

# Paired Programming with Gen AI

## GitHub Copilot X Vs AWS CodeWhisperer – The Race ahead in AI Assisted Code Development

Amit Sengupta

Senior Delivery Manager, Software Engineering, Capgemini America

### ABSTRACT

GitHub Copilot X and AWS CodeWhisperer are two AI-powered code generators that have been making waves in the developer community. Both tools use machine learning to suggest code snippets, complete functions, and even write entire classes.

But when you dive into the details, you'll notice that Copilot and CodeWhisperer work a bit differently. They also focus on different sets of use cases, which is the most important distinction between them.

**Keywords:** - Github Copilot, AWS Code Whisperer, Gen AI, Paired Programming, Inline Code Assistance, Shift Left Strategy.

### What is GitHub Copilot?

Built on top of OpenAI's Codex model and further developed by GitHub, Copilot is a code completion tool that offers complete code snippets based on context. Trained on billions of lines of code from public repositories, Copilot can make an educated guess and suggest complete lines of code, increasing efficiency.

### What is Amazon CodeWhisperer?

Announced in June 2022 by Amazon, CodeWhisperer aims to help developer productivity using a machine learning (ML) service that generates code recommendations based on contextual information in the IDE, including the code, and comments in natural language. CodeWhisperer supports only Java, JavaScript, and Python only and supports popular IDEs such as JetBrains IDEs, Visual Studio Code, and Amazon's AWS Cloud9

### GitHub Copilot vs. Amazon CodeWhisperer: Key Similarities

The key similarity is from a functional perspective, since both address the same needs, the developers can use them in same ways. With either tool, you just describe what you want your code to do, then let AI models auto-suggest the code you need to solve the purpose.

Copilot and CodeWhisperer are also similar in the fact that they are both cloud-based solutions offered by major tech vendors.

A third similarity is that neither tool claims to be capable of writing entire software programs on its own. Both Copilot and CodeWhisperer limit themselves to helping complete relatively small segments of code based on context provided by developers.

### GitHub Copilot vs. Amazon CodeWhisperer: Major Differences

The major difference lies on the fact that Copilot is designed to be more of a general-purpose AI-assisted development tool, whereas CodeWhisperer caters first and foremost to development use cases associated with Amazon platforms, such as AWS.

While it doesn't mean that CodeWhisperer can't be used to help write applications that aren't linked in any particular way to the Amazon ecosystem, still when it comes to writing code related to Amazon technologies, CodeWhisperer typically does a better job than Copilot. For instance, If the ask is to write code for moving files between Amazon S3 bucket or working with EC2 instances, the end user would probably have an easier time getting good code using CodeWhisperer.

In contrast, although Copilot is hosted on a Microsoft-owned platform, it doesn't cater in any special way to Microsoft technologies or Microsoft-related programming use cases. It's a general-purpose tool.

Another important difference between Copilot and CodeWhisperer is the fact that CodeWhisperer supports fewer programming languages and IDEs. Currently, it's compatible with C#, Java, JavaScript, Python, and TypeScript as programming languages, and most of the IDEs supported are Amazon-based (JetBrains and Visual Studio Code are the exceptions).

Copilot isn't subject to these limitations. Copilot can generate code in virtually any language, which includes Python, JavaScript, TypeScript, Ruby, Go, C#, and C++ and is longer than the list of languages supported by CodeWhisperer. Copilot can also support almost all of the major IDEs.

### **GitHub Copilot vs. Amazon CodeWhisperer: Which is a preferred tool?**

If the ask is for the best possible code generation experience, GitHub Copilot X is the way to go. It supports a wider range of languages, has more comprehensive code snippet suggestions, integrates better with GitHub repositories, and is more accurate and idiomatic. However, it's still in beta stage and can be buggy at times.

If the end user is looking for a tool stability, AWS CodeWhisperer is a great option. It's more affordable, more stable, and it integrates better with AWS services. However, it supports a narrower range of languages, and its code snippet suggestions can be less comprehensive.

To specifically categorize the above factors, below are the considerations to be made while choosing for the right AI Powered Code Generator: -

### **SCOPE AND FOCUS**

**GitHub Copilot** is more focused on general code development, assisting with a wide range of programming tasks across various languages and frameworks.

**AWS CodeWhisperer** is specialized in AWS cloud development, providing tailored guidance and suggestions for AWS-specific coding and infrastructure.

### **TARGET AUDIENCE**

**GitHub Copilot** caters to a broad audience of developers working on different types of projects.

**AWS CodeWhisperer** is most beneficial for developers building applications within the AWS ecosystem.

### **USE CASES**

**GitHub Copilot** can be used for general code writing, rapid prototyping, and implementing common coding patterns.

**AWS CodeWhisperer** shines when developing applications that leverage AWS services, ensuring optimized architecture, performance, and security.

### **INTEGRATION**

**GitHub Copilot** can be integrated into various code editors and IDEs, making it accessible to a wide range of developers.

