

AWS Certified Machine Learning Engineer - Associate

Domínios, pesos oficiales y habilidades clave del temario de MLA-C02, para repasar antes del examen.

Nivel: Intermediate (Associate) Duración: 130 min Nota de corte: 720/1000 Fuente: guía de estudio oficial de Microsoft Learn

1. Data Preparation for ML and AI

28-28%

Collect and store data

- SageMaker AI entrena con datos de Amazon S3, Amazon EFS o Amazon FSx for Lustre: el modo File descarga todo el dataset, FastFile lo transmite bajo demanda con acceso de archivo y EFS o FSx se montan desde una VPC. Para sacar datos de bases operativas sin afectarlas, la exportación de DynamoDB a S3 no consume capacidad de lectura y la exportación de instantáneas de RDS escribe Parquet en S3 en segundo plano.
- La clase de almacenamiento de S3 se elige por patrón de acceso y resiliencia: Intelligent Tiering para accesos desconocidos, las clases IA para datos poco usados (One Zone-IA solo si se pueden recrear), Express One Zone para la mínima latencia y las clases Glacier para archivo. Para requisitos normativos de inmutabilidad, S3 Object Lock en modo compliance impide que nadie, ni el usuario raíz, borre o sobrescriba versiones durante la retención. DynamoDB, incluidos sus índices secundarios globales, una clave de partición de baja cardinalidad crea particiones calientes que no se arreglan añadiendo capacidad. En Amazon S3, la tasa de peticiones escala por prefijo y de forma gradual, así que conviene repartir la carga entre prefijos y reiniciar los errores 503 Slow Down.
- Los problemas de capacidad en la ingesta suelen venir de claves mal distribuidas: en Kinesis Data Streams y en Amazon Data Firehose se entrega datos de streaming en Amazon S3 y otros destinos sin escribir aplicaciones, agrupándolos por tamaño o tiempo, y puede leer directamente de Kinesis Data Streams o de un topic de Amazon MSK. Para cálculos con estado y ventanas de tiempo se usa Amazon Managed Service for Apache Flink y cuando varias aplicaciones leen el mismo stream, enhanced fan-out da a cada una rendimiento de lectura dedicado y permiten saltar bloques, por lo que reducen los datos que leen Athena y otros motores; se puede convertir con CTAS, AWS Glue o la conversión de formato de Amazon Data Firehose, que solo acepta JSON de entrada. Los algoritmos integrados de SageMaker AI exigen formatos concretos, como CSV sin cabecera con el objetivo en la

Perform data transformation, feature engineering, and pre-processing

- AWS Glue es un servicio de integración de datos sin servidor que ejecuta trabajos ETL sobre Apache Spark y usa job bookmarks para procesar solo los datos nuevos. AWS Glue DataBrew limpia y normaliza datos sin código mediante recetas reutilizables, y EMR Serverless ejecuta Spark sin gestionar clústeres. SageMaker Data Wrangler, ahora integrado en SageMaker Canvas, prepara datos con poco código y exporta el flujo a S3. Pipelines o Feature Store
- SageMaker Feature Store organiza las características en feature groups con un identificador de registro y un event time. El online store guarda solo el registro más reciente para consultas de baja latencia y admite TTL; el offline store en Amazon S3 conserva todo el historial para entrenamiento e inferencia por lotes. Los conjuntos de entrenamiento sin fuga de datos se crean con uniones point-in-time sobre el offline store.
- Amazon Data Firehose puede invocar una función Lambda que transforma cada lote y marca los registros como Ok, Dropped o ProcessingFailed, y puede convertir JSON a Parquet u ORC con un esquema del Glue Data Catalog. Lambda mantiene estado entre invocaciones con ventanas tumbling sobre Kinesis. Los trabajos de streaming de AWS Glue usan Spark Structured Streaming, procesar por ventanas y usan checkpoints en lugar de job bookmarks: para que las variables de gran magnitud no dominen el modelo. El binning convierte un valor numérico en rangos categóricos cuando su relación con el objetivo no es lineal, y la transformación logarítmica reduce la asimetría.
- Los modelos de embeddings de Amazon Bedrock convierten texto, imágenes o ambos en vectores; los multimodales los sitúan en un espacio semántico compartido para buscar imágenes por texto. Se puede elegir una dimensión de salida menor para reducir almacenamiento y latencia, pero el índice vectorial debe tener la misma dimensión. La similitud la calcula la base de datos vectorial, y para indexar grandes volúmenes conviene usar trabajos por lotes.

