

WEBINAR

Best Practices to build Scalable AI Solutions on AWS

Terumi Laskowsky

AWS Authorized Instructor - Champion

Visit Now

www.netcomlearning.com



AGENDA

- 1 Overview of AWS Machine Learning Services
- 2 Best Practice for Data Preparation and Feature Engineering
- 3 Architectural Consideration for Scalable AI Solutions on AWS
- 4 Case Studies and Real-world Examples
- 5 Q&A Session

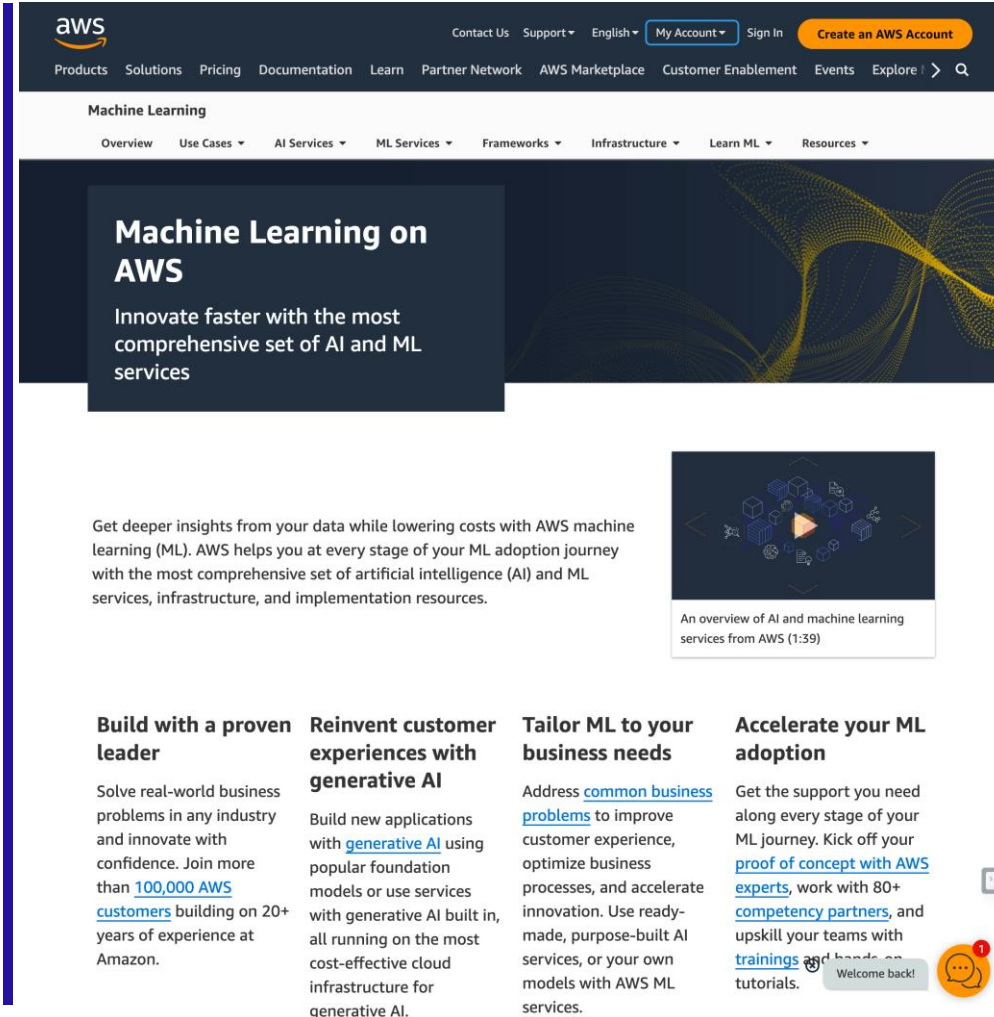
AGENDA

- 1 Overview of AWS Machine Learning Services
- 2 Best Practice for Data Preparation and Feature Engineering
- 3 Architectural Consideration for Scalable AI Solutions on AWS
- 4 Case Studies and Real-world Examples
- 5 Q&A Session

