
         modelId=model_id,
         accept=accept,
         contentType=content_type )

    response_body = json.loads(response['body'].read())
    embedding = response_body.get('embedding')
}
```

This will generate an embeddings vector consisting of numbers that look like the following output.

- **Output**

```
[0.82421875, -0.6953125, -0.115722656, 0.87890625, 0.05883789, -0.020385742,
0.32421875, -0.00078201294, -0.40234375, 0.44140625, ...]
```

Amazon Titan Multimodal Embeddings: is used for use cases like searching images by text, by image for similarity, or by a combination of text and image. It translates the input image or text into an embedding that contain the semantic meaning of both the image and text in the same semantic space. The following example demonstrates how you can create embeddings vectors from a prompt and image using the Amazon Titan Multimodal Embeddings G1.

- **Input**

```
{
   body = json.dumps({"inputText": <prompt>,
       "inputImage": <image>,
       "embeddingConfig":  { "outputEmbeddingLength":  <output_embedding_length> }
})
   model_id = 'amazon.titan-embed-image-v1'
   accept = 'application/json'
   content_type = 'application/json'

   response = bedrock_runtime.invoke_model(
        body=body,
        modelId=model_id,
        accept=accept,
        contentType=content_type )

   response_body = json.loads(response.get('body').read())
   embedding = response_body.get('message')
}
```

Amazon Titan Image Generator: is an image generation model. It comes in two versions v1 and v2.

- **With Amazon Titan Image Generator v1**, users can create images that match their text-based descriptions by inputting natural language prompts. They can upload and edit existing images, apply text-based prompts, or edit specific parts of an image using an image mask. The model also supports **outpainting**, which extends the boundaries of an image. **Inpainting** is used to fill in missing image areas.
- **Amazon Titan Image Generator v2** supports all the existing features of Titan Image Generator v1 with additional capabilities. It allows users to leverage reference images to guide image generation. The output image will then align with the layout and composition of the reference image while still following the textual prompt. It also includes an automatic background removal feature to remove backgrounds from images containing multiple objects.

AI21 Jurassic-2 (Mid and Ultra)

Common parameters for Jurassic-2 models include temperature, Top P, and stop sequences. Jurassic-2 models support the following unique parameters to control randomness, diversity, length, or repetition in the response:

Length

- **Max completion length (maxTokens):** specify the maximum number of tokens to use in the generated response.
- **Stop sequences (stopSequences):** configure stop sequences that the model recognizes and after which it stops generating further tokens.

Repetitions

- **Presence penalty (presencePenalty):** use a higher value to lower the probability of generating new tokens that already appear at least once in the prompt or in the completion.
- **Count penalty (countPenalty):** use a higher value to lower the probability of generating new tokens that already appear at least once in the prompt or in the completion. The value is proportional to the number of token appearances.
- **Frequency penalty (frequencyPenalty):** use a higher value to lower the probability of generating new tokens that already appear at least once in the prompt or in the completion. The value is proportional to the frequency of the token appearances (normalized to text length).
- **Penalize special tokens:** reduce the probability of repetition of special characters. The default values are true as follows:
  - **Whitespaces (applyToWhitespaces):** a **true** value applies the penalty to white spaces and new lines.
  - **Punctuations (applyToPunctuation):** a **true** value applies the penalty to punctuation.
  - **Numbers (applyToNumbers):** a **true** value applies the penalty to numbers.
  - **Stop words (applyToStopwords):** a **true** value applies the penalty to stop words.
  - **Emojis (applyToEmojis)**: a **true** value excludes emojis from the penalty.

Jurassic-2 Mid: this is a mid-sized model that is optimized to follow natural language instructions and context, so there is no need to provide it with any examples. It is ideal for composing human-like text and solving complex language tasks, such as question answering, and summarization.

Jurassic-2 Ultra: is a large-sized model that you can apply to language comprehension or generation tasks. Use cases include generating marketing copy, powering AI assistantsassisting with creative writing, performing summarization, and extracting information.

- **Input**

