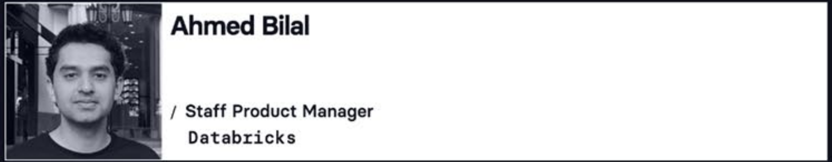
DATA+AI SUMMIT
BY databricks

Simplify GenAI App Development with secure and custom **AI agents**

Aakrati Talati

/ Senior Software Engineer
Databricks



Ahmed Bilal

/ Staff Product Manager
Databricks

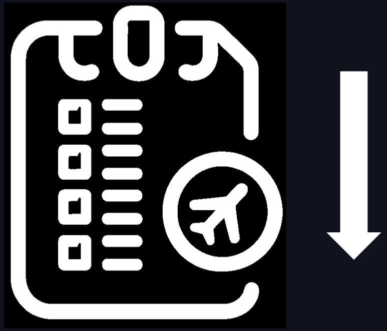
Agenda

- Overview of AI Agents
- Benefits of AI Agents
- Core Components of AI Agents
- Challenges with AI Agents
- Deploying and Governing AI Agents on Databricks
- Best Practices
- Future Developments

What are AI Agents?

