



Amazon Q Dev CLI를 이용한 바이브 코딩 Vibe Coding

윤석찬

수석 테크 에반젤리스트, AWS

1. 바이브 코딩이란?

바이브 코딩이란?

“

제가 “바이브 코딩”이라고 부르는 새로운 종류의 코딩 방식이 있는데, 느낌에 따라서 만든 엄청나게 많은 코드를 받아들이고, 실제로 코드가 존재한다는 사실조차 잊어버리는 것입니다.

LLM(예: Cursor Composer w Sonnet)이 너무 좋아지고 있기 때문에 가능합니다. 또한 SuperWhisper로 Composer와 대화하기 때문에 키보드를 거의 건드리지 않습니다.

Andrej Karpathy, OpenAI 공동창업자



바이브 코딩이란?

- 새로운 아이디어를 구현하거나 실험하기 위한 빠른 프로토타이핑
- AI가 생성한 코드를 직접 읽어보거나, 코드 리뷰를 하지도 않고, 기능 구현에만 중점
- 버그가 생겨도 수정하라고 명령만 내리고, 어떤 코드가 어떻게 수정됐는지에 상관없이 계속 진행
- 버그가 계속 되면, 디버깅에 힘을 쓸기 보다는 처음부터 다시 시작함

개발자가 아닌 사람이 아니라, 어느 정도 개발 능력을
가진 사람이 빠른 생산성을 확보하기 위한 개념입니다!
(Andrej는 이미 뛰어난 개발자였습니다!)

왜 바이브 코딩이 뜨는가?

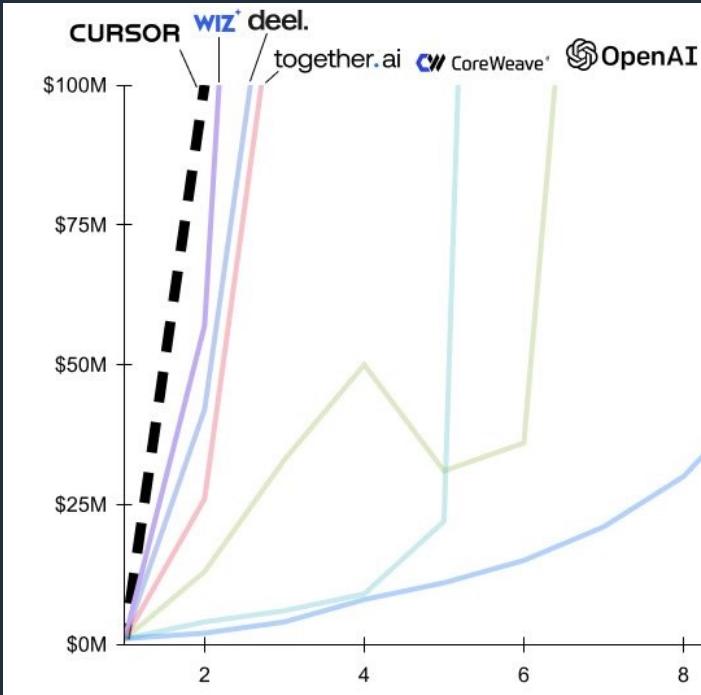
1. Reasoning 모델 등장
2. Cursor AI 도구의 등장
3. 생산성 가속화

1. Reasoning 모델의 등장

특징	Reasoning 모델	Inference 모델
정의	정보를 가지고 생각하는 과정 비가 오는데 우산이 없으니 옷이 젖을 것이다.	정보에 대한 해답을 얻는 과정 비가 오면 (옷이 젖으니) 우산을 사야한다.
주요 목적 및 강점	명시적인 단계별 문제 해결 및 논리적 추론	범용 텍스트 생성 및 이해
모델 종류	ChatGPT o1, Claude 3.7, DeepSeek-R1	ChatGPT-4o, Llama3.3, Claude 3.5
훈련 (학습)	단계별 추론 작업 및 형식 논리에 특화하여 훈련됨	다양한 목적을 가진 방대한 텍스트 데이터로 훈련
출력 구조	명확한 추론 단계 구분을 통해 고도로 구조화 됨	대화형이며, 추론과 다른 내용이 혼합될 수 있음
해석 및 오류 감지	생각의 사슬 (Chain of Thought) 불필요	블랙박스 특성으로 인해 출력 추론 추적이 어려움
활용 사례	과학적 추론 등 다단계 의사 결정 AI 에이전트	챗봇, 텍스트 요약, 콘텐츠 생성, 질의 응답, 코드 완성



2. Cursor AI 도구의 등장



Project Rules

+ Add new rule

Project-specific rules that help the AI understand your codebase and follow your project's conventions. They can be automatically included or fetched by an agent. These are synced with your codebase.