OVERVIEW OF AWS MACHINE LEARNING SERVICES

AWS Machine Learning Site

<https://aws.amazon.com/machine-learning/>



The screenshot shows the AWS Machine Learning website. The header includes the AWS logo, navigation links (Products, Solutions, Pricing, Documentation, Learn, Partner Network, AWS Marketplace, Customer Enablement, Events, Explore), and user options (Contact Us, Support, English, My Account, Sign In, Create an AWS Account). The main content area features a large banner titled "Machine Learning on AWS" with the text "Innovate faster with the most comprehensive set of AI and ML services". Below the banner, there is a section titled "Get deeper insights from your data while lowering costs with AWS machine learning (ML). AWS helps you at every stage of your ML adoption journey with the most comprehensive set of artificial intelligence (AI) and ML services, infrastructure, and implementation resources." This section includes a video thumbnail titled "An overview of AI and machine learning services from AWS (1:39)". At the bottom, there are four columns of content: "Build with a proven leader" (Solve real-world business problems in any industry and innovate with confidence. Join more than 100,000 AWS customers building on 20+ years of experience at Amazon.), "Reinvent customer experiences with generative AI" (Build new applications with generative AI using popular foundation models or use services with generative AI built in, all running on the most cost-effective cloud infrastructure for generative AI.), "Tailor ML to your business needs" (Address common business problems to improve customer experience, optimize business processes, and accelerate innovation. Use ready-made, purpose-built AI services, or your own models with AWS ML services.), and "Accelerate your ML adoption" (Get the support you need along every stage of your ML journey. Kick off your proof of concept with AWS experts, work with 80+ competency partners, and upskill your teams with trainings and hands-on tutorials.).

OVERVIEW OF ML SERVICES



Overview

Amazon Web Services (AWS) offers a variety of machine learning services and tools catering to both beginners and experienced data scientists. These services can be broadly categorized as follows:

- **High-level Services:** Services that offer built-in models and require minimal ML expertise
- **Mid-level APIs:** Services that are flexible and make common tasks easier
- **Lower-level Frameworks and Infrastructure:** Services that offer the most control and customization but require more expertise and manual setup
- **Specialized Libraries and Algorithms**
- **ML Ops and Data Pipeline**

OVERVIEW OF ML SERVICES: HIGH-LEVEL SERVICES



High-level Services

- **Amazon SageMaker:** A fully-managed platform to build, train, and deploy machine learning models. SageMaker provides a set of modular capabilities for the entire machine learning workflow.
- **Amazon Comprehend:** A natural language processing (NLP) service that uses machine learning to find insights in text.
- **Amazon Lex:** Provides automatic speech recognition (ASR) and natural language understanding (NLU) capabilities for building conversational interfaces.
- **Amazon Polly:** A text-to-speech service that turns text into lifelike speech.
- **Amazon Rekognition:** An image and video analysis service that can identify objects, scenes, faces, and even detect unsafe content.

OVERVIEW OF ML SERVICES: MID-LEVEL SERVICES



Mid-level Services

- **AWS DeepLens:** A deep learning-enabled video camera for developers.
- **Amazon Translate:** Real-time and batch text translation services.
- **Amazon Transcribe:** Automatic speech recognition to convert speech into text.
- **Amazon Forecast:** Time-series forecasting service based on machine learning.
- **Amazon Personalize:** Provides real-time personalization and recommendation.

OVERVIEW OF ML SERVICES: LOW-LEVEL SERVICES



Low-level Frameworks and Infrastructure

- **Elastic Inference:** Adds just the right amount of inference acceleration to an Amazon EC2 or SageMaker instance.
- **AWS Deep Learning AMIs (Amazon Machine Images):** Pre-installed with popular deep learning frameworks like TensorFlow, PyTorch, and MXNet.
- **AWS Inferentia:** A machine learning inference chip designed to deliver high performance at low cost.
- **EC2 Instances Optimized for Machine Learning:** Special EC2 instances like the p3 and g4 instances optimized for machine learning tasks.