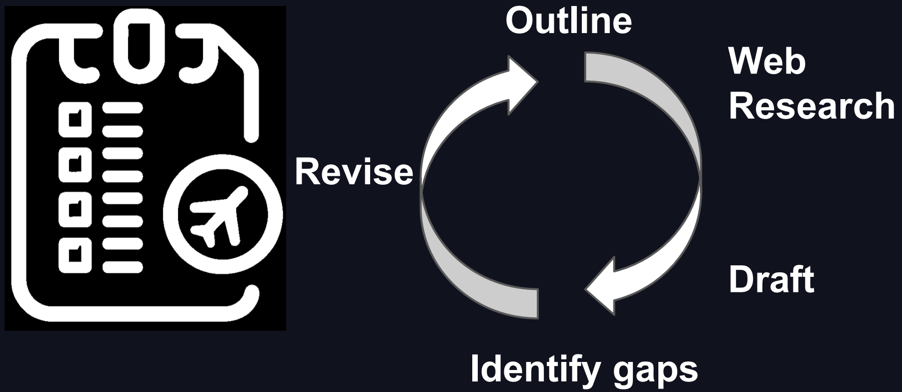
Prompt: Plan my next vacation

Non-Agentic



Writes the itinerary in one go

Agentic



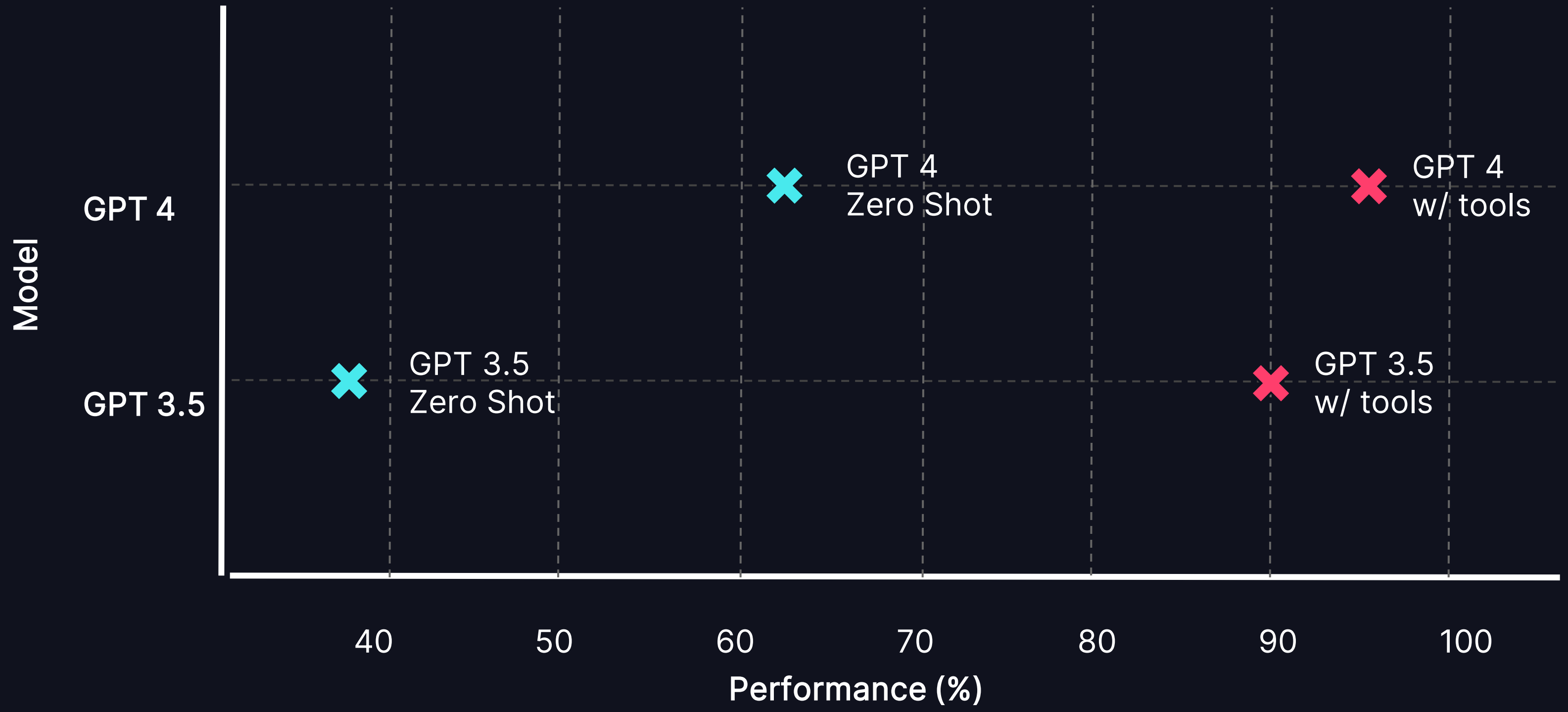
Inspired from Andrew Ng talk on 'What's next for AI agentic workflows' at Sequoia Capital's AI Ascent

Demo:
Hello Agents!

Why AI Agents?