access-vscode-services.mdc

Applies to: vscode/**/*.tsx

access-vscode-services.mdc

Add an agent tool to the codebase

add-an-agent-tool.mdc

Adding Analytics Tracking for user events

add-analytics.mdc

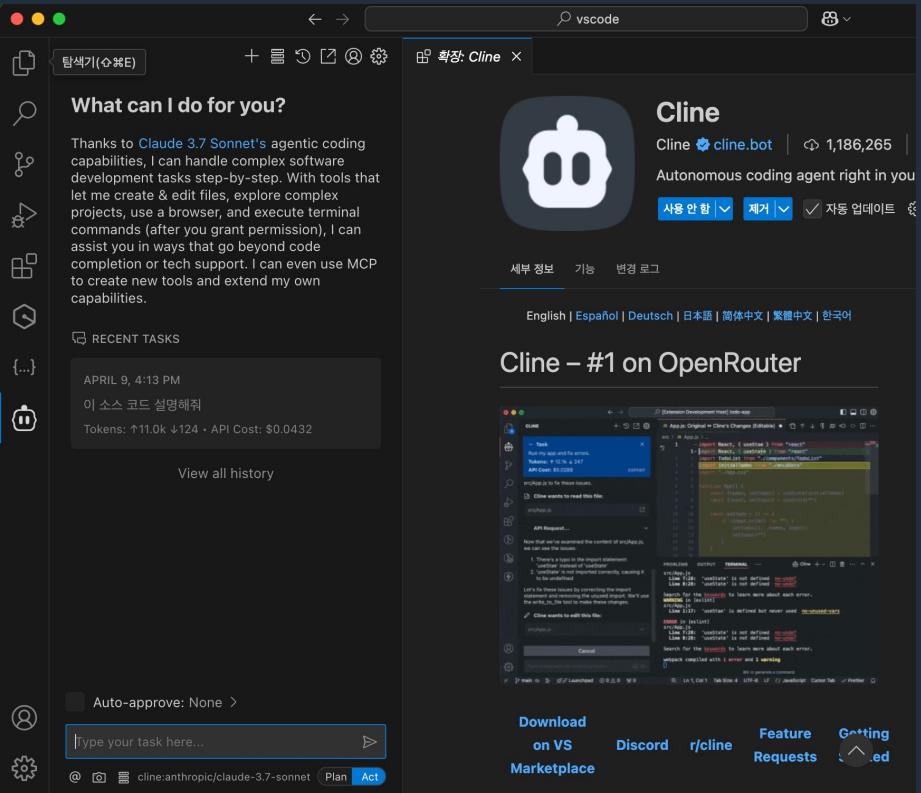
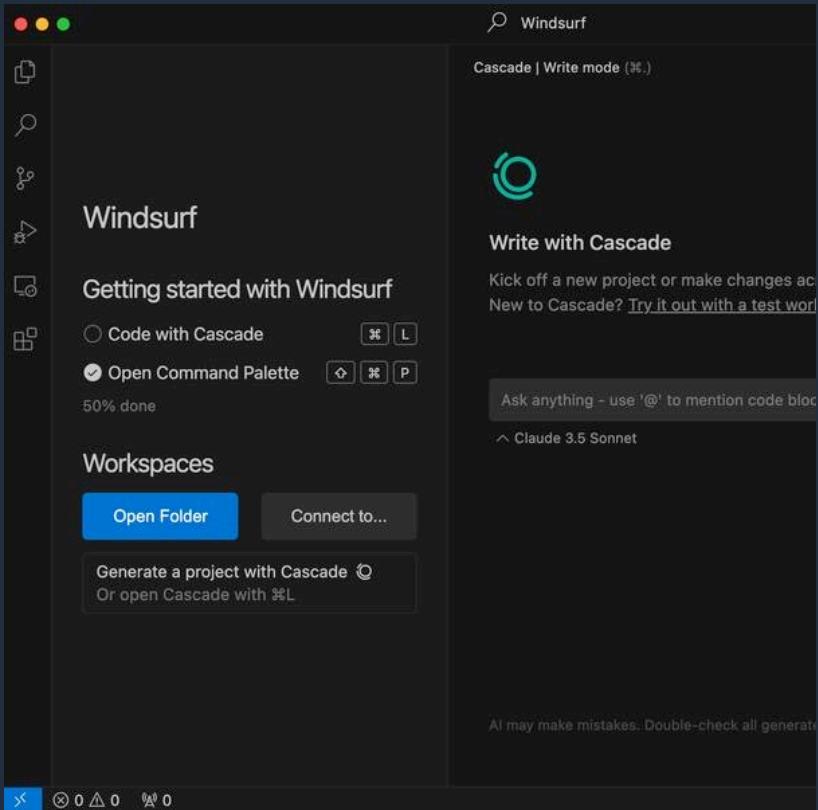
Implement a new backend service and integrate it with the existing frontend system for handling server-client communication.

add-backend-service.mdc

How to add a new composer capability. Useful to add logic that hooks onto Composer functions like stream chat, apply

add-composer-capability.mdc

Windsurf vs. Cline



3. 생산성의 극대화

- **No feature requirement**

- 간단한 기능으로부터 복잡한 기능으로 지속적으로 나아감
- 기능 구현 단계에 대한 AI의 적용 방법을 익힘

- **No human review**

- AI가 생성한 코드를 직접 읽어거나, 코드 리뷰를 하지 않음

- **No human debug**

- 오류가 나오면, 직접 디버깅 하지 않고, AI에게 맡김
- 더 많은 오류가 생기는 경우, 처음 부터 다시 시작



바이브 코딩이 유용한 케이스

- 학습을 위한 프로젝트 (개인용)

- 새로운 기술을 배워야 할 때

- 최소한의 MVP 프로젝트 (샌드박스)

- 스타트업 개발 방식
 - Claude Artifact 샌드박스 이용

- 위험성이 낮은 프로젝트 (프로덕션)

- SW 보안 및 개인 정보에 대해서 검토 필요

바이브 코딩을 위한 12가지 규칙

1. 바이브 PM으로 시작하세요
2. 기술 스택을 단순하게 유지하세요
3. AI에게 올바른 규칙과 문서를 제공하세요.
4. AI가 전부 코딩하도록 하지 마세요
5. AI에게 옵션을 물어보고 간단한 것을 선택하세요.
6. 작업을 작은 단계로 나누세요
7. 간단한 이미지로 AI에게 맥락을 제공해 보세요
8. 변경 한 후, 무자비하게 테스트하세요
9. 주저하지 말고 되돌리세요
10. 버전 관리를 위해 Github을 사용하세요
11. 직관을 느끼면서 자신의 방식을 찾아보세요
12. AI에게 코드를 설명해 달라고 요청하세요.



