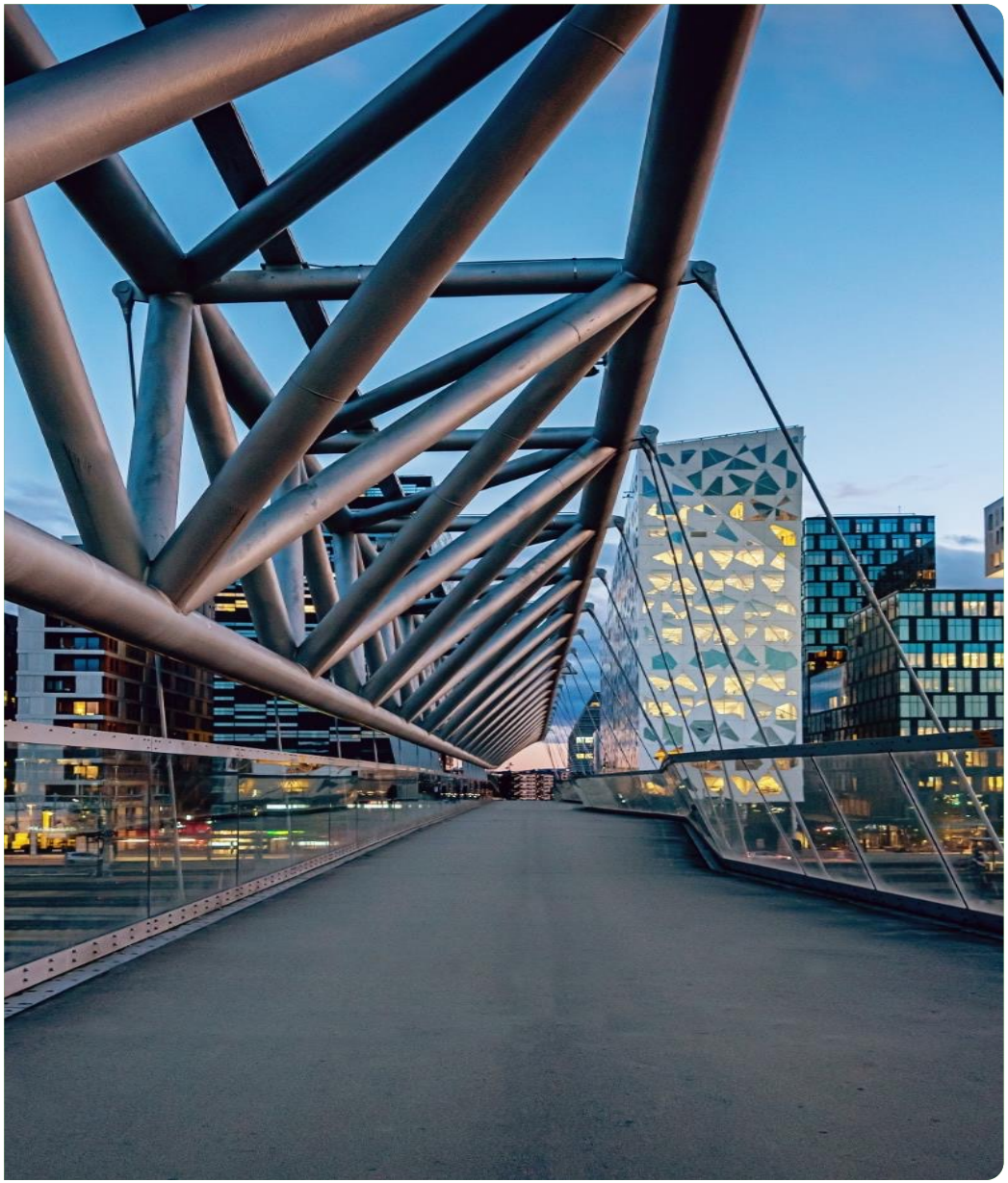
Building GenAI on AWS

Andrew Larssen
AWS technical lead @ PA Consulting

Bringing Ingenuity to Life.
paconsulting.com

01

Introduction



What is GenAI

Generative artificial intelligence (generative AI, GenAI, or GAI) is artificial intelligence capable of generating text, images, videos, or other data using generative models, often in response to prompts. Generative AI models learn the patterns and structure of their input training data and then generate new data that has similar characteristics.

– Wikipedia

AI that can **generate** new content!

Why now?