```
{
    "prompt": "<prompt>",
    "maxTokens": 200,
    "temperature": 0.5,
    "topP": 0.5,
    "stopSequences": [],
    "countPenalty": {"scale": 0.0},
    "presencePenalty": {"scale": 0.0},
    "frequencyPenalty": {"scale": 0.0}
}
```

- **Output**

```
{
    "id": 1234,
    "prompt": {
        "text": "<prompt>",
        "tokens": [
            {
                "generatedToken": {
                    "token": "\u2581who\u2581is",
                    "logprob": -12.980147361755371,
                    "raw_logprob": -12.980147361755371
                },
                "topTokens": null,
                "textRange": {"start": 0, "end": 6}
            },
            //...
        ]
    },
    "completions": [
        {
            "data": {
                "text": "<output>",
                "tokens": [
                    {
                        "generatedToken": {
                            "token": "<|newline|>",
                            "logprob": 0.0,
                            "raw_logprob": -0.01293118204921484
                        },
                        "topTokens": null,
                        "textRange": {"start": 0, "end": 1}
                    },
                    //...
                ]
            },
            "finishReason": {"reason": "endoftext"}
        }
    ]
}
```

Jamba-Instruct: offers a 256K context window for text generation, summarization, and question answering tasks for the enterprise. To call the AI21 Labs Jamba-Instruct model, you can use either ***invoke_model*** or ***converse*** API.

- **Input with *invoke_model***

```
response = bedrock.invoke_model(
modelId='ai21.jamba-instruct-v1:0',
body=json.dumps({
'messages': [
{ 'role': 'user',
'content': <prompt>
}
],
})
)
```

- **Input with *converse***

```
response = bedrock.converse(
modelId='ai21.jamba-instruct-v1:0',
'messages': [
{ 'role': 'user',
'content': [{'text':<prompt>}]
}
],
)
```

Anthropic Claude

Anthropic Claude 2 and Claude 3 are additional models available for text generation on Amazon Bedrock. Claude is a generative AI model by Anthropic. It is purpose built for conversations, summarization, question answering, workflow automation, coding, and more. It supports everything from sophisticated dialogue and creative content generation to detailed instruction following.

You can use Amazon Bedrock to send Anthropic Claude Text Completions API or Anthropic Claude Messages API inference requests. You use the messages API to create conversational applications, such as a virtual assistant or a coaching application. Use the text completion API for single-turn text generation applications. For example, generating text for a blog post or summarizing text that a user supplies.

Claude uses common parameters, such as temperature, Top P, Top K, and stop sequences. In addition, Claude models use the following unique parameter to further tune the response output.

- **Maximum length (max_tokens_to_sample)**: specify the maximum number of tokens to use in the generated response.
-

Using Messages API

You can use the **Messages API** to create AI assistant applications. The API manages the conversational exchanges between a user and an Anthropic Claude model (assistant). In Anthropic Claude Messages API, each input message must be an object with a role and content.

Here is an example with a single user message:

```
[{"role": "user", "content": "Hello, Claude"}]
```

The following example shows an input configuration used to invoke a response from Anthropic Claude 3 using Amazon Bedrock with multiple conversational turns:

- **Input**

```
[  {"role": "user", "content": "Hello there."},
  {"role": "assistant", "content": "Hi, I'm Claude. How can I help you?"},
  {"role": "user", "content": "Can you explain LLMs in plain English?"},
]
```

- **Output**

