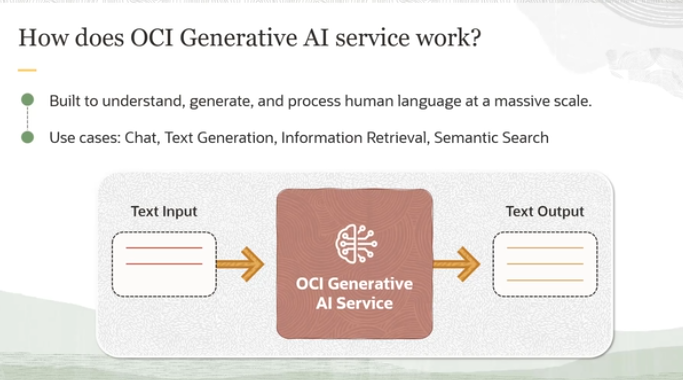


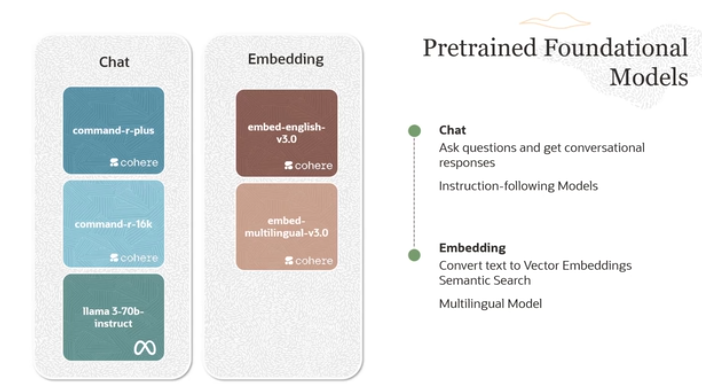
OCI generative AI service is a fully managed service that provides a set of customizable large language models available via single API to build generative AI applications. And what I mean by single API access is that you have the flexibility to use different foundational models with minimal code changes.

The service is also serverless, meaning you don't have to manage any infrastructure. There are three key characteristics of the service. The first being the service provides a choice of pre-trained foundational models from Meta and Cohere. The second is the service enables flexible fine-tuning. And you can create custom models by fine-tuning pre-trained foundational models with your own data set. And last, the service enables dedicated AI clusters. These are GPU-based compute resources that host your fine-tuning and inference workloads, and we'll go through each of them in a little bit more detail.



ow does the generative AI service really work? Well, it's you provide text input as a prompt and you get a response. You can ask questions in natural language and optionally submit text such as documents, email, product reviews, et cetera, to the generative AI service, and it reasons over the text and provides intelligent answers. So it is built to understand, generate, and process human language at a massive scale.

And some of the use cases that it unlocks are things like chat, so you could have dialogue, you could generate text. It could be used for information retrieval or even things like semantic search. We'll look into each of these in more detail.



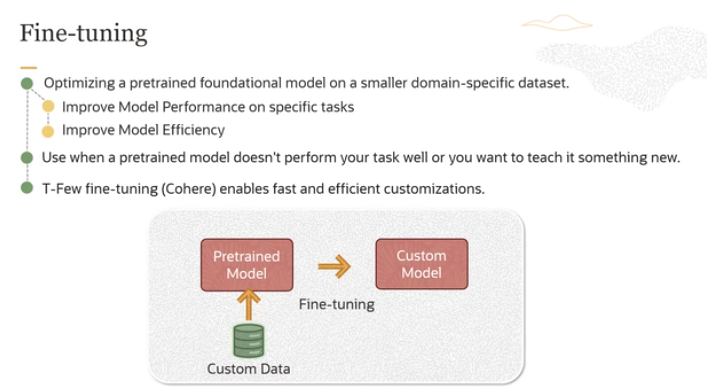
Let's break down the key points of this explanation in a simple way, focusing on two main types of **pre-trained foundational models**: **chat models** and **embedding models**.

**1. Chat Models:**

* **What are Chat Models?** Chat models are designed to engage in conversations and respond to user prompts. They remember the context of the conversation, allowing users to ask follow-up questions and maintain the flow of dialogue.
* **Examples of Chat Models**:
  + **Command-R and Command-R Plus**: These are part of the **Cohere family of large language models (LLMs)**. The difference between the two is in their power and pricing:
    - **Command-R Plus**: More advanced, handles larger requests (up to 128,000 tokens), but it’s more expensive.
    - **Command-R**: More affordable, but can handle fewer tokens (up to 16,000 tokens). It’s good for simpler use cases.
* **Instruction-Tuned Models**: Both command-r and command-r-plus are **instruction-tuned**. This means they are trained to follow human instructions well, such as generating emails, summarizing text, or answering specific questions.

