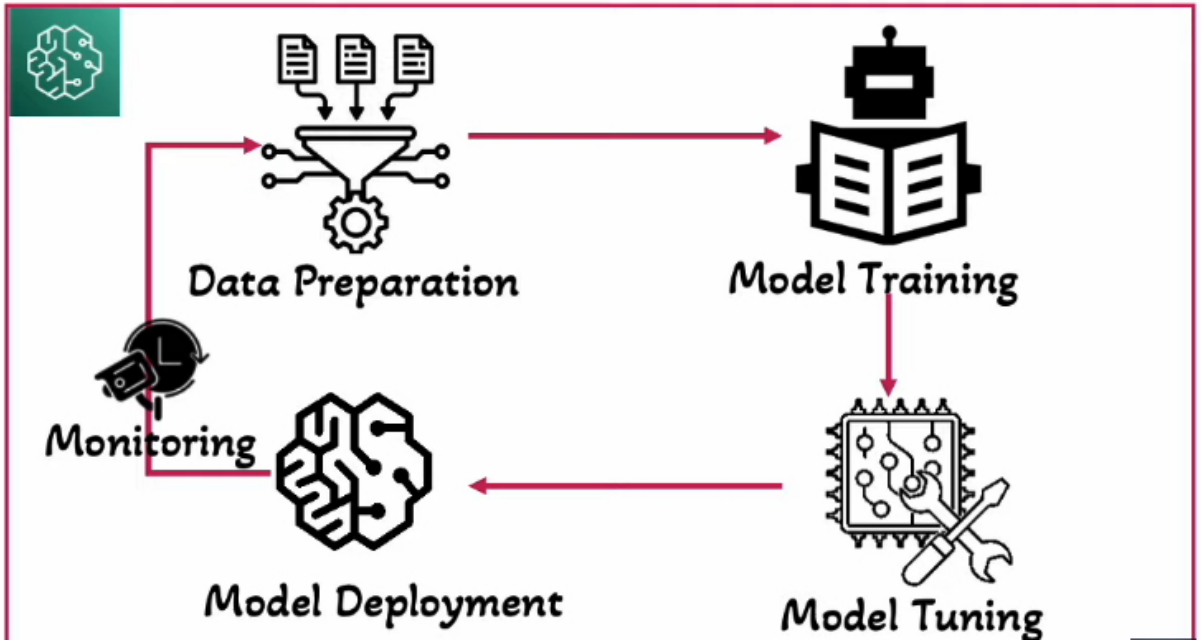
Amazon SageMaker

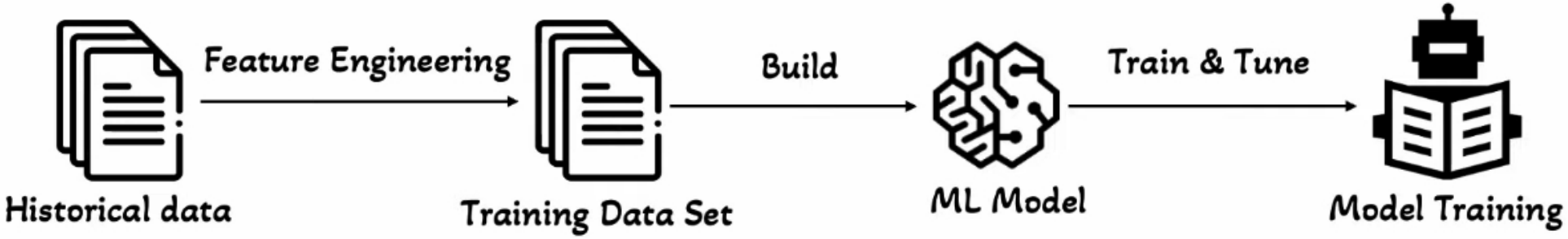
- Fully managed service for developer and data scientists to build, train and deploy ML models.
- Suitable users with varying levels of expertise.
- Provides comprehensive set of tools for end-to-end ML:
  - Data Preparation
  - Model Training
  - Model Tuning
  - Model Deployment
  - Model Monitoring



- Build end-to-end ML workflows

Amazon SageMaker ML model prediction

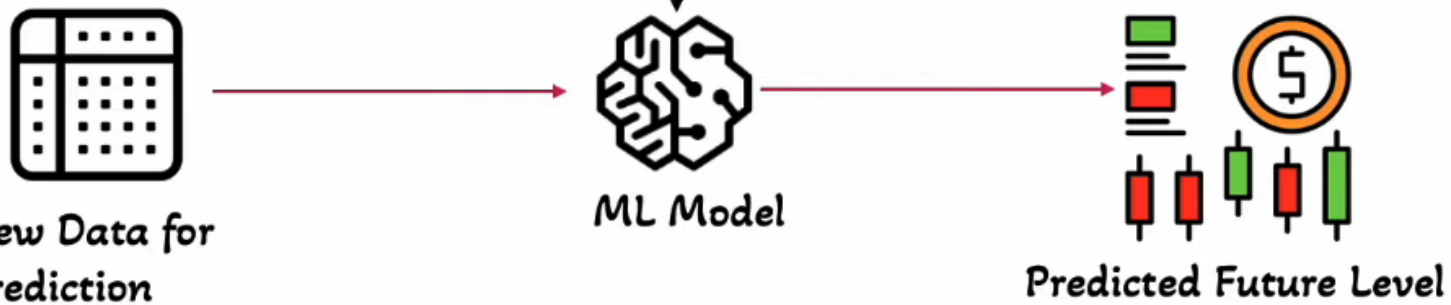
Predict the future price trends of Apple stock in NASDAQ



- Historical stock Prices
- Technical Indicators
- Market Data
- Macroeconomic Indicators

Model Ready for Prediction

- Date – March 31, 2025
- RSI – 45.12
- NASDAQ – 19,246



Amazon SageMaker Tools Suite

- SageMaker: Build, train, and deploy ML models.
- SageMaker Automatic Model Tuning: Optimize model hyperparameters automatically.
- SageMaker Deployment & Inference: Deploy models for real-time predictions.
- SageMaker Studio: Integrated development environment for ML.
- SageMaker Feature Store: Centralized repository for ML features.
- SageMaker Clarify: Detect bias and explain predictions.
- SageMaker Model Cards: Document model details and performance.
- SageMaker Ground Truth: RLHF, for model grading and data labeling
- SageMaker Data Wrangler: Simplify data preparation for ML.

Amazon SageMaker Tools Suite

- **SageMaker Model Dashboard:** Monitor and manage deployed models.
- **SageMaker Model Monitor:** Continuously monitor model performance.
- **SageMaker Model Registry:** Store and version ML models.
- **SageMaker Pipelines:** Automate and manage ML CI/CD pipeline.
- **SageMaker Role Manager:** Manage permissions for SageMaker users.
- **SageMaker JumpStart:** Pre-built solutions and models.
- **SageMaker Canvas:** No-code ML model building.
- **MLFlow on SageMaker:** Manage ML lifecycle with MLFlow.

Amazon SageMaker Built-in Algorithms

Domain	Problem Types	Input Format	Build-in Algorithm
Supervised Learning	Binary/Multi-class classification or Regression	Tabular	k-nearest neighbors (kNN) Linear Learner Algorithm XGBoot
Supervised Learning	Time-series Forecasting	Tabular	DeepAR Forecasting
Unsupervised Learning	FE: Ddimensionality reduction	Tabular	Principal Component Analysis (PCA) Algorithm
	Anomaly detection	Tabular	Random Cut Forest (RCF) Algorithm
	Clustering or grouping	Tabular	K-Means Algorithm
Textual Analysis	Text Summarization Speech-to-text	Text	Sequence-to-Sequence Algorithm
Image Processing	Image Classification	Image	Image Classification – Tensor Flow
	Object Detection	Image	MXNet, OD-Tensor Flow
	Computer Vision	Image	Semantic Segmentation Algorithm

SageMaker Automated Model Tuning (AMT)

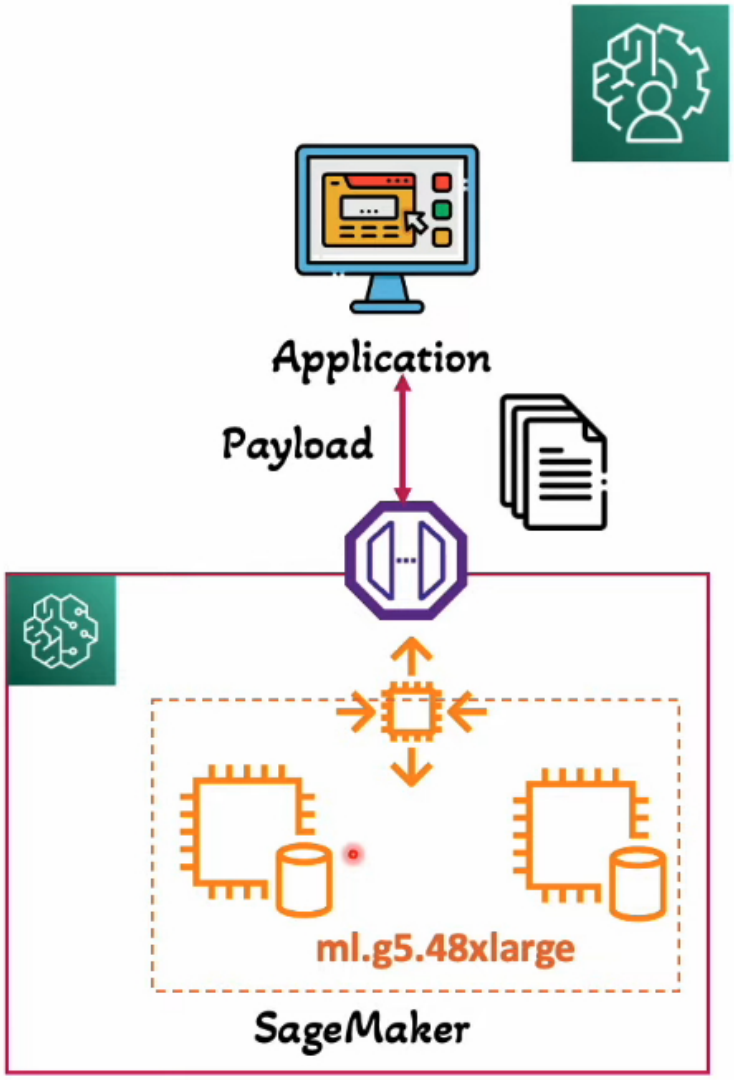
- Automatic model tuning, → hyperparameter optimization
- Hyperparameter settings control the training process and impact model performance
- SageMaker automatically tunes models by running multiple training jobs with different hyperparameter values
    - Define the hyperparameters to tune and their ranges.
    - Specify the objective metric to optimize (e.g., accuracy, loss).
    - SageMaker will run multiple training jobs and find the best hyperparameters.
- Amazon SageMaker takes care the heavy-lifting of Model Tuning.

SageMaker Model Deployment & Inferences

- Options for Deploying ML Models:
  - Real-time Inference
  - Batch Inference
  - Asynchronous Inference
  - Serverless Inference

Real-Time Inference

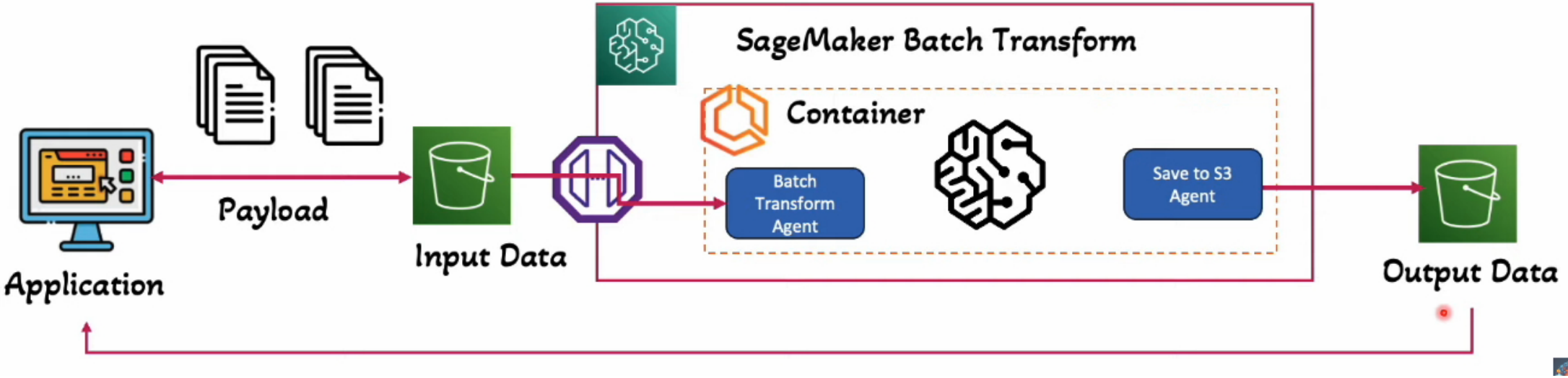
- Real-time endpoints for low-latency predictions.
- Ideal for chatbots and recommendations
- One prediction at a time
- Payload up to 6MB
- Select Instance Type (Cost/hour or Best Performance)



SageMaker Model Deployment & Inference

Batch Inference

- Useful for large dataset
- Store the input data in Amazon S3 (format can be CSV, JSON)
- Configure the batch job with input/output data location and instance type
- SageMaker launches the specified instance and process data in batches
- Retrieve output data from the S3 bucket
- No Persistent endpoint, hence cost effective

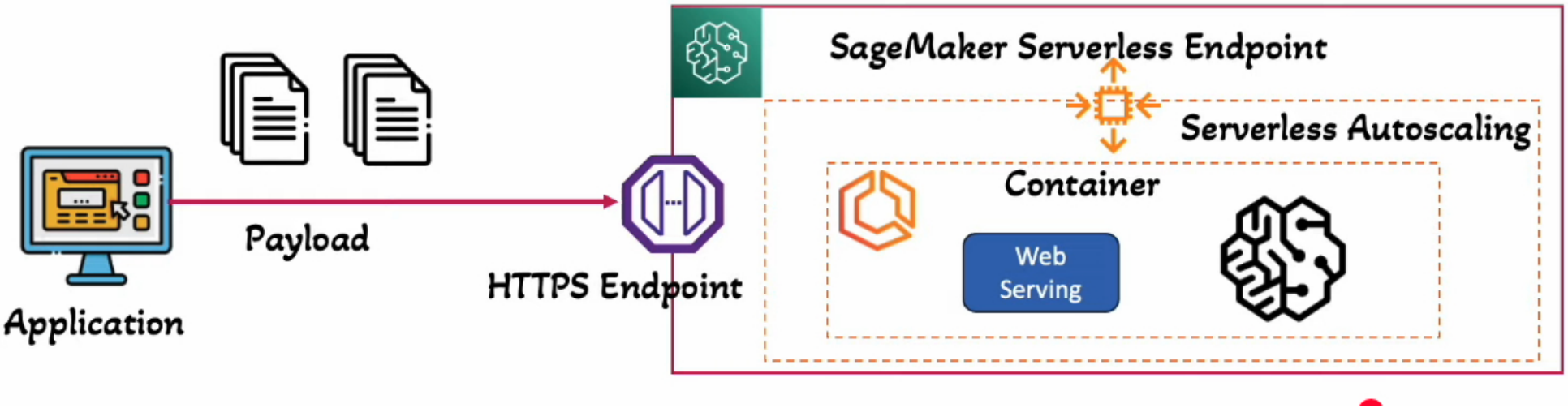


1.00

SageMaker Model Deployment & Inference

Serverless Inference

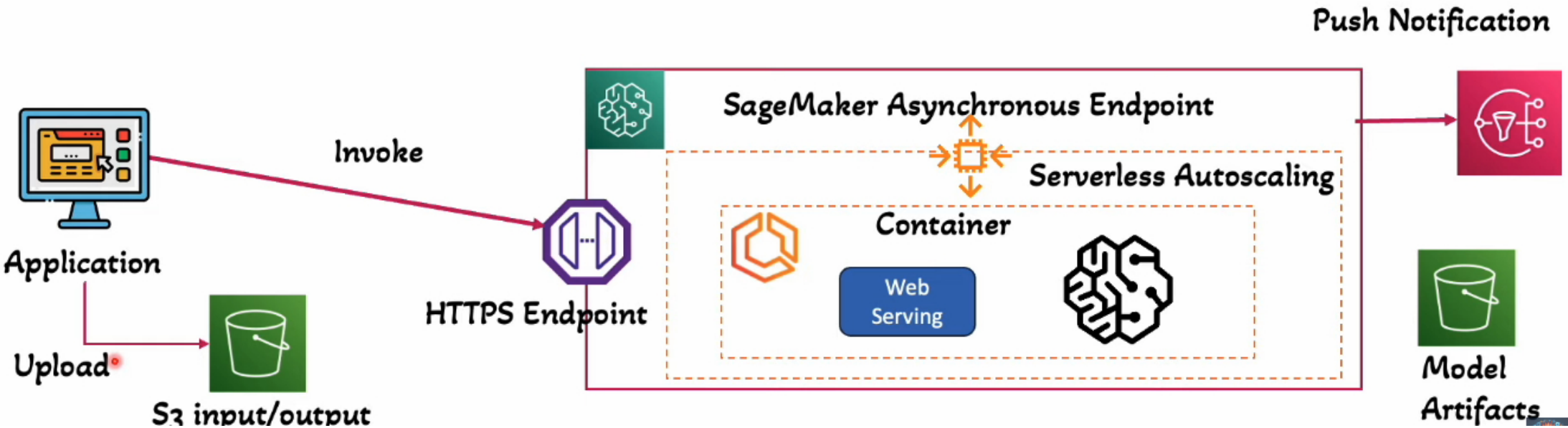
- Deploy models without managing underline infrastructure
- Configure endpoint with memory size and max concurrent invocation.
- SageMaker automatically provision necessary compute resources
- During idle periods, endpoints scales down to zero
- Ideal for unpredictable prediction traffic
- Workload tolerable to clod start (high-latency)



SageMaker Model Deployment & Inference

Asynchronous Inference

- Designed to process large payload or long processing times.
- SageMaker queues the request for processing and returns identifier & output loc
- Suitable for near real-time latency requirements.
- Ideal for large payload up to 1GB and processing time up to 15 min
- Autoscaling to zero when there are no requests, reducing costs.



SageMaker Model Deployment Comparison

Inference Type	Latency	Payload Size	Processing Time	Use Case
Real-time Inference	Low (milisec to seconds)	Up to 6 MB (One record)	Max 60 seconds	Fast, instance predictions for web/mobile apps and chatbots
Batch Transformation	High (mins to Hours)	Up to 100 MB per invocation	Max 1 Hour	Bulk processing for large datasets concurrent processing
Serverless Transformation	Low (milisec to seconds)	Up to 6 MB (One record)	Max 60 seconds	Sporadic, short-term inference without infrastructure, can tolerate cold start
Asynchronous Inference	Medium to high (near real-time)	Up to 1 GB(One record)	Max 1 Hour	Large payloads and workloads requiring longer processing time.

SageMaker Studio

- Integrated development environment (IDE) for machine learning.
- Unified interface for building, training and deploying models
- Prerequisite – Availability of SageMaker Domain
- Supports multiple IDEs, including JupyterLab, RStudio
- Supports end-to-end ML model development.
  - Deploy, tune and debug ML models
- Supports scalability for underling computing resources
- Use Cases:
  - Data Exploration and Preparation
  - Model Training and Tuning
  - Model Deployment and Monitoring

SageMaker Feature Store

What is Feature in ML?

We want to predict House Price

Factors to determine house price:

- House size (in Square Foot)
- Number of Bedrooms
- House Location
- Year of Built

} Feature

Features are input to ML models.

Features provides information that the ML model used to make prediction.

