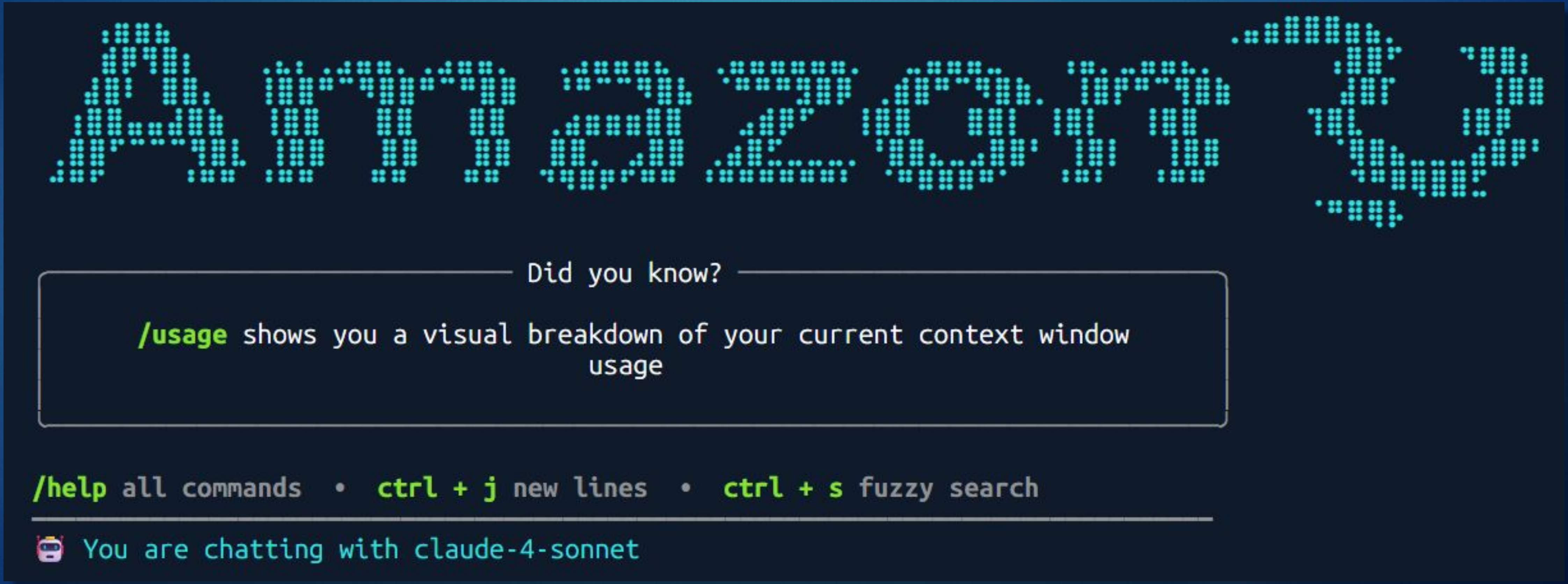
AI-Powered Serverless Development on AWS

SWEN 514/614: Engineering Cloud Software Systems

AJ Barea

Overview

- In this activity, you will use Amazon Q Developer along with AWS Free Tier services to build and deploy a serverless application



