AWS ML Engineer Associate

Curriculum Resource Guide

The AWS ML Engineer Associate Curriculum Resource Guide is a comprehensive resource designed to support your Machine Learning Engineer journey using AWS. Whether you're a beginner or an experienced practitioner, this guide offers a curated collection of materials tailored to your needs. This curated list of resources is designed to guide you on your learning journey. Familiarize yourself with these course materials and delve into them to further your understanding of how to operationalize machine learning solutions. Let this guide be your trusted resource as you embark on an enriching and transformative learning experience.

Course 1

Ingest and Store Data

This course focuses on the data preparation aspects of machine learning, including data collection, ingestion, and storage. You will learn about data fundamentals and AWS storage services, like Amazon S3. You

Resources:

- ✓ AWS Machine Learning Blog: Storage – Stay up-to-date on AWS solutions for ML workloads. This resource contains blog posts from ML experts on AWS storage solutions for various ML workloads.
- ✓ Data Collection for ML Workloads – This resource discusses how to follow AWS Well-Architected guidelines for ML data collection.
- ✓ AWS Tools for Reporting and Cost Optimization – To help you track, report, and analyze costs over time, AWS provides several reporting and cost-optimization

will also learn about services that assist with ingesting, extracting, and merging data, such as Amazon Kinesis. These services help you make effective data storage and format decisions based on your specific ML requirements.

tools. Use this resource to learn how to choose the most cost-effective storage solution for your ML use case.

✓ Data Ingestion and Preparation – This resource provides insight and best practices for data ingestion and preparation. Although this resource is written for public sector customers, these best practices apply to data ingestion in general.

Course 2
Transform Data

This course covers various transformation concepts and techniques, such as data cleaning, encoding, and feature engineering. You will discover how to use AWS services such as Amazon SageMaker Feature Store, Amazon SageMaker Data Wrangler, and AWS Glue to transform your data.

Resources:

- ✓ Time-Series Datasets – This resource explains time-series data that refers to a collection of observations or measurements recorded over regular intervals of time.
- ✓ Amazon Mechanical Turk Workforce – This resource discusses how Amazon Mechanical Turk workforce provides the most workers for your Amazon SageMaker Ground Truth labeling job and Amazon Augmented AI human review task.
- ✓ Amazon SageMaker Ground Truth – With SageMaker Ground Truth, you can build high-quality training datasets for your machine learning models. Learn more with this resource.
- ✓ Amazon SageMaker Ground Truth Plus – This resource discusses SageMaker Ground Truth Plus. It is a turnkey data labeling service that uses an expert workforce to deliver high-quality annotations quickly and reduces costs by up to 40 percent.
- ✓ Create, Store, and Share Features with Feature Store – SageMaker Feature Store simplifies how you create, store, share, and manage features. This is done by providing feature store options and reducing repetitive data processing and curation work.

- ✓ Transform Data – When you add a transform using SageMaker Data Wrangler, it adds a step to the data flow. Each transform you add modifies your dataset and produces a new dataframe. All subsequent transforms apply to the resulting dataframe. Learn more with this resource.
- ✓ How Lambda Processes Records from Amazon Kinesis Data Streams – This resource discusses how Lambda processes records from Amazon Kinesis Data Streams.
- ✓ Apache Spark on Amazon EMR – Amazon EMR is the best place to run Apache Spark. Learn more about features and benefits, use cases, and customer success stories with this documentation.

Course 3
Validate Data and Prepare for Modeling

This course covers the fundamental aspects of preparing and configuring data for model training. You will learn about data validation strategies, bias mitigation, data security, AWS services for data validation, and the importance of dataset splitting, shuffling, and augmentation.

Resources:

- ✓ AWS Key Management Service Features – With AWS Key Management Service (AWS KMS), you have centralized control over the cryptographic keys used to protect your data. The service is integrated with other AWS services to encrypt data you store in these services and to control access to the keys that decrypt it.
- ✓ At-Rest Encryption Conditions – ElastiCache supports symmetric customer-managed AWS KMS keys for encryption at rest. Learn more with this resource.
- ✓ Detecting and Redacting PII using Amazon Comprehend – Amazon Comprehend is a natural language processing (NLP) service that uses machine learning (ML) to find insights and relationships. This blog post details a specific customer use case and detecting PII in Amazon Comprehend.
- ✓ Security in AWS Glue DataBrew – This documentation helps you understand how to apply the shared responsibility model when using DataBrew. The topics will show you how to configure DataBrew to meet your security and compliance objectives.
- ✓ AWS Glue Data Quality – With AWS Glue Data Quality, you can measure and monitor the quality of your data so you can make good business decisions. AWS Glue Data

Quality provides a managed, serverless experience. This resource explains its benefits and key features, how it works, and the considerations.