피터 양(creatoreconomy.so)

바이브 코딩의 이슈

• 전문성 부족

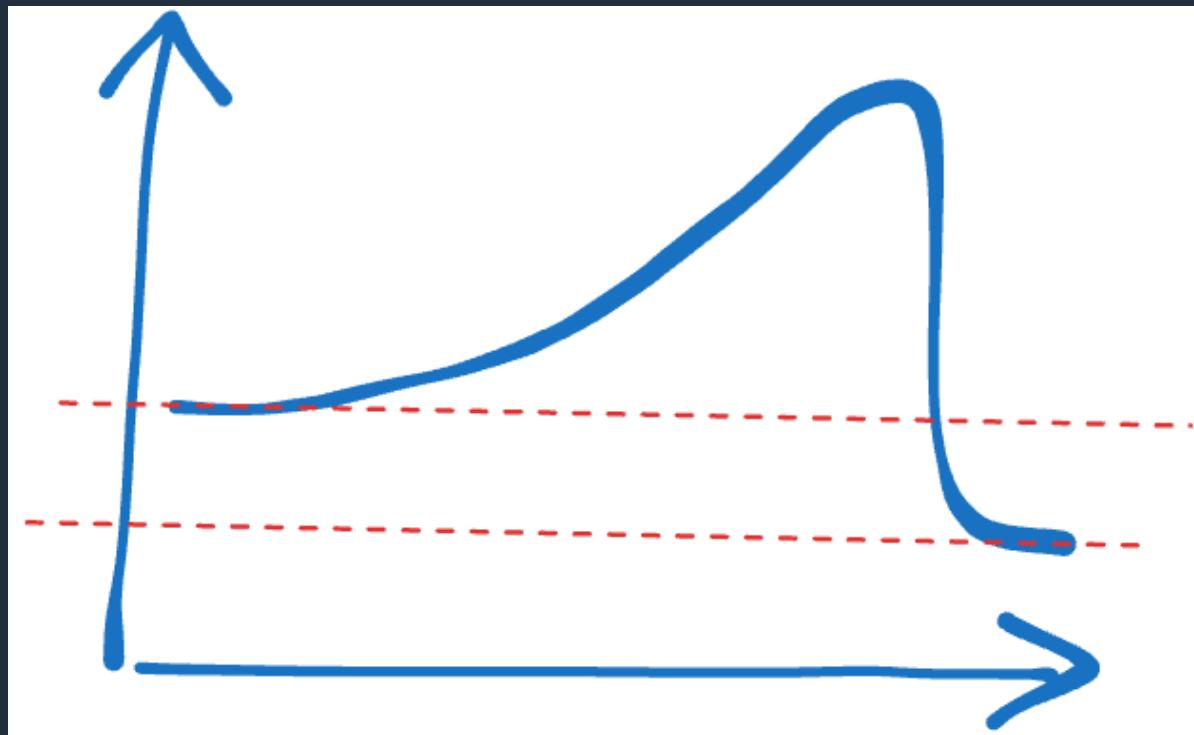
- 기술 분야에 대한 지식이 부족하면, 생성된 코드의 전체적인 구조를 이해할 수 없기 때문에 더 큰 버그를 만든다.
- 도메인 전문성이 부족하면, 버그로 인해 처음부터 다시 시작한다 해도 문제가 반복될 수 있다.
- 해당 기술에 대한 기본적인 지식을 가지기 위한 주요 개념, 용어, 프레임워크에 대한 선행 학습이 필요하다.

• 디버그 능력

- 코드가 계속 생성될 수록 버그도 계속 만들어 질 수 있다. 코드 간결성이 높을 수록 버그 숫자는 줄어든다.
- 코드를 생성하기보다는 더 간결하게 만드는 과정을 거쳐야 한다.

바이브 코딩의 이슈

버그 심각도



바이브 코딩의 이슈

:: 비용 및 사용량 정보

이번 달 비용

US\$842.16

↑ 지난달 같은 기간 대비 2,502%

예상 월말 비용

US\$880.68

↑ 지난달 총 비용 대비 1,927%

절감 기회

-

-

비용(US\$)

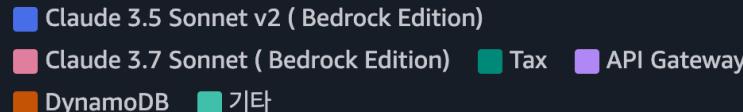
1천

500

0

24년 10월 24년 11월 24년 12월 25년 1월 25년 2월 25년 3월

월(연도)



과금 정보 및 비용 관리로 이동

바이브 코딩에서 개발자의 역할

• 제품 엔지니어

- 사용자의 요구사항을 정확하게 파악하고, AI 도구를 활용해서 빠르게 제품을 개발하는 역할. 마치 건축가처럼 전체적인 그림을 그리고, AI에게 필요한 부분을 지시할 수 있을 역량 필요.
- 화이트보드 코딩 테스트나 알고리즘 문제 풀이보다는 실제 제품을 얼마나 빠르고 효율적으로 만들고 디버깅 할 수 있는지 평가하는 방식의 채용 필요

• 시스템 아키텍처

- 애플리케이션이 동작하는 SW 아키텍처의 안정성과 성능을 유지하고, 복잡한 기술적인 문제를 해결하는 역할. 마치 건물의 안정된 설계를 통해 튼튼하게 만드는 역량 필요.
- 전체적인 시스템 구조를 이해하고, AI가 생성한 코드가 시스템에 잘 통합될 수 있도록 설계하는 능력 평가 필요

2. Amazon Q Developer 소개

Amazon Q Developer 란?

주요 기능

- 17년의 AWS 지식을 바탕으로 훈련된 데이터
- 콘솔, IDE, 문서 등 AWS를 사용하는 모든 곳에서 활용
- 신규 소스 코드 생성 뿐만 아니라 레거시 코드 변환
- 인프라를 위한 커맨드 라인 (CLI) 및 인프라 코드 (IaC) 템플릿 생성
- 사내 데이터에 대한 안전하고 책임감 있는 AI 가드레일 제공



The screenshot shows a developer wearing glasses and a denim jacket working at a computer. A purple hexagonal icon with a white question mark is in the top left corner. A speech bubble from the AI says: "Help me write a function to get a wallet repository from DynamoDB". The AI's response is: "Sure thing. Here's how I'd handle that.
1. Instantiate a new DynamoDB client
2. Make an async request with a user-defined [tableName] and [walletId]
3. Return the value of the result's Item key
Want me to make those changes?". The developer replies: "LGTM – go ahead. Thanks!".