OVERVIEW OF ML SERVICES: SPECIALIZED LIBRARIES



TensorFlow

Specialized Libraries

- **MXNet:** An open-source deep learning framework designed for both efficiency and flexibility.
- **TensorFlow:** An open-source software library for high-performance numerical computation.
- **PyTorch:** An open-source machine learning library for Python, based on Torch.
- **Chainer:** An open-source deep learning framework written in Python.
- **Keras:** A high-level neural networks API running on top of TensorFlow, CNTK, or Theano.
- **Scikit-learn:** A tool for data mining and data analysis built on NumPy, SciPy, and matplotlib.
- **Spacy:** An open-source software library for advanced natural language processing.
- **NLTK:** A leading platform for building Python programs to work with human language data.

OVERVIEW OF ML SERVICES: SPECIALIZED ALGORITHMS



Specialized Algorithms

- **XGBoost:** Gradient boosting framework that you can use for classification, regression, and ranking problems.
- **Random Forest:** An ensemble learning method for classification, regression, and other tasks.
- **K-Means Clustering:** A type of unsupervised learning used for clustering unlabelled data.
- **Linear Learner:** A method for large-scale linear models that are tailored to distributed environments.
- **DeepAR:** A forecasting method for time-series data using recurrent neural networks (RNN).
- **Seq2Seq:** Sequence-to-sequence learning for tasks like machine translation.
- **Principal Component Analysis (PCA):** Used for dimensionality reduction in machine learning.
- **Factorization Machines:** General predictors working well for sparse data sets and frequently used for recommendation.

OVERVIEW OF ML SERVICES: LAMBDA

Role of Lambda

AWS Lambda can play a significant role in a machine learning workflow, although it's not explicitly designed as a machine learning service. Here's how Lambda might fit into various stages:

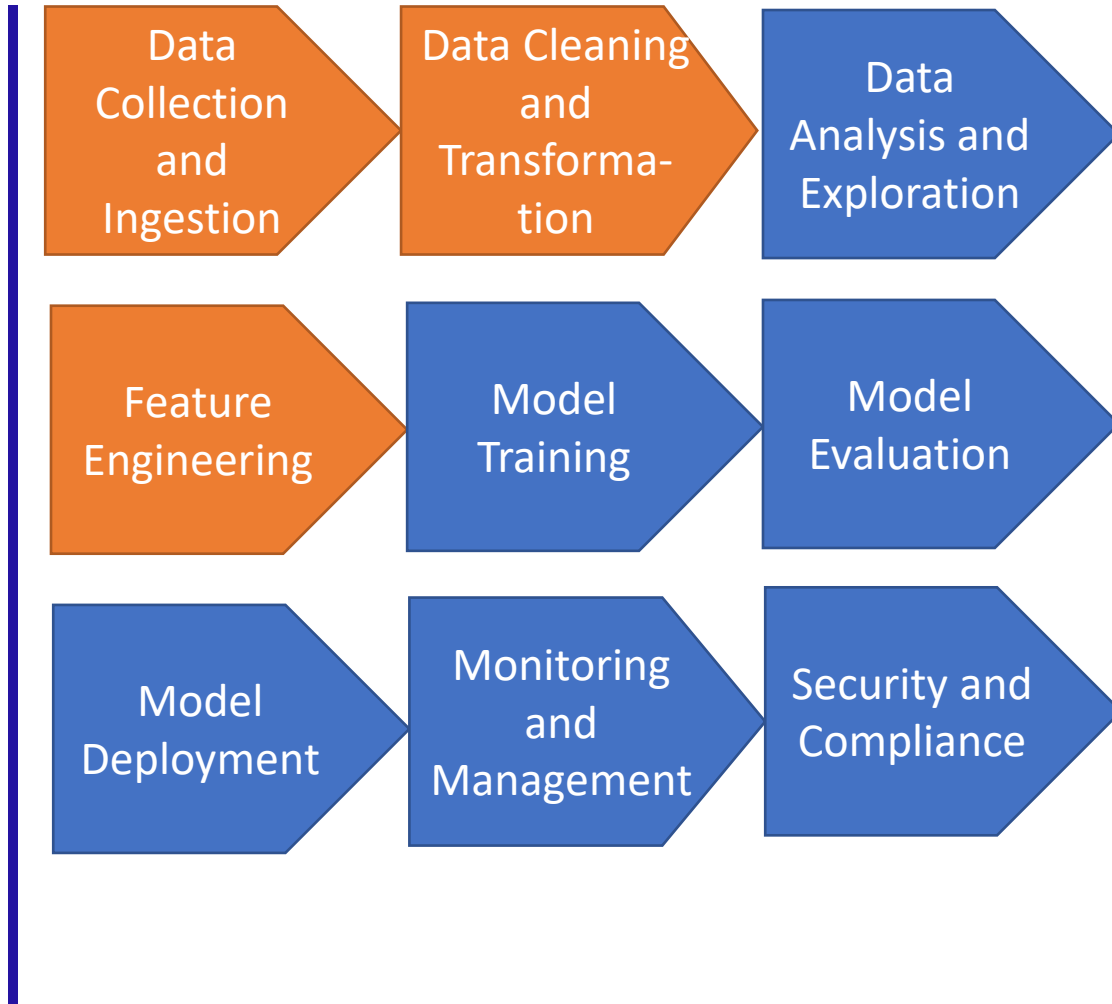
- Data Collection and Preprocessing
- Model Deployment and Inference
- Real-time Data Stream Analysis
- ETL (Extract, Transform, Load) Jobs
- Workflow Orchestration
- Event-Driven ML Pipelines
- Integrating with API Gateway