- ✓ Amazon SageMaker – Using CSV Format – Many SageMaker algorithms support training with data in CSV format. Learn more with this resource.
- ✓ Amazon SageMaker – Using RecordIO Format – This resource details SageMaker RecordIO Formation. SageMaker converts each observation in the dataset into a binary representation as a set of 4-byte floats, then loads it in the protobuf values field.
- ✓ Amazon Elastic File System – This resource explains Amazon Elastic File System (Amazon EFS). The service is designed to be highly scalable, highly available, and highly durable.
- ✓ Amazon FSx for Windows File Server – This documentation helps you understand FSx for Windows File Server. It explains file systems, backups and file shares, security and data protection, availability and durability, and price and performance flexibility.
- ✓ Amazon FSx for Lustre User Guide – This resource explains Amazon FSx for Lustre. Use Lustre for workloads where speed matters.

Course 4
Choose a Modeling Approach

T This course covers the AWS ML stack, including using SageMaker for machine learning tasks and selecting appropriate models. It also covers using pre-trained Amazon SageMaker JumpStart and Amazon Bedrock solutions with a focus on interpretability in model or algorithm selection to solve common business challenges.

Resources:

- ✓ Choosing an AWS Machine Learning Service – This is a decision guide for help with picking the right ML services and frameworks to support your work.
- ✓ SageMaker Workshops – This website lists SageMaker Workshops created by the teams at AWS. Workshops are hands-on events designed to teach or introduce practical skills, techniques, or concepts that you can use to solve business problems.
- ✓ Choose an Algorithm – This is AWS documentation about choosing an algorithm using SageMaker.
- ✓ SageMaker Jumpstart – You can deploy, fine-tune, and evaluate pre-trained models from popular model hubs through the SageMaker JumpStart landing page in the updated SageMaker Studio experience. Learn more in this developer guide.
- ✓ Amazon Bedrock – The Amazon Bedrock webpage features information and links to videos and demos.
- ✓ Model Interpretability – This is AWS documentation about model interpretability and model performance.

Course 5
Train Models

This course covers how to develop machine learning models using SageMaker, including using built-in algorithms and pre-built Docker images. It also covers creating training jobs in SageMaker Studio and developing models using script mode and supported frameworks like Apache MXNet, TensorFlow, and PyTorch. You will learn about optimizing training processes and using external models from AWS Marketplace or that are trained outside of SageMaker.

Resources:

- ✓ Train Machine Learning Models – In this SageMaker developer guide article, you can learn about the process of training a model.
- ✓ Ensure Efficient Compute Resources on Amazon SageMaker – This AWS Machine Learning blog post is about efficient compute resources on SageMaker. It includes coverage of compute environment selection.
- ✓ Select Optimal Computing Instance Size – This AWS Well-Architected Framework Machine Learning Lens article includes guidance for selecting compute resources.
- ✓ Use Pre-built SageMaker Docker Images – This SageMaker developer guide article provides an overview of the pre-built Docker images available in SageMaker.
- ✓ Increase ML Model Performance and Reduce Training Time Using Amazon SageMaker Built-in Algorithms with Pre-trained Models – This is an AWS Machine Learning blog post about training ML models with SageMaker built-in algorithms.
- ✓ Bring Your Own Model with Amazon SageMaker Script Mode – This AWS Machine Learning blog post covers how to bring your own model to SageMaker using Script Mode.

- ✓ SageMaker Script Mode Examples – This SageMaker examples documentation highlights example Jupyter notebooks for a variety of ML use cases that you can run in SageMaker.
- ✓ KT's Journey to Reduce Training Time for a Vision Transformers Model using Amazon SageMaker – In this AWS Machine Learning blog post, you will learn about how the KT Corporation paired with AWS to reduce model training time.