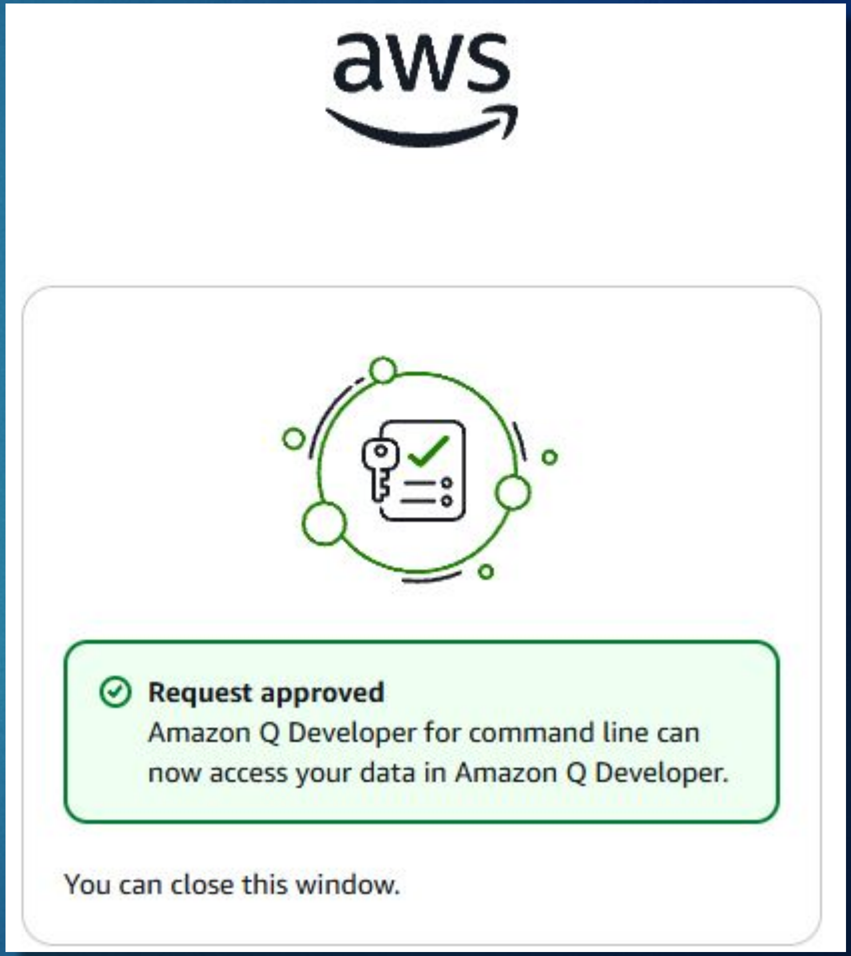
```
Did you know?

/usage shows you a visual breakdown of your current context window
usage

/help all commands  •  ctrl + j new lines  •  ctrl + s fuzzy search

You are chatting with claude-4-sonnet
```

Key Cloud Services

- AWS CloudShell
- AWS Lambda
- Amazon Simple Storage Service (S3)
- AWS Identity and Access Management (IAM)
- AWS Command Line Interface (AWS CLI)
- Amazon Q Developer

Learning Outcomes

- Practice AI-driven cloud development
- Generate and deploy a Lambda function
- Understand how AI tools can scaffold code, policies, and infrastructure



Request approved
Amazon Q Developer for command line can now access your data in Amazon Q Developer.

You can close this window.

Setup Instructions

- Launch AWS CloudShell
- Check Amazon Q availability
  - In CloudShell, run `q --version`
  - If it returns a version number, Q CLI is available. If the command is not found, Q CLI can be installed using a workaround
  - If you see the message "Q CLI integration is temporarily disabled.", Q CLI can still be installed using the workaround

```
q --version
```

Setup Instructions

- To install Q CLI when it’s not available, run:

