Lab 2: Model Providers

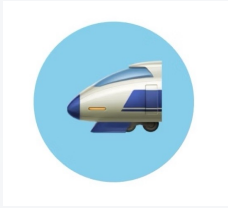
Learning Objectives

- What the ModelProvider abstraction is and why it matters
- Connecting to Anthropic's Claude models directly
- Setting up LiteLLM integration
- How to configure other models with Amazon Bedrock
- Best practices for choosing the right provider

Key Benefits

- **Provider flexibility** - Switch between cloud and local models
- **Consistent API** - Same code works across providers
- **Easy experimentation** - Test different models without rewriting logic
- **Production adaptability** - Choose optimal providers for different environments

LiteLLM



- Gateway to provide model access across 100+ LLMs
- All in the OpenAI format
- Models from providers like Databricks, Groq, ElevenLabs and others

Ollama

- Free to use
- Runs entirely offline
- Great for experimentation
- No API costs during development

AWS Credentials Setup

- Environment variables (AWS_ACCESS_KEY_ID, AWS_SECRET_ACCESS_KEY)
- AWS CLI configuration
- IAM roles (recommended for production)

Choosing the Right Provider

✓ **Use Amazon Bedrock when:**

You're already in the AWS ecosystem
You need enterprise security and compliance
You want access to multiple model providers through one API
You need managed scaling and reliability

✓ **Use Anthropic directly when:**

You want the latest Claude models first
You need specialized Claude features
You're building applications specifically around Claude's capabilities

✓ **Use Ollama for:**

Local development and testing
Privacy-sensitive applications
Offline requirements
Cost-free experimentation

Choosing the Right Provider (Cont.)

- **Cost**: Local models are free but require hardware; cloud models have usage-based pricing
- **Latency**: Local models eliminate network overhead
- **Capabilities**: Different models excel at different tasks
- **Reliability**: Managed services offer SLAs and uptime guarantees

Key Takeaways

🔑 Key Concepts

ModelProvider Abstraction: Universal interface that lets you switch between LLM providers without changing your agent code
Provider Flexibility: Same agent logic works across Anthropic, OpenAI, Amazon Bedrock, and more

🛠️ Implementation Highlights

Anthropic Direct: Access latest Claude models first with direct API integration
LiteLLM: Gateway to hundreds of specialized models from various providers
Ollama: Local development option - free, offline, great for experimentation
Amazon Bedrock: Default provider with Claude 4 Sonnet - managed, enterprise-ready

💡 Best Practices

Security: Always use environment variables for API keys, never hardcode
Provider Selection: Choose based on use case, cost, latency, and reliability needs
Easy Switching: Install provider-specific packages: strands-agents[anthropic], strands-agents[litellm]