Course 6
Refine Models

This course covers how to refine machine learning models, including techniques for bias mitigation, model performance improvement, hyperparameter tuning, and utilizing Amazon SageMaker services.

Resources:

- ✓ What Is Overfitting? – Learn more about overfitting, detecting and preventing overfitting, and how AWS can minimize overfitting errors in this guide.
- ✓ Efficiently Train, Tune, and Deploy Custom Ensembles Using Amazon SageMaker – In this blog post, learn more about the process of using a single training job to train multiple models, how to use automatic model tuning to optimize the ensemble hyperparameters, and how to deploy a single serverless endpoint that blends the inferences from multiple models.
- ✓ Perform Automatic Model Tuning with SageMaker – Learn more about hyperparameter tuning in this AWS developer guide.
- ✓ Deploy Large Models at High Performance Using FasterTransformer on Amazon SageMaker – Explore large model hosting challenges and how SageMaker large model inference (LMI) containers help you address these challenges using its low-code no-code capabilities.
- ✓ Domain Adaptation Fine-Tuning – Domain adaptation fine-tuning allows you to use pre-trained foundation models and adapt them to specific tasks using limited domain-

specific data. Learn more in this AWS documentation.

- ✓ Instruction-Based Fine-Tuning – Instruction-based fine-tuning uses labeled examples to improve the performance of a pre-trained foundation model on a specific task. Learn more in this AWS documentation.
- ✓ Automatically Retrain Neural Networks with Renate – This blog post details the problems associated with retraining neural networks and the main benefits of the Renate library in the process.

Course 7

Select a Deployment Infrastructure

This course introduces fundamental deployment concepts, essential production infrastructure components, and orchestration services for ML workflows, and deployment targets and environments. It also covers AWS compute instance types and the provisioning of compute resources in production and test environments.

Resources:

- ✓ Kubernetes Orchestration – This documentation highlights information for SageMaker Operators for Kubernetes and SageMaker Components for Kubeflow Pipelines.
- ✓ Create and Manage SageMaker Pipelines – This resource discusses how to create and manage SageMaker Pipelines.
- ✓ Model Building Pipeline Steps – SageMaker Pipelines are composed of steps. These steps define the actions that the pipeline takes and the relationships between steps using properties.
- ✓ Manage SageMaker with Step Functions – This resource provides insight into how to manage SageMaker with AWS Step Functions.
- ✓ Manage your ML Lifecycle with MLflow and Amazon SageMaker – In this blog post, explore how you can manage your machine learning lifecycle with SageMaker and MLflow.
- ✓ Best practices for Deploying Models on SageMaker Hosting Services – This documentation highlights best practices for deploying models on SageMaker Hosting services.

- ✓ Amazon SageMaker Pricing – With SageMaker, you pay only for what you use. Learn more about on-demand pricing, savings plans, total cost of ownership, and pricing examples.
- ✓ What Is Edge Computing? – Learn more about edge computing, AWS use cases, why edge computing is important, and how AWS can help you with your edge computing requirements.
- ✓ AWS IoT Greengrass – Learn more about how AWS IoT Greengrass works, its use cases, customer stories, and how to get started.
- ✓ Amazon SageMaker Neo – SageMaker Neo automatically optimizes machine learning models for inference on cloud instances and edge devices to run faster with no loss in accuracy. Learn how it works, its use cases, and its key benefits.

Course 8

Create and Script Infrastructure

This course covers building and scripting infrastructure for ML solutions with a focus on maintainability, scalability, and cost-efficiency. This includes exploring Infrastructure as Code (IaC) tools like AWS CDK, the Amazon SageMaker Python SDK, containers, and auto scaling methods.

Resources:

- ✓ Machine Learning Lens – Learn more about the AWS Well-Architected Framework through a Machine Learning Lens.
- ✓ Choosing an Infrastructure as Code Tool for Your Organization – This guide explores five IaC tools that can be used to provision AWS resources: AWS CloudFormation, AWS Serverless Application Model (AWS SAM), AWS Cloud Development Kit (AWS CDK), HashiCorp Terraform, and Pulumi.
- ✓ Refer to Resource Outputs in Another CloudFormation Stack – With Cross-stack references, you use a layered or service-oriented architecture. Explore this guide to learn more about creating a cross-stack reference.
- ✓ AWS CloudFormation – Use this guide to learn more about CloudFormation and the scenarios that demonstrate how CloudFormation can help.
- ✓ SageMaker Resource Type Reference – A user guide for SageMaker resource type references.
- ✓ AWS CDK Construct Library – The AWS Construct Library is organized into modules. Explore them in this guide.
- ✓ awslabs/aws-solutions-constructs – AWS Solutions Constructs is an extension of the

AWS CDK and provides additional patterns. Use this resource to view the available patterns.