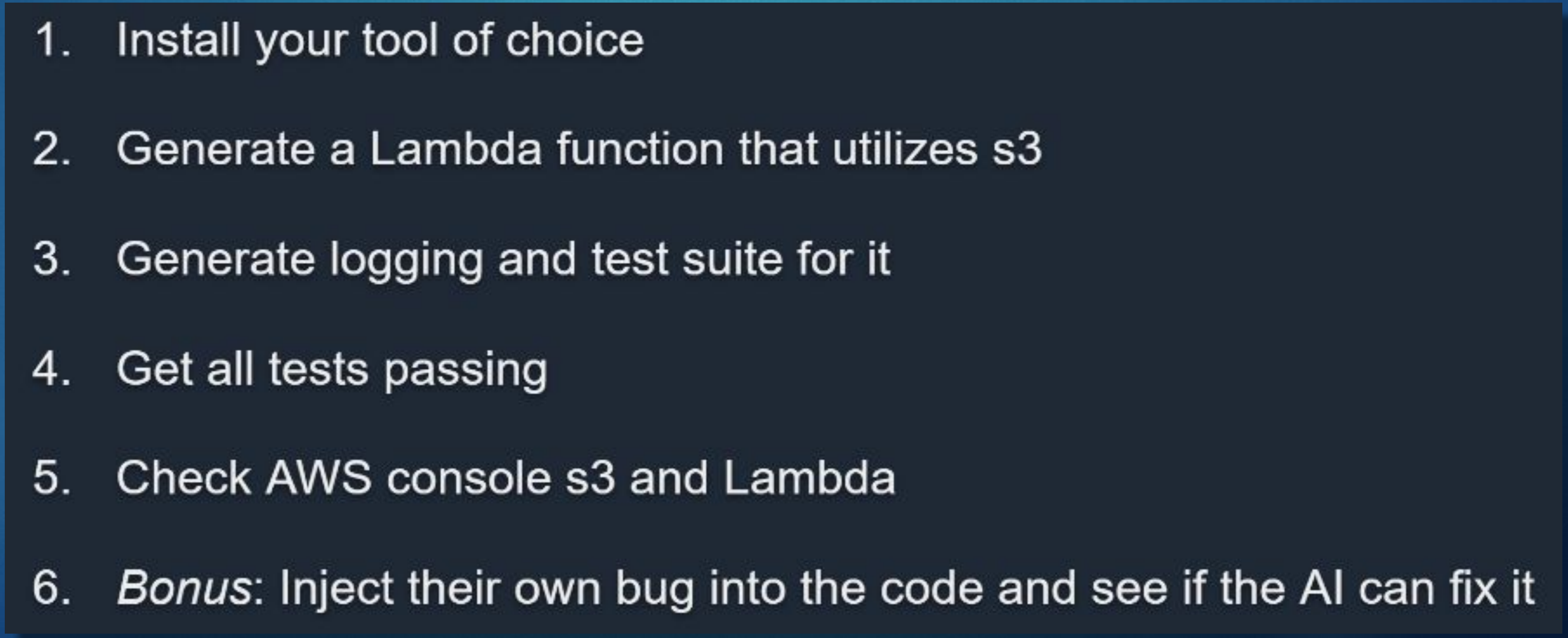
```
curl --proto '=https' --tlsv1.2 -sSf \
"https://desktop-release.q.us-east-1.amazonaws.com/latest/q-x86_64-linux.zip" \
-o q.zip

unzip q.zip && ./q/install.sh

export PATH="$HOME/.local/bin:$PATH"

q login
```