Ship AI Apps faster with Agentic Workflows

Agents Significantly Improve Quality



Source: Coding benchmark by Andrew Ng, Joaquin Dominguez and John Santerre

Agents Simplify Access to Enterprise Data

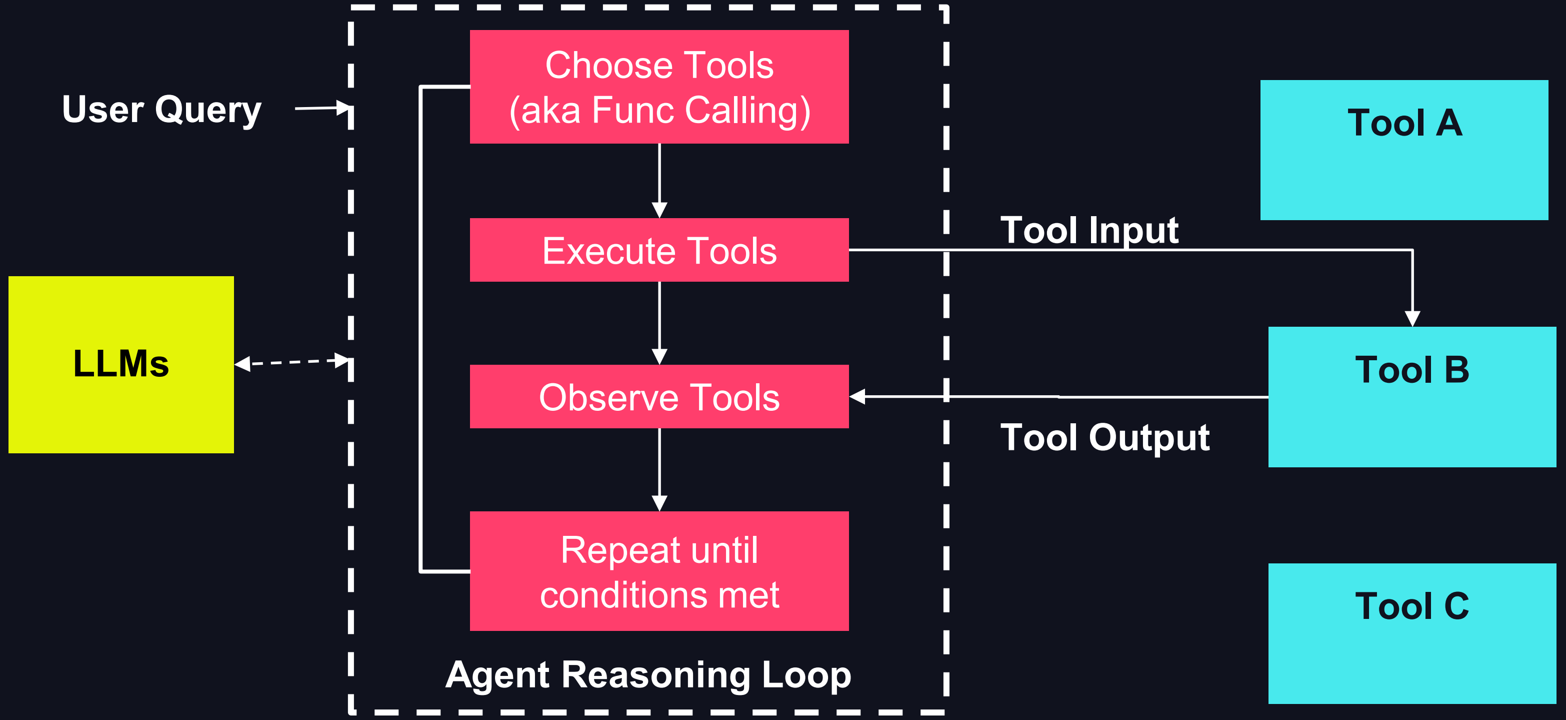
- Easily Access Company Data Sources and Systems
- Compared to Chains, Agents can
  - Automatically Recover from Errors
  - Execute Multi-Hop Prompts

Data Stores: Vector Search, SQL Warehouse, Online Tables

Company Systems: Search, Internal or External API services

Computations: Calculators, Code

Components of AI Agents



Challenges with AI Agents

Ensuring Consistent Quality is Difficult

Deploying Agents Requires Disparate Tools

Governing Agent Workflows is Complex

Agentic Capabilities on Databricks

Improve Quality with Function Calling and Agent Evaluation

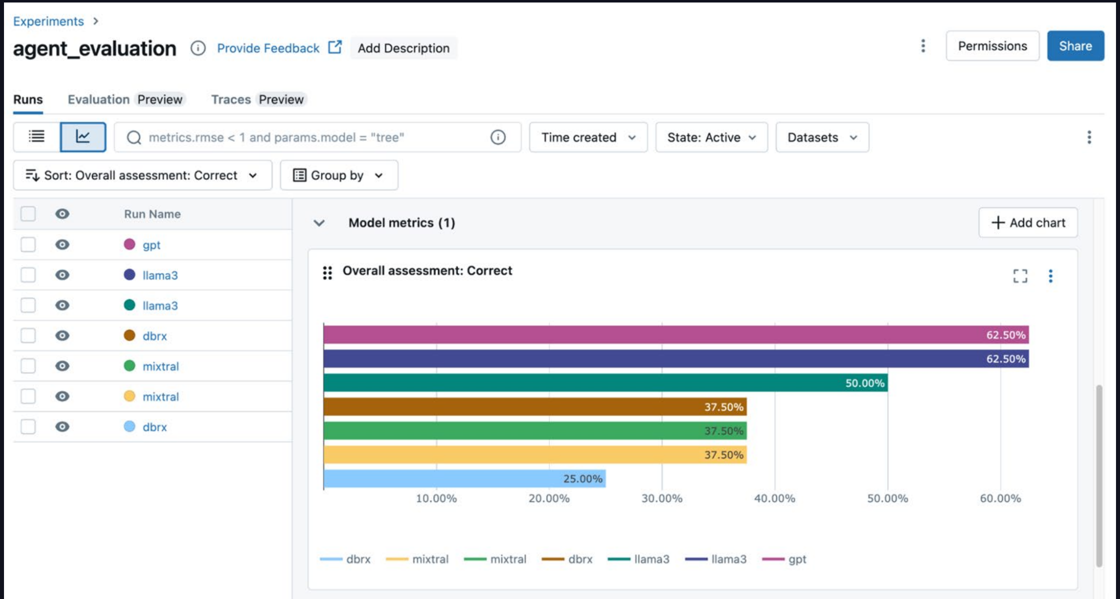
Deploy Agents with Agent Framework and Unified Serving

