



Event ends in 2 days 15 hours 12 minutes.



[Event dashboard](#) > **Open Source With Bedrock**

# Open Source With Bedrock

## Overview

This module focuses on bringing the open source modules into [Bedrock](#). Open source community has been at the fore-front of innovation like RAG, Agents to solve complex business problems. We will showcase major technologies like [Langchain](#) and [LlamaIndex](#). We will showcase RAG and Agents using these technologies.

Entity extraction is an NLP technique that allows us to automatically extract specific data from naturally written text, such as news, emails, books, etc. That data can then later be saved to a database, used for lookup, or any other type of processing. There are many different techniques and approaches for entity extraction, but in this module we will focus on using LLMs for entity extraction. LLM entity extraction works by sending a prompt instruction to the model, asking it to extract entities you specify:

## Why it is Relevant

**Open Source** Within this series of labs, you will be taken through some of the most common usage patterns we are seeing with our customers for Generative AI. We will explore techniques for generating text and images, contextual chats, RAG, creating value for organizations by improving productivity. This is achieved by leveraging foundation models to help in composing emails, summarizing text, answering questions, building chatbots, creating images and generating code. You will gain hands-on experience using Bedrock APIs, SDKs, and open-source software for example LangChain and FAISS to implement these usage patterns.

This module can be executed by any developer familiar with Python, as well as data scientists and other technical people.

- Financial, Data and Business Analysts
- Gen Ai developers
- C-suite executives: Understand how guardrails enhance data security and compliance.
- Marketing and sales teams: Learn how guardrails maintain brand voice and customer engagement.
- IT and DevOps teams: Explore the technical implementation of guardrails.
- Legal teams: Delve into how guardrails can help in compliance with laws and regulations.
- Data scientists, Machine Learning engineers, Developers, Product managers: Dive into the technical details and implementation of guardrails.
- Security & Compliance teams: Understand how guardrails enhance data protection and compliance.

## Challenges

Implementing guardrails for chatbots presents several challenges:

- Ensuring guardrails do not hinder chatbot performance or user experience.
- Keeping guardrails updated as the chatbot evolves.
- Balancing guardrails with flexibility and adaptability in chatbot responses.
- Consistent application of guardrails across all chatbot instances and platforms.

## Sub-patterns

1. [Node JS with Bedrock](#) : In this notebook you will use JavaScript to generate the image using Titan Image generator.
2. [Introduction to Agents](#) : Understanding Inner workings of Agents: In this module we will learn how the agents concept works in general. The files for this are under weather\_service\_agent Agents Intro
3. [Agents with Tools](#) : In these set of examples we will use Claude v3 models with LangChain to showcase function calling and providing tools to Agents to execute the end to end user workflow.



4. [lang Graph Supervisor Agents](#) : In these set of examples we will use Claude v3 models with Lang-graph to demonstrate inter agents communication and leverage agents for complex orchestration. We will demonstrate multi-agent query and examples
5. [Langchain with Chatbot](#) : In these set of examples we will use Titan and Claude and Llama models to show contextual chat and open source vector databases to answer query grounded in facts and reduce hallucinations.
6. [Chatbot with Pinecone](#) : In these of examples we will showcase Knowledgebase with Claude, Titan with open source Postgres and Pinecone and FAISS databases.
7. [Bedrock & NeMo Guardrails](#) : In this notebook, you will utilize Amazon Bedrock and NVIDIA's NeMo guardrails.
8. [Chat with your Long documents with LangChain](#) : In this notebook, you will leverage the key patterns with Amazon Bedrock Claude V3 with ability to process long documents while reducing latency issues.

---

[Previous](#)[Next](#)