AGENDA

- 1 Overview of AWS Machine Learning Services
- 2 Best Practice for Data Preparation and Feature Engineering
- 3 Architectural Consideration for Scalable AI Solutions on AWS
- 4 Case Studies and Real-world Examples
- 5 Q&A Session

DATA PROCESSING AND FEATURE ENGINEERING



ML Pipeline Stages

AWS Machine Learning (ML) pipeline typically consists of several stages designed to move data from its raw form to a trained model that can make predictions or classifications. Here are the key components of a typical AWS ML pipeline:

- Data Collection and Ingestion
- Data Cleaning and Transformation
- Data Analysis and Exploration
- Feature Engineering
- Model Training
- Model Evaluation
- Model Deployment
- Monitoring and Management
- Security and Compliance

DATA PREPARATION

Data Preparation

- **Data Storage and Accessibility:** Use services like Amazon S3 for durable, secure, and scalable storage of raw data.
- **Data Cleaning:**
 - Use AWS Glue for ETL (Extract, Transform, Load) processes.
 - Use Amazon SageMaker Data Wrangler for quick data cleaning and transformation.
- **Data Sampling:** Create small, representative samples of your data for initial experimentation to save time and resources.
- **Schema Definition:** Clearly define the schema and ensure that it's consistently applied across the data set.
- **Data Versioning:** Use version control mechanisms to manage different versions of the dataset, especially if it's frequently updated.



FEATURE ENGINEERING



Feature Engineering (1)

- **Feature Selection:**
 - Identify the most important variables that affect the model's prediction.
 - Use built-in algorithms in SageMaker to automatically select features.
- **Feature Transformation:**
 - Normalize or scale numerical features.
 - Encode categorical features.
 - Use SageMaker's built-in algorithms for automatic transformation.
- **Feature Creation:**
 - Derive new features that might give additional insights.
 - Use SageMaker to create and test new features.