Govern with Tools Catalog

Improve Quality with Function Calling and Agent Evaluation

- NEW! Perform **Function Calling** with OSS or proprietary models with unified API.

- NEW!: **Agent Evaluation:** Evaluate agents with experts, utilize LLM judges, debug with traces.



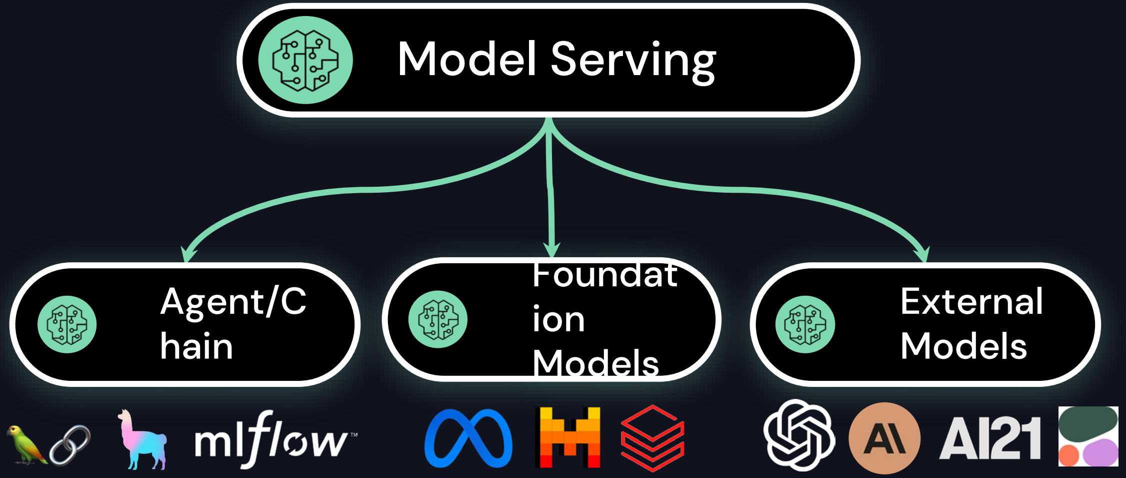
metrics.rmse < 1 and params.model = "tree"

Sort: Overall assessment: Correct

Run Name
gpt
llama3
llama3
dbrx
mixtral
mixtral
dbrx

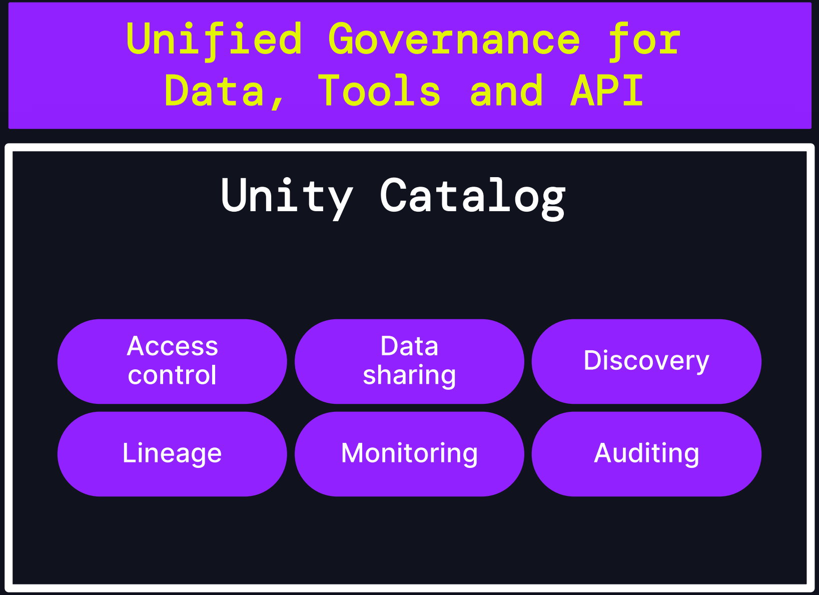
Deploy Agents with Agent Framework and Unified Serving

- Deploy any AI entities: traditional, GenAI, agents.
- NEW! Agent Framework: simplified deployment, streaming and tracing.
- NEW! Securely access data stores from serving endpoints.