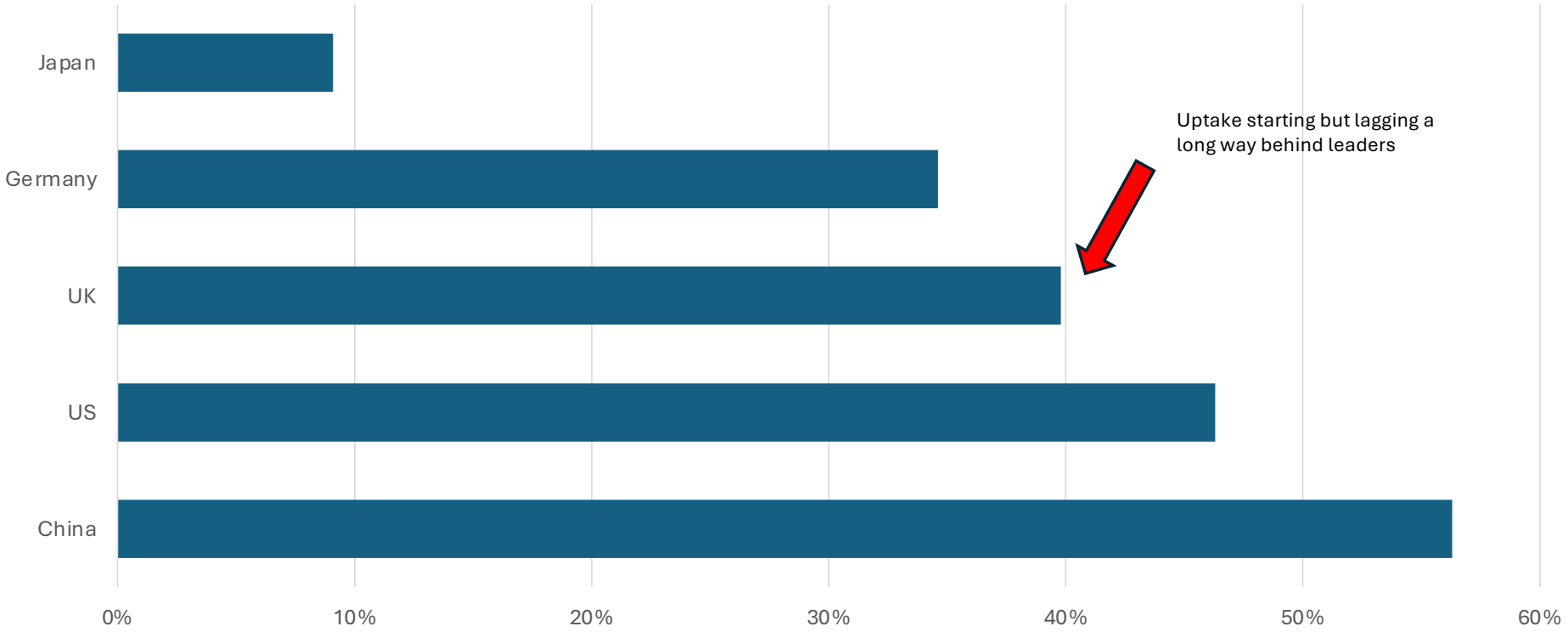
- The concept has been around since the 90’s
- 2017-18 GPT-1 Generative pretrained transformer
  - Semi supervised learning
  - Transformer architecture (quicker to train)
- 2022-23 GenAI explosion (hype)
- 2024 GenAI starting to see real use

