

Oracle Cloud Infrastructure Enterprise AI Professional

Volume I

Table of Contents

Module 1

Introduction to OCI Enterprise AI

- OCI Enterprise AI: A Flexible, Scalable, Simplified Approach One comprehensive service for building, managing, and deploying AI solutions
- Introducing OCI Enterprise AI
- Three Model Categories
- What Models Enable — All Seven Use Cases
- The Playground — Try Before You Build
- 1.10 Developer Toolkit — Your Language, Your Framework
- From Intelligence to Action — LLM vs. Agent
- Two Paths to Building Agents on OCI
- OCI Responses API — Tools, Memory, Building Blocks
- Deploying Agents — From Code to Live Endpoint
- Effective AI Safety is Multi-Faceted
- Built-in Security, Privacy, and Compliance with OCI Enterprise AI

Module 2

Introduction to Large Language Models

- What is a Large Language Model?
- This Module

LLM Architectures

- Encoders and Decoders
- Model Ontology

Prompting and Prompt Engineering

- Affecting the distribution over Vocabulary
- Prompting
- Prompt Engineering
- In-context Learning and Few-shot Prompting
- Example Prompts
- Advanced Prompting Strategies

Issues with Prompting

- Prompt Injection
- Memorization

Training

- Training
- Hardware Costs

Decoding

- Decoding
- Greedy Decoding
- Non-Deterministic Decoding
- Temperature

Hallucination

- Hallucination
- Groundedness and Attributability

LLM Applications

- Retrieval Augmented Generation
- Code Models
- Multi-Modal
- Language Agents

Module 3

Agents: Overview Part-I

- Objectives

Agents: Overview Part-II

- Summary

Agent Loop and Reasoning Cycle - I

- Objectives

Agent Loop and Reasoning Cycle - II

- A Full Loop Walkthrough: Three Turns
- Summary

Tools Part - I

- Objectives

- What Tools Are and Why Agents Need Them
- Built-in Tools vs Custom Function Tools
- Function Calling: How the LLM Decides Which Tool to Invoke
- Tool Schemas: Defining Tools the LLM Can Understand

Tools Part - II

- Tool Results: Feeding Outputs Back into the Agent Loop
- End-to-End: A Tool Call from Request to Answer
- Common Tool-Calling Pitfalls
- Code: Adding Tools to the Agent Loop
- Summary

Model Context Protocol (MCP) - I

- Objectives
- What MCP Is, Why It Exists, and Why It Matters
- The Problem MCP Solves
- After MCP: Agent Architecture
- MCP Architecture: Hosts, Clients, and Servers

Model Context Protocol (MCP) - II

- Tool Discovery: How Agents Find Tools at Run Time
- Parallel Tool Calls: When the LLM Calls Multiple Tools at Once
- MCP vs Custom Function Calling: When to Use Each
- Code: Adding MCP to the Agent Loop
- Summary

Memory: Concepts and Retrieval - I

- Objectives
- Why Stateless Agents Fail at Long-Horizon Tasks
- The Four Memory Types
- In-Context Memory: The Messages Array
- In-Context (Sessions: Persistent Working Memory)

Memory: Concepts and Retrieval - II

- Semantic Tool Memory: Retrieving Only Relevant Tools
- Episodic and Procedural Memory Patterns
- Summary

Responses API - Part I

Objectives

What is the Responses API?

Responses API: Agentic by Design

Memory Without the Messages Array

Built-in Tools: Zero Schema Required

Responses API - Part II

Streaming and Structured Outputs

Responses API – The Same Agent, Less Code

Summary

Module 4

Enterprise AI Models

Objectives

2.1 Three Model Categories — Chat, Embed, Rerank

2.2 Five Provider Families — Design Philosophy and Enterprise Strength

2.3 On-Demand vs. Dedicated AI Cluster — Choosing Your Serving Mode

2.4 Model Lifecycle — Deprecation, Retirement, and Long-Term Support

Summary

Offered and Imported Models

Objectives

3.1 Cohere Chat Models — Enterprise Reasoning & RAG

3.2 Chat Model Catalog — Meta, Google, xAI, OpenAI

3.3 Embed & Rerank Models — The Intelligence Behind Search

3.4a Model Selection — Criterion 1: Task Type

3.4c Model Selection — Criteria 6–7: Customization & Serving Mode

3.5 Model Import — Bring Any Supported Model to OCI

3.6 Model Import Workflow — From Source to Live Endpoint

Summary

Customizing Models with Your Data

Objectives

4.1a Why Fine-Tune? — When Prompting and RAG Are Not Enough

- 4.1b Training Data — Format, Rules, and Storage
- 4.2 Fine-Tunable Models & the T-Few Method
 - 4.2b Fine-Tuning Workflow — Dataset to Production Endpoint (7 Steps)
- 4.3 Hyperparameter Configuration — The Four Controls
- Summary