```
{
   "id": "<identifier>",
   "type": "message",
   "role": "assistant",
   "model": "claude-3-sonnet-20240229",
   "content": [
       {
           "type": "text",
           "text": " Sure, I'll try to explain …"
       }
   ],
   "stop_reason": "end_turn",
   "stop_sequence": null,
   "usage": {
       "input_tokens": <token count>,
       "output_tokens": <token count>
   }
}
```

Using Converse API

The Converse API provides a unified set of parameters that work across all models that support messages.

Stability AI (SDXL)

This is a text-to-image model used to generate detailed images. SDXL includes support for the following types of image creation:

- **Image-to-image prompting:** this involves inputting one image to get variations of that image.
- **Inpainting:** this involves reconstructing the missing parts of an image.
- **Outpainting:** this involves constructing a seamless extension of an existing image.

Stability AI Diffusion models support the following controls:

- **Prompt strength (cfg_scale):** this control determines how much the final image portrays the prompt. Use a lower number to increase randomness in the generation.
- **Generation step (steps):** this control determines how many times the image is sampled. More steps can result in a more accurate result.
- **Seed (seed):** this control determines the initial noise setting. Use the same seed and the same settings as a previous run so inference can create a similar image. If you don't set this value, it is set as a random number.

The following example shows an input configuration used to invoke a response from SDXL using Amazon Bedrock.

- **Input**

```
{       "text_prompts": [
            {
```

```
            "text": string,
            "weight": float
         }
      ],
      "height": int,
      "width": int,
      "cfg_scale": float,
      "clip_guidance_preset": string,
      "sampler": string,
      "samples",
      "seed": int,
      "steps": int,
      "style_preset": string,
      "extras" :JSON object

}
```

- **Output**

```
{
    "result": "success",
    "artifacts": [
        {
            "seed": 123,
            "base64": "<image in base64>",
            "finishReason": "SUCCESS"
        },
        //...
    ]
}
```

Cohere Command

Command is the flagship text generation model by Cohere. It is trained to follow user commands and be useful instantly in practical business applications, such as summarization, copywriting, dialogue, extraction, and question answering. Optimized for business priorities, Cohere is System and Organizations Control (SOC) 2 compliant and emphasizes security, privacy, and responsible AI.

In addition to temperature, Top P, Top K, maximum length, and stop sequences, the Cohere Command model supports the following unique controls:

- **Return likelihoods (return_likelihoods):** specify how and if the token likelihoods are returned with the response. You can specify the following options:
  a. **GENERATION:** this option only returns likelihoods for generated tokens.
  b. **ALL:** this option returns likelihoods for all tokens.
  c. **NONE:** this option doesn’t return any likelihoods. This is the default option.
- **Stream (stream):** specify *true* to return the response piece by piece in real time and *false* to return the complete response after the process finishes.

The following example shows an input configuration used to invoke a response from Cohere Command using Amazon Bedrock.

- **Input**