High-quality features are important for high-quality prediction

SageMaker Feature Store

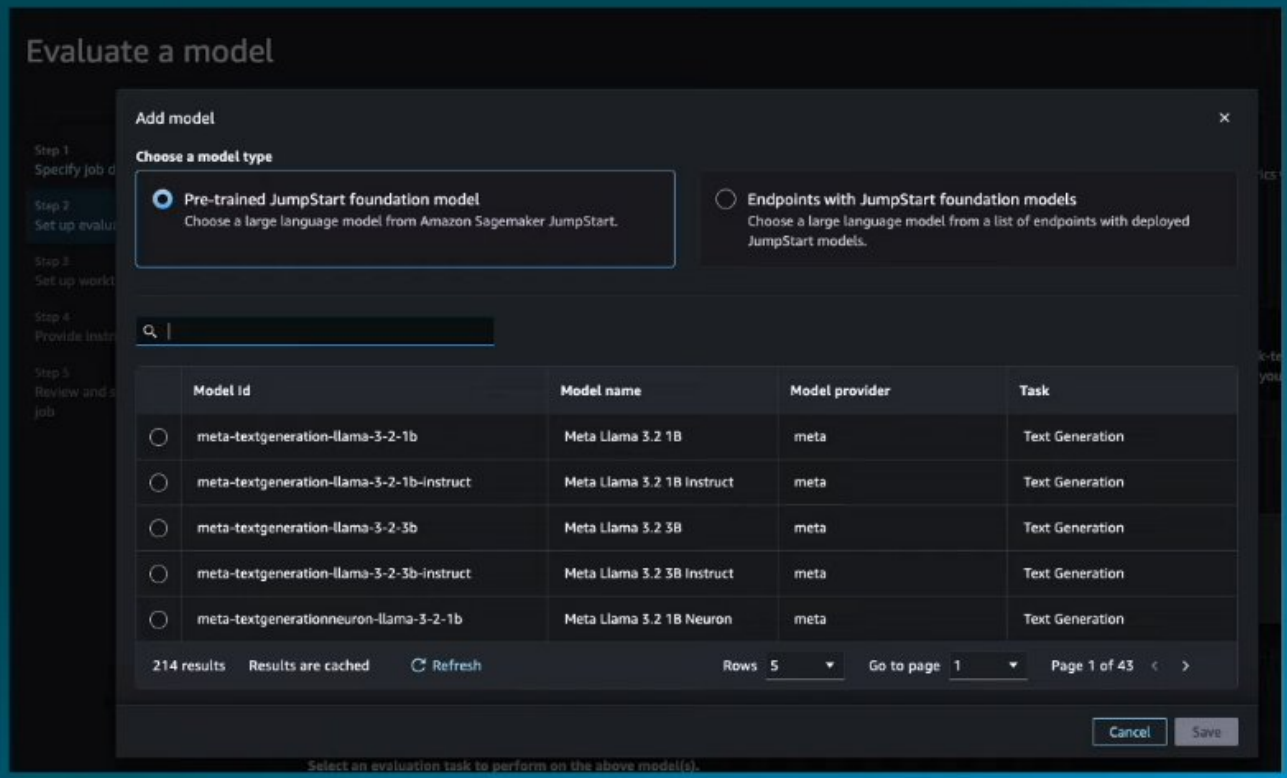
- Fully managed repo to store, share and manage ML features.
- Store features from various streaming and batch data sources.
- Capability to access historical feature values.
- Store and share features across different ML projects
- Provides both online and offline storage capabilities

SageMaker Clarify

- Detect Bias and explains predictions in ML model.
- Identify potential bias during data preparation
- Ensure that your models are fair and unbiased
- Understand how input features contribute to model predictions
- Provide insights into model behavior
- Provide compliance support

SageMaker Clarify – Model Evaluation

- Evaluate Foundation Models
- Option evaluate using curated alog (automatic) or work-team (human)
- Use built-in dataset or bring your own dataset
- Option to use AWS Managed team or use own workforce
- Built-in metrics and algorithms for model evaluation



Biases in ML Models

- Systematic errors in ML models that leads to incorrect/unfair predictions
- Selection Bias: When the training data is not representative of overall population.
- Example: Building a model with only data from approved loans to predict loan approval
- Sampling Bias: Data sets with overrepresented or under-representative groups.
- Example: Training data for facial recognition model with people from one ethnic group.
- Measurement Bias: Errors in the data collection process.
- Example: Training data collected with a faulty sensor.
- Confirmation Bias: Model reinforces existing biases.
- Example: Historical hiring data shows preference for certain demography.
- Observation Bias: Person collecting/interpreting the result has bias for result.
- Example: Study only includes successful startups.

SageMaker Clarify – Model Explainability

- Supports explainability for tabular data, NLP, and computer vision models
- Provides both Global and Local explanations:
  - Global → Overview of feature importance across dataset
  - Local → Explain individual predictions
- PDP helps visualize how changes in feature value impact model's prediction
- Uses SHAP (SHapley Adaptive exPlanations) values to measure influence of a feature for a prediction.
- Use Cases:
  - Building Trust
  - Debugging Models
  - Regulatory compliances
  - Fairness and Bias detection before model deployment.

Amazon SageMaker - Model Cards

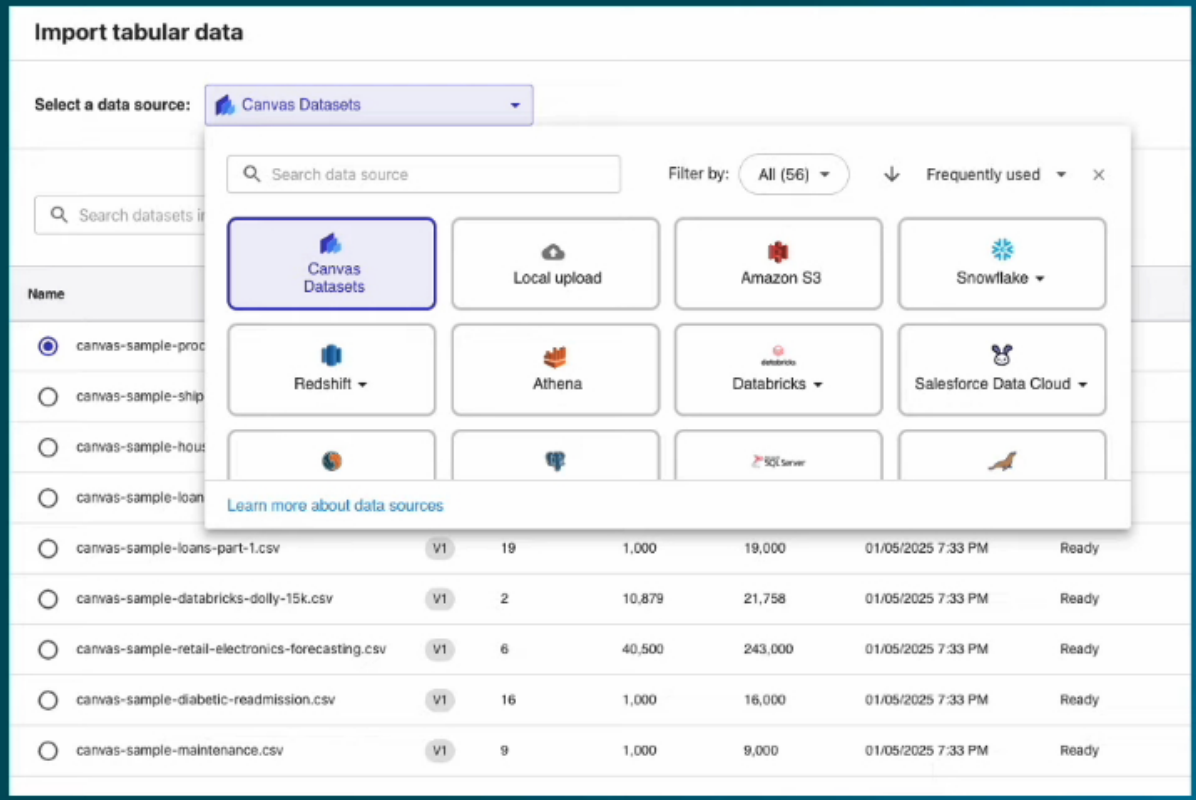
- Document critical details about ML model in a single place
- Capture key information about your models, including intended use, risk rating, training details, and performance metrics
- Provide detailed documentation for audit and compliance
- Maintain an immutable record of model changes
- Communicate how models are intended to support business goals and their limitations

SageMaker Data Wrangler

- Streamline the process of data preparation and feature engineering.
- Visual interface to clean, transform and visualize data.

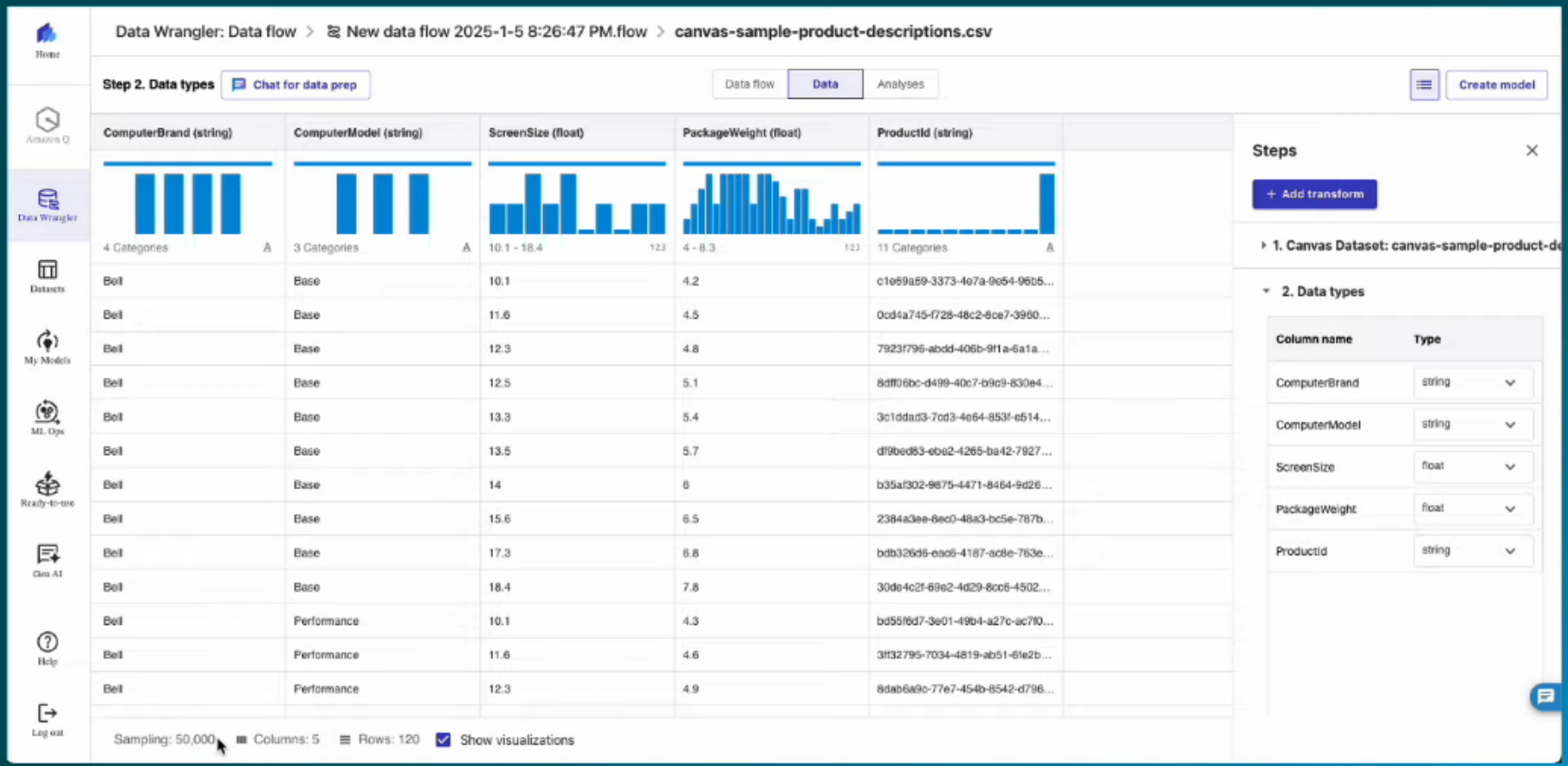
Data Import:

- Supports multiple data sources
- Import data from Amazon S3, Redshift, Athena, and others



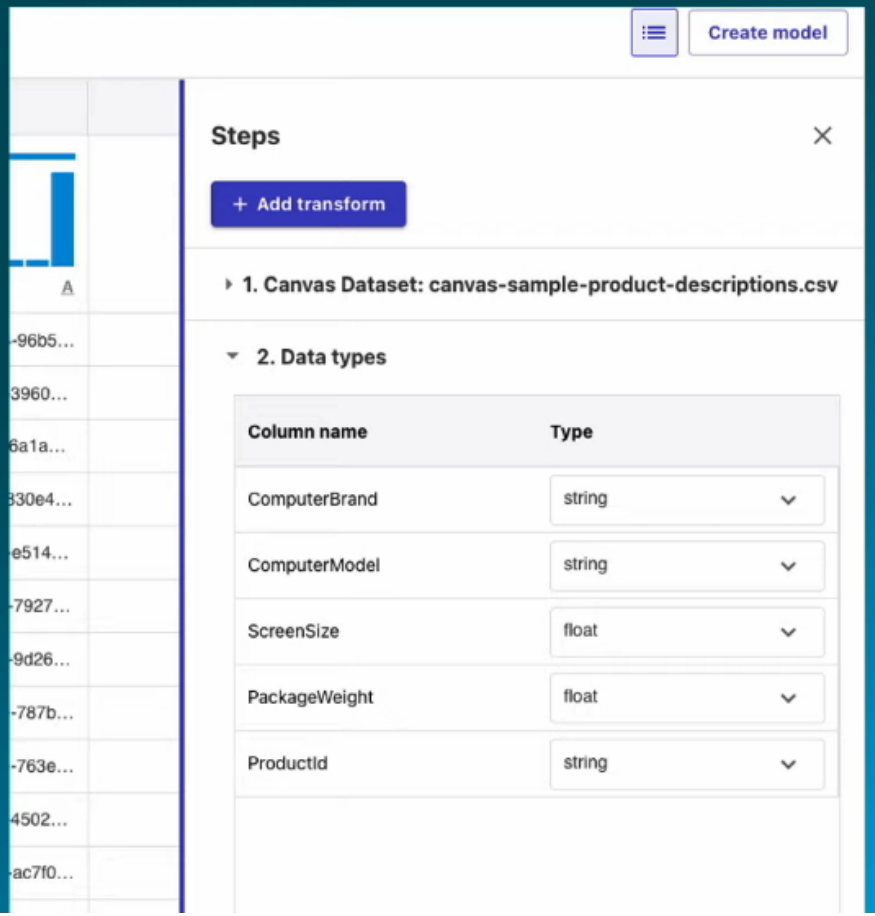
SageMaker Data Wrangler – Data Preview

- Preview data sample, understand its structure, data issues
- Spot missing values, outliers, and other anomalies



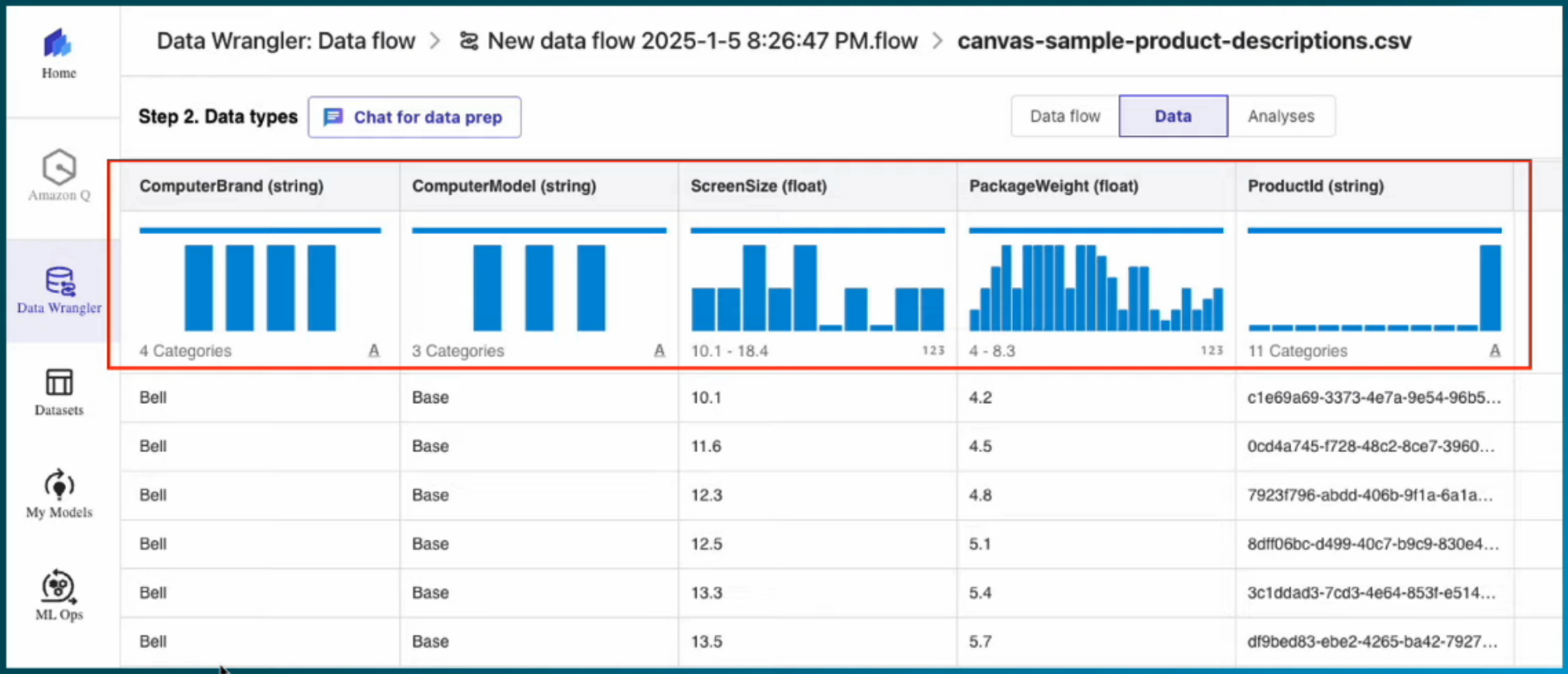
SageMaker Data Wrangler – Data Transform

- Over 300 built-in transformations
- Perform Data Transform operations:
  - Handling Missing Data
  - Normalize
  - Feature Engineering
  - Data Cleaning
- Data Quality Tool
- SQL Support



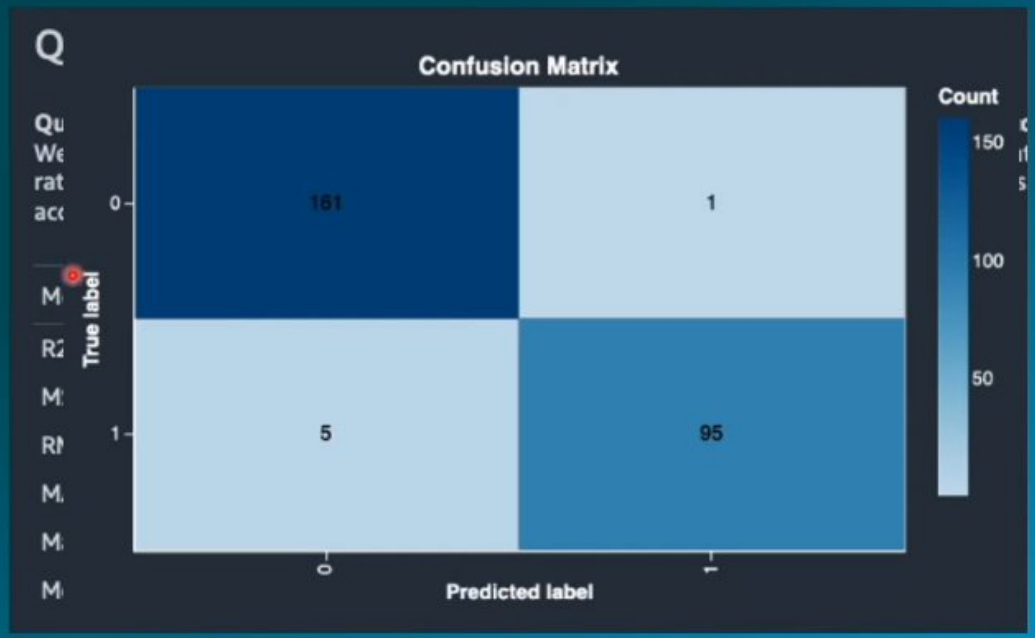
SageMaker Data Wrangler – Data Visualization

- Visualization tool to explore data distribution, correlations, summary
- Create histograms, scatter plots, box plots