- ✓ Steps Class – This is a resource guide for SageMaker Pipelines steps class workflows.
- ✓ SageMaker Automatic Model Tuning – You can use all the estimators with SageMaker Automatic Model Tuning with SageMaker Automatic Model Tuning, which performs hyperparameter tuning jobs. Learn more with this resource.
- ✓ Amazon SageMaker Neo – SageMaker Neo automatically optimizes machine learning models for inference on cloud instances and edge devices to run faster with no loss in accuracy. Learn how it works and about its use cases and key benefits.
- ✓ SageMaker Model Building Pipeline – This is a resource documentation to support Amazon SageMaker Model Building Pipeline, which includes background, code examples, and more.
- ✓ SageMaker Environment Variables – This table summarizes input and output paths for training datasets, checkpoints, model artifacts, and outputs that are managed by the SageMaker training platform.
- ✓ AWS Fargate Pricing – This is a list of available Fargate CPU capacity settings and

their corresponding memory capacity settings.

- ✓ How Step Scaling for Application Auto Scaling Works – This topic describes how step scaling works and introduces the key elements of a step scaling policy.

Course 9
Automate Deployment

This course introduces MLOps, which applies DevOps practices to machine learning workflows. It includes strategies for deploying, monitoring, and retraining models and creating automated pipelines for building, testing, and continuously deploying high-quality models.

Resources:

- ✓ Git – Git is a distributed version control system (VCS) that enables you to store code, track revision history, merge code changes, and revert to earlier code version when needed. Learn more about Git basics, benefits, and more. For more information, refer to What is Source Control?
- ✓ GitHub – GitHub is a widely used web-based hosting service for code repositories using Git.
- ✓ MLOps Template for Model Building, Training, and Deployment – Follow this resource to use the MLOps template for model building, training, and deployment.
- ✓ MLOps Template for Model Building, Training, and Deployment with Third-Party Git Repositories Using CodePipeline – Explore this resource to use the MLOps template for model building, training, and deployment with third-party Git repositories using CodePipeline.
- ✓ MLOps Template for Model Building, Training, and Deployment with Third-Party Git Repositories Using Jenkins – Explore this resource to use the MLOps template for model building, training, and deployment with third-party Git repositories using Jenkins.

- ✓ SageMaker Model Building Pipelines – Amazon SageMaker Model Building Pipelines is a tool for building machine learning pipelines that take advantage of direct SageMaker integration. Learn more in this developer guide.
- ✓ AWS CodePipeline – If you are new to CodePipeline, you can follow the tutorials in this guide after following the steps in this chapter to get set up.
- ✓ AWS CodePipeline Security – This documentation helps you understand how to apply the shared responsibility model when using CodePipeline.
- ✓ AWS Code Pipeline Troubleshooting – The following information might help you troubleshoot common issues in AWS CodePipeline.
- ✓ AWS CodeBuild – In this tutorial, you use AWS CodeBuild to build a collection of sample source code input files into a deployable version of the source code.
- ✓ AWS CodeBuild Security – This documentation helps you understand how to apply the shared responsibility model when using CodeBuild.
- ✓ AWS CodeBuild Troubleshooting – Use the information in this topic to help you

identify, diagnose, and address issues pertaining to CodeBuild.