FEATURE ENGINEERING



Feature Engineering (2)

- **Data Imbalance:** If your classes are imbalanced, consider techniques like oversampling the minority class or using different evaluation metrics.
- **Data Splitting:** Use SageMaker to automatically split your data into training, validation, and test sets.
- **Pipeline Creation:**
 - Use SageMaker Pipelines to automate and streamline the steps.
 - Include data preparation and feature engineering steps in the pipeline for end-to-end machine learning.
- **Iteration:** Always re-evaluate the features as you improve the model or gather more data.
- **Monitoring:** Once the model is deployed, continuously monitor its performance and the quality of the incoming data.

AGENDA

- 1 Overview of AWS Machine Learning Services
- 2 Best Practice for Data Preparation and Feature Engineering
- 3 Architectural Consideration for Scalable AI Solutions on AWS
- 4 Case Studies and Real-world Examples
- 5 Q&A Session

ARCHITECTURAL CONSIDERATIONS



Architectural Considerations for Scalable AI Solutions on AWS

When designing scalable AI solutions on AWS, several architectural considerations come into play to ensure that the system can handle increased load, is resilient, and cost-effective.

Here are some key points to consider:

- Compute Resources
- Data Storage
- Data Pipeline
- Model Training
- Model Deployment
- Networking
- Security
- Monitoring and Logging
- Cost Management
- Compliance and Governance