AWS 콘솔



통합 개발 도구 (IDE)



AWS 문서 사이트



AWS 콘솔 모바일 앱



Slack 및 MS Teams
(AWS Chatbot 활용)



Amazon
CodeCatalyst

소프트웨어 개발 단계

기획 → 설계 → 코드 구현 → 테스트 → 배포 → 개선



소프트웨어 개발 단계

기획 → 설계 → 코드 구현 → 테스트 → 배포 → 개선

The screenshot shows the AWS Console Home interface. On the left, there's a sidebar with 'Recently visited' services like IAM, Lambda, and DynamoDB. The main area has a 'Welcome to AWS' section with a 'Getting started with AWS' button. A prominent feature is the 'Amazon Q' window, which displays a question: 'How do I build a web app with AWS? What are my options?' Below the question, it lists four main options for building a web app with AWS:

- Host a static website using AWS Amplify. Amplify provides continuous deployment of your static files like HTML, CSS, JavaScript directly to an S3 bucket. This is good for basic marketing sites or static content sites.
- Use AWS Lambda along with API Gateway to build serverless backends. You can write code to handle requests and responses without managing servers. Lambda works well for microservices, APIs, and backend processing.
- Use AWS Elastic Beanstalk for deploying and scaling web applications. It supports several programming languages and frameworks out of the box like Java, .NET, PHP, Node.js, Python and Ruby.
- Build monolithic or microservice architectures using EC2 instances, ECS, EKS or Lambda. EC2 is a good option for virtual servers. ECS and EKS are good for containerized applications.

At the bottom of the Amazon Q window, there's a text input field with placeholder 'Ask me anything about AWS' and a character limit of 'Max 1000 characters'. The footer of the page includes links for 'CloudShell', 'Feedback', and 'Cookie preferences'.

AWS 콘솔에서 웹 애플리케이션 개발 방식 질문에 대한 응답

The screenshot shows a VSCode interface with an 'AMAZON Q: CHAT' panel open. The panel lists several commands:

- /dev: Plan and implement new functionality across multiple files in your workspace.
- /transform: Transform your Java 8 or 11 Maven project to Java 17
- /help: Learn more about Amazon Q
- /clear: Clear this session

Below the commands, there's a text input field with a placeholder 'Ask me anything' and a character limit of 'Max 1000 characters'. At the bottom right, it shows '3999/4000' and a note: 'Use of Amazon Q is subject to the AWS Responsible AI Policy.'

VSCode에서 자바 기반 앱 구현 설계 요청 시 응답



my_project

AMAZON Q

> CODE ISSUES

CHAT

Q - Dev +

S3에 이미지 업로드 하는 AWS lambda 함수를 닷넷프로젝트로 만들어줘

Okay, I'll generate code for that.

This might take a few minutes. You can navigate away from this chat, but please keep this tab open. I'll notify you when I'm done.

Generating code...

I'll share updates here as I work on this.

Summary of changes

- First let's create necessary files for a .NET AWS Lambda function that handles image uploads to S3. I'll create a new Lambda function project with required AWS SDK NuGet packages.
- I have created a complete .NET AWS Lambda function project that handles image uploads to S3, including the main function code, project file with required dependencies, configuration file, and documentation. Since all necessary files are created and properly configured, we can mark this task as complete.

Files used

These fields can change throughout code generation

- Reviewed: 0
- Created: 4
 - ImageUploaderFunction.cs
 - ImageUploader.csproj
 - aws-lambda-tools-defaults.json
 - README.md
- Changed: 0
- Deleted: 0

Code suggestions 4 file(s)

Changes 4 file(s)

- ImageUploaderFunction.cs
- ImageUploader.csproj
- aws-lambda-tools-defaults.json
- README.md

Accept all changes

Provide feedback & regenerate

C empty ↔ ImageUploaderFunction.cs

```
1+ using Amazon.Lambda.Core;
2+ using Amazon.Lambda.APIGatewayEvents;
3+ using Amazon.S3;
4+ using Amazon.S3.Model;
5+ using System;
6+ using System.IO;
7+ using System.Threading.Tasks;
8+
9+ [assembly: LambdaSerializer(typeof(Amazon.Lambda.Serial
10+
11+ namespace ImageUploader
12+
13+ public class Function
14+
15+
16+
17+
18+
19+
20+
21+
22+
23+
24+
25+
26+
27+
28+
29+
30+
31+
32+
33+
34+
35+
36+
37+ public async Task<APIGatewayProxyResponse> Function
{
    private readonly IAmazonS3 _s3Client;

    public Function()
    {
        _s3Client = new AmazonS3Client();
    }

    public async Task<APIGatewayProxyResponse> Function
    {
        try
        {
            // Check if the request contains a file
            if (string.IsNullOrEmpty(request.Body))
            {
                return new APIGatewayProxyResponse
                {
                    StatusCode = 400,
                    Body = "No file data provided"
                };
            }

            // Convert base64 string to bytes
            byte[] imageBytes = Convert.FromBase64S
}
```

소프트웨어 개발 단계

기획 → 설계 → **코드 구현** → 테스트 → 배포 → 개선

The screenshot shows the AWS Cloud9 IDE interface. On the left is the Explorer sidebar with project files like .mvn, .vscode, src, main/java/com/example/project, test/java/com/example/project, target, .gitignore, pom.xml, README.md, and summary.md. The center is the code editor with two tabs: Calculator.java and CalculatorTests.java. The code editor displays Java code for a calculator:

```
// Function to multiply numbers
public Double multiply(Double a, Double b) {
    /**
     * Multiplies two numbers
     * @param a the first number
     * @param b the second number
     * @return the product of a and b
     */
    return a * b;
}

// Function to divide numbers
public Double divide(Double a, Double b) {
    /**
     * Divides two numbers
     * @param a the first number
     * @param b the second number
     * @return the quotient of a and b
     */
    return a / b;
}

// Function to square a number
public Double square(Double a) {
    return a * a;
}
```

On the right side of the code editor, there are three callout boxes with Korean text:

- 코드 자동 생성 (Code Generation)
- 기존 코드 설명 (Existing Code Documentation)
- 주석 및 문서 작성 (Comment and Document Writing)

At the bottom of the code editor, it says: "Your organization controls whether suggestions include code with references. To update these..."