- ✓ AWS Step Functions – Explore AWS Step Functions in this developer guide.
- ✓ Traffic Shifting Modes – Learn more about the various traffic shifting modes in blue/green deployment.
- ✓ Rolling Deployments – Rolling deployments gradually replace the previous deployment of your model version with the new version by updating your endpoint in configurable batch sizes. Learn more in this AWS documentation.
- ✓ Elastic Weight Consolidation – For an implementation of Elastic Weight Consolidation algorithm, check out this GitHub resource.
- ✓ Continual Learning through Synaptic Intelligence – For code that prevents catastrophic forgetting in continual learning, check out this GitHub resource.
- ✓ Real-Time Inferencing – Real-time inference is ideal for inference workloads where you have real-time, interactive, low latency requirements. Learn more in this developer guide.
- ✓ Serverless Inference – In this developer guide, explore Amazon SageMaker Serverless Inference. This is a purpose-built

inference option you can use to deploy and scale ML models without configuring or managing any of the underlying infrastructure.

- ✓ Asynchronous Inference – Learn more about how asynchronous inference works and how to get started.
- ✓ Batch Transform – Learn more about when and how to use batch transform in this developer guide.

Course 10
Monitor Model Performance and Data Quality

This course introduces monitoring techniques to detect the four types of drift and how to correct and mitigate issues with your model's accuracy and performance. This course shares best practices for continuous monitoring, creating notifications, and automated remediation.

Resources:

- ✓ ML Lifecycle Phase – Monitoring – This AWS Well-Architected Framework Machine Learning Lens documentation describes the monitoring phase and lists key components of the ML lifecycle.
- ✓ Detecting Data Drift Using Amazon SageMaker – This AWS Architecture Blog post describes detecting data drift using Amazon SageMaker. It includes guidance for how to detect drift, build a feedback loop, and gain insights into data and models.
- ✓ Monitor Data and Model Quality with Amazon SageMaker Model Monitor – This Amazon SageMaker developer guide article provides an overview of the Amazon SageMaker Model Monitor capabilities.
- ✓ Monitor Data Quality – For guidance on how to use Amazon SageMaker Model Monitor to monitor data quality, review this documentation.
- ✓ Monitor Model Quality – For guidance on how to use Amazon SageMaker Model Monitor to monitor model quality, review this documentation.
- ✓ Monitor Bias Drift for Models in Production – For guidance on how to use Amazon SageMaker Model Monitor to monitor for bias drift, review this documentation.

- ✓ Monitor Feature Attribution Drift for Models in Production – For guidance on how to use Amazon SageMaker Model Monitor to monitor for feature attribution drift, review this documentation.
- ✓ Amazon SageMaker Model Monitor – This Amazon SageMaker Examples documentation highlights example Jupyter notebooks for a variety of machine learning monitoring use cases for Amazon SageMaker Model Monitor.
- ✓ Detecting Data Drift Using Amazon SageMaker – This AWS Architecture Blog post covers detecting NLP data drift using Amazon SageMaker Model Monitor. The post includes guidance for how to use SageMaker Model Monitor to measure data drift.
- ✓ How LotteON Built Dynamic A/B testing for Their Personalized Recommendation System – This AWS Machine Learning Blog post documents how LotteON implemented dynamic A/B testing for their personalized recommendation system. They used AWS services like SageMaker, Lambda, and Amazon API Gateway.
- ✓ A/B Testing with Amazon SageMaker – This site highlights example Jupyter notebooks to use for A/B testing using Amazon SageMaker.

- ✓ Improve Governance of Your Machine Learning Models with Amazon SageMaker – This AWS Machine Learning Blog post covers guidance on using Amazon SageMaker Model Cards and the SageMaker Model Dashboard to improve visibility and governance over ML models.
- ✓ Automate Model Retraining with Amazon SageMaker Pipelines When Drift Is Detected – This AWS Machine Learning Blog post covers automated model retraining. It includes guidance for how to set up an automated model retraining solution using Amazon SageMaker Pipelines.

Course 11
Monitor and Optimize Infrastructure and Costs

This course covers the key performance indicators, monitoring tools, and cost analysis tools available on AWS for optimizing and monitoring

Resources:

- ✓ Build a Centralized Monitoring and Reporting Solution for Amazon SageMaker Using Amazon CloudWatch – In this blog, a cross-account observability dashboard is presented that provides a centralized view for monitoring SageMaker user activities and resources across multiple accounts.
- ✓ Techniques and Approaches for Monitoring Large Language Models on AWS – This blog demonstrates metrics for online Large Language Model (LLM) monitoring and their respective architecture for scale using AWS services such as CloudWatch and Lambda.

infrastructure and ML solutions.