Validate data quality and manage bias

- AWS Glue Data Quality define reglas en DQDL y las evalúa sobre tablas del Data Catalog (con recomendaciones de reglas) o dentro de jobs de ETL, donde puede aislar los registros que fallan antes de cargarlos. Sus analyzers y la detección de anomalías con ML aprenden tendencias y estacionalidad. En DataBrew, los rulesets se validan en prome jobs y emiten eventos que EventBridge puede iniciar.
- Amazon SageMaker Ground Truth etiqueta con plantillas privadas o de proveedores, consolida las anotaciones de varios trabajadores y ofrece etiquetado automatizado con aprendizaje activo para tareas integradas. Los trabajos de verificación califican etiquetas y los de ajuste las corrigen. Para preetiquetar grandes volúmenes con un modelo fundacional, la inferencia por lotes de Amazon Bedrock procesa prompts desde S3.
- Las métricas previas al entrenamiento de SageMaker Clarifai, como Class Imbalance y DPL, detectan infrarrepresentación y diferencias de etiquetas entre grupos sin necesidad de modelo. Para mitigar el sesgo: dividir por clave o de forma estratificada en Data Wrangler, barajar los datos en cada época con ShuffleConfig y aumentar imágenes con transformaciones de color, rotación y recorte.
- SageMaker Clarifai calcula métricas de sesgo sobre datos tabulares: las facetas numéricas se dividen con un umbral y las categóricas por valor. Para texto o imágenes se crea una tabla de atributos por elemento (por ejemplo, el idioma detectado con Amazon Comprehend). JS compara distribuciones multiclase de forma simétrica y CDD condiciona la dispersión a subgrupos para descartar la paradoja de Simpson.
- Data Wrangler equilibra clases con random oversampling o SMOTE, que interpola muestras sintéticas solo con variables numéricas y copia las no numéricas. Sin tocar los datos, XGBoost usa scale_pos_weight y Image Classification multietiqueta usa use_weighted_loss. Para evaluar la clase minoritaria conviene la división estratificada y métricas como balanced accuracy en lugar de la accuracy global.

2. ML Model and Foundation Model (FM) Development

24–24%

Choose appropriate modeling approaches for ML and AI solutions

- Amazon Bedrock Evaluations compara modelos con tres enfoques: trabajos programáticos con conjuntos integrados o propios, un modelo juez que puntúa con métricas como corrección, fidelidad o coherencia lógica, y equipos de evaluadores humanos para criterios subjetivos. Con el modelo juez se pueden aportar las respuestas de un modelo externo a Amazon Bedrock. Antes, filtra candidatos por modalidad, ventana de contexto, latencia y costo.
- Amazon Bedrock ofrece supervised fine-tuning con datos etiquetados, reinforcement fine-tuning con funciones de recompensa (por ejemplo en AWS Lambda) cuando el éxito se puede verificar y etiquetar es caro, y destilación para pasar el conocimiento de un modelo grande a otro pequeño, incluso desde registros de invocación. En SageMaker
- Al técnicas PEFT como LoRA abaratan y aceleran el ajuste al actualizar menos parámetros.
- Elige el modelo según el problema y las restricciones: SageMaker AI ofrece algoritmos integrados por tipo de tarea (XGBoost para tablas, DeepAR para series temporales, Random Cut Forest para anomalías) y SageMaker JumpStart aporta modelos preentrenados y plantillas de solución. Si la explicabilidad es crítica, conviene un modelo interpretable por naturaleza; si prima la precisión, un modelo complejo explicado con SageMaker Clarify.
- Cuanto más alto el nivel, menos esfuerzo y menos control los servicios de IA como Amazon Translate o Amazon Textract no requieren experiencia en ML; Amazon Bedrock da acceso serverless a foundation models con prompts y RAG; SageMaker AI permite entrenar y personalizar con control total, y la infraestructura propia da la máxima flexibilidad. Un camino habitual es empezar en Bedrock y pasar a SageMaker AI al crecer la personalización.
- Amazon Bedrock Knowledge Bases cubre varios patrones de RAG: Knowledge bases gestionadas con conectores, filtrado por ACL y acceso MCP vía AgentCore Gateway; knowledge bases gestionadas por el cliente con su propio almacén vectorial; conexión a datos estructurados en Amazon Redshift, que traduce preguntas a SQL; y GraphRAG con Amazon Neptune Analytics para relacionar entidades entre documentos.