There is still a lot of hype but there are real benefits

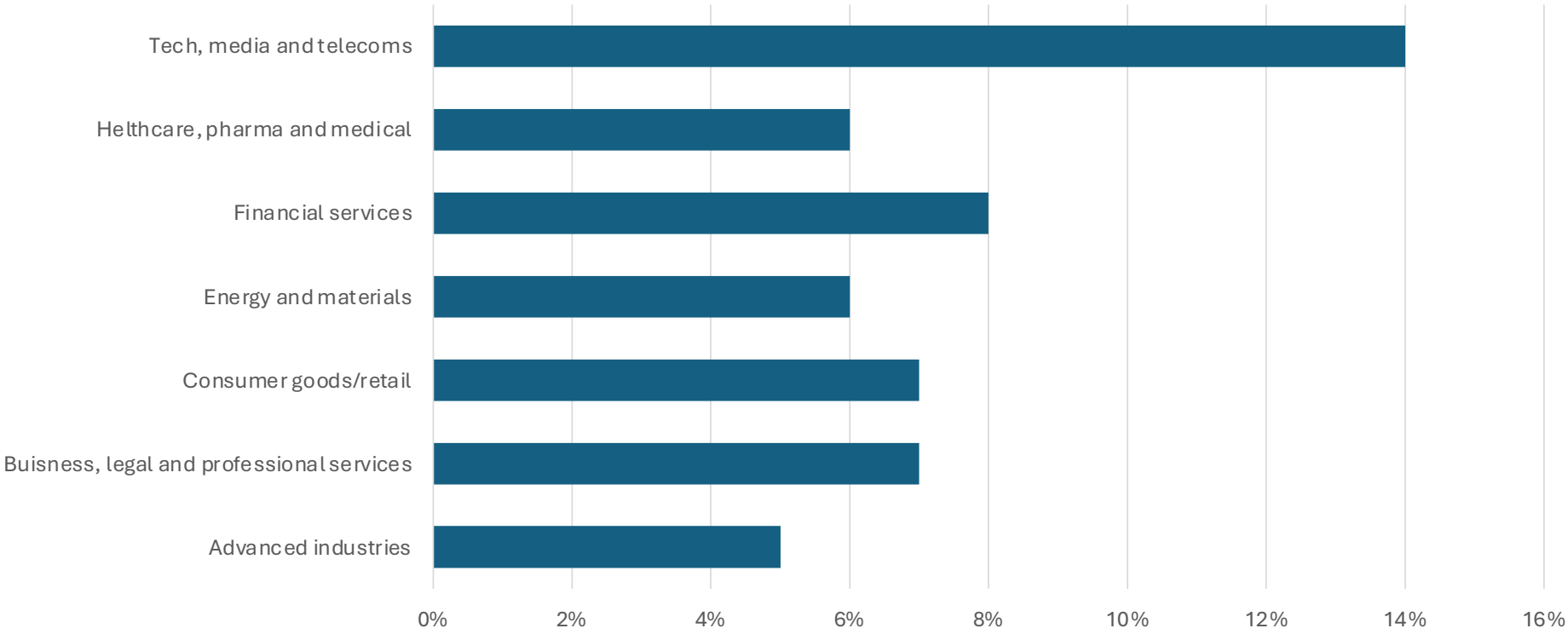
Opportunity

- Automation
- Customer insight
- Training
- Intelligent assistants (business/developers)
- Reading structured documents and graphics