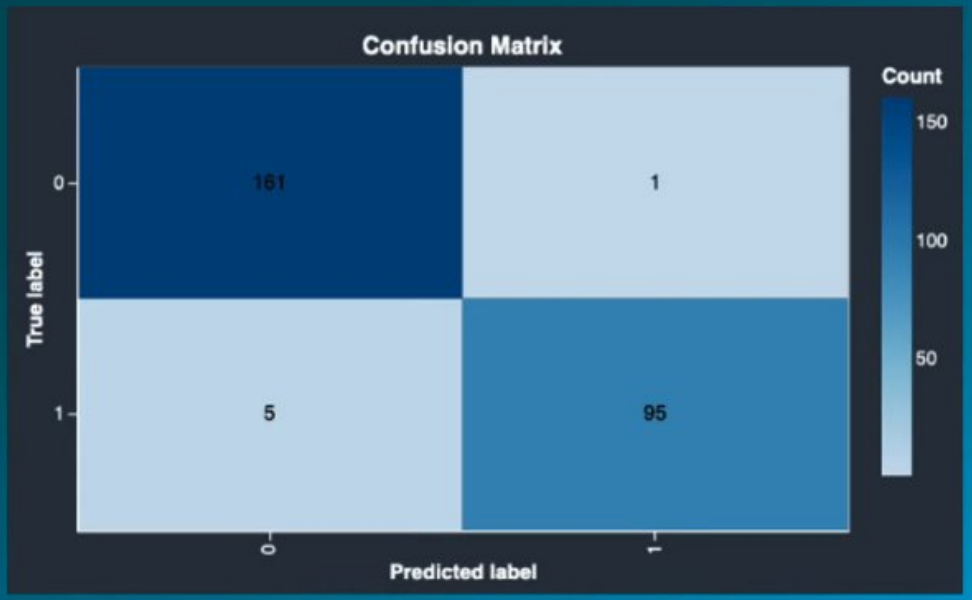
SageMaker Data Wrangler – Quick Model

- Provides and estimate of data's predictive power
- Splits data into training and validation set
- Provides insights into model accuracy, feature importance, confusion matrix.
- Validate data before building complex models
- Confusion Matrix:
  - Number of times predicted label matches true label
  - Number of times predicted label doesn't match true label



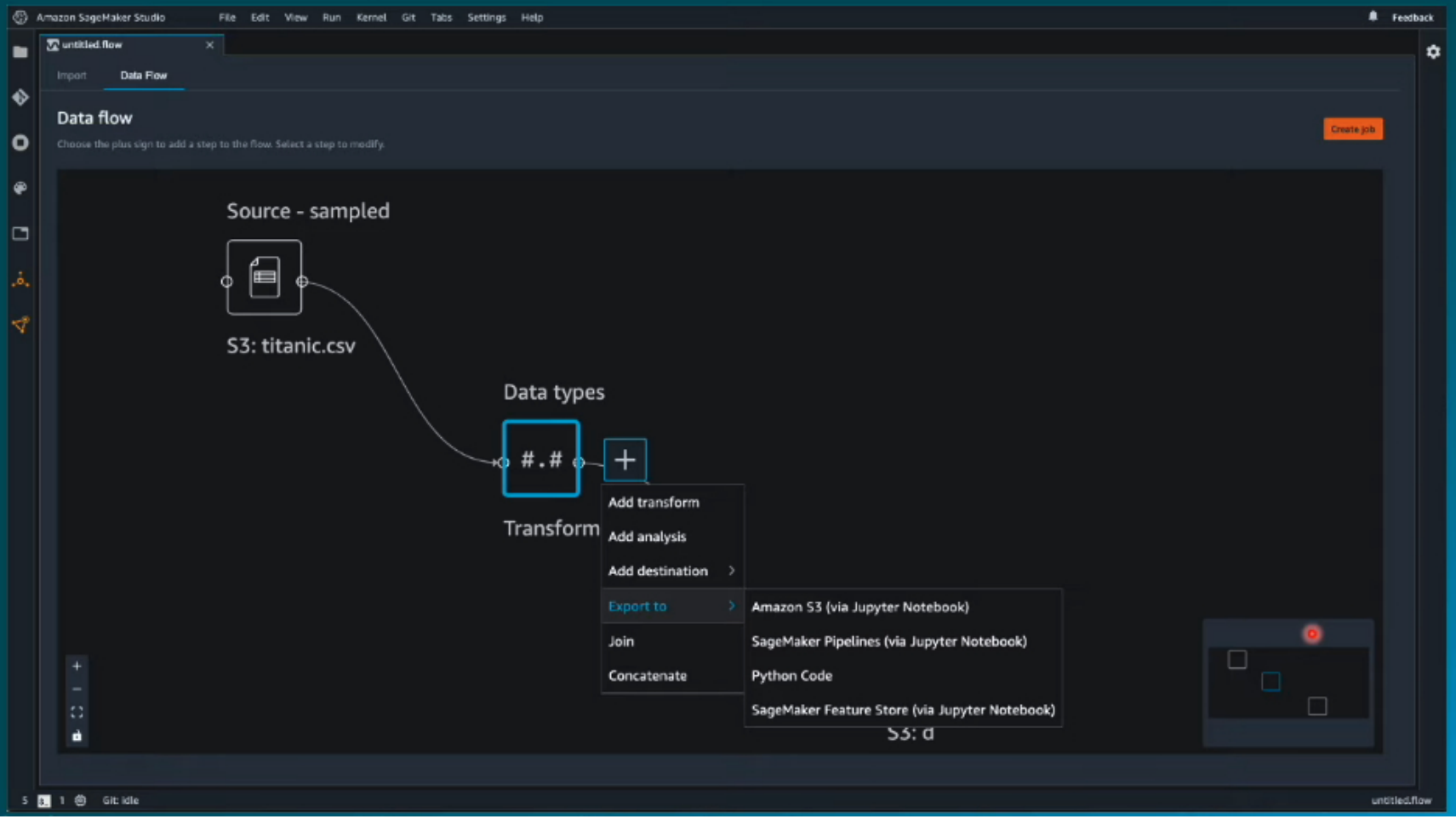
SageMaker Data Wrangler – Quick Model

- True Label → Actual observation in your data
- Example: Actual fraudulent transaction in fraudulent transactions detection model
- Predicted Label → Label that model assigns to the data
- Model's Behavior:
  - Sensitivity - model's ability to detect fraudulent transactions.
  - Specificity - model's ability to avoid detecting non-fraudulent transactions as fraudulent



SageMaker Data Wrangler – Data Export

- Export the data to Amazon S3 or to SageMaker
- Generates code to reproduce transformation



SageMaker Ground Truth

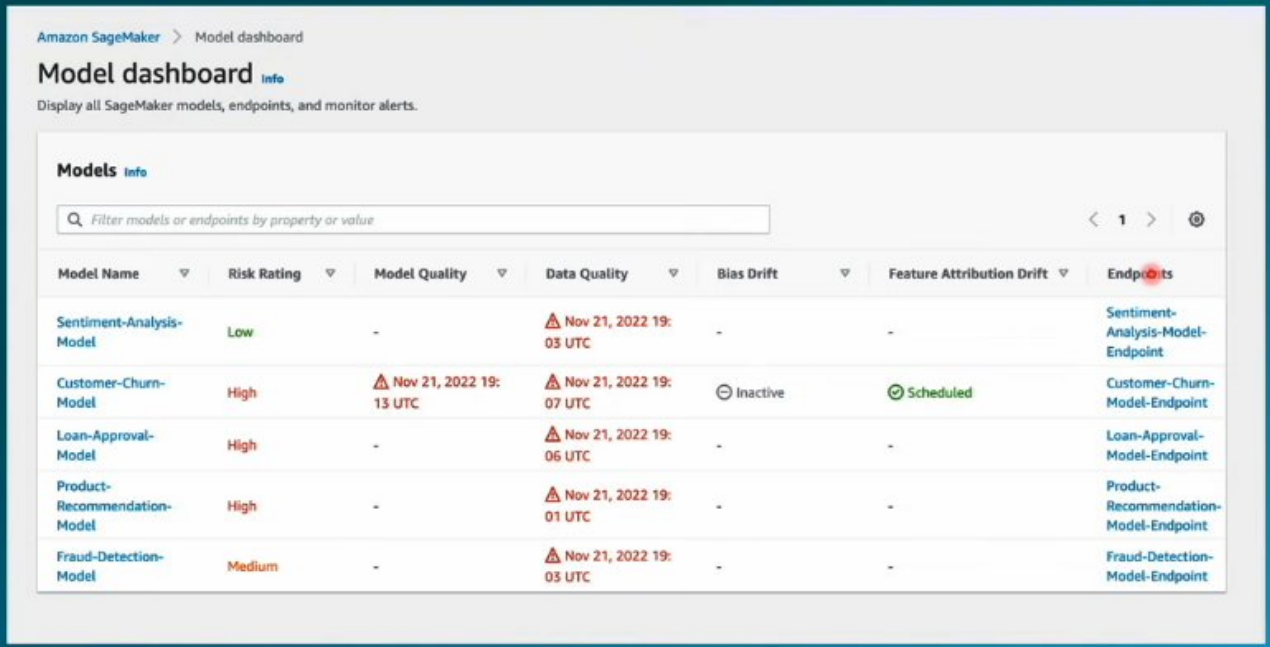
- Create and manage labeled dataset for machine learning
- Built-in workflows for data labeling
- Human-in-the-loop (RLHF)
- Human feedback to evaluate and improve model performance
- Automated Labeling
- Pre-built labeling template
- Flexible Workforce Option
  - Amazon Mechanical Turk, 3rd-party vendor, Own workforce
- Generate custom dataset
- RLHF => SageMaker Ground Truth

SageMaker Model Governance

- Framework to provide systematic visibility into ML models
- Ensures compliance, enhance transparency, managed risks
- SageMaker Model Dashboard:
  - Centralized visual overview of all the ML model
  - Integrates Model Monitor, ML Lineage Tracking, CloudWatch
- SageMaker Role Manager:
  - Manages permissions and access controls
  - Allows admin to define user permissions
- SageMaker Model Cards:
  - Document essential model information
  - Captures model uses, risk rating, training details

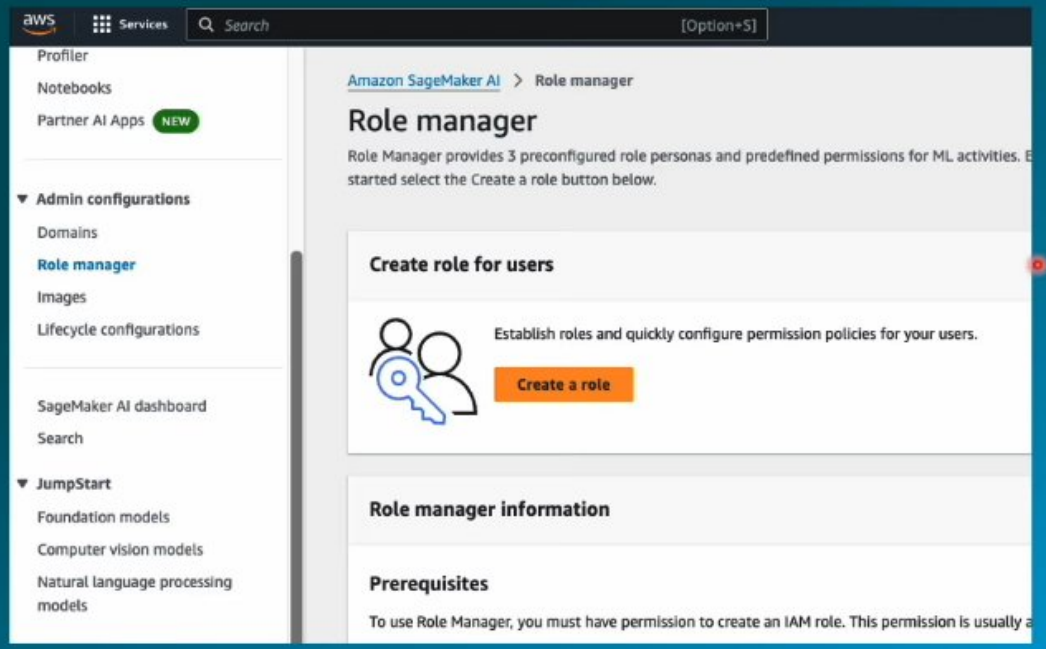
SageMaker Model Dashboard

- Unified view for IT administrators, model risk managers, & business leaders
- Monitor model performance in real-time, track data quality, model quality
- Set up automated alerts for deviations from expected behavior
- Access to model's detailed insights, access logs



SageMaker Role Manager

- Manage permissions and access controls for ML personas.
- Provides three preconfigured role personas with predefined permissions
  - Data Scientist
  - MLOps
  - Sagemaker Compute Role
- Create and maintain custom roles
- Integrates with AWS Identity and Access Management (IAM)
- Ensures least privilege access, reducing the risk of unauthorized access

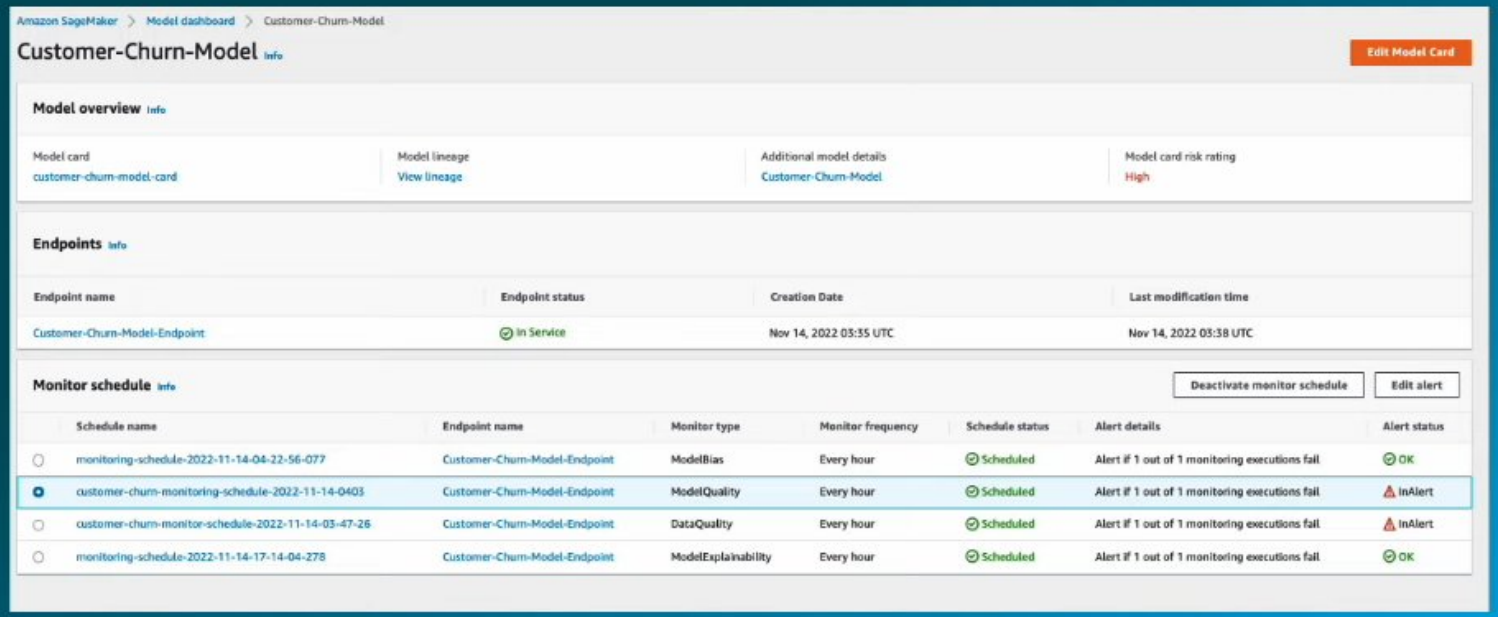


Exam Tips

Phrases in the Exam question	Probable AWS Services/Option
Data Preparation, ML Data ETL, confusion Metrix	SageMaker Data Wrangler
Human Review, RLHF in Machine Learning	SageMaker Ground Truth
Minimize risk of incorrect annotations	SageMaker Ground Truth
Share and manage variables for model deployment	SageMaker Feature Store

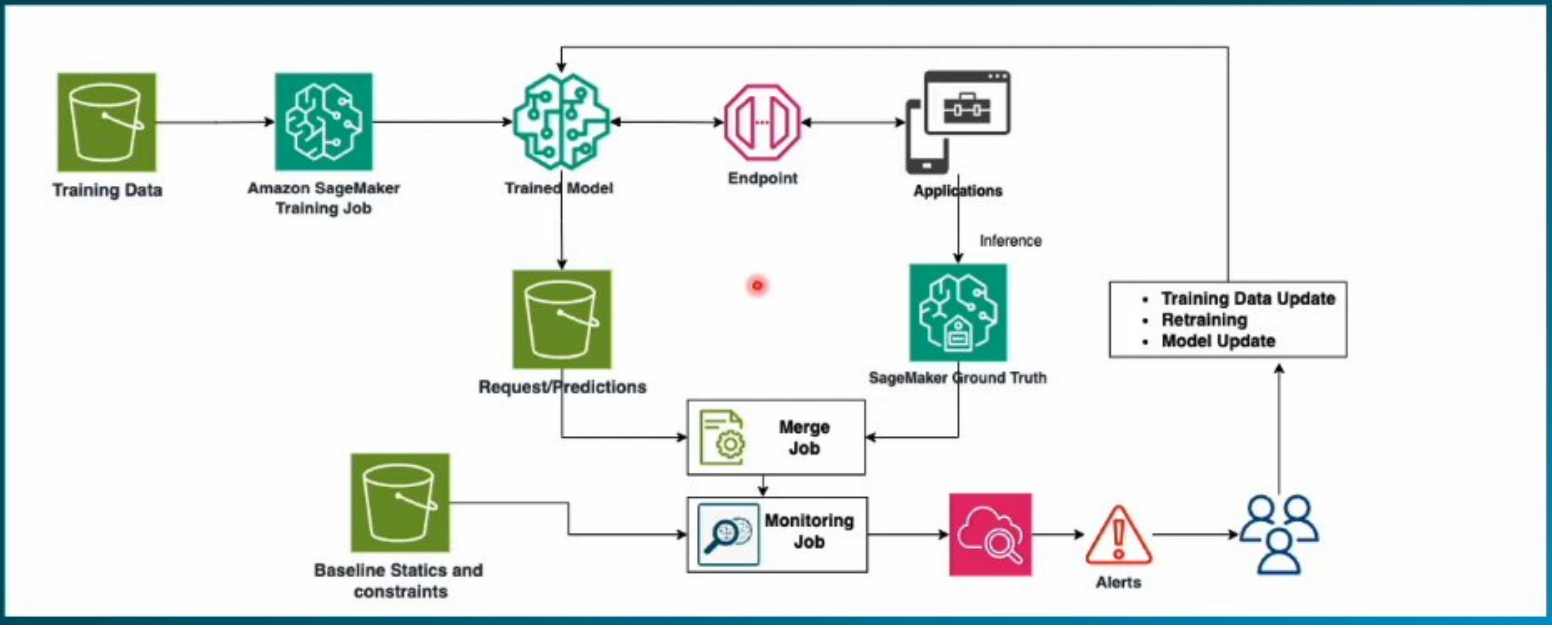
SageMaker Model Monitor

- Real-time monitoring of data quality, model quality, bias drift.
- Automatic alerts for deviation from predefined thresholds.
- Integrate with SageMaker Clarify to identify potential bias
- Ensuring model accuracy over time and takes proactive remediations.
- Proactive Actions:
  - Retrain model
  - Audit Upstream systems
  - Fix quality issues