Code Customization

내부 코드 및 모범 사례를 기반으로
개발자의 요구에 맞는 맞춤형 제안

내부 코드 베이스와 작동 방식을
인식하는 채팅 기능 제공

my_project

AMAZON Q

> CODE ISSUES

CHAT

Q - Doc X +

Welcome to doc generation!

I can help generate documentation for your code. To get started, choose what type of doc update you'd like to make.

Create a README

Create a README for this project?

my_project

✓ Yes C Change folder X Cancel

Choose an option to continue

Amazon Q Developer uses generative AI. You may need to verify responses. See the AWS Responsible AI Policy.

... Yes

Okay, I'm creating a README for your project. This may take a few minutes.

Scanning source files

Summarizing source files

Generating documentation

Documents ready 1 file(s)

Documentation 1 file(s)

README.md

✓ Accept C Make changes X Reject

Choose an option to continue

Amazon Q Developer uses generative AI. You may need to verify responses. See the AWS Responsible AI Policy.

1+ # Django Note-Taking Application with AWS Lambda

2+

3+ This project is a Django-based note-taking app + S3.

4+

5+ The application allows users to create, read, + It features a Django web interface for managing + uploads to an S3 bucket.

6+

7+ The note-taking functionality includes a list + forms for creating and editing notes. The app + ensure efficient storage and transfer.

8+

9+ On the AWS side, the Lambda function processes + encoded image data. It generates unique filenames + specified S3 bucket.

10+

11+ ## Repository Structure

12+

13+ - `lambda_function.py`: AWS Lambda function for

14+ - `urls.py`: Main URL configuration for the Django

15+ - `my_app/`: Main application directory

16+ - `forms.py`: Form definitions for the note-taking

17+ - `models.py`: Data models for the Note entity

18+ - `views.py`: View functions for CRUD operations

19+ - `urls.py`: URL patterns for the note-taking

20+ - `templates/my_app/`: HTML templates for rendering

21+ - `note_confirm_delete.html`: Template for confirmation

22+ - `note_detail.html`: Template for displaying a note

23+ - `note_form.html`: Template for note creation

24+ - `note_list.html`: Template for displaying a list of notes

25+

26+ ## Usage Instructions

27+

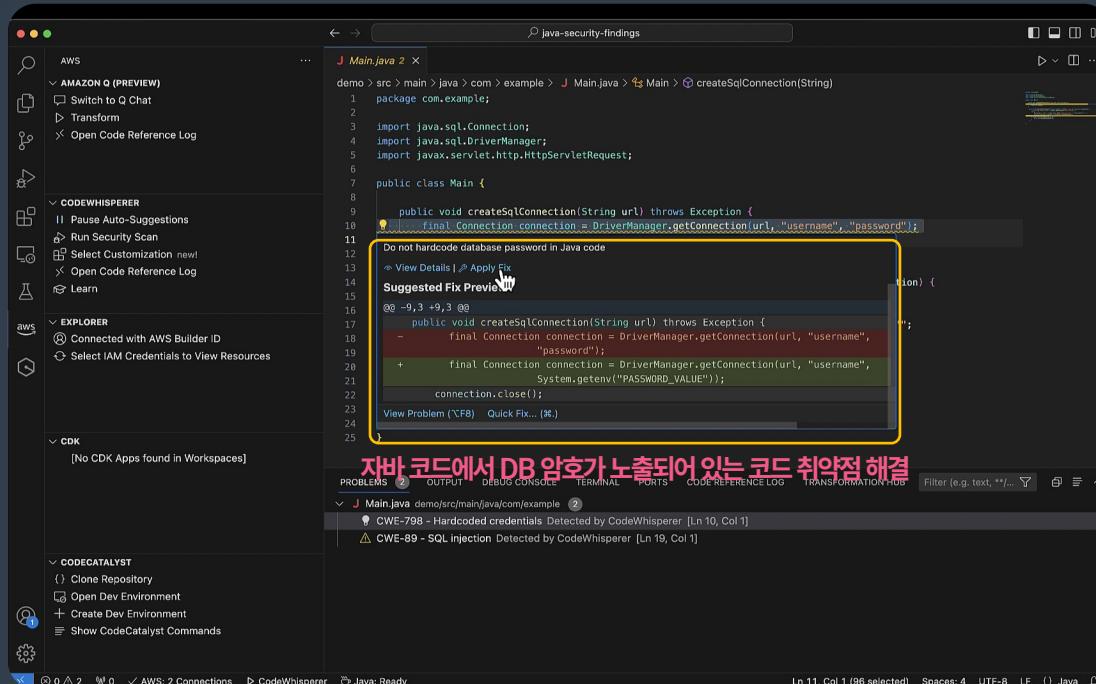
28+ ### Installation

29+

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

소프트웨어 개발 단계

기획 → 설계 → 코드 구현 → 테스트 → 배포 → 개선



- 소스 코드 내 보안 취약점 검사
- 코드 품질을 개선하기 위한 제안

AMAZON Q

> CODE ISSUES

CHAT Chat X Q - Test X +

/test
Included in your Q Developer Agent subscription

I can generate unit tests for your active file.

After you select the functions or methods I should focus on, I will:

1. Generate unit tests
2. Place them into relevant test file

To learn more, check out our [user guide](#).

Sure. This may take a few minutes. I'll share updates here as I work on this.

Generating unit tests for the following methods in

`lambda_function.py`

`lambda_handler`

Select View diff to see the generated unit tests. ➤

> CODE ISSUES

CHAT Chat X Q - Test X +

Please see the unit tests generated below. Click "View diff" to review the changes in the code editor.