UC & AI Gateway for Governing Data, Tools, APIs

- Unified governance for data, models, APIs via UC and AI Gateway.
- NEW! AI Tools Catalog: Share and deploy tools securely.
- NEW! AI Gateway: guardrails, logging, usage tracking across endpoints.



Unified Governance for Data, Tools and API

Unity Catalog

Access control

Data sharing

Discovery

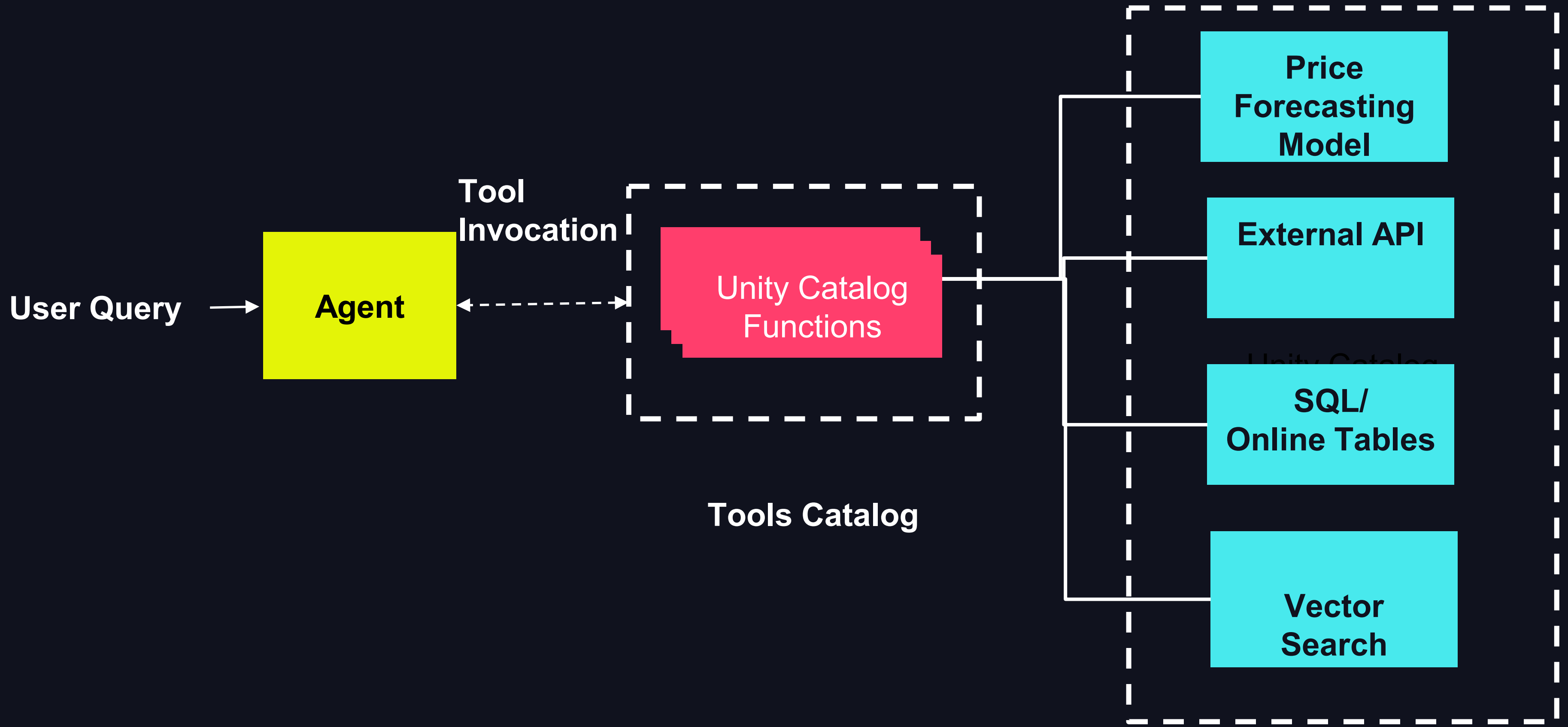
Lineage

Monitoring

Auditing

Travel Bot Demo: Deploying Agents on Databricks

Travel Bot

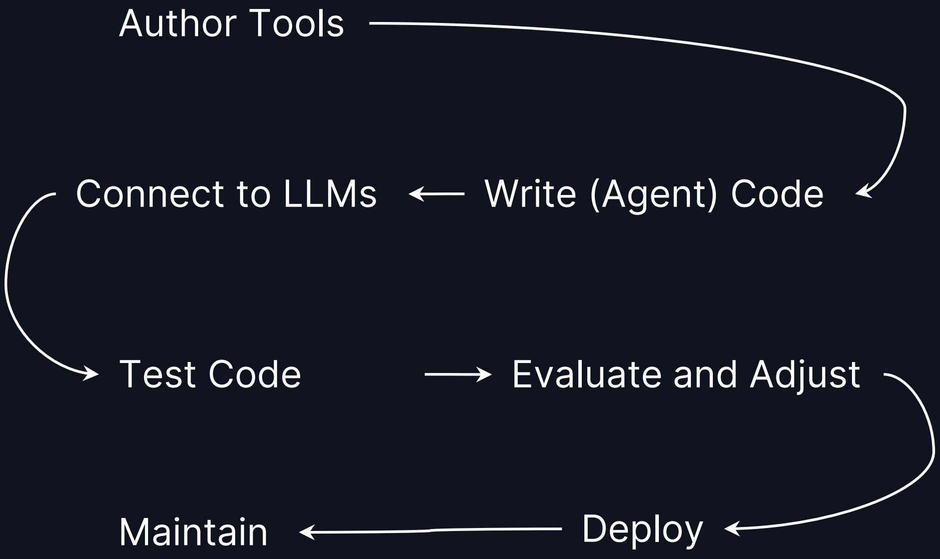

Enterprise Sources
User Query
Agent
Tool Invocation
Unity Catalog Functions
Tools Catalog
Price Forecasting Model
External API
Unity Catalog
SQL/ Online Tables
Vector Search

Demo Recap

- Unified catalog for all Data Sources and Tools!
  - Easy sharing of functions from UC Tools Catalog
- Low maintenance and reduced operational overhead!
  - Managed Credentials
  - Unified API
  - Operational monitoring and metrics
- Built-in tracing and payload logging!

Future Agent Roadmap

Current Agent DevEx



Author Tools

Write (Agent) Code