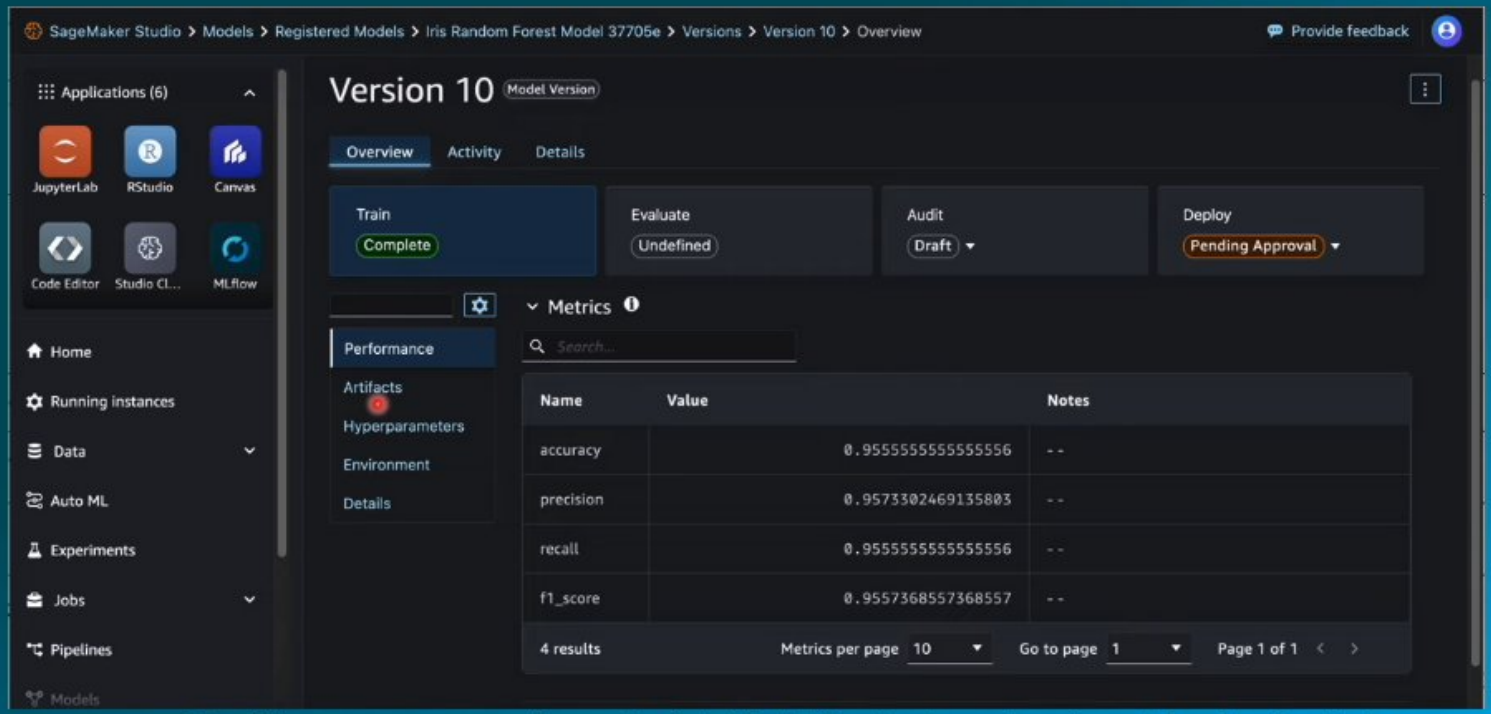
SageMaker Model Monitor

- Model monitor's pre-built monitoring capabilities:
  - Data Quality – Monitor drift in the data quality
  - Model Quality - Monitor drift in the model quality (accuracy)
  - Bias drift for model in production – Monitor bias in model's prediction
  - Feature attribution drift for model in production - Monitor feature drifts



SageMaker Model Registry

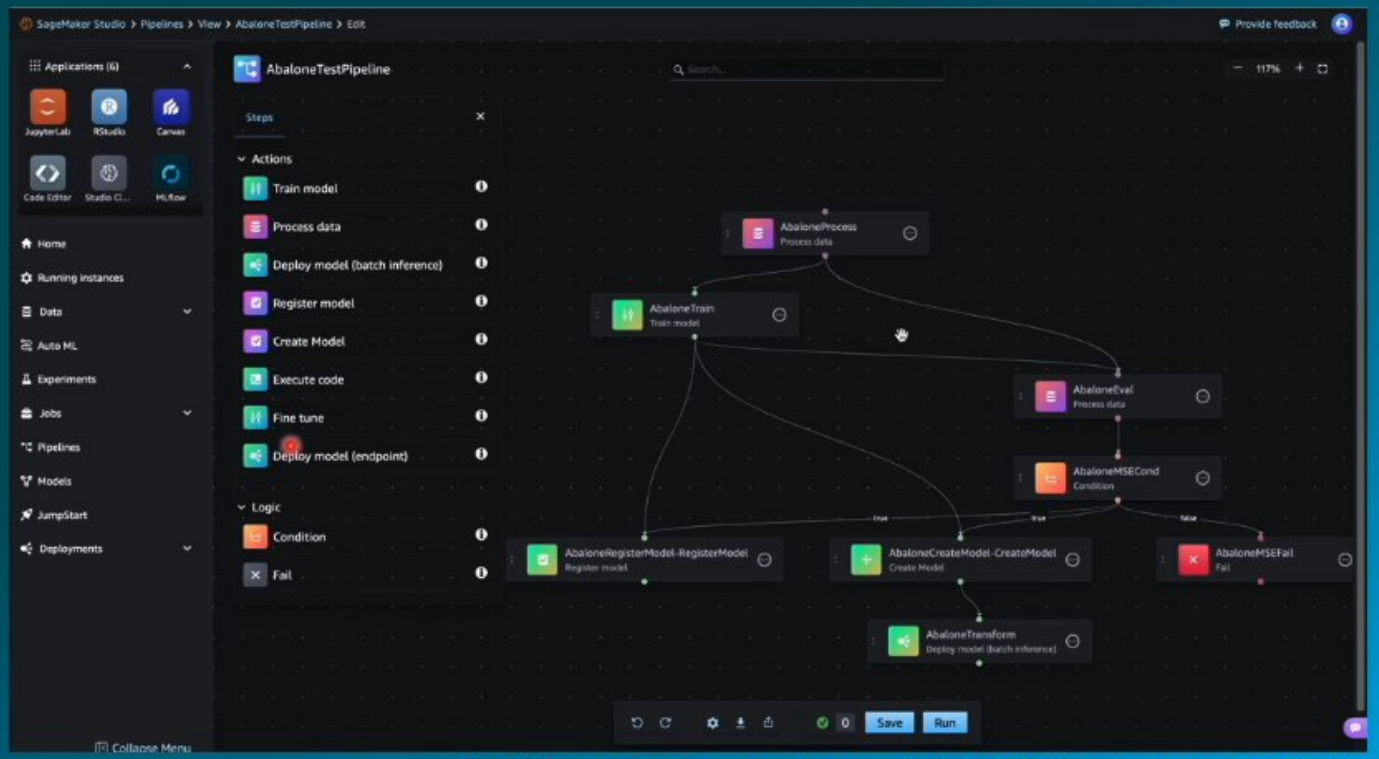
- Centralized repo to manage and catalog ML models
- Catalog models for production, track and manage model version
- Associate metadata with ML models
- Integrate Model Pipeline
- Streamline model deployment process
- Enhance collaboration with model share
- Use collections to group models



https://docs.aws.amazon.com/sagemaker/latest/dg/mlflow-track-experiments-model-registration.html

SageMaker Pipelines

- End-to-end workflow orchestration for ML projects
- Integration with SageMaker Data Wrangler and Feature Store for data preparation
- Add pipeline steps (training, processing, model registration)
- Automate feature engineering & training
- Ensure reproducibility and consistency in ML workflow.



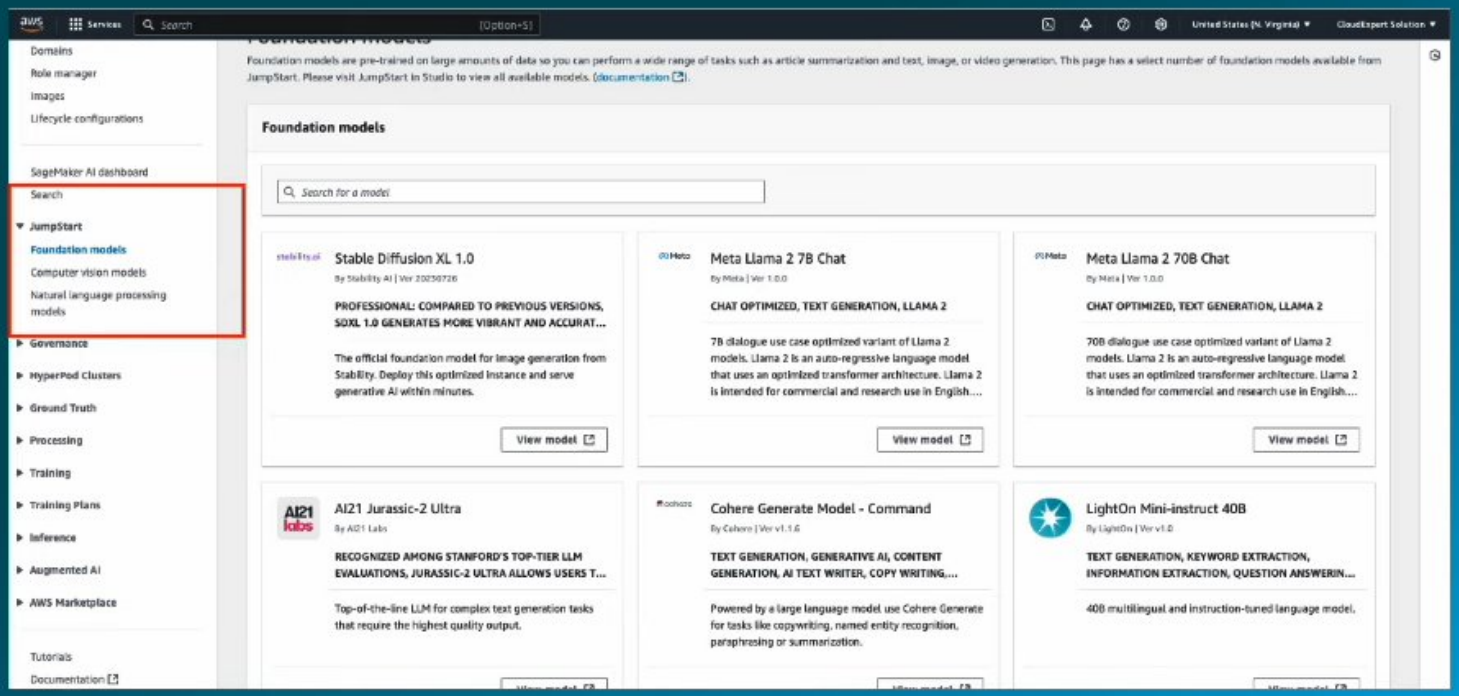
https://docs.aws.amazon.com/sagemaker/latest/dg/pipelines-overview.html

SageMaker Pipelines – Supported Steps

- Processing – data processing, feature engineering
- Training – for training a model
- AutoML – for automatic model training
- Tuning – for Hyperparameter tuning
- Model – to create or register a SageMaker Model
- ClarifyCheck - perform drift check against baseline (Data bias, model bias, model explainability)
- Quality Check – perform quality check against baselines (Data Quality, Model Quality).

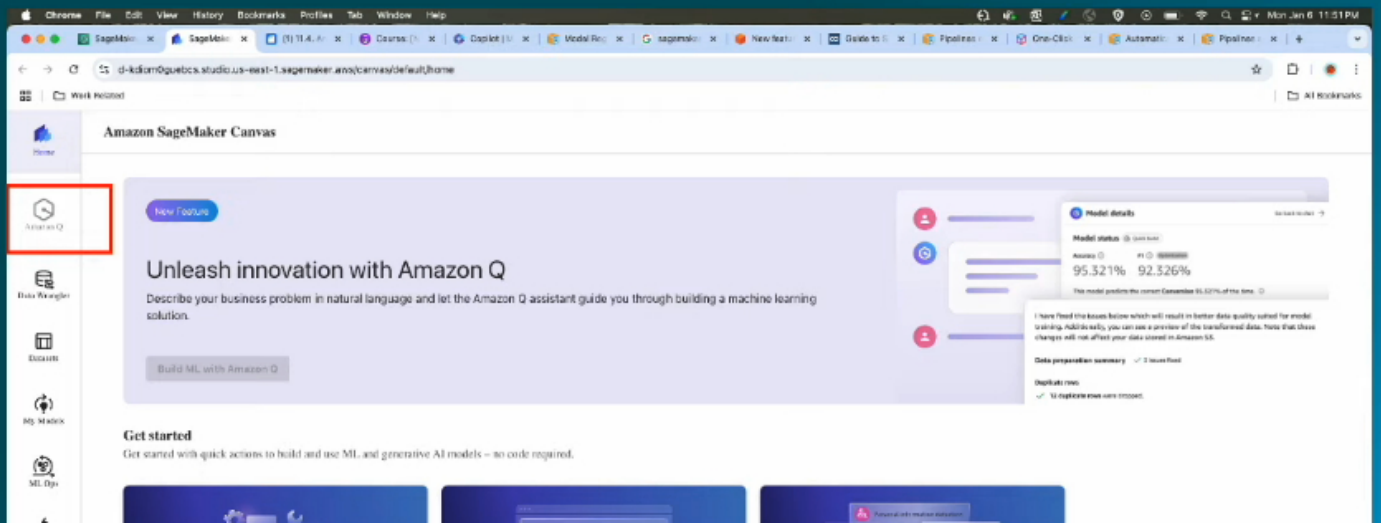
SageMaker JumpStart

- Access to foundation models and built-in algorithms
- Prebuilt ML solutions that can be deployed with just a few clicks
- Customizable models for specific use cases
- Model Options:
  - Foundation
  - Computer Vision
  - NLP
- Rapidly deploy ML models for common tasks
- Accelerating the ML journey with ready-to-use solutions



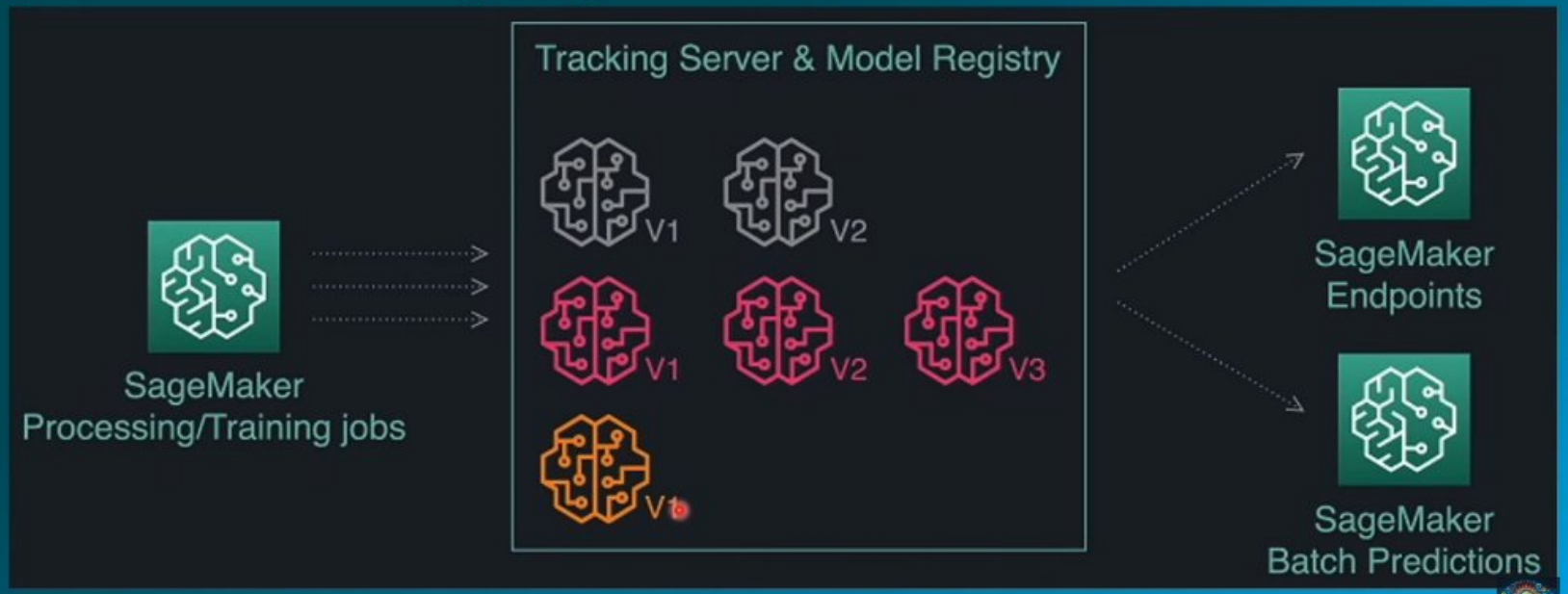
SageMaker Canvas

- Visual interface for building, evaluating, and deploying ML models
- Integration with SageMaker Model Registry
- Generative AI-powered assistance with Amazon Q Developer
- No-code AutoML
- Collaboration & Governance



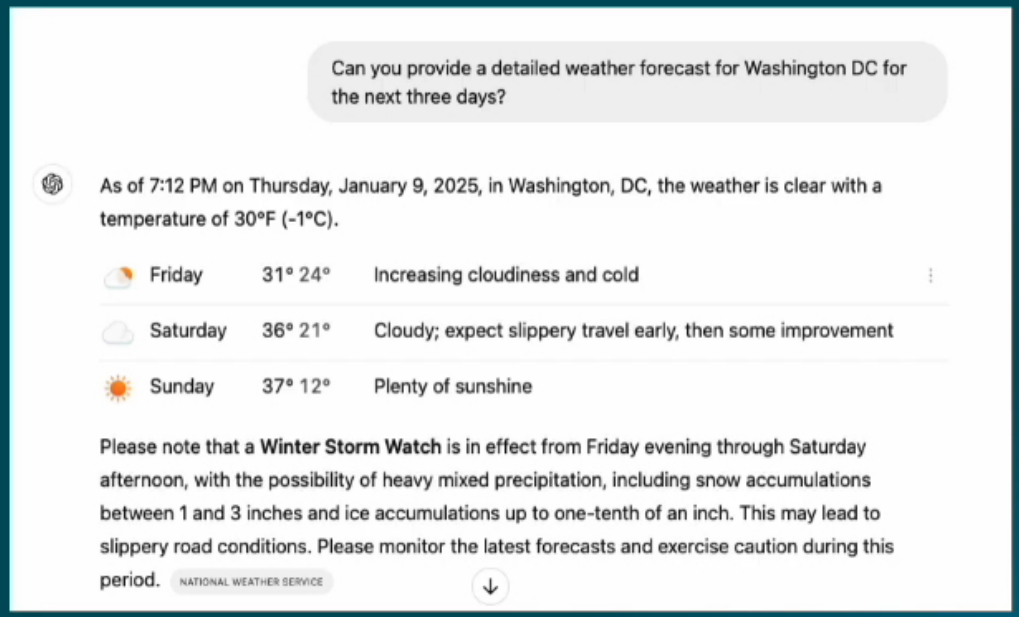
MLflow on SageMaker

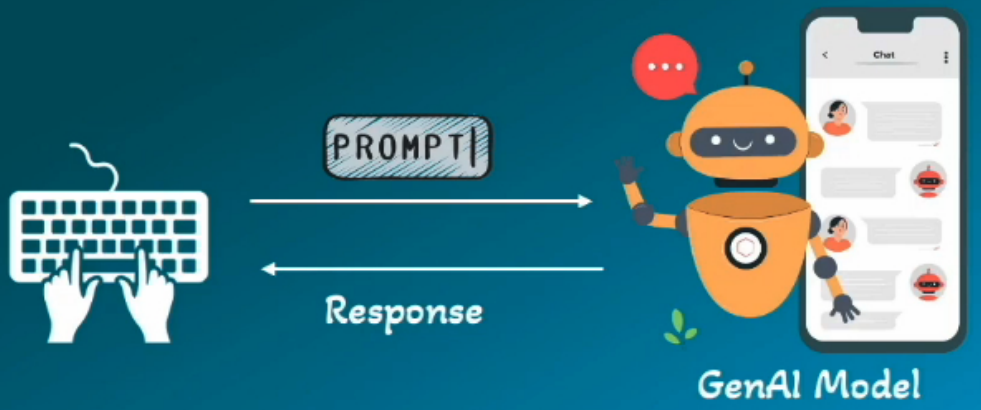
- Open-source platform to manage the ML lifecycle
- MLflow component – Tracking, Projects, Models, Registry
- Deploy MLflow on AWS Fargate
- Track, Organize and compare ML experiments
- Integrate with SageMaker model registry



What is Prompt Engineering?

- **Prompt** - Natural language text that requests the **generative AI** to perform a specific task





- Process of designing and refining prompts to effectively communicate with Generative AI models.

What is Prompt Engineering?

- Naïve Prompt – What is weather?
- Enhanced Prompt – Can you provide a detailed weather forecast for Washington DC for the next three days??
- Include specific instructions or context to guide the GenAI model's response.
- Naïve Prompt – Explain machine learning
- Naïve Prompt – Explain the concept of supervised learning in machine learning with examples
- Enhance Prompt - Explain the concept of supervised learning in machine learning, including its definition, key components, and a real-world example.

What is Prompt Engineering?