GenAI adoption by country

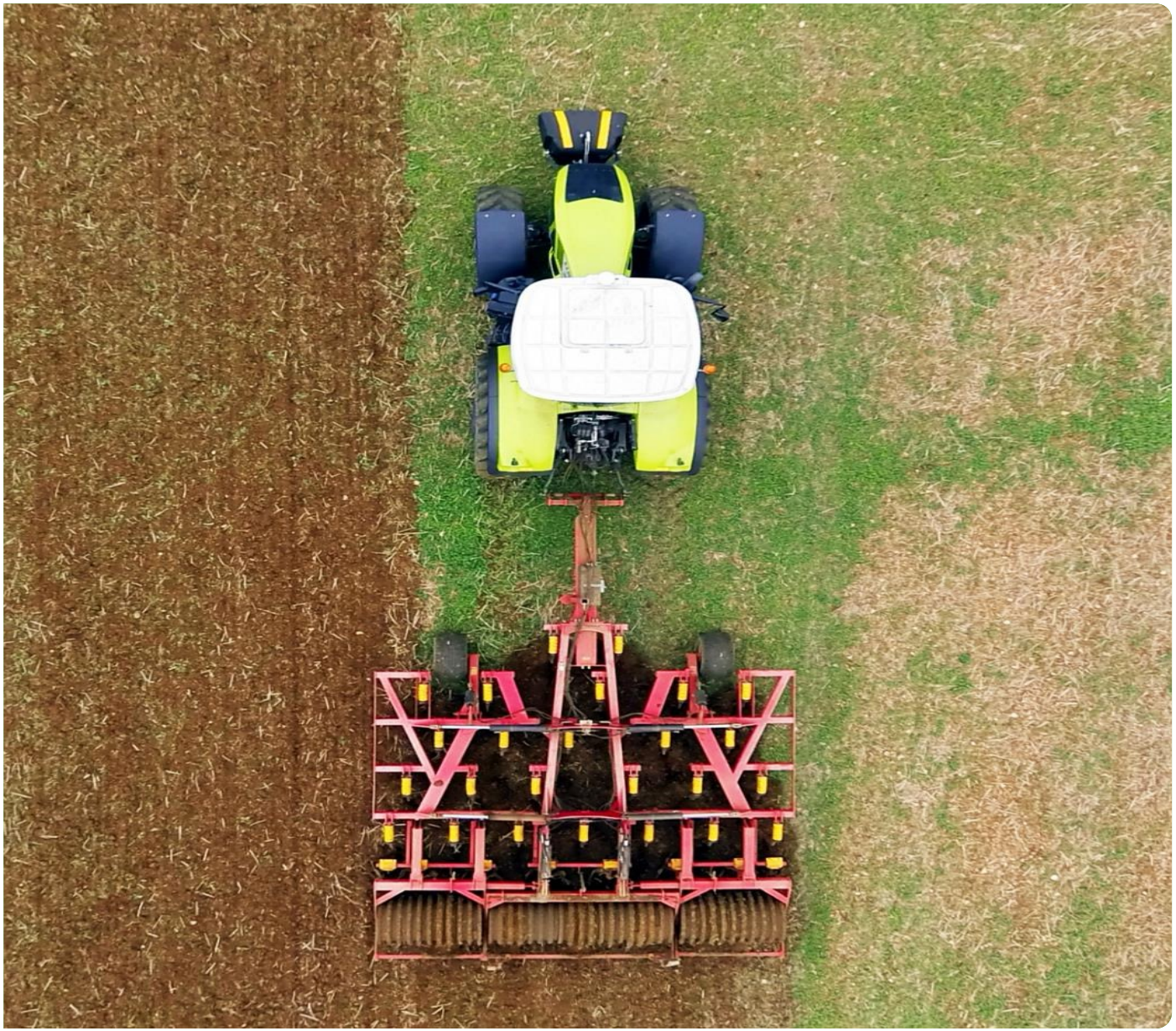


GenAI adoption by market vertical



What are PA doing?

- Genie – Our generative AI product foundation for building multiple products including RAG solutions
- Kaiwa – Train customer service representatives using GenAI
- Fraud detection
- Developer tools
- Intelligent airports



Why AWS?

- We already love AWS - Worlds biggest cloud provider (and you are here)
- Best hardware
- Model availability (Bedrock)
  - Anthorpic
  - Titan
  - Llama – and more
- The data tools I need

RAG???

Retrieval-Augmented Generation (RAG) is the process of optimizing the output of a large language model, so it references an authoritative knowledge base outside of its training data sources before generating a response. – AWS

RAG allows a model to answer your questions!

What is RAG?

- User request
- Vector search of knowledge base
- Retrieve citations
- Create prompt with original request, additional instructions and citations as context
- Inference
- Quality control
- Response

Bedrock model types

- Text generating
  - Take text (or binary files) as an input and generate text
- Image generating
  - Take text or image as an input and generate an image
- Embedding
  - Take text (or sometimes also images) as an input and generate a vector

02

Setup

Objectives

- Jupyter/Sagemaker intro
- Use Amazon Bedrock Knowledge bases
- Prompt engineering primer
- Basic steps for building a RAG product

Now onto the hands-on element

- https://github.com/pa-digital/pa-aws-genai-workshop-2024
- https://event.genie-demo.co.uk/#/login/0osRTQwgRz

https://tinyurl.com/bdkydd2x

https://tinyurl.com/222adpxb

Step 1 – Sign in

- Download the Github repo
- MAKE SURE YOU ARE SIGNED OUT OF AWS CONSOLE BEFORE YOU START
- Go to the account URL
  - Sign up and verify your email address
  - Log in to AWS console

Step 2 - Sagemaker

- Setup SageMaker
  - Search for SageMaker
  - “New to SageMaker?” >> “Set up for single user”” on RHS
  - Wait for setup
  - Create a user profile (all default options)
  - Create a space
  - Launch studio
  - Fix the IAM role
    - Click on the IAM role and add bedrock full access policy

Step 3 – S3

- Create a s3 bucket
  - Navigate to s3
  - Create bucket
  - Choose a unique name e.g. genai-comsum-andrewl
- Upload EuRegs to bucket (from git checkout)
  - Click on upload
  - Drag and drop eu_it_regs folder from git checkout

