



Serverless x GenAI BKK Workshop

Generative AI at scale: Serverless workflows for enterprise-ready apps

[Pre-requisite] Enable foundation model access in Amazon Bedrock

[Pre-requisite] Configuring the front-end application

► Playground

▼ Use cases

▼ Building a RAG pipeline

Testing the need for RAG

Building the RAG Pipeline

Running the Pipeline

Testing the RAG Inference

High level Code Walkthrough

Summary

► Document extraction and summarization

► Workshop Cleanup

▼ AWS account access

[Open AWS console \(us-west-2\)](#)

[Get AWS CLI credentials](#)

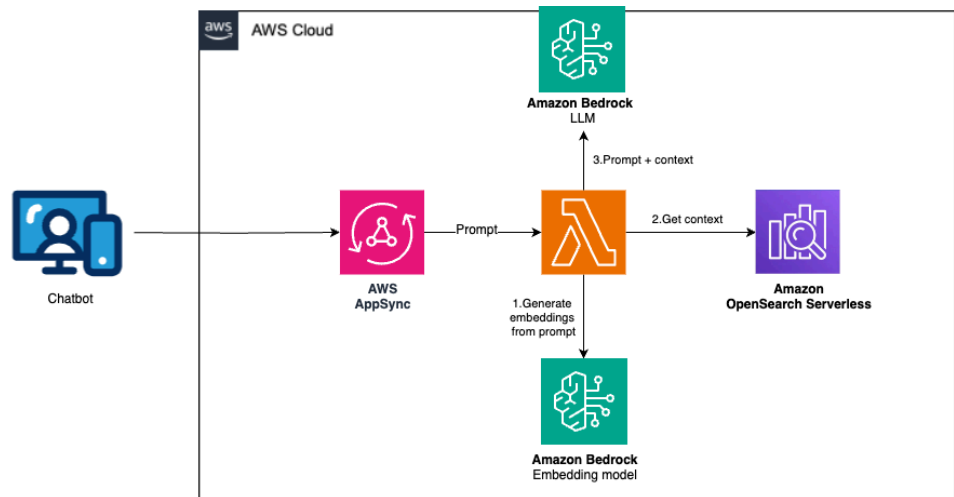
Exit event

[Event dashboard](#) > [Use cases](#) > [Building a RAG pipeline](#) > Testing the RAG Inference

Testing the RAG Inference

In this section, you will use the chatbot to check if it is able to retrieve the correct information for the shareholder related question using the vector data loaded through the pipeline.

To refresh your memory, here is how the retrieval part works. The solution is already deployed. It uses the vector data created through your pipeline.



Test using chatbot

1. Navigate to the chatbot UI.
2. Select **Chatbot** from left menu under **RAG Pipeline**

© 2008 - 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved. [Privacy policy](#) [Terms of use](#) [Cookie preferences](#)

1 How does building in primitives meaningfully expand your degrees of freedom as per Amazon shareholder letter 2023?

You can see the chatbot fetching the relevant answers thanks to the pipeline you built.

USER

How does building in primitives meaningfully expand your degrees of freedom as per Amazon shareholder letter 2023?

ASSISTANT

According to the context provided, building in primitives meaningfully expands one's degrees of freedom in a few key ways: 1. Primitives enable faster innovation and iteration. By creating discrete, foundational building blocks that can be easily combined, developers can rapidly improve customer experiences without being constrained by tightly coupled components. 2. Primitives facilitate external innovation. By exposing these primitives through well-documented, stable APIs, Amazon was able to enable external developers to build anything they imagined on top of AWS, leading to the rapid growth of many companies and industries. 3. Primitives accelerate internal innovation. Amazon's own consumer businesses were able to innovate dramatically across retail, advertising, devices, and more by leveraging the speed and flexibility provided by the primitives in AWS. 4. Primitives allow for more modular and adaptable systems. As Amazon's business grew, the ability to decouple and refactor components into primitives helped overcome challenges with tightly coupled systems and bureaucratic bottlenecks. In summary, the context highlights how building with a primitives-based approach significantly expanded Amazon's degrees of freedom to innovate quickly, enable external builders, and adapt its own systems - key advantages that were central to the company's growth and success.

Using RAG

Summary

You have come to the end of the RAG content aware interaction Module. In this module, you have learnt to build a data ingestion pipeline that generates vector embedding of knowledge base at large scale.

✔ **Congratulations!** You used Distributed Map state for RAG pipeline and interacted with RAG to get the specific answers

References

[Previous](#)

[Next](#)