- Enhance Prompt - Explain the concept of supervised learning in machine learning, including its definition, key components, and a real-world example.
- Improved Prompt - Explain the concept of supervised learning in machine learning. Start with a definition, then describe the key components such as labeled data and algorithms, and finally, provide a real-world example like spam email detection, and please keep the answer less than 200 words.
- Improved Prompt Components:
  - Instruction
  - Context
  - Input Data (optional)
  - Output Indicator

What is Prompt Engineering? – Improved Prompt

Instruction:	Explain the concept of supervised learning in machine learning. Start with a definition, then describe the key components such as labeled data and algorithms, and finally, provide a real-world example like spam email detection
Context:	I am a High School Teacher and wants to teach the concept to 10^{th} grade students.
Output Indicator:	Please keep the answer less than 200 words.
Negative Prompt:	Do not include any technical jargon or complex mathematical formulas.

Prompt Engineering – Negative Prompting

- A technique used in prompt engineering where you instruct the AI model on what not to include in its response.
- Negative Prompting helps ensure that the output meets specific requirements and avoids unwanted content.
- **Negative Prompting is useful for:**
  - Content Filtering
    - **Example:** "Generate a product review, but do not include any negative comments."
    - **Use Case:** "This can be useful for marketing purposes where you want to highlight only the positive aspects of a product."

Prompt Engineering – Negative Prompting

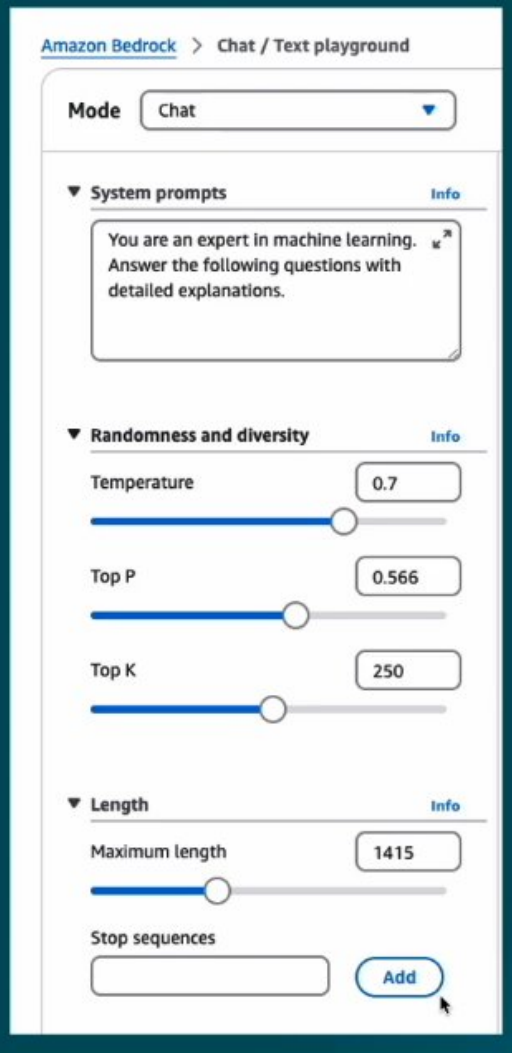
- A technique used in prompt engineering where you instruct the AI model on what not to include in its response.
- Negative Prompting helps ensure that the output meets specific requirements and avoids unwanted content.
- <u>Negative Prompting is useful for</u>:
  - Content Filtering
  - Simplifying Technical Explanation/Educational Content
  - Bias Mitigation

- **Example: "Provide an analysis of the election results, but do not include any biased or politically charged language."**

Prompt Engineering – Negative Prompting

- A technique used in prompt engineering where you instruct the AI model on what not to include in its response.
- Negative Prompting helps ensure that the output meets specific requirements and avoids unwanted content.
- **Negative Prompting is useful for:**
  - Content Filtering
  - Simplifying Technical Explanation/Educational Content
  - Bias Mitigation
  - Customer Service

Prompt Performance Optimization



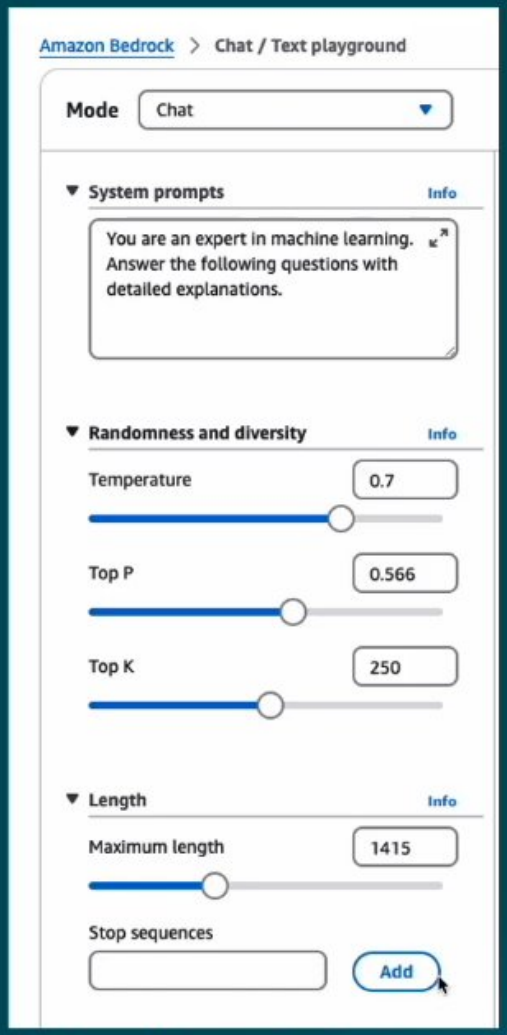
System Prompts: Predefined instruction given to the GenAI model to set context or behavior

- Example – "You are an expert in machine learning."

Temperature: Controls randomness of the model's response

- Low Value (0.2) – more deterministic and focused responses
- Example - Supervised learning is a type of machine learning where the model is trained on labeled data.
- High Value (1.0) – diverse, captive and unpredictable responses
- Example - Supervised learning involves training a model using labeled data, where each example is paired with an output label, guiding the learning process.

Prompt Performance Optimization – Top P (Nucleus Sampling)

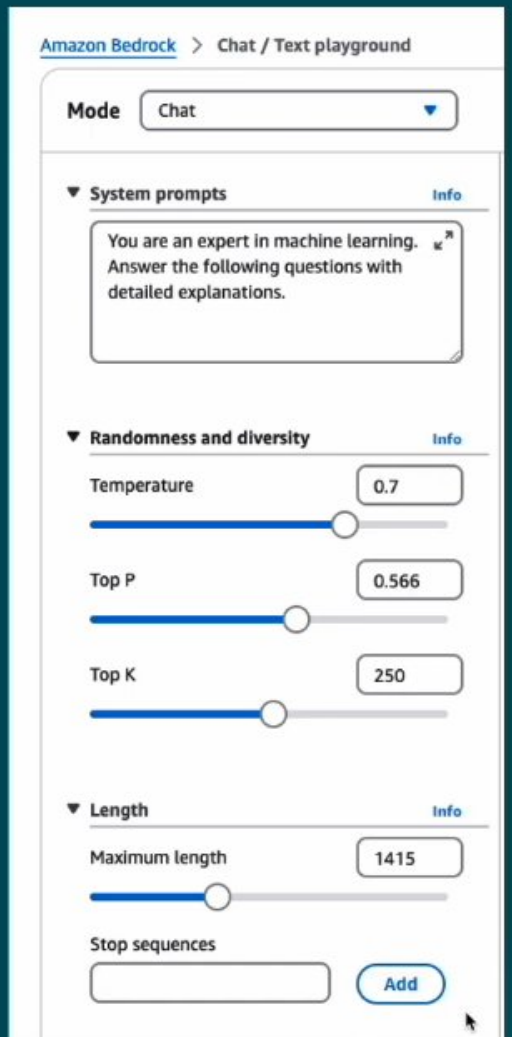


Top P (0 to 1): Controls the diversity of response

- Low (e.g., 0.1) – top 10% of probable words, more focused and predictable response
- Example - Supervised learning is a method where the model learns from labeled data.

- High (e.g., 0.9) – Up to 90% of probable words, diverse and varied responses

Prompt Performance Optimization



Top K (0 to 500): Limit top K most probable words

- Low (e.g., 10) – top 10 most probable words, more focused and predictable response
- Example - Supervised learning is a type of machine learning where the model is trained on labeled data.
- High (e.g., 250) – top 250 most probable words, diverse and varied responses
- Example - Supervised learning involves training a model using labeled data, where each example is paired with an output label, guiding the learning process.

Length: - Controls the maximum number of token

Stop Sequence: - specific tokens signal the model to stop

Prompt Latency

- Time it takes for an AI model to generate a response after receiving a prompt.
- High latency → delays in interactions
- Low latency →responses are generated quickly
- Factors influences Prompt Latency
  - Model Size and Complexity
  - Model Types
  - Number of Input Tokens
  - Number of Output Tokens
- Latency is NOT influenced by Temperature, Top P, Top K.

Prompt Engineering – Zero Shot Prompting

- Ask AI model to perform a task without providing example.
- Model relies on pre-trained knowledge to generate response
- Zero-shot prompting is useful when you want the model to perform tasks based on its general knowledge without needing specific examples.

- Efficient for straightforward tasks like translations, summaries, and factual questions.

Prompt Engineering – Few Shot Prompting

- Providing a few examples to guide AI model's response.
- Help model understand the desired format and context
- Few-shot prompting is effective for tasks where the model benefits from seeing a few examples to understand the desired output.

- If you provide one example → "Single-Shot" or "One-Shot".

Prompt Engineering – Chain of Thoughts Prompting

- Encourages the AI model to think through a problem step-by-step, breaking down the reasoning process.
- Chain of thought prompting is valuable for tasks that require logical reasoning and step-by-step problem-solving.
- Helps the model follow a structured approach to arrive at the correct solution.
- Look for words "Think through step-by-step" or "breaking down reasoning process".

logarithmic equation is: $\log_b(x)=y$ This equation means that (b^y = x), where:
•(b) is the base of the logarithm,
•(x) is the number we want to find the logarithm of,
•(y) is the exponent to which the base (b) must be raised to get (x).
•Using this steps please solve $\log_2(8)$

Prompt Template

- Predefined structures that can be reused for different tasks
- Helps maintain consistency and efficiency in prompt engineering.
- Benefits of Prompt Templates:
  - Consistency
  - Efficiency
  - Clarity
  - Scalability

aws

Amazon Bedrock Prompt Template

```
Prompt template for Titan
"""{{Text}}


{{Question}}? Choose from the following:
{{Choice 1}}
{{Choice 2}}
{{Choice 3}}"""
```

```
User prompt:
San Francisco, officially the City and County
of San Francisco, is the commercial, financial, and cultural
center of Northern California. The city proper is the fourth
most populous city in California, with 808,437 residents,
and the 17th most populous city in the United States as of 2022.

What is the paragraph above about? Choose from the following:

A city
A person
An event
```

```
Output:
A city
```

Amazon Q Business

- Fully managed, GenAI-powered assistance for enterprises
- Can integrate with internal and external data sources
- Provide answers, generate summaries, automate routine tasks
- Offers a conversational interface to ask questions and receive answers
  - Example: What are the remote work policies?
- Personalized Responses – Tailored responses based on user's profile and permissions. (Example – A manager might receive different project budget information compared to team members)
- Provides file upload capabilities
- Amazon Q business built-on Amazon Bedrock, uses more than one FMs
- Amazon Q business doesn't allow to choose the underline FM

Amazon Q Business – Use Cases

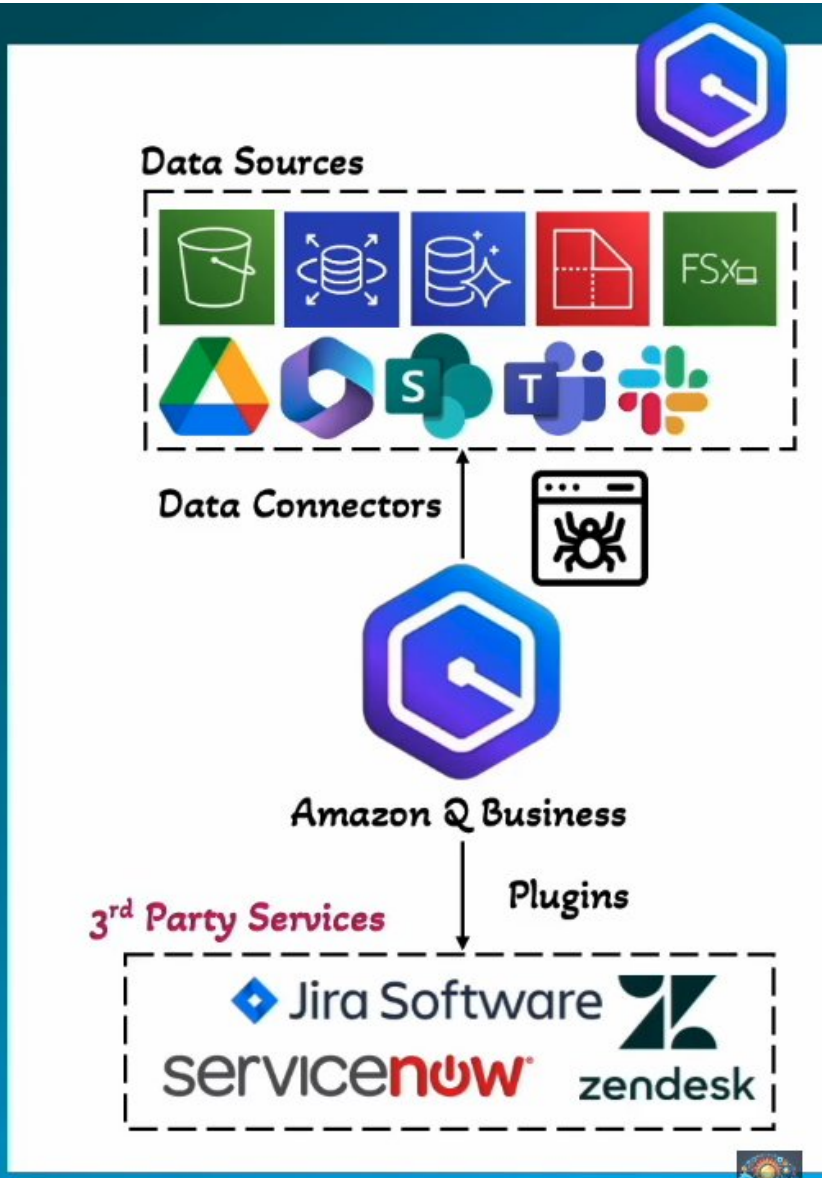
- Sales and Marketing- Generate content, summarize reports.
  - Example - A sales manager can ask, 'What were our top-performing products last quarter?' and get a detailed analysis.

- HR Support- Provide instant answers to HR-related questions.
  - Example - An employee can ask, 'How many vacation days do I have left?' and get an immediate answer.

- IT Help Desk- Automate responses to common IT queries.
  - Example - An employee can ask, 'How do I reset my password?' and receive step-by-step instructions.

Amazon Q Data Source Integrations

- Amazon Q provides 40+ fully managed connectors
  - Integrate with AWS services like Amazon S3, RDS, Amazon Aurora, WorkDocs, FSx for Windows, and others...
  - Integrate with databases, data warehouses, and data lakes through Amazon Quick Sight
  - External data sources Gmail, Google Drive, MS Teams, SharePoint, slack, and others...
  - Custom Data Connector – Develop custom connectors to integrate with company's internal CRM system.
- 50+ ready-to-use plugins library
  - Integrate with third-party applications, like Jira, ServiceNow, Salesforce, PagerDuty
  - Develop custom plugins to extend functionalities or connect using APIs (Example - Create custom plugin to automate weekly sales report)



Amazon Q Business - User Management

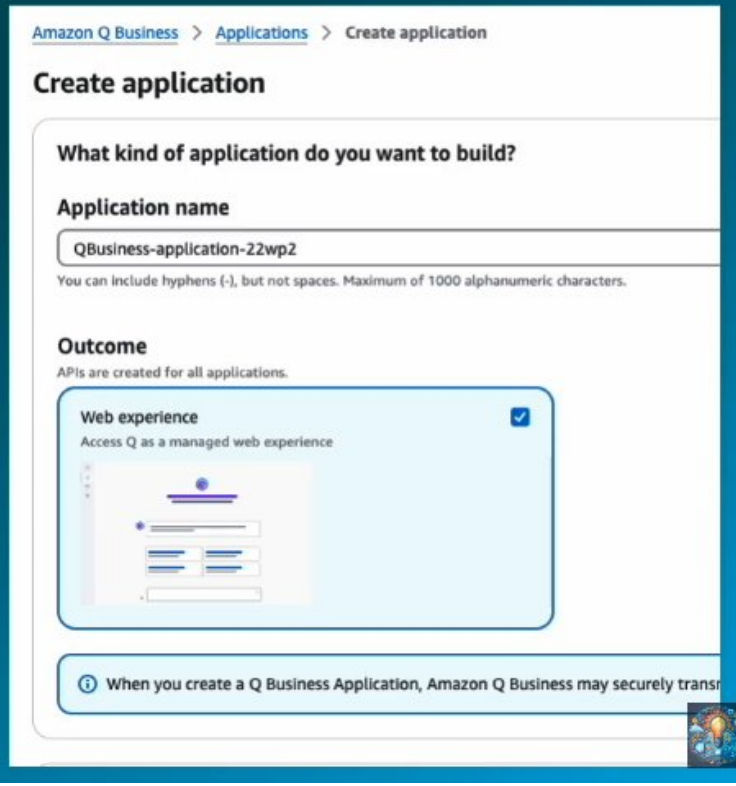
- Manage user access and permissions using AWS IAM Identity Center.
  - Example – Ensure only HR personnel can access employee records.
- Users receive responses only from the content that the user has access to.
- Integrate AWS CloudTrail for audit trial.
- Integrate External IDP with AWS IAM Identity Center using SAML or Open ID Connect (example – Google IDP, Microsoft Entra ID).

Amazon Q Business – Admin Control

- Amin Control = Guardrails.
- Define Global Control and Topic-level controls for your application.
  - Control how Amazon Q Business responds to specific topics
  - Control whether end users can upload files in chat
  - Configure where all Amazon Q Business responses will be generated using enterprise data or use underline LLM
- Global Controls apply to all users and interactions (e.g., file upload, data usages, personalization.
- Topic-level controls allows you to define specific rules for different topic or departments.
  - Example – Set up a control to ensure HR-related queries handled differently from IT support.
- Configure these controls through Admin Console or using *UpdateChatControlsConfiguration* API.