**AWS CodeWhisperer** is tightly integrated with AWS services and workflows, enhancing cloud development within the AWS environment. Presently it has good support with VSCode and IntelliJ IDE's.

### **CONCLUSION**

GitHub Copilot X and AWS CodeWhisperer are both powerful tools that can help developers write code faster and more efficiently. However, they have different strengths and weaknesses, and the best fitment depends on specific needs and preferences.

## REFERENCES

- [1]. <https://github.blog/ai-and-ml/github-copilot/how-github-copilot-is-getting-better-at-understanding-your-code/>
- [2]. Sravan Kumar Pala, Use and Applications of Data Analytics in Human Resource Management and Talent Acquisition, International Journal of Enhanced Research in Management & Computer Applications ISSN: 2319-7463, Vol. 10 Issue 6, June-2021.
- [3]. <https://devops.com/measuring-github-copilots-impact-on-engineering-productivity/>
- [4]. Goswami, Maloy Jyoti. "Challenges and Solutions in Integrating AI with Multi-Cloud Architectures." International Journal of Enhanced Research in Management & Computer Applications ISSN: 2319-7471, Vol. 10 Issue 10, October, 2021.
- [5]. Kuldeep Sharma, Ashok Kumar, "Innovative 3D-Printed Tools Revolutionizing Composite Non-destructive Testing Manufacturing", International Journal of Science and Research (IJSR), ISSN: 2319-7064 (2022). Available at: <https://www.ijsr.net/archive/v12i11/SR231115222845.pdf>
- [6]. Neha Yadav, Vivek Singh, "Probabilistic Modeling of Workload Patterns for Capacity Planning in Data Center Environments" (2022). International Journal of Business Management and Visuals, ISSN: 3006-2705, 5(1), 42-48. <https://ijbmv.com/index.php/home/article/view/73>
- [7]. Bharath Kumar. (2022). AI Implementation for Predictive Maintenance in Software Releases. International Journal of Research and Review Techniques, 1(1), 37–42. Retrieved from <https://ijrrt.com/index.php/ijrrt/article/view/175> <https://analyticsindiamag.com/developers-corner/why-developers-are-ditching-github-copilot/> <https://docs.aws.amazon.com/codewhisperer/>
- [8]. Goswami, Maloy Jyoti. "Utilizing AI for Automated Vulnerability Assessment and Patch Management." EDUZONE, Volume 8, Issue 2, July-December 2019, Available online at: [www.eduzonejournal.com](http://www.eduzonejournal.com)
- [8]. Sravan Kumar Pala. (2016). Credit Risk Modeling with Big Data Analytics: Regulatory Compliance and Data Analytics in Credit Risk Modeling. (2016). International Journal of Transcontinental Discoveries, ISSN: 3006-628X, 3(1), 33-39.
- [9]. <https://www.hatica.io/blog/amazon-codewhisperer/>
- [10]. Bharath Kumar. (2022). Integration of AI and Neuroscience for Advancing Brain-Machine Interfaces: A Study. International Journal of New Media Studies: International Peer Reviewed Scholarly Indexed Journal, 9(1), 25–30. Retrieved from <https://ijnms.com/index.php/ijnms/article/view/246>
- [11]. Jatin Vaghela, A Comparative Study of NoSQL Database Performance in Big Data Analytics. (2017). International Journal of Open Publication and Exploration, ISSN: 3006-2853, 5(2), 40-45. <https://ijope.com/index.php/home/article/view/110>
- [12]. Sravan Kumar Pala, "Detecting and Preventing Fraud in Banking with Data Analytics tools like SASAML, Shell Scripting and Data Integration Studio", *IJBMV*, vol. 2, no. 2, pp. 34–40, Aug. 2019. Available: <https://ijbmv.com/index.php/home/article/view/61>
- [13]. Jatin Vaghela, Security Analysis and Implementation in Distributed Databases: A Review. (2019). International Journal of Transcontinental Discoveries, ISSN: 3006-628X, 6(1), 35-42. <https://internationaljournals.org/index.php/ijtd/article/view/54>
- [14]. Anand R. Mehta, Srikarthick Vijayakumar, DevOps in 2020: Navigating the Modern Software Landscape, International Journal of Enhanced Research in Management & Computer Applications ISSN: 2319-7471, Vol. 9 Issue 1, January, 2020. Available at: [https://www.erpublications.com/uploaded\\_files/download/anand-r-mehta-srikarthick-vijayakumar\\_THosT.pdf](https://www.erpublications.com/uploaded_files/download/anand-r-mehta-srikarthick-vijayakumar_THosT.pdf)
- [15]. Goswami, Maloy Jyoti. "Leveraging AI for Cost Efficiency and Optimized Cloud Resource Management." International Journal of New Media Studies: International Peer Reviewed Scholarly Indexed Journal 7.1 (2020): 21-27.