- ✓ Enable Pod-Based GPU Metrics in Amazon CloudWatch – This post details how to set up container-based GPU metrics and provides an example of collecting these metrics from Amazon Elastic Kubernetes Service (Amazon EKS) pods.
- ✓ Monitoring Amazon ML with Amazon CloudWatch Metrics – Amazon ML automatically sends metrics to CloudWatch so that you can gather and analyze usage statistics for your ML models. Learn more in this documentation.
- ✓ What Is Amazon CloudWatch Logs? – You can use Amazon CloudWatch Logs to monitor, store, and access your log files from Amazon Elastic Compute Cloud (Amazon EC2) instances, AWS CloudTrail, Amazon Route 53, and other sources. Learn more in this user guide.
- ✓ Amazon CloudWatch Logs Now Offers Automated Pattern Analytics and Anomaly Detection – In this blog post, you will learn about the automated pattern analytics and anomaly detection that CloudWatch Logs offers.
- ✓ Minimize Real-Time Inference Latency by Using Amazon SageMaker Routing Strategies – This blog post explores SageMaker least outstanding requests (LOR) routing strategy and its benefits over the

default routing mechanism. It also covers how you can enable LOR for your model deployments.

- ✓ What Is AWS X-Ray? – This documentation reviews how AWS X-Ray works.
- ✓ Monitor Function Performance with Amazon CloudWatch Lambda Insights – Amazon CloudWatch Lambda Insights collects and aggregates Lambda function runtime performance metrics and logs for your serverless applications. This page describes how to enable and use Lambda Insights to diagnose issues with your Lambda functions.
- ✓ Using CloudWatch Anomaly Detection – When you enable *anomaly detection* for a metric, CloudWatch applies statistical and machine learning algorithms. This user guide explores this more.
- ✓ Analyzing Log Data with CloudWatch Logs Insights – With CloudWatch Logs Insights, you can interactively search and analyze your log data in CloudWatch Logs. Learn more in this user guide.
- ✓ What Is AWS CloudTrail? – CloudTrail is an AWS service that you can use to enable operational and risk auditing, governance, and compliance of your AWS account.

- ✓ Working with CloudTrail Trails – This guidance explores how to work with CloudTrail trails.
- ✓ How Amazon QuickSight Works – This guidance explores how you can access data and prepare it for use in reporting by using QuickSight.
- ✓ Automating Amazon SageMaker with Amazon EventBridge – This guidance explores how EventBridge monitors status change events in SageMaker.
- ✓ What Is Amazon EventBridge? – EventBridge is a serverless service that uses events to connect application components together. This helps you to build scalable event-driven applications. Explore more in this user guide.
- ✓ Amazon SageMaker Inference Recommender – SageMaker Inference Recommender reduces the time required to get ML models in production by automating load testing and model tuning across SageMaker ML instances.
- ✓ AWS Compute Optimizer – Rightsizing Recommendation Preferences – This documentation outlines the details of each feature and how to set your rightsizing recommendation preferences in AWS Compute Optimizer.

- ✓ Set Up Enterprise-Level Cost Allocation for ML Environments and Workloads Using Resource Tagging in Amazon SageMaker – Best practices regarding cost allocation for your SageMaker environment and workloads.
- ✓ AWS Service Quotas – Your AWS account has default quotas. This documentation provides detailed information regarding the quotas.
- ✓ Announcing Provisioned Concurrency for Amazon SageMaker Serverless Inference – This blog post discusses provisioned concurrency and Application Auto Scaling, how to use them, and some best practices and guidance for your inference workloads.
- ✓ Provisioned Throughput for Amazon Bedrock – If you customized a model, you must purchase Provisioned Throughput to be able to use it. This guidance discusses pricing.
- ✓ Automatically Scale Provisioned Concurrency for a Serverless Endpoint – This guidance details auto scaling the provisioned concurrency for cost optimization.
- ✓ AWS Tools for Reporting and Cost Optimization – This guidance details the different AWS tools for reporting and cost optimization.

- ✓ Managing Your Costs with AWS Budgets – You can use AWS Budgets to track and act on your AWS costs and usage. Learn more in this user guide.
- ✓ Announcing Amazon EC2 Capacity Blocks for ML to Reserve GPU Capacity for Your Machine Learning Workloads – A new Amazon EC2 usage model that further democratizes ML, so you can access GPU instances to train and deploy ML and generative AI models.
- ✓ Machine Learning Savings Plans – Learn more about how Savings Plans offer a flexible usage-based pricing model for SageMaker.