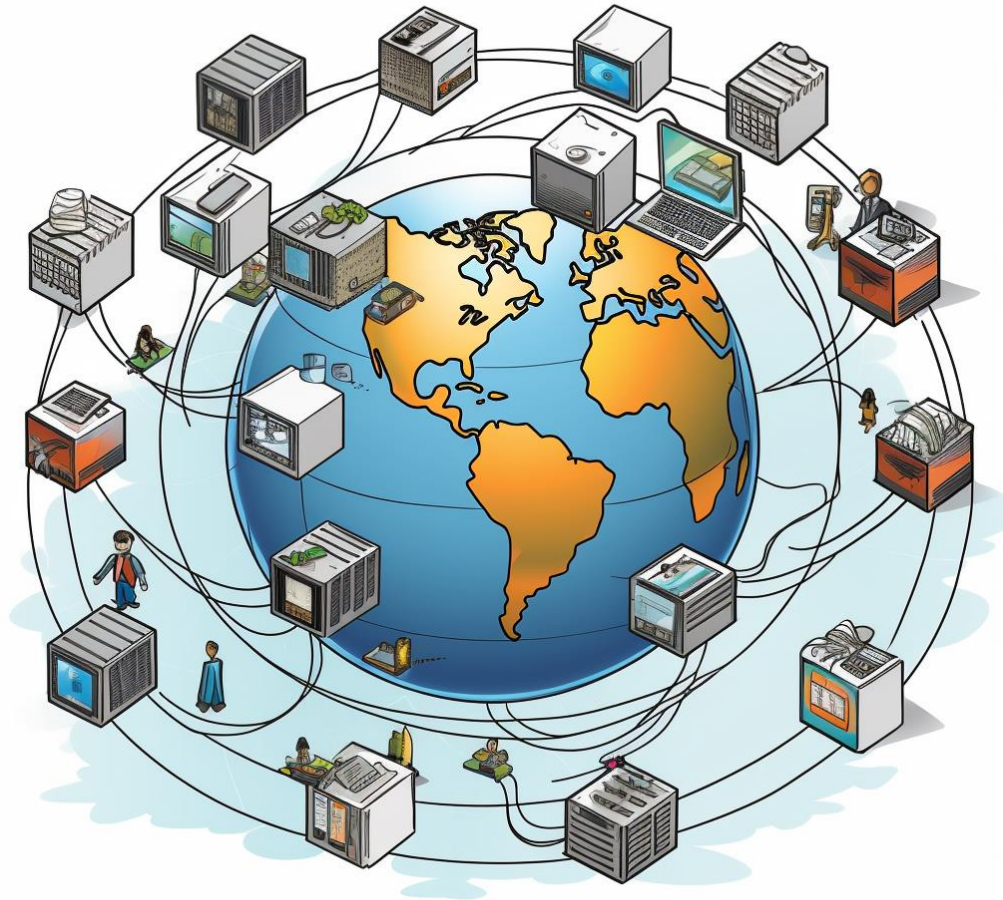
ARCHITECTURAL CONSIDERATIONS: COMPUTE



Compute Resources

- **Elasticity:** Using Amazon EC2 Auto Scaling Groups or AWS Fargate to automatically adjust the number of compute resources.
- **Resource Types:** Selecting the right type of EC2 instance or hardware accelerator (like GPU instances) for your specific AI workload.

ARCHITECTURAL CONSIDERATIONS: DATA STORAGE



Data Storage

- **Data Partitioning:** Implementing sharding or partitioning strategies in databases like Amazon DynamoDB.
- **Data Caching:** Using services like Amazon ElastiCache to cache frequent queries.
- **Optimized Data Formats:** Using optimized formats like Parquet for analytics.

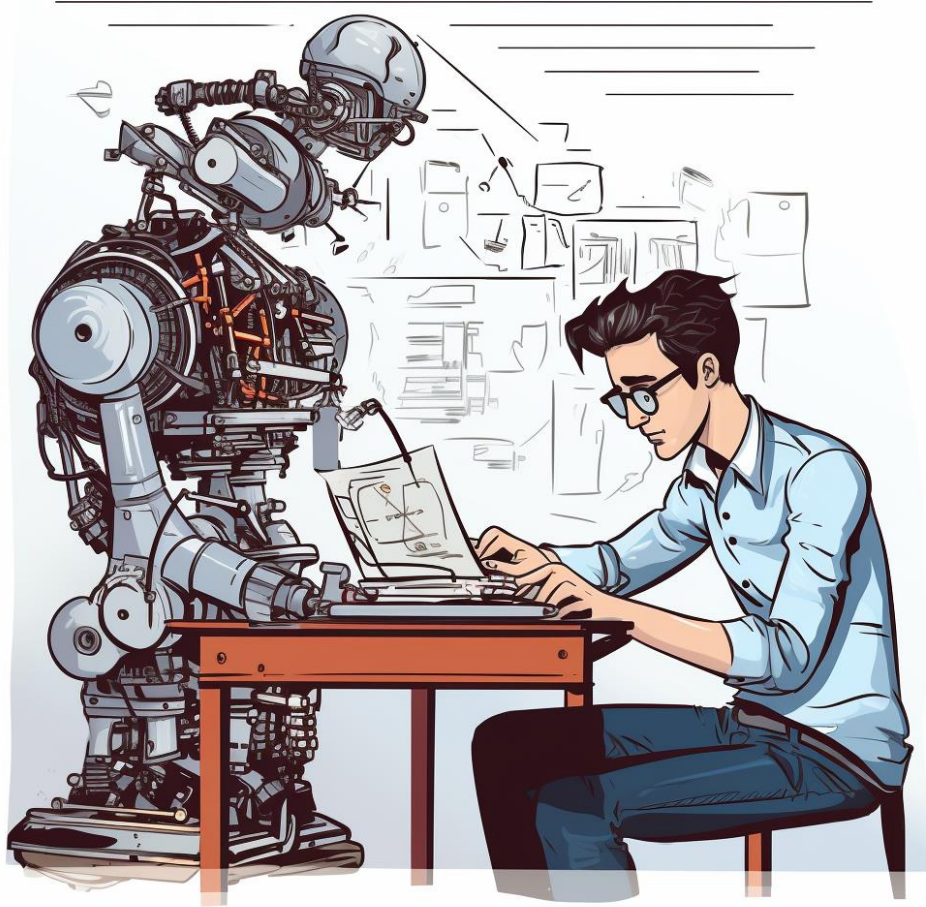
ARCHITECTURAL CONSIDERATIONS: DATA PIPELINE

Data Pipeline

- **Batch vs Real-time:** Considering whether data processing should be real-time (Amazon Kinesis) or batch-based (AWS Batch).
- **ETL Operations:** Utilizing services like AWS Glue for ETL transformations.