Train, fine-tune, and customize models for ML and AI solutions

- SageMaker AI ofrece algoritmos integrados por tipo de problema: XGBoost, LightGBM o Linear Learner para datos tabulares, DeepAR para prever muchas series temporales relacionadas, Factorization Machines para datos dispersos como la predicción de clics, IP Insights para relacionar entidades con direcciones IPv4 y Random Cut Forest para anomalías. Cada algoritmo exige formatos de entrada concretos, como recordIO protobuf.
- El modo Script ejecuta tu propio código de entrenamiento en contenedores gestionados de SageMaker AI (PyTorch, TensorFlow, scikit-learn, XGBoost) sin crear imágenes de Docker. Las dependencias se instalan desde requirements.txt, los datos de cada canal llegan en SM_CHANNEL_ <NOMBRE> y el modelo guardado en SM_MODEL_DIR se empaqueta como model.tar.gz. Ampliar o crear una imagen solo compensa si el contenedor gestionado no basta.
- El ajuste automático de modelos (AMT) de SageMaker AI lanza trabajos de entrenamiento con estrategias Bayesian, Random, Grid (solo valores categóricos) o Hyperband, que detiene configuraciones débiles y reasigna épocas. Conviene usar escala logarítmica en rangos de varios órdenes de magnitud, limitar el número de hiperparámetros y reutilizar trabajos anteriores con warm start (TRANSFER LEARNING si cambian los datos).
- Para entrenar más rápido en SageMaker AI: paralelismo de datos cuando el modelo cabe en una GPU y paralelismo de modelo cuando no cabe; parada anticipada en el ajuste (TrainingJobEarlyStoppingType en AUTO, con la métrica emitida en cada época) o Hyperband; y grupos de calentamiento gestionados para reutilizar instancias entre trabajos consecutivos con la misma configuración.
- Sobreajuste: buen resultado en entrenamiento y malo en validación; se corrige con más regularización, menos complejidad (por ejemplo, menor max_depth en XGBoost), menos épocas o parada anticipada. Infraajuste: malo en ambos; se añaden características y se reduce la regularización. El olvido catastrófico en el ajuste fino de modelos fundacionales se mitiga mezclando datos del dominio con datos generales.

Analyze and evaluate the performance of ML and AI systems

- MLflow administrado en SageMaker AI registra parámetros, métricas y artefactos de cada ejecución, permite compararlas y pasar el mejor modelo a SageMaker Model Registry. En Bedrock, Prompt Management guarda versiones inmutables del prompt y variantes para compararlas, y las evaluaciones de Bedrock reutilizan el mismo conjunto de datos en S3 para repetir cada prueba.
- La deriva de datos se detecta comparando el tráfico capturado con una referencia de estadísticas y restricciones calculada sobre los datos de entrenamiento; la deriva de calidad del modelo exige combinar las predicciones con etiquetas reales que llegan después. Las alarmas de detección de anomalías de CloudWatch aprenden la estacionalidad de una métrica y evitan los umbrales estáticos.
- Una variante shadow recibe una copia del tráfico de la variante de producción del mismo endpoint y sus respuestas no llegan al llamante, así que se prueba con tráfico real sin afectar a los usuarios. El panel de la prueba compara métricas operativas; para comparar predicciones hay que activar la captura de datos. Al promoverla con sus propias instancias, conviene probar con el 100 % del tráfico.
- Los valores de SHapley (SHAP) asignan a cada característica su contribución a una predicción concreta y, agregados, dan la importancia global; los gráficos de dependencia parcial muestran el efecto medio de una característica. Las explicaciones SHAP dependen de la referencia elegida. En RAG, RetrieveAndGenerate de Amazon Bedrock Knowledge Bases devuelve citas que enlazan cada parte de la respuesta con su fuente.
- Con un algoritmo propio, las definiciones de métricas (nombre y expresión regular) hacen que SageMaker AI envíe la pérdida de entrenamiento y validación a CloudWatch en tiempo real. TensorBoard en SageMaker AI muestra distribuciones de gradientes y pesos para diagnosticar gradientes que se desvanecen o explotan. Si la validación empeora mientras el entrenamiento mejora, hay sobreajuste: más regularización o parar antes.