Infrastructure: Clusters, Endpoints, & Private Access

- Objectives
- 5.1a Dedicated AI Clusters — Two Types, Unit Sizing, and Hour-Based Billing
- 5.1b Performance Benchmarks — Sizing for Production SLAs
- 5.2a Creating a Fine-Tuning Cluster — Step by Step
- 5.2b Creating a Hosting Cluster and Configuring an Endpoint
- 5.3 Private Endpoints — Zero Public Internet Model Access
- Summary

Module 5

Enterprise AI Agents: Overview

- Objectives
- OCI Enterprise AI Agents: Platform Overview
- The OCI Responses API
- Agent Tools
- Agent Memory
- Low-Level Agentic Building Blocks
- Platform Architecture: Build and Deploy Together
- Getting Started: Required Resources
- Summary

OCI Responses API vs Chat Completion API

- Objectives
- The OCI OpenAI-Compatible Package
- OCI Responses API vs Chat Completions API
- Agentic Inference: Supported Models
- Authentication: Two Methods
- Regions and Endpoint Reference

Summary

OCI Responses API: Tools

Objectives

OCI Responses API: Four Tool Types

Tool 1: File Search

Tool 2: Code Interpreter

Tool 3: Remote MCP Calling

Tool 4: Function Tools

Choosing the Right Tool

Summary

OCI Responses API: MCP Tools

Objectives

What is MCP and How does it work in OCI?

Key Features of OCI MCP Tool Support

Declaring MCP Tools: Field-by-Field Reference

Tool Filtering: Controlling Cost and Latency

MCP Authentication: OAuth and Security

Hosting MCP Servers on OCI

Streaming Responses and Tracing MCP Calls

Summary

OCI Responses API: Agent Memory

Objectives

Three Memory Mechanisms at a Glance

Projects: The Foundation for Agent Memory

Short-Term Memory: Responses Chaining

Short-Term Memory: Conversations API

Long-Term Memory: Across Conversations

Short-Term Memory Compaction

Memory Architecture: Putting It All Together

Summary

OCI Responses API: Vector Stores

Objectives

What are Vector Stores?

Control Plane vs Data Plane Clients

Step 1: Create a Vector Store

Step 2: Add Files to the Vector Store

Step 3: Search the Vector Store

Vector Store Connector: Production Scale

End-to-End RAG Pipeline Reference

Summary

Oracle Cloud Infrastructure Enterprise AI Professional

Volume II

Table of Contents

Module 6

Hosted Agents Overview

- Objectives
- What Are Hosted Agents?
- How It Works: End-to-End Architecture
- Deployment Configuration Options
- Generative AI Applications and Deployments
- Invoking the Hosted Agent Endpoint
- Summary

Build and Deploy Hosted Agents

- Objectives
- The Complete Build and Deploy Pipeline
- Prerequisites Before You Deploy
- Steps 1 & 2: Develop and Containerize Your Agent
- Step 4: Create a Generative AI Application
- Summary

Module 7

OCI Generative AI: Enterprise Governance

- Objectives
- How Governance Layers Work Together
- Why Governance Matters in AI Deployments
- Network Layer: Private Endpoints and ZPR
- Zero Trust Packet Routing: How It Works
- Identity Layer: IAM Policies
- IAM: Generative AI Resource Types
- AI Behavior Layer: Guardrails
- Summary

Data Handling and Network Security

- Objectives

What data does OCI Generative AI handle?

Data Handling Commitments: What Oracle Does and Does Not Do

Encryption: At Rest and In Transit

Customer Controls: Deletion and Key Management

Network Security: Private Endpoints

Zero Trust Packet Routing: Intent-Based Network Policy

Private Endpoints and ZPR: Layered Network Defense

Summary

API Keys for Generative AI

Objectives

What is an OCI Generative AI API key?

The Two-Secret Design: Zero-Downtime Rotation

IAM Permissions for API Keys

Using API Keys with the OpenAI SDK

Summary

IAM Policies and Authentication

Objectives

IAM Policy Syntax: How Subject, Verb, Resource, and Location Connect

IAM Policy Syntax: The Four Verbs

generative-ai-family: The Aggregate Resource Type

Individual Resource Types: generative-ai-family

Fine-Tuning and Object Storage Policies: Who Needs What

Agent Resource Permissions

Permissions for Deploying Applications

Authentication Methods: API Keys vs. IAM

Summary

Observability

Objectives

The Observability Stack: What OCI Provides

Responses API: The Built-In Trace - output Array

Debug Logging: Capturing opc-request-id and Raw HTTP Calls

Third-Party Tracing: Langfuse Integration

OCI Metrics for Generative AI: Clusters and Endpoints

Summary

Calculating Cost in Generative AI

Objectives

Two Pricing Models: On-Demand vs. Dedicated AI Clusters

On-Demand: Pricing Tiers by Model Family

Search and Retrieval Pricing

Worked Example: Calculating On-Demand Cost

Dedicated AI Clusters: Unit-Hours, Commitments, and Multipliers

Worked Example: Calculating Dedicated AI Cluster Cost

OCI Cost Estimator: AI and Machine Learning Category

Summary