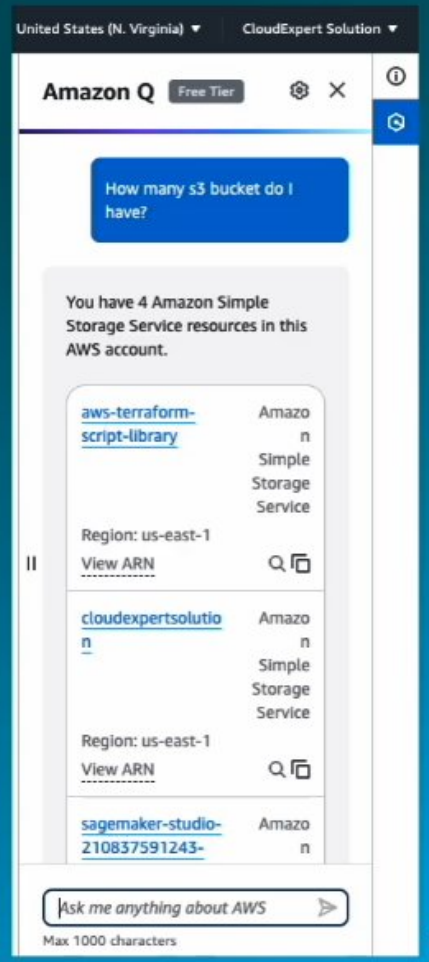
Amazon Q Apps

- Create Amazon Q Business GenAI-powered apps without coding.
- Amazon Q App only available for Amazon Q Business Pro users.
- User can directly create an app using natural language prompts.
- Amazon Q App is enabled by default when you create new Q Business application using IAM or IAM federation.
- Built-in plugins for common use cases
  - Confluence – Search Page
  - Google Callender – Find and list events
  - Microsoft Exchange – Get events, calendar, events
  - Salesforce – Manage Cases
  - ServiceNow – create, update, read, delete incidents
  - Zendex Suite – create, get and update tickets



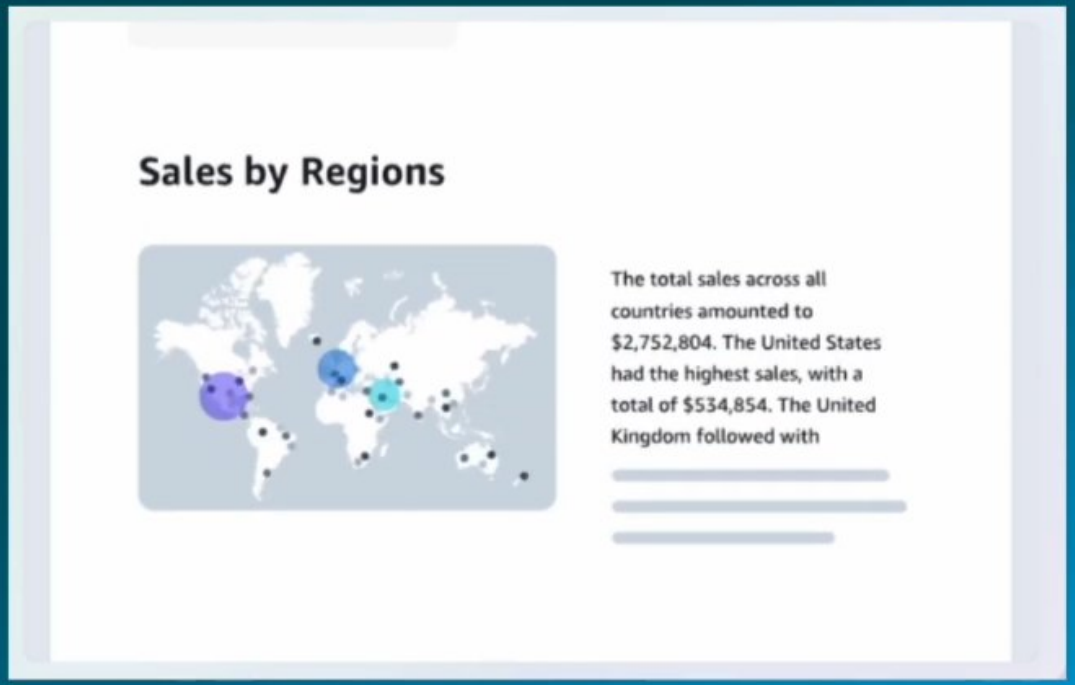
Amazon Q Developer

- Amazon Q Developer to help developers write code, troubleshoot issues, optimize workflows.
- Example: - Answer question about AWS documentations
- Answer question about resources in your AWS account
- Code Completion and Suggestions – offers intelligent code completion and suggestions
- Error Detection and Fixes – detects errors in code and suggests fix
- Code Refactoring – refactor code to improve performance
- Documentation and Examples – Code examples directly within IDE
- Cloud Architecture Design – expert advice for architecture design
- Integration with IDEs – integrates seamlessly with VS Code, Visual Studio, JetBrains,
- Data is encrypted in transit and at rest and IAM access controls



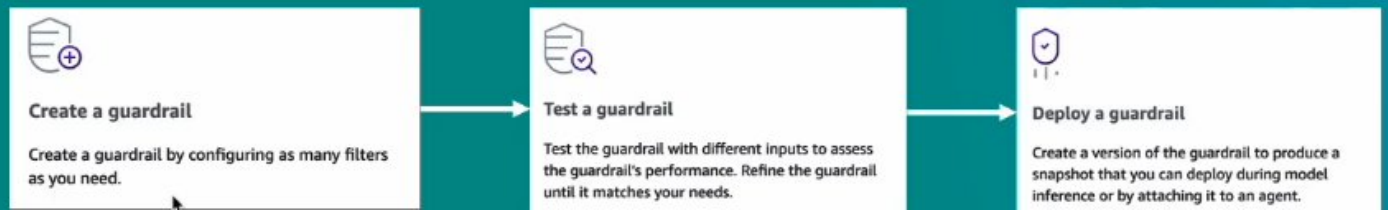
Amazon Q for QuickSight

- Amazon QuickSight is a cloud-based BI service to create visualizations, perform analysis, and share insights.
- Amazon QuickSight Q integrated NLP capability in BI Dashboard.
- Allows you to talk with data.
- Generate and edit visuals using Natural language prompts.
- Create executive summary and compelling story using data.



Amazon Bedrock

- Fully Managed service provide access to leading foundational models.
- AI Model Providers – AI 21 Labs, Amazon, Anthropic, Cohere, HugginFace, and more
- Option to select Serverless models and Model from Bedrock Marketplace
- Option to create custom model and import the model to Amazon Bedrock
- Option to choose and simultaneously operate with more than one models
- Prompt Router – efficiently route requests between different foundational models
- Serverless experience – No need to manage infrastructure
- Provides Playgrounds to evaluate and validate Foundational Model without deployment
- Customization – Fine-tune models using your own dataset to improve performance
  - Option to configure – System Prompts, temperature, Top P, Top K, Length
- Amazon Bedrock Guardrails – Content Filtering, Bias Detection, Mitigation, Privacy Protection



Available Foundational Models

aws BEDROCK

Providers

- [x] AI 21 Labs (5)
- [x] Amazon (13)
- [x] Anthropic (9)
- [ ] Arcee AI (5)
- [ ] Camb.ai (1)
- [x] Cohere (6)
- [ ] EvolutionaryScale, PBC (1)
- [ ] Gretel (1)
- [x] HuggingFace (83)
- [ ] IBM Data and AI (6)
- [ ] John Snow Labs (3)
- [ ] Karakuri, Inc. (1)
- [ ] LG CNS (1)
- [ ] Liquidai (3)
- [x] Meta (9)
- [ ] Mistral AI (4)
- [ ] NCSoft (2)
- [ ] NVIDIA (1)
- [ ] Preferred Networks, Inc. (1)
- [x] Stability AI (1)

- **AI21:** Text generation and summarization
- **Amazon Titan:** Text generation, embeddings, and summarization
- **Anthropic Claude:** Conversational AI and NLP tasks, chatbots and virtual assistants.
- **Cohere:** Natural language understanding and generation, sentiment analysis and text classification.
- **Hugging Face:** NLP tasks, including translation and text generation.
- **Meta:** Language understanding and multimodal applications (text + images).
- **Mistral AI:** Real-time text generation and analysis.
- **Stability AI:** High-quality image generation from text prompts.

How to Identify the Optimal Foundational Model?

- Task Requirement:
  - Nature of the task – Text generation, image generation, embeddings
  - Domain Specificity – Medical, legal, creative writing
- Model Capabilities:
  - Parameter Count – Complex models require more compute power
  - Supported Modalities – Text, image, video
- Customization Options:
  - Customization – Ability to fine-tune the model
  - Provisioned Throughput – requirement for high performance & customization
- Licensing and Costs:
  - Licensing Agreements: restrictions on commercial use.
  - Cost – Usages cost (including API calls, customization fees)
- Latency and Performance:
  - Response Time: Evaluate how quickly the model can generate outputs.

Amazon Bedrock - Foundation Models

- Amazon Bedrock makes a copy of the Foundation Model. The model is only available to you. Which can further be fine-tunes with custom data.

- None of your data is used to train the Foundational Model.

Compare Foundational Models

Attributes	Amazon Titan	Meta Llama 3	Claude 3	Stable Diffusion
Max Token	4K Tokens	128K Tokens	200K Tokens	77 Tokens/prompt
Features	Text generation, Summarization, Semantic Search, Image Generation	Language translate, coding assistance, synthetic data generation	Long context handling, text and image input, multilingual chat	Image generation, high customization
Use Cases	Content Creation, Recommendation, Image editing	Multilingual chat, coding, test data generation	Complex tasks, multilingual tasks, long-context tasks	Professional Image Generation
Pricing (100 K Tokens)	Input: $0.08 Output: $0.16	Input: $0.19 Output: $0.25	Input: $0.8 Output: $2.4	$0.04 – 0.08 per Image

Model Customization

- Process of improving model's performance for specific use-cases.

Patient Record,
Medical Journal

Training

Amazon Titan

Amazon Titan Text G1
General-Purpose Language model

Amazon Titan

Amazon Titan Text G1
Specialized in Healthcare

Expert in medical terminology

- You can customize Amazon Bedrock foundation models to improve their performance.
- All FMs doesn't support customization.

Model Customization

- Fine-Tuning
  Provide labeled data
- Continued pre-training
  Provide unlabeled data

Distillation (Preview)

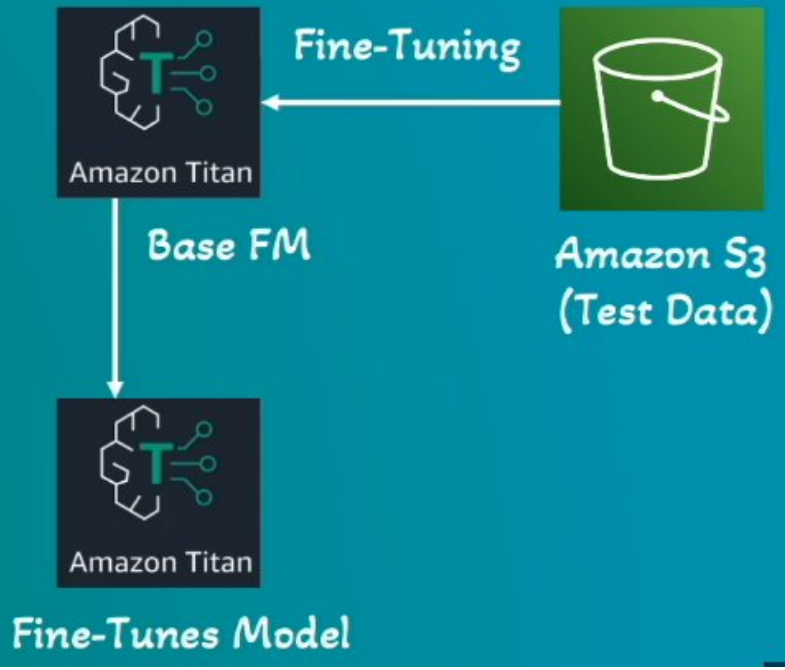
Generate synthetic data from an LLM (Teacher)

User the synthetic data to fine-tune a smaller model

Amazon Bedrock - Fine-Tuning

- Choose a model from the list of Available Foundation Models in Amazon Bedrock.
- Note - All foundation models can't be fine-tuned
- You can fine-tune one model/fine-tuning job.
- Amazon Bedrock creates a copy of the selected Foundation Model.
- Optionally encrypt model's artifact with Customer-Managed key.
- Training Data and Validation dataset (optional) must be stored in an Amazon S3 bucket.
- Training data must be in the specified format

  {"prompt": "what is AWS", "completion": "it's Amazon Web Services"}
- Bedrock supports hyperparameters customization (Steps, Epochs, Batch Size, etc.)
- Must purchase provisioned throughput to use fine-tuned model.



Model Fine-Tuning Use Cases

- Customer Support:
  - Fine-tune a model to answer FAQs for an e-commerce website.
  - Designed chatbot with a specific persona/tone/purpose.
- Content Generation:
  - Adapt a model to write marketing copy in your brand's tone.
- Healthcare:
  - Train a model to analyze patient data and suggest treatments.
  - Fine-tunes an image recognition model to detect anomalies in X-rays or MRIs.
- Fraud Detection:
  - Detect fake reviews or fraudulent accounts in e-commerce platform
  - Suspicious transactions, large withdrawals or purchase.

Fine-Tuning: Data Preparation

- Fine-Tuning is a supervised learning/Labeled Data:
- Fine-tuning: Text-to-text
  - Each data object must be in JSON format
  - Each sample must contain prompt and completion filed

```
{"prompt": "<prompt1>", "completion": "<expected generated text>"}
{"prompt": "<prompt2>", "completion": "<expected generated text>"}
{"prompt": "<prompt3>", "completion": "<expected generated text>"}
```

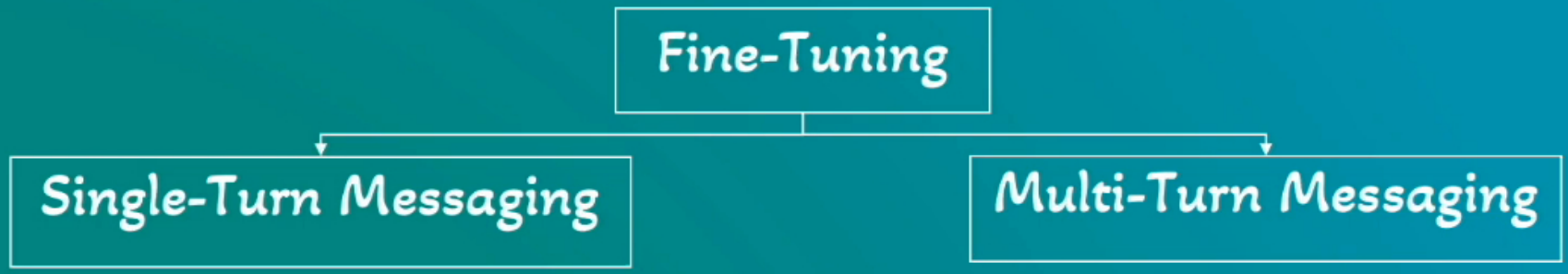
```
{"prompt": "what is AWS", "completion": "it's Amazon Web Services"}
```

- Fine-tuning: Text-to-image & Image-to-embedding
  - Each sample must image-ref and caption filed

```
{"image-ref": "s3://bucket/path/to/image001.png", "caption": "<prompt text>"}
{"image-ref": "s3://bucket/path/to/image002.png", "caption": "<prompt text>"}
{"image-ref": "s3://bucket/path/to/image003.png", "caption": "<prompt text>"}
```

```
{"image-ref": "s3://amzn-s3-demo-bucket/my-pets/cat.png", "caption": "an orange cat with white spots"}
```

Fine-Tuning: Single-turn messaging



- Train the model on individual prompt-response pairs.
- Tasks where context contain single interaction.

- Train the model on conversations with multiple back-and-forth exchanges.
- More complex tasks where context from previous message is important.

Fine-Tuning: Single-turn messaging Template

```
{
    "system": "You are a helpful assistant."
    "messages": [
        {"role": "user", "content": "what is AWS"}
        {"role": "assistant", "content": ”it’s Amazon Web Services"}
        ]
}
```

Fields:

- system (optional) : sets the context for the conversation.
- messages : An array of message objects, each containing:
  - role : Either user or assistant
  - content : message text

Rules:

- The message array must contain 2 messages.
- First message must have a role of the user
- Last message must have a role of the assistant

Fine-Tuning: Multi-turn messaging Template

```
{ "system": "You are a helpful assistant."
    "messages": [{"role": "user", "content": “Hello there.”}
                 {"role": "assistant", "content": “Hi, how can I help you?”}
                 {"role": "user", "content": “What are LLMs?”}
                 {"role": "assistant", "content": “LLM means large language model.”}
}
```

Fields:

- system (optional) : sets the context for the conversation.
- messages : An array of message objects, each containing:
  - role : Either user or assistant
  - content : message text

Rules:

- The message array must contain 2 messages.
- First message must have a role of the user
- Last message must have a role of the assistant
- Message must alternative between user and assistant roles

Continues Pre-training

- Involves updating a model with new data over time to keep the model relevant and accurate.
- Provide **unlabeled data** for the model training
- Domain-adaption fine-tuning (proficient in specific domain)

Collection of medical records, patient interactions, and healthcare articles.

Amazon Bedrock to continuously pre-train the model with this data.

Proficient in understanding and generating text related to healthcare.

Benefits

- Domain-Specific Knowledge
- Improved Performance
- Customization

Fine-Tuning vs Continues Pre-training

aws BEDROCK

Fine-Tuning

Fine-tuning involves training a pre-trained model on a smaller, domain-specific dataset.

Requires intermediate ML skills.

Relatively quick to set up, especially if using pre-trained models.

Can be completed in hours to days, depending on the dataset size and model complexity

Cost-effective

Continuous Pre-training

Continuous pre-training involves updating a model with a large amount of new, unlabeled data.

Advance ML skills are needed.

Longer setup time for extensive data collection and preprocessing

Can take weeks to months, depending on the scale of data and model size

Significantly more expensive due to the extensive computational resources required

Transfer Learning

- Leverage pre-trained model and adapt to a specific task or domain.
- Example: Create a customer support chatbot for e-commerce website

Pre-trained language model trained on general conversational data.	Upload customer support dataset to Amazon S3.	Use fine-tuning feature in Amazon Bedrock to train the model

- Fine-tuning is a specific types of transfer learning
- Benefits of Transfer Learning
- Efficiency: Reduces the amount of data and computational resources needed compared to training a model from scratch.
- Performance: Better performance on specific tasks due to the model's prior knowledge.
- Speed: Faster to deploy as it leverages existing pre-trained models.

Why do we need Model Evaluation?

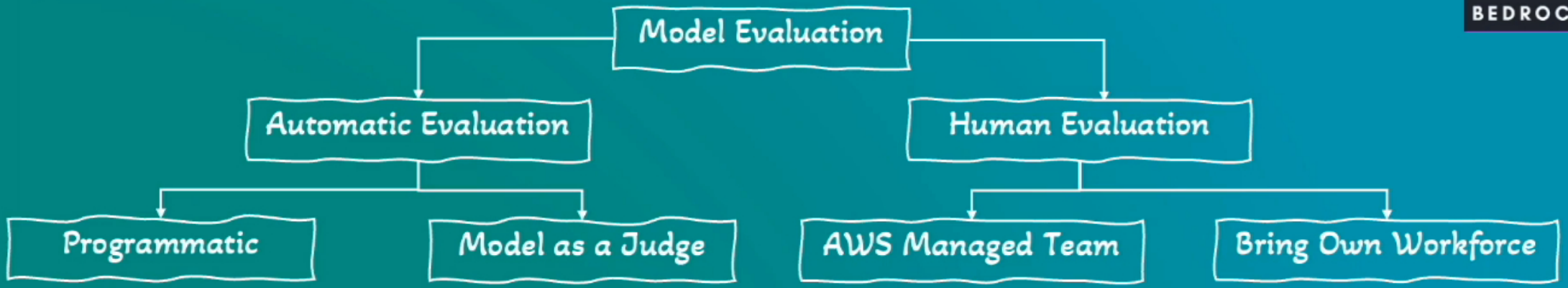
amazon BEDROCK

- Use Case: Chatbot → evaluate models like **Anthropic Claude** vs **Cohere Command**
- Evaluate multiple models on task-specific metrics
- Model Evaluation helps you select right model by comparing performance.
- Without evaluation, you cannot make informed decisions .

Customer Support Chatbot

Model Selection

Amazon Bedrock Model Evaluation



Model evaluation Info

Create and review model evaluation jobs

▼ How it works

Automatic

The automatic approach offers 2 options for evaluation:

- **Programmatic**: Evaluate performances using just the model and metrics you select.
- Public Preview **Model as a judge**: A pre-trained model evaluates your model's responses using metrics you've selected.