READY FOR REVIEW 1 file(s)

my_project 1 file(s)

test 1 file(s)

test_lambda_function.py

View diff

PROBLEMS 3 OUTPUT DEBUG CONSOLE TERMINAL PORTS

Other diff options

test_lambda_function.py (Generated by Amazon Q) 2 X

tmp > aws-toolkit-vscode > q-testgen > resultArtifacts > test > test_lambda_func

```
1+ from my_project.lambda_function import lambda_handler
2+ from unittest.mock import MagicMock, patch
3+ from unittest.mock import patch, MagicMock
4+ import base64
5+ import json
6+ import os
7+ import pytest
8+
9+ class TestLambdaFunction:
10+
11+     @pytest.fixture
12+     def mock_context(self):
13+         context = MagicMock()
14+         context.aws_request_id = 'test_request_id'
15+         return context
16+
17+     @pytest.fixture
18+     def mock_s3(self):
19+         with patch('my_project.lambda_function.boto3.client'):
20+             yield mock_client.return_value
21+
22+     def test_lambda_handler_2(self):
23+         """
24+             Test lambda_handler with non-base64 encoded image
25+         """
26+         # Mock the AWS Lambda context
27+         mock_context = MagicMock()
28+         mock_context.aws_request_id = "test-request-id"
29+
30+         # Mock the S3 client
31+         mock_s3 = MagicMock()
32+
33+         # Mock environment variable
34+         mock_env = {"S3_BUCKET_NAME": "XXXXXXXXXX"}
35+
36+         # Prepare test event data
37+         event = {
```

AMAZON Q

> CODE ISSUES

> CHAT

Chat X Q - Review X +

/review
Included in your Q Developer subscription

I can review your workspace for vulnerabilities and issues.

After you begin a review, I will:

1. Review all relevant code in your workspace or your current file
2. Provide a list of issues for your review

You can then investigate, fix, or ignore issues.

To learn more, check out our [user guide](#).

Which type of review would you like to run?

Project Review File Review Project Review

Waiting on your inputs... ➤

Amazon Q Developer uses generative AI. You may need to verify responses. See the [AWS Responsible AI Policy](#).

> CODE ISSUES

Critical 1 issues
High 1 issues
Medium 3 issues
Low 0 issues
Info 0 issues

> CHAT

Chat X Q - Review X +

Sure. Reviewing your project for code issues.

This may take a few minutes. I'll share updates here as I work on this.

Initiating code review
 Waiting for review to finish
 Processing review results

Reviewing your project is complete. Here's what I found:

- Critical: 1 issue
- High: 1 issues

Ask a question or enter "/" for quick actions ➤

Amazon Q Developer uses generative AI. You may need to verify responses. See the [AWS Responsible AI Policy](#).

Code Issue Details X

CWE-94 - Unsanitized input is run as code

Critical

Passing user-provided input to `eval` and `exec` functions without sanitization makes your code vulnerable to code injection. Make sure you implement input validation or use secure functions.

[Learn more](#)

Common Weakness Enumeration (CWE) CWE-94

Detector library Unsanitized input is run as code

File path main.py [Ln 3]

Suggested code fix preview

```
1 def main():
2     sum = 0
3     numbers = ast.literal_eval(input("Enter numbers separated by commas: "))
4     #import ast
```

Open

Accept Fix Regenerate Fix Explain Ignore Ignore All

소프트웨어 개발 단계

기획 → 설계 → 코드 구현 → 테스트 → 배포 → 개선

The screenshot shows the AWS Lambda console's 'Test' tab for a function named 'Test-Create-Case-API'. It displays a 'Test event' configuration section with a 'Create new event' button and an 'Edit saved event' button (which is selected). Below it is an 'Event JSON' section containing a JSON object:

```
1 - {
2 -   "version": "2.0",
3 -   "routeKey": "$default",
4 -   "rawPath": "/create_case",
5 -   "rawQueryString": "parameter1=value1&parameter1=value2&parameter2=value",
6 -   "cookies": [
7 -     "cookie1",
8 -     "cookie2"
9 -   ]
0 - }
```

At the bottom, there are links for 'CloudWatch' and 'Feedback', and copyright information: © 2023, Amazon Web Services, Inc. or its affiliates.

앱 배포 시, AWS 콘솔에서 오류 진단 및 문제 해결
VPC 네트워크 연결 문제 해결

The screenshot shows the Amazon Q interface with a query: "List services in my account which increased the most from February to March." The response below lists the top 3 AWS services with the largest cost increase:

1. Amazon QuickSight - Cost increased from \$ [redacted] in February to \$ [redacted] in March, an increase of \$266.18.
2. Amazon Relational Database Service - Cost increased from \$ [redacted] in February to \$ [redacted] in March, an increase of \$170.25.
3. Amazon Elastic Container Service - Cost increased from \$ [redacted] in February to \$ [redacted] in March, an increase of \$142.45.

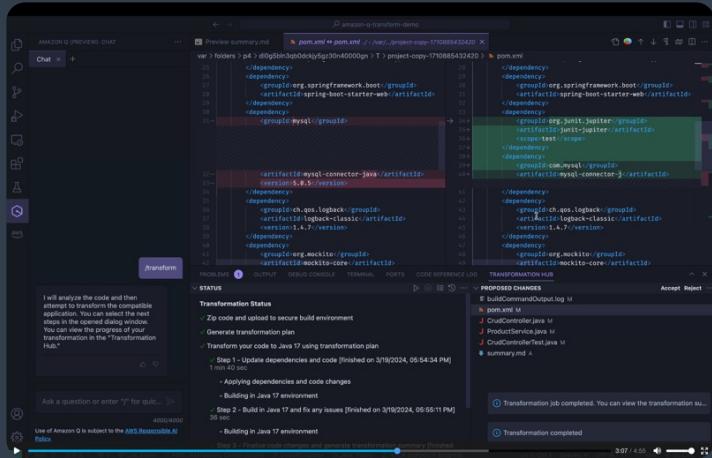
Below the results, it says: "The following Cost Explorer parameters were used to answer your question:" followed by a list of parameters. At the bottom, it says: "Amazon Q's cost analysis capability is in preview and can make mistakes. Please verify your cost data with AWS Cost Explorer. Use the thumb icon to provide feedback and help us improve."