Connect to LLMs

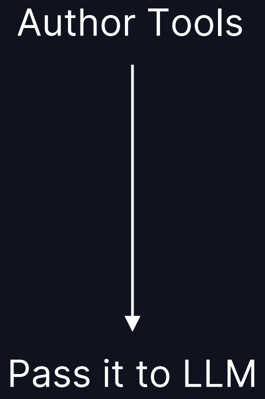
Test Code

Evaluate and Adjust

Deploy

Maintain

Hosted Agent Capabilities



Author Tools

Pass it to LLM

Sneak Preview: Hosted Agents Capabilities

Summary

- Agent open up new GenAI use cases by improving quality and making it easy for LLM to talk to enterprise data sources and systems

- Start deploying agentic workflows on Databricks with new capabilities: Function Calling, Agent Framework, Tools Catalog.

- Sign up for the private preview of hosted agent capabilities.

Hosted Agent sign-up form

End of Slides

Best Practices

Tool Usage

- Provide clear tool prompts with descriptions,
- Return data details, arguments list, and example values.
- Make inputs optional when they can be inferred.

Error Handling

- Detect and prompt for agent errors.
- Use simple, deterministic functions for common tasks, and ensure proper error handling to keep agents on track.

Fine-Tuning

- Fine-tune models to improve tool usage quality and use return values to guide the agent’s subsequent actions.