3. Deployment and Orchestration of ML and AI Workflows

24-24%

Manage deployment infrastructure for ML and AI model types

- El destino de despliegue depende de dónde y cómo se infiere. Inference Recommender automatiza pruebas de carga para elegir tipo y número de instancias con el mejor coste y rendimiento; Serverless Inference evita gestionar instancias pero no admite GPU ni configuración de VPC; y SageMaker Neo compila modelos para dispositivos edge
- Los endpoints en SageMaker AI permiten un contenedor y carga desde S3 bajo demanda muchos modelos del mismo framework, a cambio de arranques en frío; los multicontenedor alojan contenedores de frameworks distintos, invocados de forma directa o en secuencia. Para orquestar, SageMaker Pipelines es el servicio propio de ML, Step Functions integra los trabajos de SageMaker AI y Amazon MWAA ejecuta DAG de Airflow sin gestionar infraestructura
- SageMaker AI ofrece cuatro estrategias de inferencia: tiempo real para baja latencia con tráfico sostenido, Serverless Inference para tráfico intermitente (con Provisioned Concurrency si las ráfagas son previsibles), Asynchronous Inference para cargas grandes y procesos largos con cola y escalado a cero, y batch transform para conjuntos de datos completos en S3 sin endpoints persistentes
- Amazon Bedrock ofrece varias formas de consumir un modelo fundacional: bajo demanda, con perfiles de inferencia entre regiones (geográficos para residencia de datos o globales), Provisioned Throughput para rendimiento fijo facturado por horas, batch inference para lotes de peticiones en S3 y Bedrock Marketplace, que despliega modelos especializados en endpoints de SageMaker AI accesibles con las APIs de Bedrock
- En modelos entrenados fuera de AWS se carga en SageMaker AI empaquetado como tar.gz en S3 con un contenedor precompilado (más requirements.txt) o con un contenedor propio que cumpla el contrato: servidor en el puerto 8080 con /invocations y /ping. Amazon Bedrock Custom Model Import importa desde S3 modelos fundacionales personalizados en formato Hugging Face (safetensors, config y tokenizador) para invocarlos bajo demanda.

Provision and configure resources for ML and AI workloads based on existing architecture and requirements

- Para tráfico intermitente, Serverless Inference de SageMaker AI cobra solo por uso y escala a cero; con picos predecibles, Provisioned Concurrency evita arranques en frío y se puede programar con Application Auto Scaling. Asynchronous Inference encola cargas grandes y escala a cero. En Amazon Bedrock, Provisioned Throughput da rendimiento fijo por hora, más barato cuanto más largo es el compromiso
- En CloudFormation, Export y Fn::ImportValue comparan salidas entre pilas de la misma cuenta y Región con una referencia fuerte que impide borrar o cambiar lo importado; Fn::GetStackOutput es una referencia débil. Step Functions orquesta trabajos de SageMaker AI con el patrón .sync, y en SageMaker Pipelines el paso Callback usa Amazon SQS
- SageMaker AI siempre usa contenedores: imágenes prediseñadas, imágenes prediseñadas con requirements.txt o extendidas, o contenedores propios. Un contenedor de inferencia propio debe servir /invocations y /ping en el puerto 8080, leer el modelo de /opt/ml/model, usar ENTRYPOINT en forma exec para recibir SIGTERM y ofrecer en /ping que informe de fallos reales por que se sustituya la instancia
- El parámetro informConfig del create_endpoint crea una subred y grupos de seguridad de la VPC mediante interfaces de red. Sin internet, hace falta un VPC endpoint de S3 y, para recursos fuera de AWS, un NAT gateway. Los clientes invocan el endpoint en privado con un interface endpoint de SageMaker AI Runtime, y el endpoint de la red invoca al contenedor basándose en un modelo que sigue create_model, create_endpoint_config y create_endpoint; con inference components se añade create_inference_component para asignar recursos y copias a cada modelo en un endpoint compartido. El SageMaker Python SDK usa ModelBuilder y deploy(). UpdateEndpoint aplica una nueva configuración sin cambiar el nombre del endpoint.