ARCHITECTURAL CONSIDERATIONS: MODEL TRAINING



Model Training

- **Distributed Training:** Using SageMaker's capabilities for distributed training to speed up model training.
- **Pipeline Parallelism:** Breaking down training pipelines into parallel tasks for faster processing.

ARCHITECTURAL CONSIDERATIONS: MODEL DEPLOYMENT



Model Deployment

- **Multi-AZ Deployment:** Deploying models across multiple availability zones for high availability.
- **Auto Scaling:** Utilizing auto-scaling features of Amazon SageMaker or EC2 instances.
- **Model Versioning:** Managing multiple versions of models and routing traffic using SageMaker endpoints.

ARCHITECTURAL CONSIDERATIONS: NETWORKING

Networking

- **Low Latency:** Leveraging AWS Global Accelerator or Amazon CloudFront for low-latency access.
- **VPC Design:** Structuring the Virtual Private Cloud (VPC) to separate different components efficiently.



ARCHITECTURAL CONSIDERATIONS: SECURITY



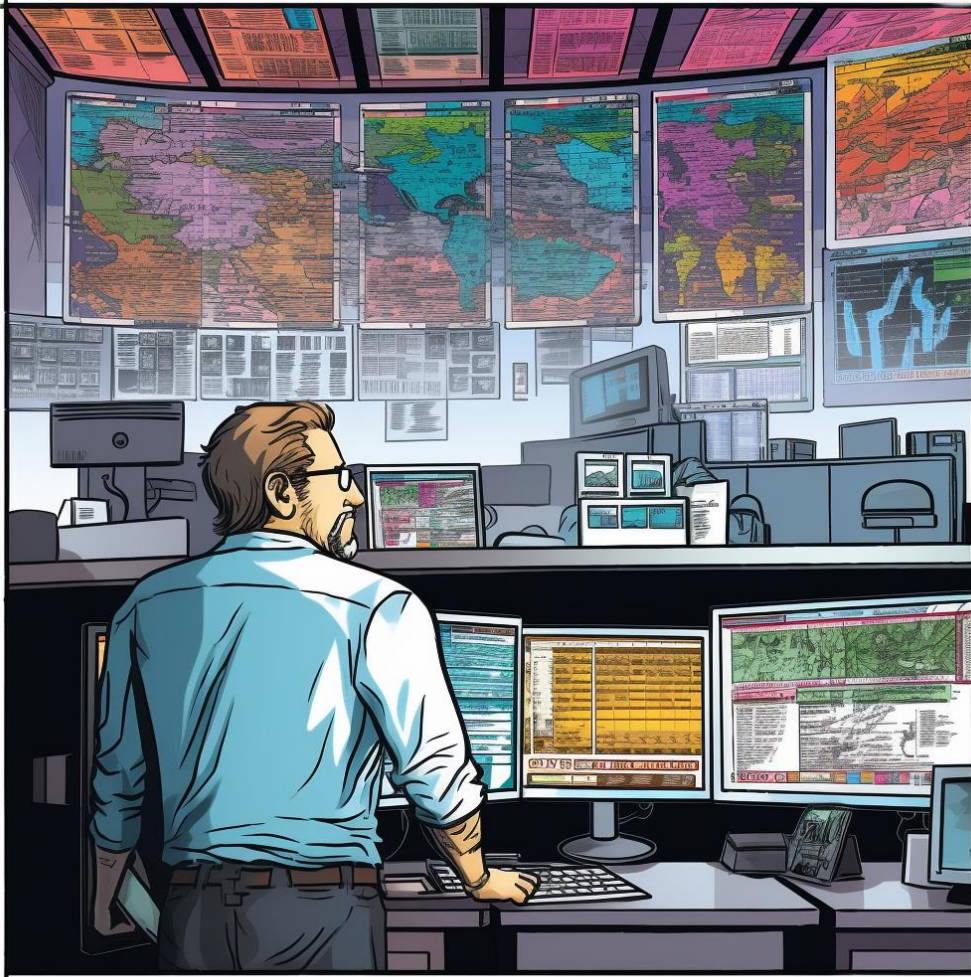
Security

- **IAM Policies:** Using IAM roles and policies for fine-grained access control.
- **Data Encryption:** Using encryption at rest and in transit.