Create ▼

Human

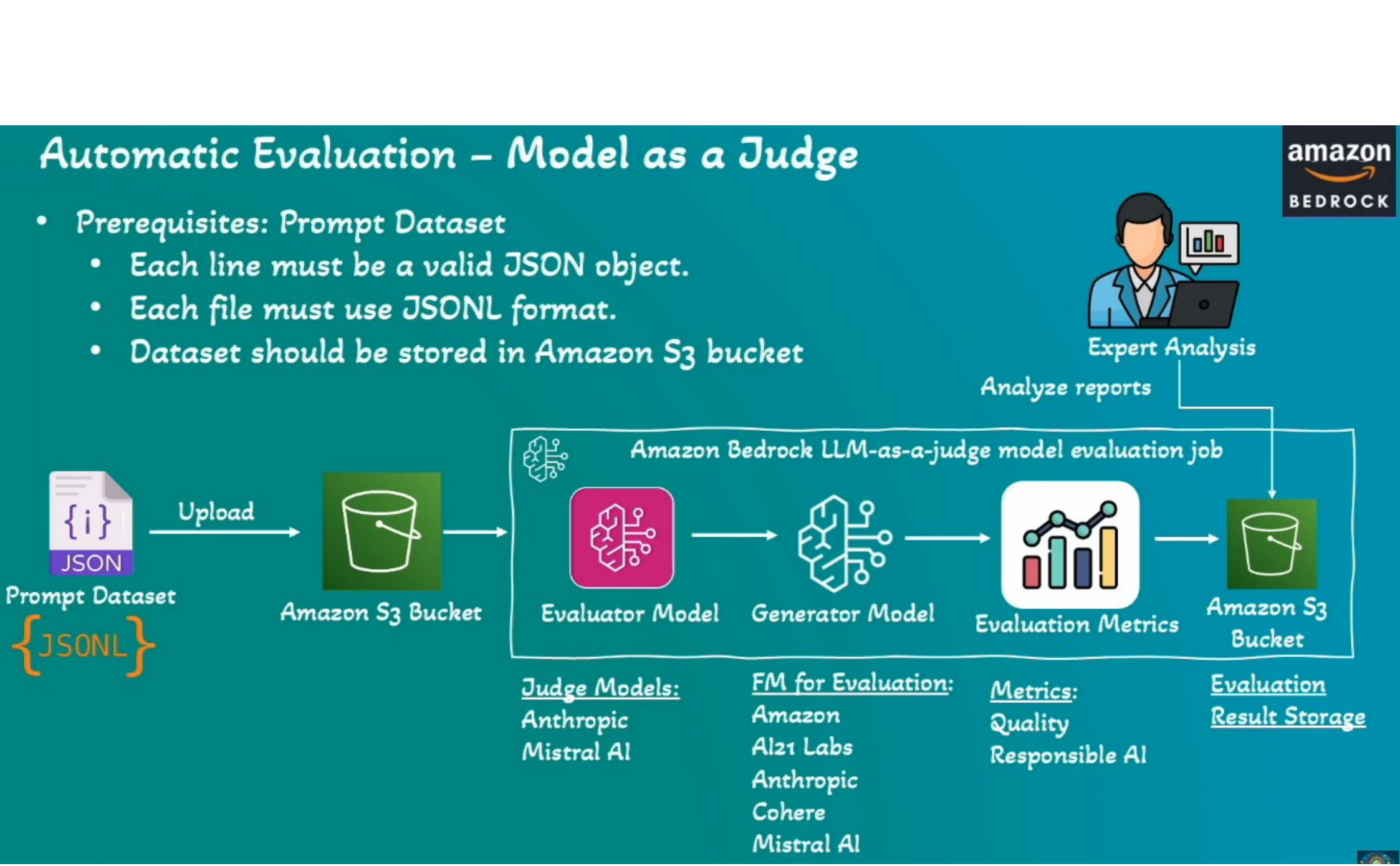
The human approach offers 2 options for evaluation:

- **AWS Managed work team**: Use an AWS curated work team to evaluate responses from up to 2 models. You can define evaluation metrics specific to your job.
- **Bring your own work team**: Evaluate responses from up to 2 models using your own work team. You can define evaluation metrics specific to your job.

Create ▼

Automatic Evaluation – Model as a Judge

- Prerequisites: Prompt Dataset
  - Each line must be a valid JSON object.
  - Each file must use JSONL format.
  - Dataset should be stored in Amazon S3 bucket

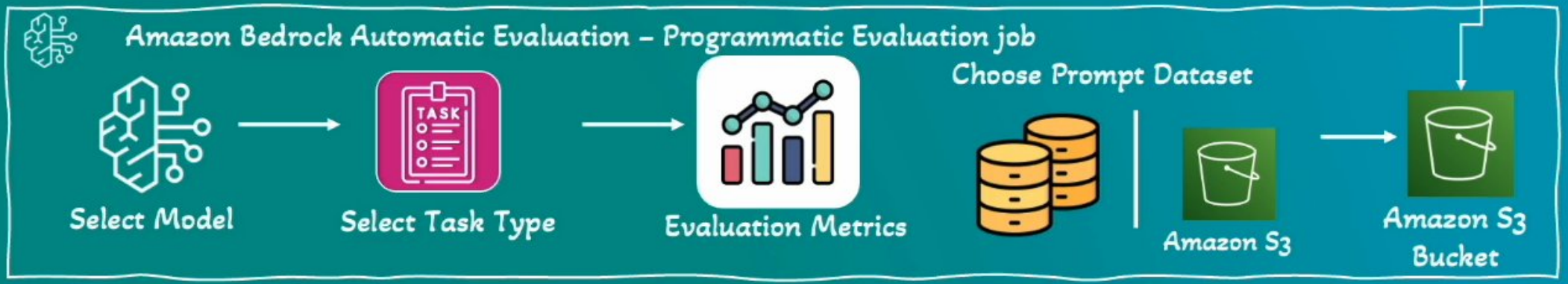


Automatic Evaluation – Programmatic

amazon BEDROCK

- Uses traditional NLP algorithms and metrics to evaluate model
- For tasks like classification, regression or structured prediction.
- When you have labeled data and need objective metrics (BLEU/ ROUGE).
- Option to use built-in prompt dataset or own custom datasets.

Expert Analysis



FM for Evaluation:
Amazon
AI21 Labs
Anthropic
Cohere
Mistral AI

Task Type:
General text generation
Text Summarization
Text Classification
Question and answer

Metrics:
Toxicity
Accuracy
Robustness

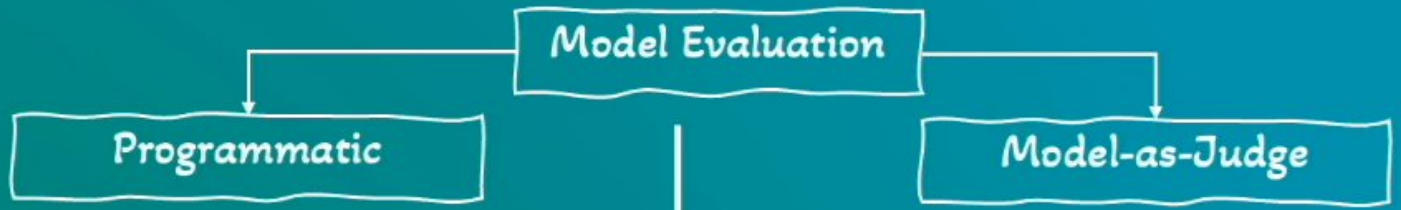
Built-in Dataset

Own Custom Dataset

Evaluation Result Storage

Automatic Evaluation – Programmatic vs LLM-as-Judge

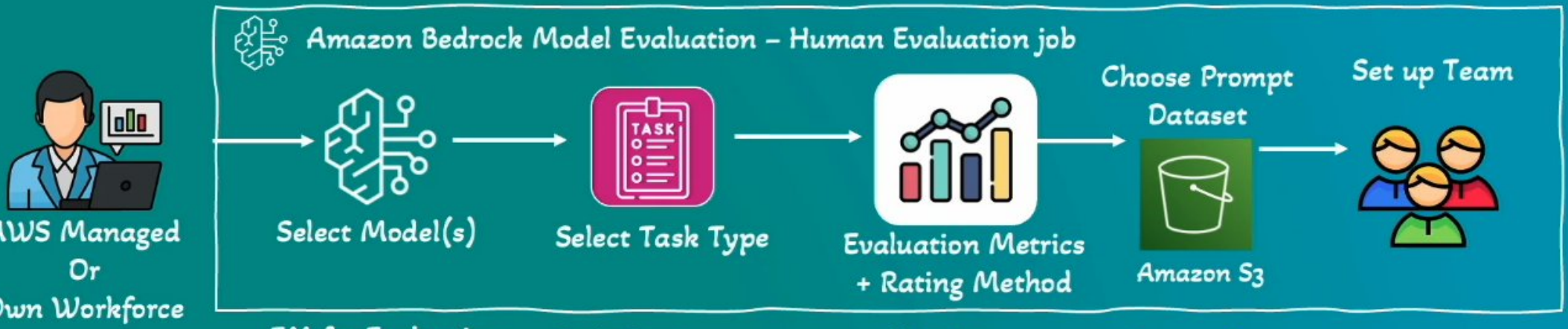
amazon BEDROCK



	Programmatic	Model-as-Judge
• Use Case	Objective Tasks (e.g., classification, regression)	Subjective Tasks, requires human judgement (e.g., text quality, creativity)
• Metrics	Objective metrics (e.g., F1-score, BLEU, ROUGE)	Objective metrics are hard to define (e.g., fluency, relevance)
• Example	Spam detection, Sentiment analysis, Image classification	Chatbot response evaluation, Creative writing assessment
• Dataset	Requires labeled dataset	No labeled data needed, uses LLM judgement
• Scalability	Highly scalable for large dataset (when you need to evaluate thousands of output quickly)	

Model Evaluation – Human Evaluation

- Subjective Tasks – quantity of output is subjective and requires human judgement.
- Complex Tasks – automatic metrics may not capture full context or nuances.
- High-Stakes Applications – apps where errors can have significant consequences.
- Option to select AWS Managed work team or Own workforce



Can select two models for Evaluation

FM for Evaluation:
Amazon
AI21 Labs
Anthropic
Cohere
Mistral AI

Task Type:
General text generation
Text Summarization
Text Classification
Question and answer
Custom

Metrics:
Toxicity
Accuracy
Robustness

Prompt Dataset
+
Evaluation Result

Provide Detailed Instruction

Result 1
Result 2
Result 3

Automated Model Evaluation Metrics - ROUGE

amazon BEDROCK

- ROUGE (Recall-Oriented Understudy for Gisting Evaluation).
  - Evaluates the quality of text summarization by comparing the overlap between generated text and reference text.
  - ROUGE-N: Measures overlap of n-grams (ROUGE-1 for unigrams, ROUGE-2 for bigrams)
  - ROUGE-L: Measures the longest common subsequence between generated and reference

Example:

Reference Text: "The cat sat on the mat" || Generated Text: "The cat was on the mat"

ROUGE-1 (Unigrams):

•Reference unigrams: {"The", "cat", "sat", "on", "the", "mat"}.

•Generated unigrams: {"The", "cat", "was", "on", "the", "mat"}.

•Overlapping unigrams: {"The", "cat", "on", "the", "mat"}.

•ROUGE-1 Recall: $\frac{5}{6} = 0.83$

ROUGE-2 (Bigrams):

•Reference bigrams: {"The cat", "cat sat", "sat on", "on the", "the mat"}.

•Generated bigrams: {"The cat", "cat was", "was on", "on the", "the mat"}.

•Overlapping bigrams: {"The cat", "on the", "the mat"}.

•ROUGE-2 Recall: $\frac{3}{5} = 0.6$.

ROUGE-L (Longest Common Subsequence):

•LCS: "The cat on the mat"

•Total Words: 6

•ROUGE-L: $\frac{Length\ of\ LCS}{Total\ number\ of\ words\ in\ the\ reference\ text}$

•ROUGE-L Recall: $\frac{5}{6} = 0.83$

Automated Model Evaluation Metrics - BLEU

amazon BEDROCK

- BLEU (Bilingual Evaluation Understudy).
  - Evaluates the quality of machine translation or text generation by comparing n-gram overlap between generated text and reference text.
  - It focuses on precision (how much of the generated text matches the reference).
  - Key Features – uses a brevity penalty to penalize overly short translations.
  - Typically calculated for n-grams (e.g., BLUE-1 to BLUE-4)
  - Use Case – Machine translation, text generation
  - BLUE-1 = Precision X Brevity Penalty

Example:

Reference Text: "The cat sat on the mat" || Generated Text: "The cat was on the mat"

BLUE-1 (Unigrams):

- Reference unigrams: {"The", "cat", "sat", "on", "the", "mat"}.
- Generated unigrams: {"The", "cat", "was", "on", "the", "mat"}.
- Matching unigrams: {"The", "cat", "on", "the", "mat"}.
- Precision = $\frac{\text{Number of matching unigrams}}{\text{Total unigrams in generated text}} = \frac{5}{6} = 0.83$

Step 2: Calculate Brevity Penalty

- Length of Generated Text (c): 6 words.
- Length of Reference Text (r): 6 words.
- Since c=r, the brevity penalty is: 1

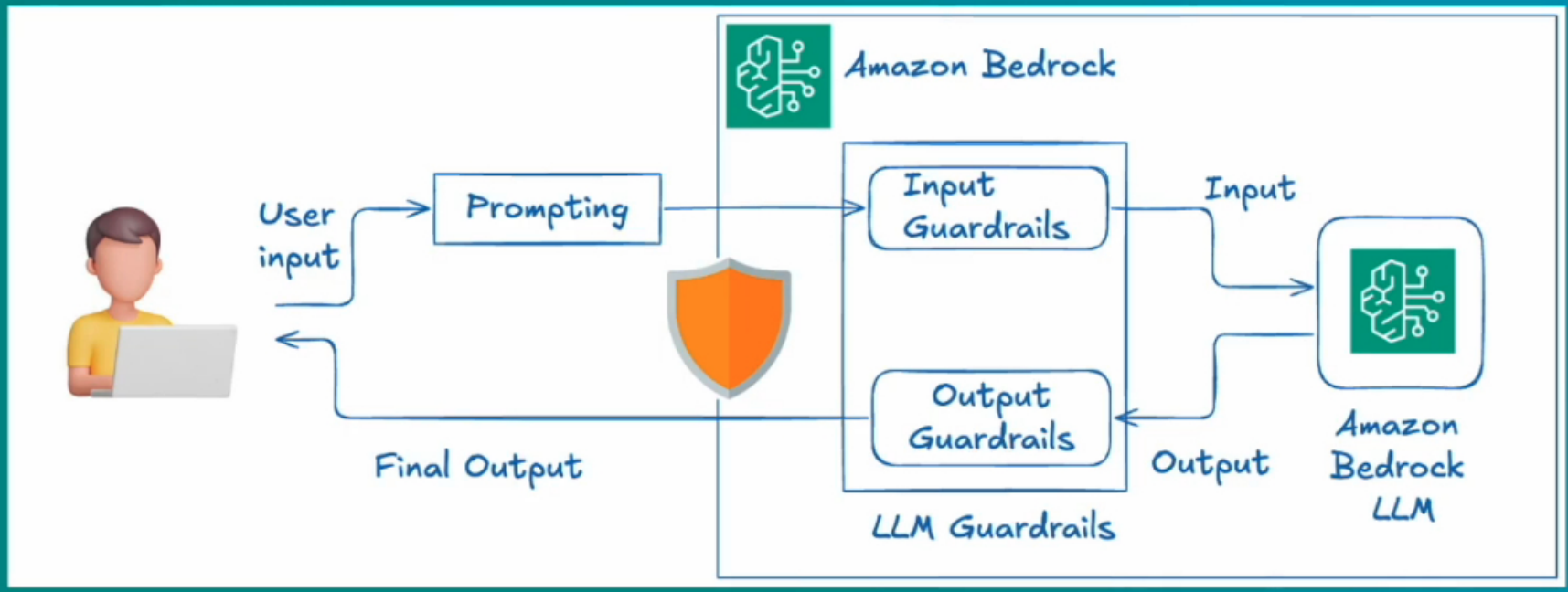
BLEU-1= Precision × Brevity Penalty = 0.83×1 = 0.83

Automated Model Evaluation Metrics

amazon BEDROCK

- **F1 Score**
  - F1 score is the harmonic mean of precision and recall.
  - Commonly used for classification task (Binary or multi-class classification).
    - Example - Spam detection, sentiment analysis.
  - $F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$
  - Precision – How many predicted positives are actually positive.
  - Recall – How many actual positives are correctly predicted
- **BERTScore**
  - Evaluates generated text by comparing semantic similarity.
  - Captures contextual meaning rather than just n-gram overlap.
    - Use Cases – Text generation, summarization, translation.
  - When evaluating the semantic quality of generated text, especially for task where meaning is more important than exact word overlap.
- **Perplexity**
  - Evaluates language models by measuring how well the model predicts a sequence of words.
  - Lower perplexity indicates better performance. (Perplexity 20 is better than perplexity 50).

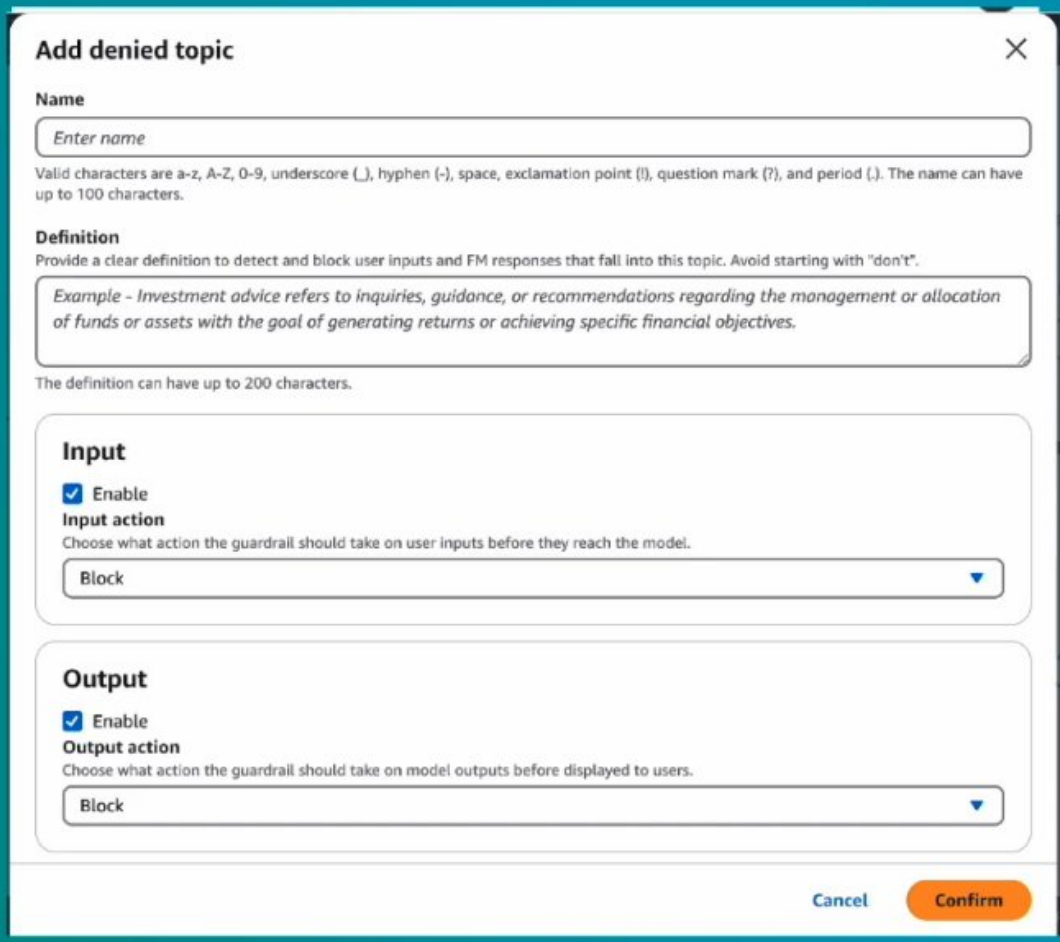
Why is Amazon Bedrock Guardrail?



- Amazon Bedrock Guardrails allows you to define safety controls for your AI models.
- Block harmful content, filter sensitive topics, enforce company policies.
- Amazon Bedrock Guardrail = Firewall for GenAI Model .

Amazon Bedrock Guardrail Key Features

- Content Filtering - Block hate speech, violence, or inappropriate content.
- Prompt Attacks Filtering – Detects and block prompt attacks. Use Input tagging.
- PII Redaction – Automatically mask sensitive info like emails or credit card numbers.
- Custom Policies – Define your own rules (e.g., no medical advice, no legal opinions).
- Multi-Model Support – Works with Claude, Llama 2, Titan, and more on Bedrock.
- Ability to create multiple Guardrails



Best Practices – Amazon Guardrails

amazon BEDROCK

Start with AWS Predefined Filters, the Customize	• First, enable AWS's default filters for quick protection. • Later, refine with custom rules based on your industry (e.g., finance, healthcare).
Use Denied Topics for Industry-Specific Restrictions	• Block sensitive keywords (e.g., "confidential," "SSN," "trade secrets"). • Example: A financial AI chatbot should block prompts like *"How to evade taxes?"*
Enable PII Redaction for Privacy Compliance	• Redact emails, phone numbers, credit cards, SSNs. • Test with prompts like *"My email is test@example.com"* → Output should show "[EMAIL REDACTED]".
Combine Guardrails with Prompt Engineering	• Use system prompts like *"You are a helpful assistant. Never provide medical advice."* • Guardrails act as a safety net if the AI still misbehaves.
Monitor & Log Violations for Continuous Improvement	• Use AWS CloudWatch Logs to track blocked requests. • Update guardrails monthly based on trends.