Implement automated orchestration and continuous integration and continuous delivery (CI/CD) pipelines for MLOps and AI workloads

- Las deployment guardrails de SageMaker AI actualizan endpoints en tiempo real y asíncronos con blue/green (all at once, canary o linear) o con despliegues rolling por lotes. Durante el baking period, unas alarmas de CloudWatch indicadas en AutoRollbackConfiguration vigilan la nueva flota y, si saltan, el tráfico vuelve a la flota anterior. Rolling update hace transiciones (antes AWS CodeStar Connections) conecta CodePipeline, CodeBuild y CodeDeploy con repositorios de terceros como GitHub o Bitbucket; una conexión creada con la CLI o CloudFormation queda en PENDING hasta completarla en la consola. Fallos típicos de CodeBuild: falta de privileged mode al construir imágenes Docker, por falta del permiso ProcessConnection, et al; y de servicio al usar el modo GitJob de SageMaker AI; el paso Transform ejecuta inferencia por lotes sobre un conjunto completo. El step caching, desactivado por defecto y con caducidad, reutiliza ejecuciones correctas con la misma configuración. Managed spot training reduce el coste;
- Las pruebas automáticas combinan pruebas de código en CodeBuild con pruebas de producción en CloudFormation report groups, y puertas de calidad en SageMaker Pipelines: un paso Condition lee métricas con JsonGet y un paso Fail detiene la ejecución. Los pasos QualityCheck y ClarifyCheck comparan datos y modelos con las baselines de la última versión
- SageMaker Pipelines es un cliente de Amazon EventBridge: una regla por patrón de eventos (por ejemplo, nuevos objetos en Amazon S3) o por programación inicia StartPipelineExecution y pasa parámetros estáticos o dinámicos extraídos del evento. Con el SDK de Python, PipelineSchedule y put_triggers crean programaciones de EventBridge Scheduler. En reentrenamientos por deriva, skip_check y register_new_baseline controlan las baselines.

Monitor ML and AI model inference and performance

- CloudWatch generative AI observability ofrece vistas predefinidas de invocaciones, latencia, tokens, errores y throttling de los modelos de Bedrock; para ver el contenido de cada prompt hay que activar model invocation logging. Las evaluaciones de Bedrock puntúan respuestas con un LLM como juez, incluso respuestas propias de producción.
- Para modelos clásicos, una canalización programada compara los datos con la línea base y avisa por SNS.
- SageMaker AI envía a EventBridge eventos de cambio de estado de trabajos, canalizaciones y pasos; los fallos incluyen el motivo y pueden disparar SNS, Lambda o Step Functions. En CloudWatch, ModelLatency mide el tiempo dentro del contenedor, OverheadLatency el que añade SageMaker AI y MidStreamErrors los fallos durante el streaming. La detección de anomalías de CloudWatch aprende bandas con estacionalidad.
- La deriva de datos se detecta comparando las entradas capturadas en producción con una línea base calculada sobre los datos de entrenamiento; no necesita etiquetas, a diferencia de la calidad del modelo. SageMaker Clarify detecta además la deriva de atribución de características con la puntuación NDCG. Como opción ligera, se publican estadísticas como métricas personalizadas con alarmas de detección de anomalías de CloudWatch.
- En SageMaker AI, una prueba A/B aloja varios modelos como variantes de producción de un mismo endpoint y reparte el tráfico según sus pesos; TargetVariant fuerza una variante concreta por solicitud. CloudWatch publica métricas por variante y UpdateEndpointWeightsAndCapacities cambia los pesos sin actualizar el endpoint. Una prueba shadow, en cambio, no devuelve al usuario las respuestas del modelo nuevo.
- AgentCore publica en CloudWatch métricas de runtime y de gateway (errores de sistema y de usuario por herramienta, latencia, tiempo del destino) y, con instrumentación ADOT, sesiones, trazas y spans para localizar dónde falla una coordinación entre agentes. AgentCore Evaluations puntúa sesiones de forma continua con evaluación online. En ConverseStream, un stopReason max_tokens revela una respuesta truncada sin error.