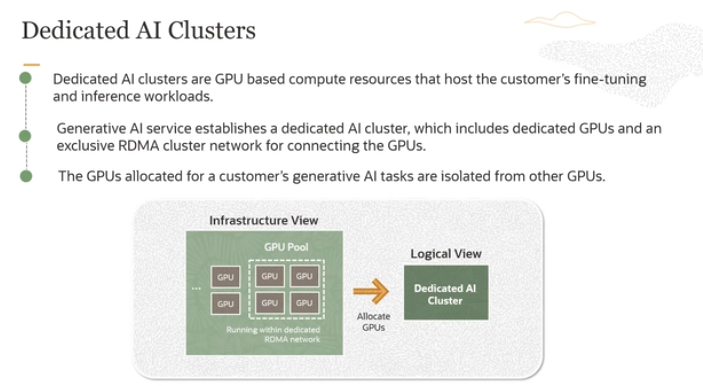
**2. Embedding Models:**

* **What are Embeddings?** Embeddings convert text into a vector of numbers, allowing computers to understand and measure relationships between pieces of text. Think of it as representing words or phrases as numerical data that the model can easily compare.
* **Uses of Embedding Models**: Embeddings are primarily used in **semantic search**, where the search focuses on the **meaning** of the text rather than just matching keywords (which is what **lexical search** does). In other words, if you search for "dog," a semantic search would also understand related terms like "puppy" or "pet" rather than just finding the word "dog."
* **Types of Embedding Models**:
  + **Embed English Model**: Focuses on English text.
  + **Embed Multilingual Model**: Works with over 100 languages. It allows searching **within the same language** (e.g., French query on French documents) or **across languages** (e.g., Chinese query on French documents).



The service also enables fine-tuning, and fine-tuning is nothing but optimizing a pre-trained foundational model on a smaller domain specific data set. So you can see, we take a pre-trained model, provide custom data, and then we end up with a custom model. And the process is called fine-tuning. There are two main benefits, fine-tuning improves model performance on specific domains, specific tasks, and it also improves model efficiency.

You use fine-tuning when a pre-trained model doesn't perform your task well or you want to teach it something new. And the service, OCI generative AI service, enables something called t-few fine-tuning, which enables fast and efficient customization. In this case, what we do is the fine-tuning method inserts-- this t-few fine-tuning method inserts new layers in the base model and selectively updates only a fraction of the models weight. Doing so, it reduces the overall training time and the cost compared to doing something like a vanilla fine-tuning, where you update all the layers or most of the layers in the base model.



**What are Dedicated AI Clusters?**

* **Dedicated AI clusters** are specialized computing setups designed to handle demanding tasks related to **AI** (Artificial Intelligence) like **fine-tuning models** or running **inference** (which means using an AI model to make predictions).
* These clusters are made up of powerful hardware, particularly **GPUs** (Graphics Processing Units), which are very good at handling the complex calculations required for AI tasks.

**Key Features of the Service:**

1. **GPU Pool**:
   * The **GPUs** in these clusters are dedicated to each customer’s specific task. This means that the GPUs being used for your work are **not shared** with anyone else. This ensures better performance and security.
2. **RDMA Cluster Networking**:
   * The GPUs in the cluster are connected through a specialized network called **RDMA (Remote Direct Memory Access)**, which allows the GPUs to communicate with each other at extremely high speeds and with **very low delay** (low latency). This is important for large-scale AI tasks that need to process a lot of data quickly.
   * This **low-latency networking** ensures that when multiple GPUs work together in a large cluster, they can exchange data very fast, making the whole process more efficient.
3. **Isolation and Security**:
   * The GPUs that are allocated for your AI tasks are **isolated**, meaning they are completely separate from GPUs being used by other customers. This adds an important layer of **security** and ensures that your data and tasks remain private.

**Why Is This Important?**

* **Dedicated resources** mean faster performance, since the GPUs are fully available for your AI tasks without sharing.
* **Low-latency networking** allows large clusters of GPUs to work together seamlessly, making it possible to handle huge AI models and tasks efficiently.
* **Security** is ensured by isolating your GPUs from others, so your AI tasks remain secure.