

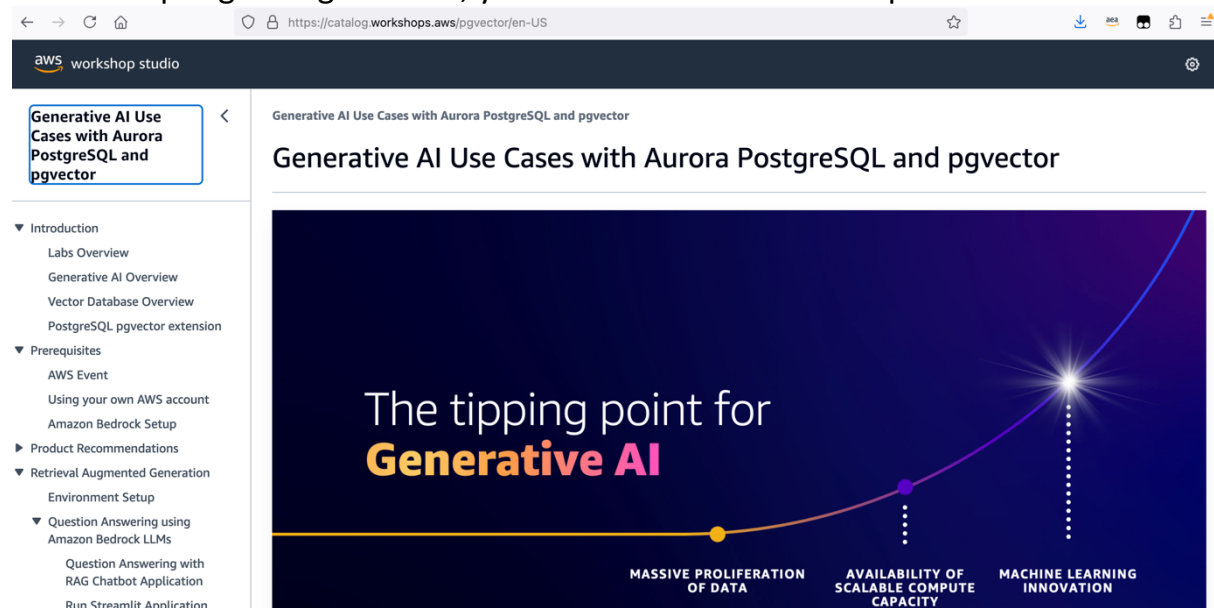
Generative AI and Data Day Workshop (London)

Generative AI Use Cases with Aurora PostgreSQL and pgvector

*** One Click link to access the workshop - <https://catalog.us-east-1.prod.workshops.aws/join>

*** Event Access code will be provided separately. Please ask the AWS staff if you can't get the access code.

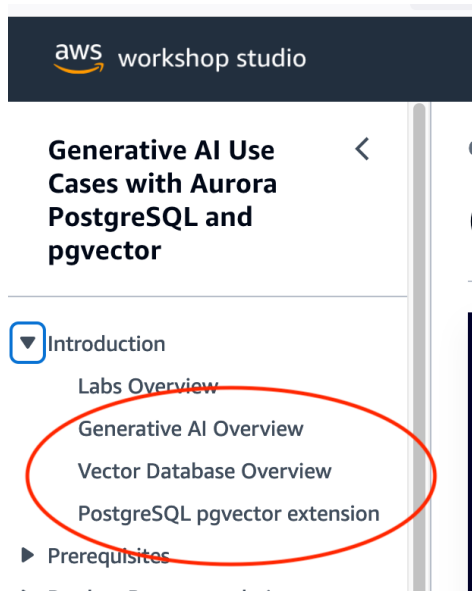
After accepting the agreement, you should see the workshop environment



README First: There are 5 major labs in this workshop but you only need to do the first use case of 'Retrieval Augmented Generation' (RAG) which is **Question Answering using Amazon Bedrock LLMs** and the lab of **Knowledge Bases for Amazon Bedrock with Aurora PostgreSQL**. Each of these labs will take you roughly 30-45 mins apart from the introduction and Prerequisites. You can also just complete one of these labs based on your preference (basic RAG with LangChain or managed RAG with Bedrock Knowledge Bases) as there is no dependency between the 2 labs.