AWS 계정내 리소스 검색 및 AWS Cost Explorer를
통한 비용 데이터 검색 및 분석(미리보기)

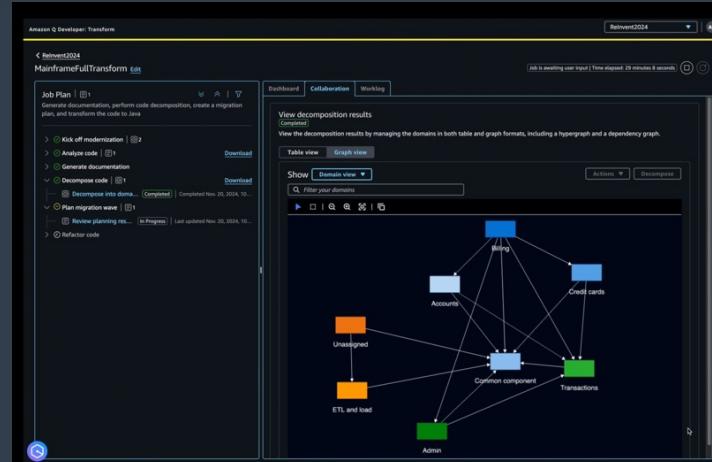


소프트웨어 개발 단계

기획 → 설계 → 코드 구현 → 테스트 → 배포 → 개선



IDE 기반 Java 8/11 코드 17으로 자동 변환 기능 제공
닷넷 프레임워크 지원 (리눅스 마이그레이션 포함)

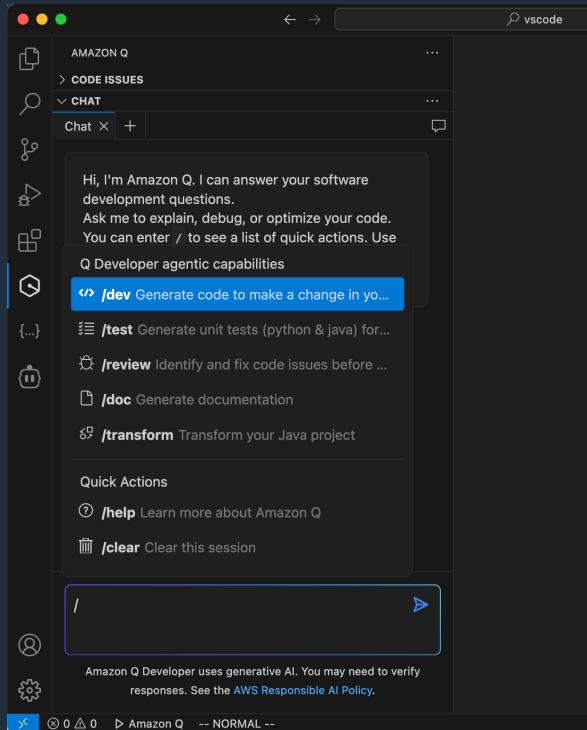


웹 기반 메인프레임 및 VMware 워크로드의
대규모 앱 현대화 기능 제공



3. Amazon Q Developer CLI 실습

Amazon Q Developer for IDE vs CLI



A screenshot of the Amazon Q CLI running in a terminal window titled 'zsh'. The terminal shows the user's last login information: 'Last login: Tue Apr 8 21:09:35 on ttys000'. The command 'q chat' was run. The output is a welcome message from Amazon Q: 'Hi, I'm Amazon Q. Ask me anything.' Below this, a section titled 'Things to try' lists several commands: 'Fix the build failures in this project.', 'List my s3 buckets in us-west-2.', 'Write unit tests for my application.', and 'Help me understand my git status'. At the bottom, a table provides a key for common CLI commands:

/acceptall	Toggles acceptance prompting
/profile	(Beta) Manage profiles for th
/context	(Beta) Manage context files f
/help	Show the help dialogue
/quit	Quit the application

단계 1. AWS Builder ID 가입하기

AWS Builder ID에 가입하세요

AWS Builder ID로 동료 빌더와 연결하고, 기술적 과제에 대한 조언을 얻고, 600+ 무료 과정에 액세스하고, Amazon Q Developer와 같은 도구를 활용하고, 새로운 AWS 제품을 가장 먼저 탐색할 수 있습니다.

[AWS Builder ID 가입하기](#)



bit.ly/aws-builderid-now

Builder ID 혜택

더 많은 방법으로 배우고, 연결하고, 문제를 해결하고, 실험할 수 있습니다.



Amazon Q 개발자
Amazon Q Developer를 코딩, 테스트



AWS re:Post
기술 관련 질문을 하고, 지식을 공유하고,



AWS 스킬 빌더
600+ 이상의 무료 디지털 과정, 학습 계



단계 2. Amazon Q Developer CLI 설치하기

aws

시작하기 서비스 가이드 개발자 도구 AI 리소스

Amazon Q Developer < 사용자 가이드

Amazon Q Developer란 무엇인가요? Features

- ▶ 시작
- ▶ 변환(미리 보기)
- ▶ 의 AWS
- ▶ IDE에서
- ▼ 명령줄에서
 - 지원 환경
 - 설치
 - 검사 중
 - 설치 제거
 - Chat
- ▶ 명령줄 완료
 - 인라인
 - 변환
 - 디버깅
 - 제공
- ▶ GitLab Duo(미리 보기)

설명서 > [Amazon Q](#) > 사용자 가이드

명령줄용 Amazon Q 설치

[PDF](#) [RSS](#) 포커스 모드

ⓘ 기계 번역으로 제공되는 번역입니다. 제공된 번역과 원본 영어의 내용이 상충하는 경우에는 영어 버전이 우선합니다.