Amazon Bedrock Guardrails – Use Cases

- **Banking & Financial Services AI Chatbots**
  - Problem: Customer might ask for fraudulent advice
  - Solution:
    - Block financial crime-related keywords (e.g., “money laundering”, “fake checks”)
    - Redact account numbers, SSNs, credit card details
  - Example:
    - User Input: "How do I transfer money anonymously?"
    - Guardrail Action: Blocked + Warning: "This violates financial regulations."
- **Healthcare & Medical AI Assistants**
  - Problem: AI giving unverified medical advice could be dangerous.
  - Solution:
    - Block diagnosis & treatment suggestions unless from a verified source
  - Example:
    - User Input: ”What’s the best treatment for COVID?"
    - Guardrail Action: Redirect to CDC.gov instead of generating an answer"

Amazon Bedrock Guardrails – Use Cases

- Customer Support AI (Preventing Offensive Responses)
  - Problem: Angry customers might use abusive language, triggering harmful replies
  - Solution:
    - Filter profanity, threats, hate speech
  - Example:
    - User Input: "Your product is trash, you idiots!"
    - Guardrail Action: Block Response → Show: "let me help professionally. What's the issue?"
- Enterprise AI (Preventing Data Leaks)
  - Problem: Employees might accidentally leak secrets via AI.
  - Solution:
    - Block internal project names, confidential data.
  - Example:
    - User Input: "Summarize the Q2 earnings report before release. "
    - Guardrail Action: Block + Alert compliance team"

What is RAG

aws BEDROCK

- RAG stands for ***Retrieval-Augmented Generation.***
  - **Retrieval:** Fetching relevant data from a knowledge base.
  - **Generation:** Using an LLM to generate accurate answers.

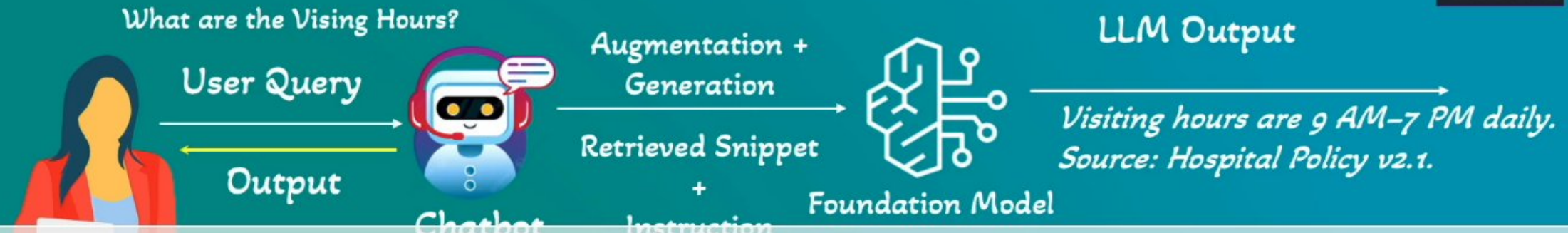
Real-Life Example: Healthcare Customer Support Chatbot

A hospital's customer support team is overwhelmed with repetitive questions like:

- *"What are the visiting hours?"*
- *"How do I reschedule an appointment?"*
- *"What's covered under my insurance plan?"*

Solution: Use Amazon Bedrock RAG to create a chatbot that pulls answers from the hospital's latest documents (PDFs, websites, databases).

What is RAG



How RAG Works in Amazon Bedrock

- Prerequisite – Setup the Knowledge Base
- User Query: “What are the visiting hours?”
- Retrieval: Bedrock searched knowledge base for Guest Hours.
- Augmentation: The FM (Amazon Titan) reads the retrieved docs.
- Generation: The AI responds with an accurate, cited answer.

Why we use RAG?

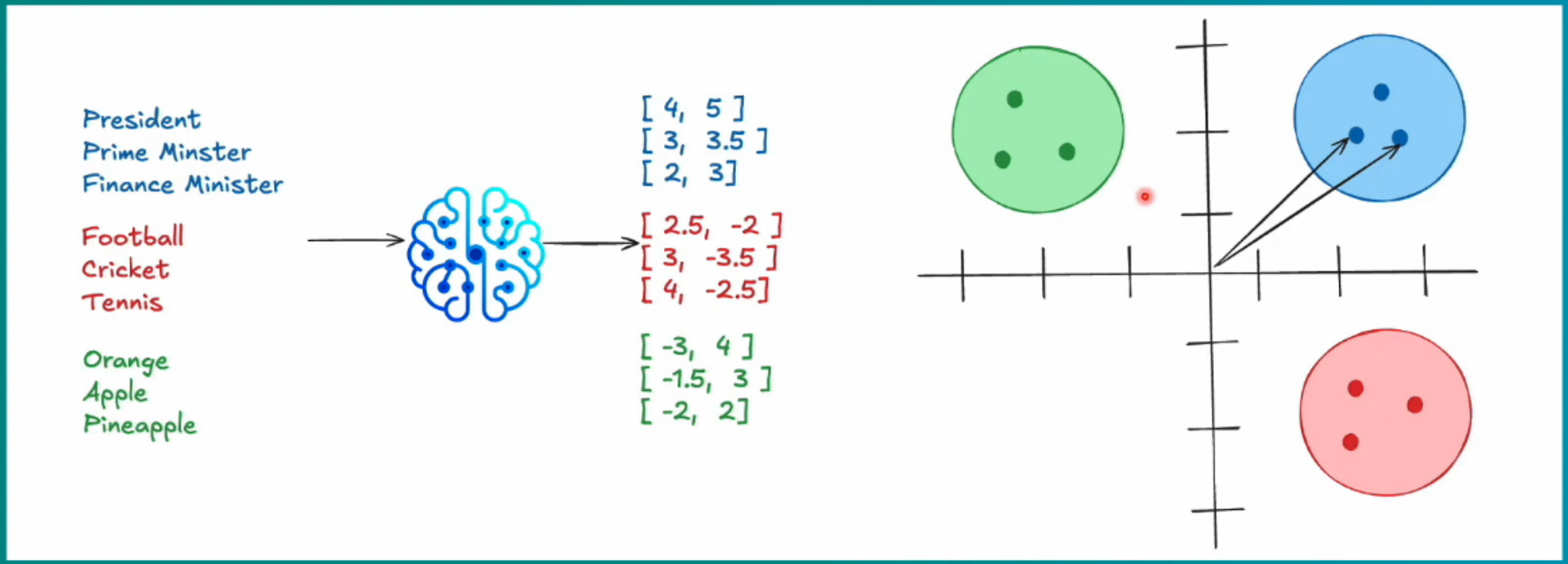
aws BEDROCK

- **Avoids Hallucinations:** LLMs sometimes make up answers. RAG ensures responses are based on real data.
- **Domain-Specific Accuracy:** Need AI for healthcare or legal docs? RAG pulls from specialized sources.
- **Cost-Effective:** Fine-tuning an LLM is expensive. RAG is cheaper and faster to implement.
- **Answer Accuracy:** The answers are accurate, including citation.

Without RAG, the LLM might guess. With RAG, it cites the official policy!"

Text Embedding / Vector Embedding

- A vector embedding is a numeric representation of text, images, or other data as a series of numbers
- Each word is assigned a vector (array of numeric values) that represents the word in an abstract vector space.



How are Embedding Created?

Why Numbers?

Computers understand math, not words. Numbers allow machines to calculate similarity

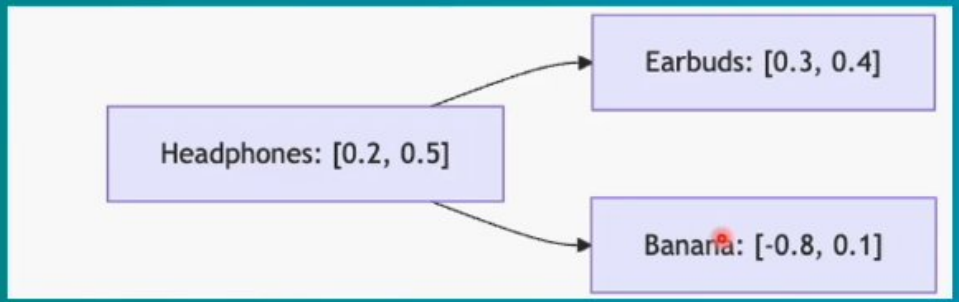
Dimensionality - AWS Titan Embeddings use 1536 dimensions

More dimensionality = Richer Meaning

How Embeddings Power RAG in Amazon Bedrock?

"Which headphones have the best noise cancellation?"

- Bedrock converts the query into a vector.
- Searches the vector database for similar vectors (e.g., product docs mentioning *"noise-canceling earbuds"*).
- Retrieves the closest matches, even if the words don't exactly match.

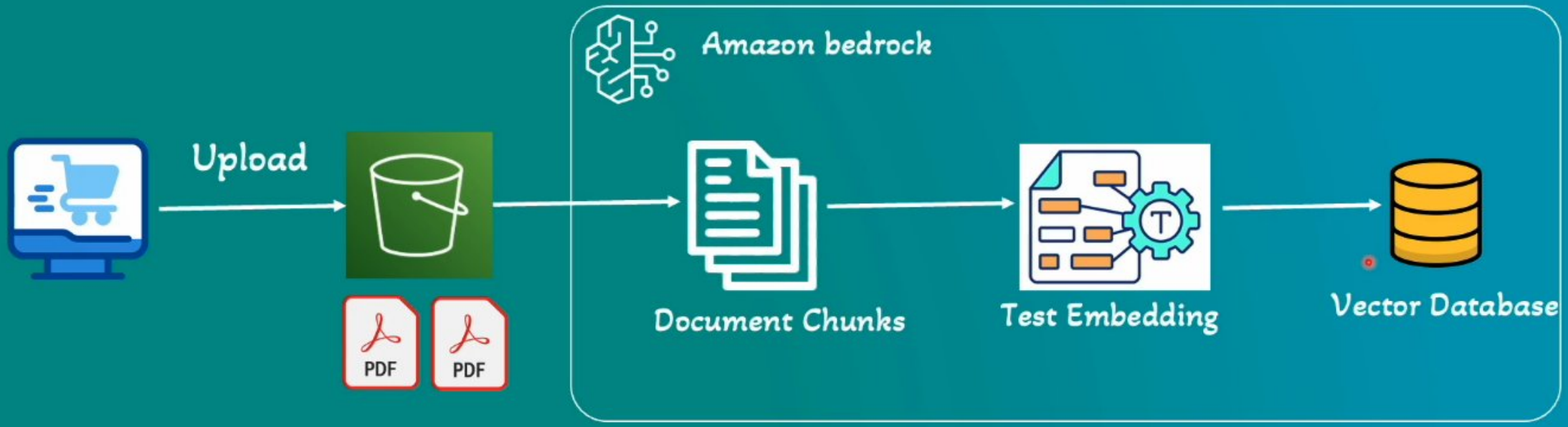


Amazon Bedrock Vector Database

- A specialized database that stores data as vector embeddings
- Enables semantic search (finding meaning, not just keywords).

Real-Life Example: E-Commerce Product Search

Scenario: An online store wants its chatbot to answer product questions.



Supported Vector Databases in Amazon bedrock

aws BEDROCK

Vector Database	AWS Integration	Best For	Key Features
Amazon OpenSearch	Native (Serverless option available)	Full-text + vector hybrid search	• Supports filtering by metadata • Built-in AWS security (IAM)
Pinecone	External (API-based)	High-scale semantic search	• Optimized for low-latency • Auto-scaling
Redis Enterprise Cloud	External (API-based)	Real-time applications	• Ultra-fast caching • Supports JSON/vector hybrid queries
Aurora PostgreSQL	Native (with pgvector extension)	Existing PostgreSQL users	• SQL + vector queries in one database • Cost-effective

Supported Vector Databases in Amazon bedrock

Vector Database	AWS Integration	Best For	Key Features
Amazon OpenSearch	Native (Serverless option available)	Full-text + vector hybrid search	• Supports filtering by metadata • Built-in AWS security (IAM)
Pinecone	External (API-based)	High-scale semantic search	• Optimized for low-latency • Auto-scaling

When to use which Vector Database?

- OpenSearch: General-purpose RAG (balanced performance/features).
- Pinecone: Large-scale, low-latency needs.
- Redis: Real-time apps with caching.
- Aurora: Existing PostgreSQL workloads.

Exam Tips:

aws BEDROCK

- **Bedrock RAG Source Database – Amazon S3, Confluence, SharePoint, Salesforce, Web pages (website, social media page).**
- **RAG Vector Database - Pinecone and Redis are external, while OpenSearch/Aurora are AWS-native.**
- **RAG is ideal for dynamic, document-heavy use cases (support, legal, healthcare).**
- **RAG vs. Fine-Tuning:**
  1. **RAG is cheaper/faster for dynamic data (e.g., policy updates).**
  2. **Fine-tuning is better for teaching the model *new behaviors* (e.g., medical jargon).**
- ***Know the supported data sources: S3, Salesforce, SharePoint, and websites."***
- ***For the AWS exam, remember: RAG improves accuracy by grounding responses in external data."***
- ***Expect scenario-based questions on when to use RAG vs. fine-tuning***

What is Amazon Bedrock Agent?

aws certified AI Practitioner FOUNDATIONAL

- A managed orchestration tool for generative AI apps
- Build GenAI applications using Foundation Model (FM) like Claude, Llama 2
- Helps build agents that can reason, plan, and act
- Abstracts the heavy lifting, and no infrastructure to manage

Amazon Bedrock Agent vs Traditional Chatbots

Bedrock Agent

VS

Traditional Chatbots

Traditional chatbots follow predefined rules, but Bedrock Agents understand intent, retrieve knowledge, and execute tasks dynamically.

Use Case: Check refund status of an amazon.com return order

aws certified AI Practitioner FOUNDATIONAL

Support Associate

Talking to a concierge

Deciphers your request

Concierge checks records

Concierge Calls the billing department

Concierge provides the refund status update

Traditional Chatbots

Authenticates you

Understand your intent

Forward you the link to visit order portal or transfer to Manual Agent

Bedrock Agent

What's my refund status?

Understand your intent

Searches S3/PDFs/DBs for refund policy

Triggers an API to check payment system

Your refund was processed today!

Amazon Bedrock Agent = Your Super-Smart Personal Assistant

- Understands you (natural language).
- Knows company policies (connected to knowledge bases).
- Can take actions (like checking systems or placing orders).

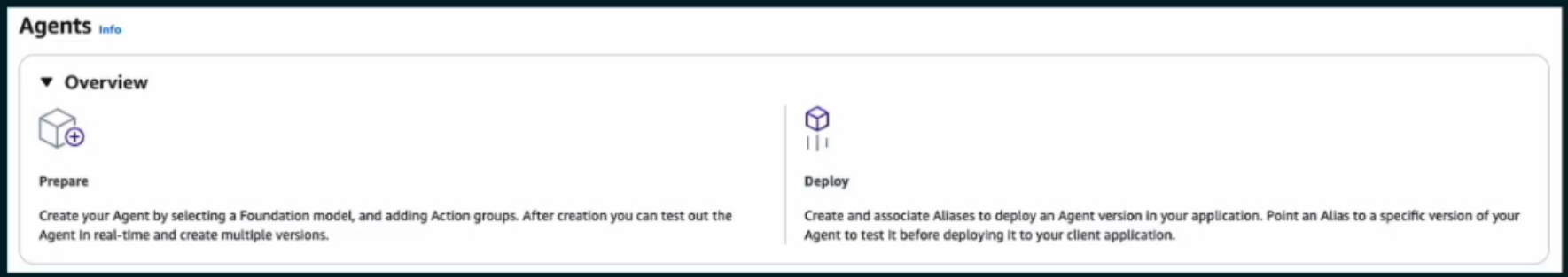
Key Features

- No-Code/Low-Code Setup
- Knowledge Base Integration
- Action Execution
- Multi-Model Support
- Secure & Scalable

Business Benefits:

Companies use Bedrock Agents to reduce manual work, improve response times, and personalize interactions—all without deep AI expertise.

How to create a Bedrock Agent



Step 1: Choose a Foundation Model - *Select a model (e.g., Claude, Llama 2) in Bedrock*

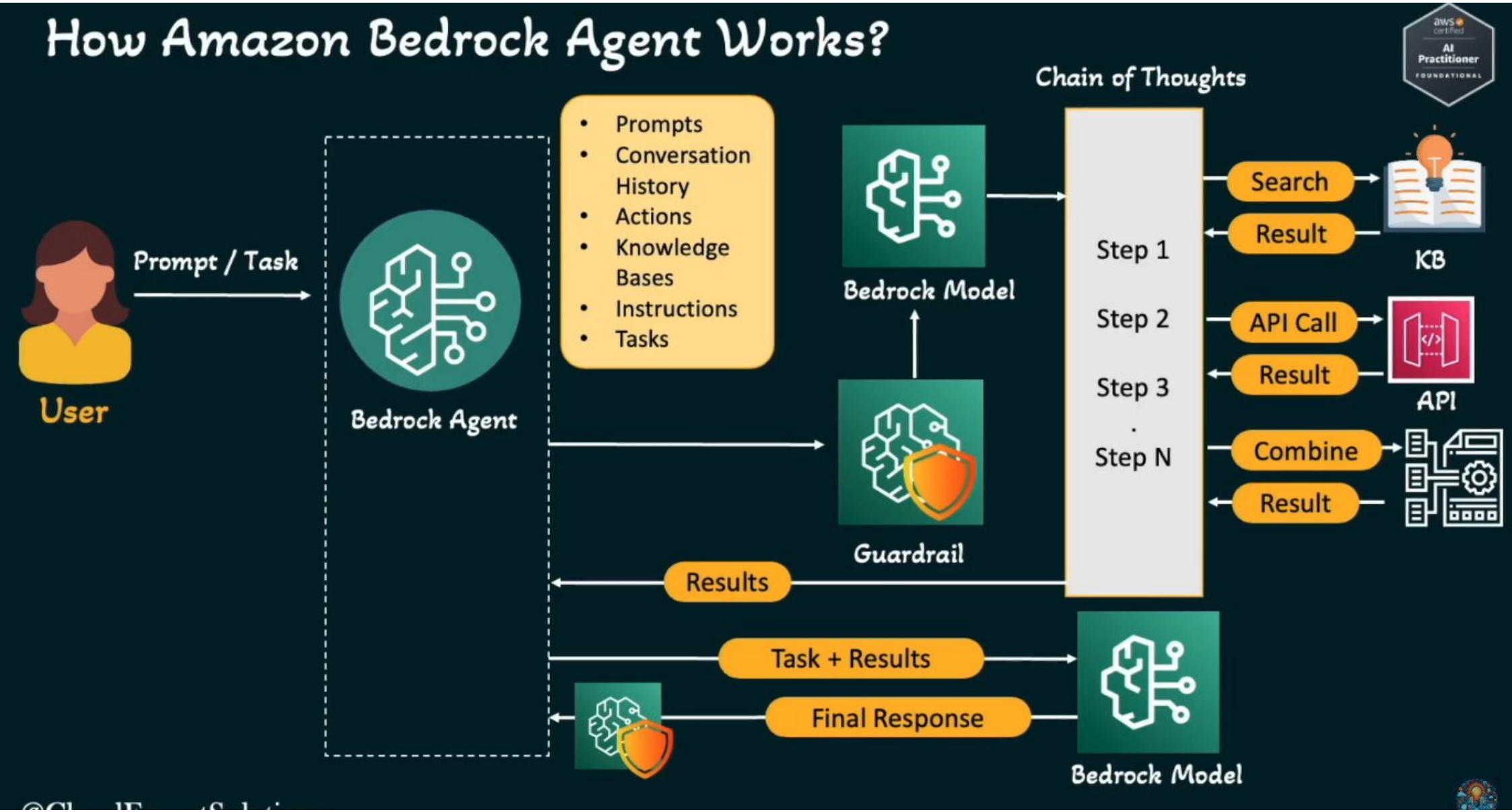
Step 2: Define Instructions - *Tell the agent its role (e.g., 'You are an IT support bot').*

Step 3: Connect Knowledge Bases - *Link S3, databases, or FAQs for the agent to reference.*

Step 4: Add Actions (APIs) - *Define APIs it can call (e.g., 'Check order status').*

Step 5: Deploy and Test

How Amazon Bedrock Agent Works?



Amazon Bedrock Agent Use Cases

Customer Support – Answering FAQs, Troubleshooting

IT Helpdesk – Reset Password, diagnosing issues

eCommerce – order tracking, product recommendations

Exam Tips:

- Question is asking to automate set of steps or chain of thoughts
- Reduce human agent to automate customer interactions
- GenAI chatbot for order tracking
- Thousands of repetitive tasks handled by support agents
- Remember the benefits of Amazon Bedrock Agent