```
{
"prompt": "string",
"temperature": float,
"p": float,
"k": float,
"max_tokens": int,
"stop_sequences": ["string"],
"return_likelihoods": "GENERATION|ALL|NONE",
"stream": boolean,
"num_generations": int
}
```

Module 3: Application Components

Objectives	Topics
In this module, you will learn how to do the following: • Describe the components of a generative AI application. • Work with embeddings and vector databases. • Customize a foundation model using RAG and model fine-tuning.	The module is organized into the following topics: • Overview of Generative AI Application Components • Foundation Models and the FM Interface • Working with Datasets and Embeddings • Additional Application Components • RAG

- Describe the text generation and text summarization architecture patterns. - Explain how to use the question answering pattern for generative AI applications. - Describe how the AI assistant pattern is used to enhance the user experience.	- Model Fine-Tuning - Securing Generative AI Applications - Generative AI Application Architecture

Module 4: Using LangChain

Objectives	**Topics**
In this module, you will learn how to do the following: - Identify common challenges practitioners face when developing LLMs. - Describe the benefits of using LangChain for training LLMs. - Integrate LangChain with LLMs, prompt templates, chains, chat models, text embeddings models, document loaders, retrievers, and agents in Amazon Bedrock. - Use LangChain agents to manage external resources.	The module is organized into the following topics: - Optimizing LLM Performance - Integrating AWS and LangChain - Using Models with LangChain - Constructing Prompts - Structuring Documents with Indexes - Storing and Retrieving Data with Memory - Using Chains to Sequence Components - Managing External Resources with LangChain Agents

Demo 1: Explore Generative AI Use Cases using LangChain and Amazon Bedrock

Objectives	Topics
In this lab/demo, you will explore several generative AI use cases using Amazon Bedrock API and AWS-LangChain integration. These examples can be modified to build practical applications. You will employ the architecture patterns from Module 3 to build these applications.	The module is organized into the following topics: - Introduction to Labs - Task 1: Performing Text Generation - Task 2: Creating Text Summarization - Task 3: Using Amazon Bedrock for Question Answering - Task 4: Building a Chatbot - Task 5: Using Amazon Bedrock Models for Code Generation - Task 6: Integrating Amazon Bedrock Models with LangChain Agents

Module 5: Using Knowledge Bases

Objectives	Topics
In this module, you will learn how to do the following: - Identify Retrieval-Augmented Generation (RAG) applications and use cases. - Explore RAG Architecture.	The module is organized into the following topics: - Retrieval-Augmented Generation (RAG) overview, use cases, architecture, and challenges while building RAG applications. - Amazon Bedrock Knowledge Bases

- Understand the challenges that come with building RAG applications. - Explore Amazon Bedrock Knowledge Bases. - Recognize best practices and advanced techniques for RAG.	- Data Ingestion into Knowledge Bases - Fully managed and customized RAG using Amazon Bedrock Knowledge Bases, - Evaluating RAG applications. - Best practices and advanced techniques for RAG.

Demo 2: Build and Evaluate Retrieval Augmented Generation (RAG) Applications Using Amazon Bedrock Knowledge Bases

Objectives	Topics
In this lab/demo, you will build a question-answering application using the AnyCompany knowledge base and Amazon Bedrock's RetrieveAndGenerate API. You leverage the existing knowledge base, which contains comprehensive information about AnyCompany's products, services, corporate details like history, leadership, financial performance, sustainability efforts, and more. You run through various notebooks that can effectively answer questions related to AnyCompany's products, services, and corporate information.	The module is organized into the following topics: - Set up the environment - Task 1: Leverage a fully-managed RAG application with Amazon Bedrock's RetrieveAndGenerate API method. - Task 2: Build a Q&A application using Amazon Bedrock Knowledge Bases with Retrieve API method. - Task 3: Test the Query Reformulation process supported by Amazon Bedrock Knowledge Bases. - Task 4: Build and evaluate Q&A Application using Amazon Bedrock Knowledge Bases using RAG Assessment (RAGAS) framework. - Task 5: Test the guardrail functionality on Amazon Bedrock Knowledge Base using RetrieveAndGenerate API method.

Module 6: Using Agents

Objectives	Topics
In this module, you will learn how to do the following: - Understand the concept of Amazon Bedrock Agents. - Describe different use cases for agents. - Describe how Amazon Bedrock Knowledge Bases and Amazon Bedrock Guardrails work in conjunction with agents. - Understand how to create and deploy agents using different methods. - Understand how to create and invoke action groups.	The module is organized into the following topics: - Introduction to agents - Use cases for agents - Overview of Amazon Bedrock Agents - Creating and deploying agents - Agent action groups - Deploying and invoking agents

Demo 3: Explore Amazon Bedrock Agents integrated with Amazon Bedrock Knowledge Bases and Amazon Bedrock Guardrails

Objectives	Topics
In this lab/demo, you use a shopping assistant agent to answer questions about lawn maintenance products from two companies.	The module is organized into the following topics: - Task 1: Configure the Agent's Short-Term Memory

You will use the console to study and test the prebuilt knowledge base, guardrails, and the agent application.	• Task 2: Setup the SageMaker Studio Environment • Task 3: Test and Trace the Agent Steps With and Without Amazon Guardrails