Step 4 - Bedrock

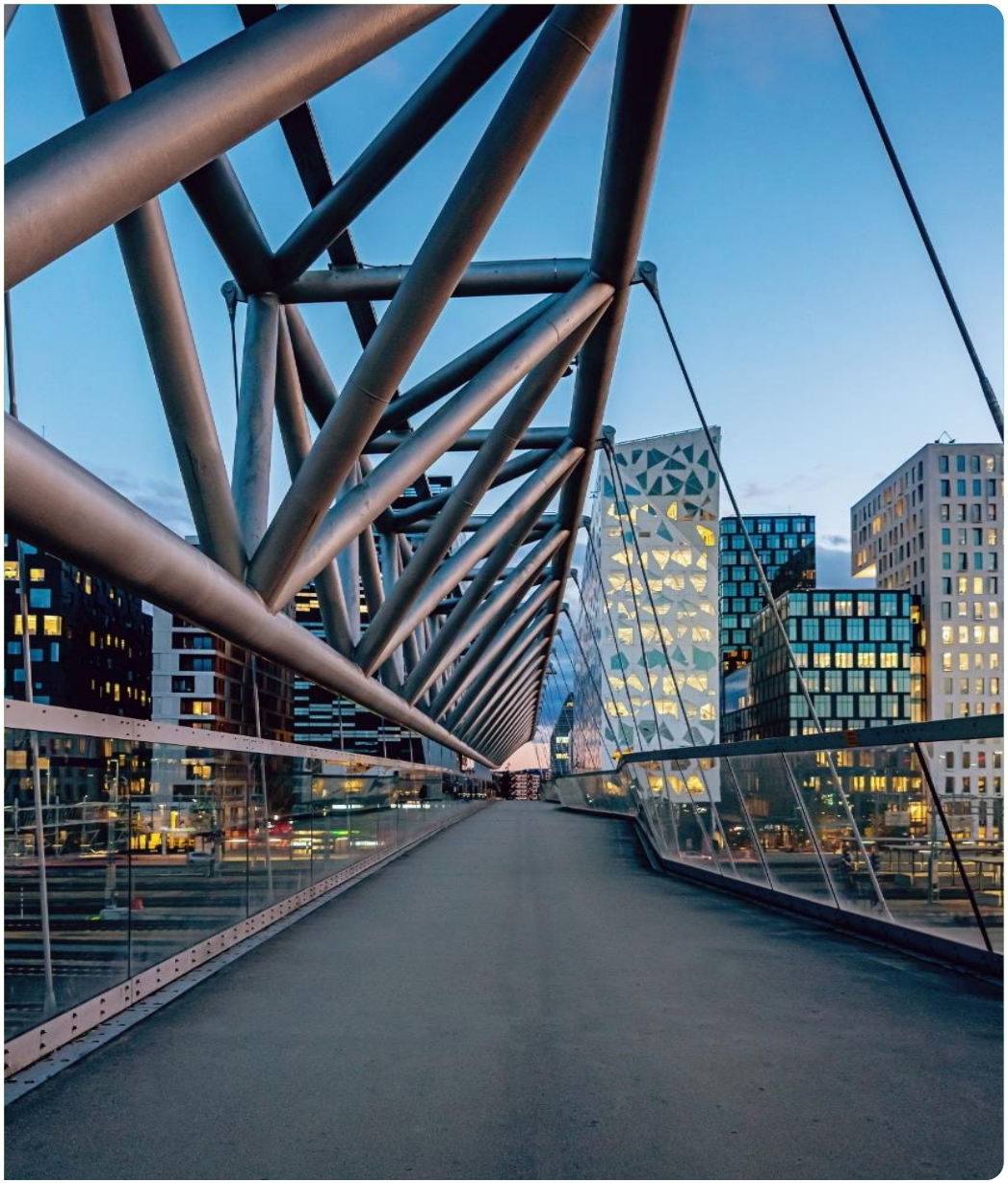
- Setup Bedrock
  - Get started
  - Enable model access (Titan embedding, Claude 3 and Cohere embedding)
  - Anthropic details
  - Submit
- Setup knowledgebase
  - Click on knowledgebase
  - Add the correct folder in your s3 bucket as the data source
  - Default chunking
  - Titan text embedding v2 with 256 vector dimension
  - Quick create a new vector store
  - Create knowledge base
  - Sync data source

What is chunking?

- With large documents you need to split them up to access the relevant sections
  - Standard (size and syntax)
  - Hierarchical (reference to parents for more context)
  - Semantic (natural language)
  - Custom
- Add metadata to chunks

03

Lets play in the console



Try some different searches

- Change the user input
- Default, hybrid, semantic search
- Query decomposition on/off
- Change the query template
- Change the temperature

04

Let’s play in SageMaker