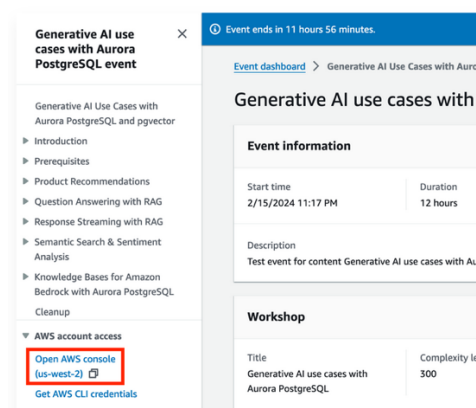
Task 1 - Introduction (~15 minutes)

Go through the introduction and basic concept of GenAI, Vector Database and pgvector extension if you are not familiar with these topics. **You can skip it if you think you have fair amount of knowledge on these topics.**

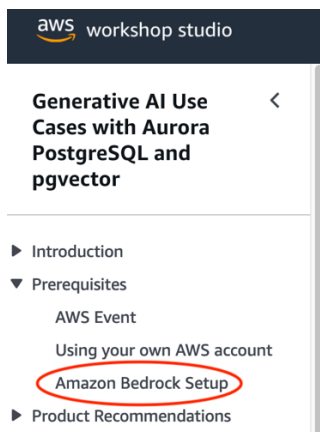


Task 2 Prerequisites (5-10 minutes)

- Click **Open AWS console** on the left navigation bar to log into the AWS Management Console.

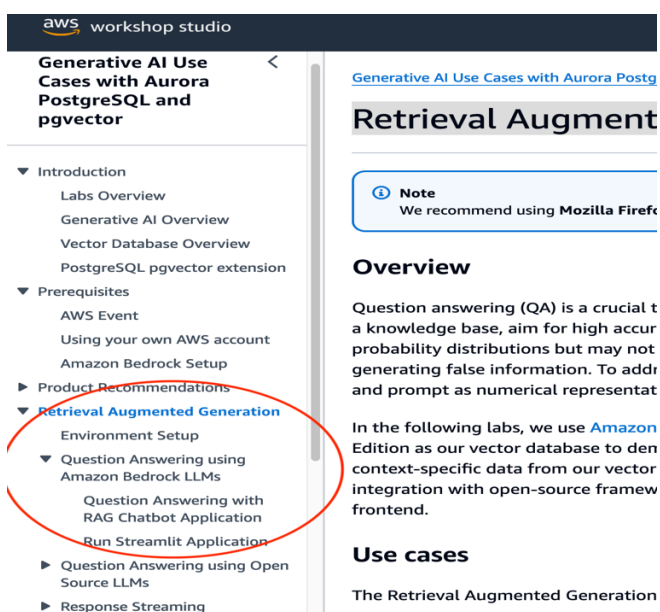


- Go to Amazon Bedrock setup and requested the required models



Task 3 - Retrieval Augmented Generation (RAG) - Question Answering using Amazon Bedrock LLMs (30-45 minutes)

- For this lab, you will experience how to build Generative AI application with Retrieval Augmented Generation (RAG) using opensource framework LangChain and popular UI Streamlit connecting with LLM hosted in Amazon Bedrock.
- From this example app, you will experience the code required to synchronise the data to a Vector stored in PostgreSQL and how to use LangChain framework to initiate the search against the Vector and pass the result to LLM to provide the result and manage the conversational state.
- You **don't need** to complete the labs for Question Answering using Open Source LLMs and Response Streaming but if you want to do so, you may need another 30-45 minutes.



Task 4 - Knowledge Bases for Amazon Bedrock with Aurora PostgreSQL (30-45 minutes)

- For this lab, you will also build a RAG Generative AI app using Vector data stored in PostgreSQL. The major difference from Task 3 is it will leverage the managed RAG capability via Amazon Bedrock and you don't need to rely on other frameworks like LangChain to build the RAG app and you don't need to deal with the complexity or write custom code for Vector Data synchronisation and queries management. The lab will use Amazon Bedrock native test interface instead of a web GUI like StreamLit but this can be easily built with any popular UI like Streamlit or Gradio.

The screenshot displays the AWS Workshop Studio interface. On the left, a sidebar lists various labs under the heading 'aws workshop studio'. The lab 'Knowledge Bases for Amazon Bedrock with Aurora PostgreSQL' is highlighted with a red circle. The main content area shows the lab title 'Knowledge Bases for Amazon Bedrock with Aurora PostgreSQL' and a brief introduction to Knowledge Bases for Amazon Bedrock. The text describes it as a fully managed RAG capability that allows customization of FM responses with company data. It also mentions integrating Amazon Bedrock with Amazon Aurora PostgreSQL to utilize features like pgvector's HNSW indexing and Amazon Bedrock's performance improvements. The lab aims to explore building generative AI applications using the RAG pattern with Knowledge Bases for Amazon Bedrock, setting up an Aurora PostgreSQL cluster as the knowledge base.