Optimize and manage ML and AI infrastructure costs and performance

- Elige la familia de instancias según el modelo y la latencia: CPU suele bastar para ML tradicional y por lotes, las optimizadas para cómputo o GPU sirven para tiempo real exigente, y AWS Inferentia (con el AWS Neuron SDK) da mejor precio-rendimiento que GPU en deep learning. SageMaker AI Inference Recommender compara configuraciones con pruebas de carga y devuelve coste, latencia y rendimiento.
- En SageMaker AI, compara ModelLatency (tiempo del contenedor) con OverheadLatency (sobrecarga del servicio) en CloudWatch para localizar la latencia. AgentCore Observability guarda métricas, spans y logs en CloudWatch; para ver trazas activa CloudWatch Transaction Search e instrumenta el agente con el ADOT SDK. AWS X-Ray muestra en el trace map qué servicio de la cadena añade latencia.
- Los paneles de CloudWatch reúnen métricas y alarmas en una sola vista; con la observabilidad entre cuentas, una cuenta de monitorización representa métricas de varias cuentas. Para Amazon Bedrock, la vista Model Invocations muestra invocaciones, latencia y tokens por modelo, y con metric math puedes calcular tokens de salida por segundo para separar lentitud del modelo de respuestas más largas.
- Para fiabilidad, despliega varias instancias por endpoint de producción y, en una VPC, subredes en al menos dos zonas de disponibilidad. El auto scaling de SageMaker AI recomienda seguimiento de objetivo, acciones programadas para picos previsibles y escalado por pasos cuando hace falta lógica avanzada, como escalar desde cero instancias con componentes de inferencia.
- AWS Budgets fija límites y puede ejecutar acciones, como aplicar una política de IAM o una SCP, al superar un umbral real o previsto. AWS Cost Anomaly Detection usa machine learning para detectar gastos inusuales y señalar su causa. Para repartir costes de Amazon Bedrock por equipo, etiqueta perfiles de inferencia de aplicación y activa esas etiquetas de asignación de costes.

Secure ML and AI workloads and model endpoints

- Amazon Inspector analiza imágenes de Amazon ECR (paquetes del sistema y de lenguajes) y, con análisis continuo, las revisa de nuevo al publicarse CVE. En CI/CD, sus plugins o la acción InspectorScan de CodePipeline generan un SBOM y hacen fallar la compilación si se supera un umbral de severidad. Inspector Code Security revisa el código fuente con SAST, SCA e IaC.
- El rol de ejecución de SageMaker AI debe limitarse a los ARN exactos que usa cada trabajo: bucket y objetos de entrada, ruta de salida y repositorio de ECR. Las etiquetas con condiciones ResourceTag separan los recursos de cada equipo sin editar políticas. Para artefactos compartidos entre cuentas, la etiqueta de sesión sagemaker:ModelPackageArn permite leerlos solo durante el despliegue del paquete.
- Las aplicaciones deben usar roles de IAM con credenciales temporales y permisos limitados a recursos concretos, como el ARN de un modelo de Amazon Bedrock. Las claves de condición imponen la configuración: sagemaker:VpcSubnets, sagemaker:NetworkIsolation o sagemaker:VolumeKmsKey en los trabajos de SageMaker AI, y bedrock:GuardrailIdentifier para exigir una barrera de protección en cada inferencia.
- CloudTrail registra quien llama a las API de SageMaker AI y Amazon Bedrock; InvokeModel y Converse son eventos de administración, mientras que InvokeAgent o Retrieve son eventos de datos que hay que activar con selectores avanzados. El registro de invocaciones de Bedrock guarda prompts y respuestas en S3 o CloudWatch Logs, y las reglas administradas de AWS Config evalúan la configuración de los recursos de SageMaker AI.
- Muchos errores de acceso en SageMaker AI vienen de iam:PassRole: quien crea un trabajo necesita permiso para pasar el rol de ejecución al servicio, limitado con iam:PassedToService. Si los datos de S3 usan una clave KMS administrada por el cliente, el rol de ejecución necesita kms:Decrypt también en la política de la clave. En Amazon Bedrock, exigir una barrera con bedrock:GuardrailIdentifier puede dar AccessDenied en RetrieveAndGenerate.