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.

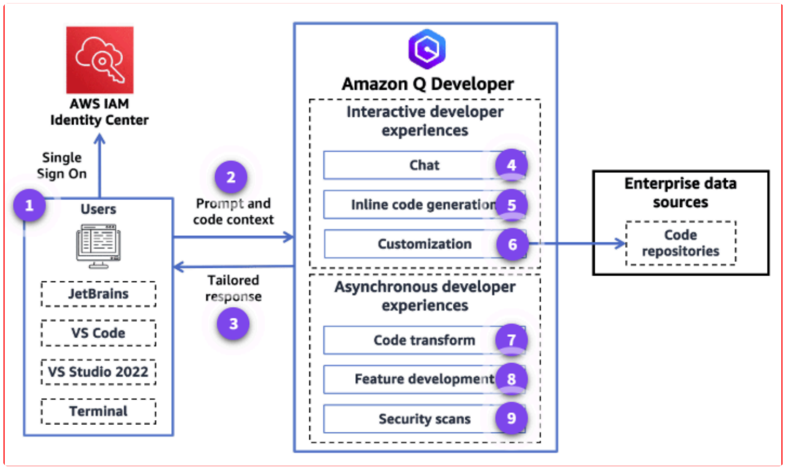
Quiz:

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.