- Start an interactive session with Q:

```
q
```

Activity Tasks Overview

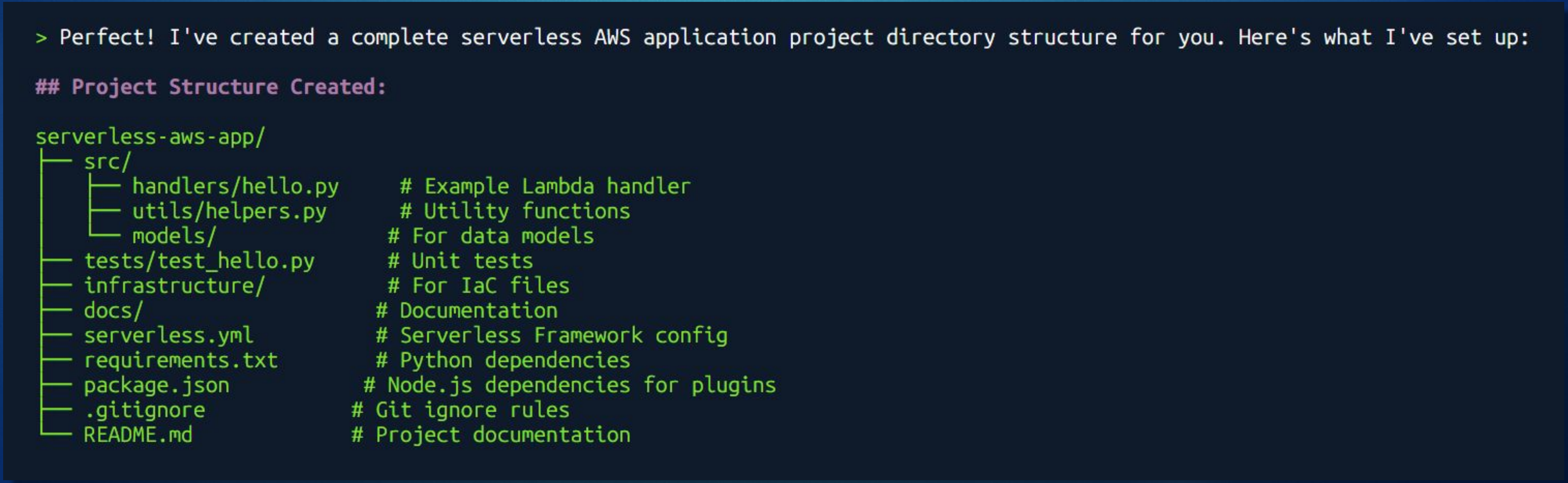
- Three challenge levels: Easy🟢, Medium🟠, Hard🔴
- Each level uses AI prompts to scaffold, integrate, test, and deploy



1. Install your tool of choice
2. Generate a Lambda function that utilizes s3
3. Generate logging and test suite for it
4. Get all tests passing
5. Check AWS console s3 and Lambda
6. *Bonus*: Inject their own bug into the code and see if the AI can fix it

EASY 🟢 Scaffold a Serverless Project

- Prompt examples:
    - Scaffold a new project directory for my serverless AWS app
    - Generate a Lambda function that writes a message to CloudWatch logs