ARCHITECTURAL CONSIDERATIONS: MONITORING/LOGGING

Monitoring and Logging

- **Monitoring:** Utilizing Amazon CloudWatch to monitor metrics and set up alarms.
- **Logging:** Implementing comprehensive logging using AWS CloudTrail and AWS X-Ray.



ARCHITECTURAL CONSIDERATIONS: COST MANAGEMENT



Cost Management

- **Cost Optimization:** Monitoring costs with AWS Cost Explorer and using cost allocation tags.
- **Reserved Instances:** Purchasing reserved instances for resources that will have consistent usage.

ARCHITECTURAL CONSIDERATIONS: COMPLIANCE AND GOVERNANCE



Compliance and Governance

- **Data Governance:** Using services like AWS Config and AWS Audit Manager for compliance checks.
- **Data Lineage:** Keeping track of data sources and transformations for compliance purposes.

AGENDA

- 1 Overview of AWS Machine Learning Services
- 2 Best Practice for Data Preparation and Feature Engineering
- 3 Architectural Consideration for Scalable AI Solutions on AWS
- 4 Case Studies and Real-world Examples
- 5 Q&A Session

REAL-WORLD EXAMPLE: NETFLIX



Netflix's Recommendation System

Netflix has over 200 million subscribers who generate a massive amount of data through their viewing habits, searches, and ratings. The company uses machine learning algorithms to analyze this data and provide personalized content recommendations to enhance user engagement and satisfaction. Given the scale, this operation must be highly reliable, quick, and able to handle huge datasets.

By leveraging AWS's scalable infrastructure and services, Netflix is able to handle millions of requests for recommendations per minute, and the system can automatically adapt to increasing demand, ensuring that users get a personalized, low-latency experience.

<https://www.youtube.com/watch?v=DJVJx2fSf90>

REAL-WORLD EXAMPLE: NETFLIX



Architecture (1)

- **Compute Resources:**
 - Netflix uses EC2 instances optimized for compute-heavy tasks and also leverages GPU instances for specific machine learning workloads.
- **Data Storage:**
 - They use Amazon S3 for storing historical user data and movie metadata.
 - Real-time data might be stored in Amazon DynamoDB for quick retrieval.
- **Data Pipeline:**
 - ETL tasks are orchestrated through AWS Glue and AWS Data Pipeline.
 - Real-time data streams are managed through Amazon Kinesis.

REAL-WORLD EXAMPLE: NETFLIX



Architecture (2)

- **Model Training:**
 - SageMaker may be used for training models using large datasets, leveraging its distributed training capabilities.
- **Model Deployment:**
 - SageMaker Endpoints for deploying the trained models, auto-scaled based on the number of incoming requests.
- **Networking:**
 - VPCs for network isolation and AWS Global Accelerator for low latency.
- **Security:**
 - IAM roles for restricting access to various services and resources, alongside encryption at rest and in transit.
- **Monitoring and Logging:**
 - CloudWatch for monitoring the system's health and performance.
 - AWS CloudTrail and AWS X-Ray for logging and debugging.

AGENDA

- 1 Overview of AWS Machine Learning Services
- 2 Best Practice for Data Preparation and Feature Engineering
- 3 Architectural Consideration for Scalable AI Solutions on AWS
- 4 Case Studies and Real-world Examples
- 5 Q&A Session

Q&A SESSION

Questions and Answers

Thank you for your attention.
Any questions?



Thank you