Course 12
Secure AWS ML Resources

This course equips you with the skills to secure your AWS resources for machine learning solutions. You can do this by implementing security controls, configuring IAM policies and roles, and exploring the

Resources:

- ✓ Shared Responsibility Model – Security and compliance is a shared responsibility between AWS and the customer. This documentation explores this further.
- ✓ Policies and Permissions in IAM – In the AWS Identity and Access Management User Guide, explore details pertaining to policies and permissions in IAM.
- ✓ Bucket Policies for Amazon S3 – Privately store and manage assets in the cloud using this version control service. Learn more with the Amazon Simple Storage Service User Guide.

Amazon SageMaker security and compliance features to meet your company's security requirements.

- ✓ What Are VPC Endpoints? – The AWS Whitepaper details a VPC endpoint as it helps customers to privately connect to supported AWS services and VPC endpoint services powered by AWS PrivateLink.
- ✓ Control Access to VPC Endpoints Using VPC Endpoint Policies – An endpoint policy is a resource-based policy that you attach to a virtual private cloud (VPC) endpoint to control which AWS principals can use the endpoint to access an AWS service. Learn more in this AWS documentation.
- ✓ What Is Amazon VPC? – Explore Amazon Virtual Private Cloud (Amazon VPC).
- ✓ AWS PrivateLink Concepts – You can use Amazon VPC to define a VPC. The guidance details key aspects of Amazon VPC.
- ✓ NAT Gateways – A NAT gateway as a Network Address Translation (NAT) service. This user guide details the basics, use cases, and more.
- ✓ Control Traffic to Your AWS Resources Using Security Groups – A security group controls the traffic that is allowed to reach and leave the resources it is associated with. This documentation showcases architectural diagrams and use cases.
- ✓ Security Group Rules – The rules of a security group control the inbound traffic that's allowed to reach the resources that

are associated with the security group. Learn about security group rules.

- ✓ Control Subnet Traffic with Network Access Control Lists – This user guide details how a network access control list (ACL) allows or denies specific inbound or outbound traffic at the subnet level.
- ✓ Firewall Policies in AWS Network Firewall – An AWS Network Firewall policy defines the monitoring and protection behavior for a firewall. This developer guide offers topics for this subject.
- ✓ AWS Managed Policies for SageMaker – AWS services maintain and update AWS managed policies. Learn more and explore different topics in the developer guide.
- ✓ Data Protection in Amazon SageMaker – This documentation details the AWS shared responsibility model for data protection in SageMaker.
- ✓ Log Amazon SageMaker Events with Amazon CloudWatch – This table lists Log Group Name and Log Stream Name.
- ✓ Monitor Amazon SageMaker with Amazon CloudWatch – You can monitor SageMaker using CloudWatch, which collects raw data and processes it into readable, near real-time metrics.

- ✓ ML Governance with Amazon SageMaker – This guidance describes why to choose ML Governance with SageMaker. It also covers the benefits and integrations.
- ✓ AWS Service Catalog Documentation – With this resource, you can learn more about AWS Service Catalog.
- ✓ What Is AWS Artifact? – This user guide information explores AWS Artifact.
- ✓ What Is AWS Config? – This documentation explains AWS Config, its considerations, the ways to use it, and its features.
- ✓ What Is AWS Security Hub? – Use this documentation to explore the AWS Security Hub benefits, security, pricing, and more.
- ✓ Security in AWS CodePipeline – This documentation helps you understand how to apply the shared responsibility model when using CodePipeline.
- ✓ Monitor AWS Resources Provisioned While Using Amazon SageMaker – This developer guide discusses how monitoring is an important part of maintaining the reliability, availability, and performance of SageMaker and your other AWS solutions.
- ✓ Security Pillar - AWS Well Architected Framework – The focus of this paper is the security pillar of the AWS Well-Architected Framework. It provides guidance to help you

apply best practices and the current recommendations in the design, delivery, and maintenance of secure AWS workloads. For more information, refer to AWS Well-Architected Framework.