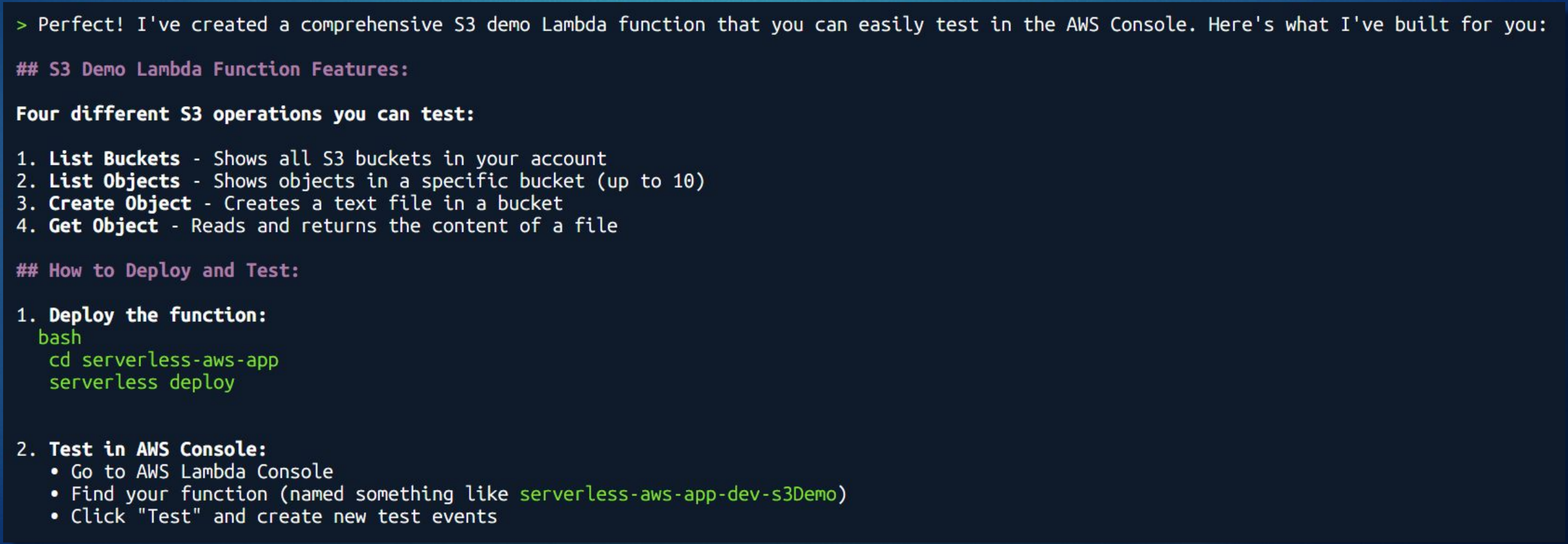
```
> Perfect! I've created a complete serverless AWS application project directory structure for you. Here's what I've set up:

## Project Structure Created:

serverless-aws-app/
├── src/
│   ├── handlers/hello.py     # Example Lambda handler
│   ├── utils/helpers.py      # Utility functions
│   └── models/              # For data models
├── tests/test_hello.py      # Unit tests
├── infrastructure/          # For IaC files
├── docs/                   # Documentation
├── serverless.yml          # Serverless Framework config
├── requirements.txt        # Python dependencies
├── package.json           # Node.js dependencies for plugins
├── .gitignore            # Git ignore rules
└── README.md             # Project documentation
```

MEDIUM 🟠 Integrate S3 and IAM

- Prompt examples:
  - Make a Lambda that reads from and writes to an S3 bucket
  - Generate an IAM policy granting read/write access to that S3 bucket

HARD 🔴 Testing & Deployment

- Prompt examples:
  - Create a test suite for my Lambda, covering success and error cases
  - Deploy my function with AWS CLI and trigger it with test event JSON
  - ***BONUS****: Inject your own bug and see if Q can fix it!*



```
> Perfect! I've created a comprehensive S3 demo Lambda function that you can easily test in the AWS Console. Here's what I've built for you:

## S3 Demo Lambda Function Features:

Four different S3 operations you can test:

1. List Buckets - Shows all S3 buckets in your account
2. List Objects - Shows objects in a specific bucket (up to 10)
3. Create Object - Creates a text file in a bucket
4. Get Object - Reads and returns the content of a file

## How to Deploy and Test:

1. Deploy the function:
  bash
   cd serverless-aws-app
   serverless deploy


2. Test in AWS Console:
   • Go to AWS Lambda Console
   • Find your function (named something like serverless-aws-app-dev-s3Demo)
   • Click "Test" and create new test events
```

Expected Outcomes

- A deployed Lambda function integrated with S3
- Generated IAM policy and test suite
- Better understanding of AI-assisted cloud workflows
- Discussion Questions:
  - What benefits did AI assistance provide during your activity?
  - Did you ask Amazon Q any programming questions that you didn’t know the answer to beforehand?
  - Do you like the CLI developer experience or are IDE extensions with UIs more your speed?
  - What were Amazon Q Developer’s limitations or challenges?

References

- Amazon Web Services. (n.d.). What is Amazon Q Developer? Retrieved from https://docs.aws.amazon.com/amazonq/latest/qdeveloper-ug/what-is.html
- Amazon Web Services. (n.d.). Installing Amazon Q for command line. Retrieved from https://docs.aws.amazon.com/amazonq/latest/qdeveloper-ug/command-line-installing.html
- Amazon Web Services. (n.d.). AWS Management Console. Retrieved from https://console.aws.amazon.com