Amazon Q 애플리케이션에 대한 파일 다운로드를 시작하여 macOS용 명령줄용 Amazon Q를 설치할 수 있습니다. 자세한 내용은 [자동 완성에 지원되는 명령줄 환경](#) 단원을 참조하십시오.

명령줄용 Amazon Q를 설치할 때 고려해야 할 두 가지 방법이 있습니다. 최소 설치는 Linux for Amazon Q 채팅 및 자동 완성 기능이 SSH(q 및)를 통해 작동하는 데 필요한 바이너리만 배포합니다 qterm. 전체 배포에는 데스크톱 애플리케이션이 포함되어 자동 완성 기능을 사용해야 합니다. Amazon Q 채팅만 사용 하려는 경우 최소 배포도 쉘 qterm에 배송되고 설치된다는 점을 고려하세요. ZShell qterm 용을 사용하여 인라인 완료를 지원할 수 있으며 전체 버전은 필요하지 않습니다. 자세한 내용은 [자동 완성 지원을 위한 ZIP 설정](#) 단원을 참조하십시오.

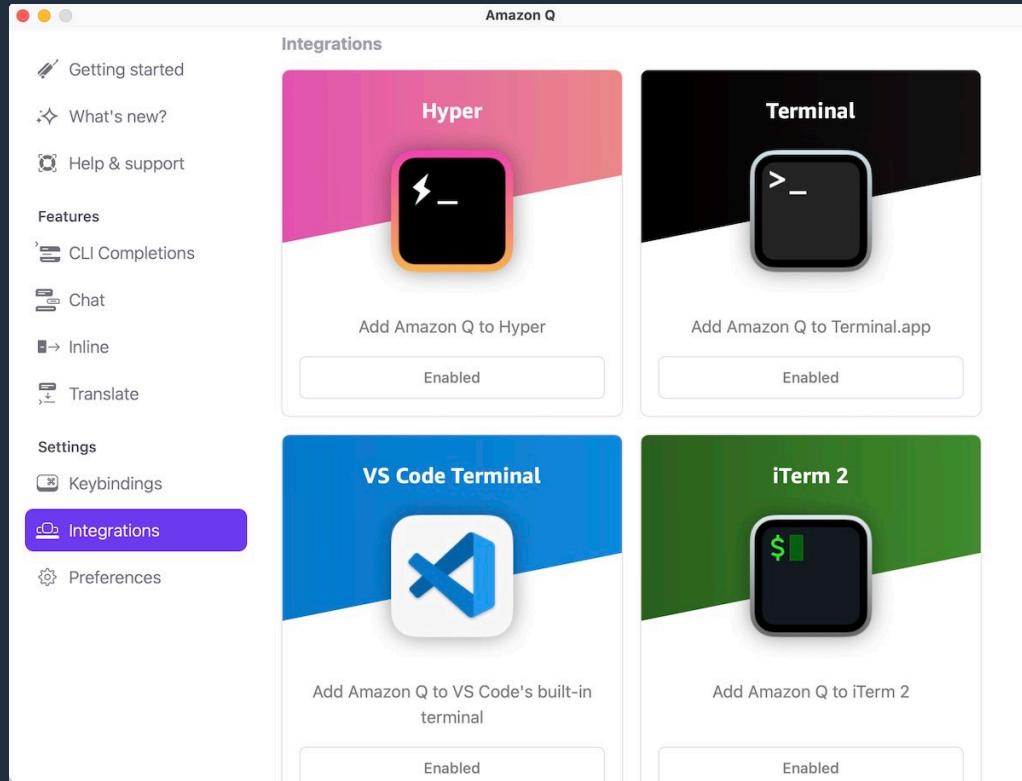
macOS

애플리케이션을 다운로드하거나 Homebrew를 사용하여 macOS용 명령줄용 Amazon Q를 설치할 수 있



go.aws/3YmHfL8

단계 2. Amazon Q Developer CLI 설치하기



```
Last login: Tue Apr  8 21:09:35 on ttys000
channyun@3c063014a57b ~ % q chat

Hi, I'm Amazon Q. Ask me anything.

Things to try
• Fix the build failures in this project.
• List my s3 buckets in us-west-2.
• Write unit tests for my application.
• Help me understand my git status

/acceptall Toggles acceptance prompting
/profile (Beta) Manage profiles for th...
/context (Beta) Manage context files f...
/help Show the help dialogue
/quit Quit the application

> |
```

단계 3. Q Developer CLI로 바이브 코딩하기

1. React와 Vite를 사용하여 강연 정보를 수집하는 call-for-content라는 새로운 애플리케이션을 만들어주세요.
2. 현재 앱을 사용자 이름과 이메일 주소, 강연 제목, 강연 소개, 강연 레벨을 수집하는 간단한 양식으로 바꾸세요. 강연 제목은 100자로, 강연 소개는 100단어로 제한해야 합니다. 강연 레벨은 100, 200, 300 또는 400이어야 합니다.
3. "강연 제안하기" 글자 헤딩을 h2로 변경해주세요.
4. 첫화면을 강연 목록으로 바꾸고, 강연 제안하기를 링크로 추가하도록 해주세요.
5. "강연 목록" 글자 헤딩을 흰색으로 변경해주세요.
6. "강연 제안하기" 페이지의 항목을 웹 브라우저 로컬 스토리지에 저장하고, 이를 "강연 목록" 페이지에서 불러서 보여주도록 바꾸어 주세요.
7. 현재 폴더의 소스 코드에 대한 README 파일을 만들어주세요.
8. 현재 /src 폴더에 있는 모든 소스 코드에 대해 코드 리뷰를 해주세요.
9. 현재 /src 폴더에 있는 코드에 대해 단위 테스트를 작성해주세요.
10. 현재 /src 폴더에 있는 코드에 대해 보안 취약점 점검을 해주세요.



bit.ly/q-dev-cli-vibe

다음 소모임 주제:

Model Context Protocol (MCP) on AWS

2025년 5월 7일(수)