机器学习剖析

AWS 架构完善的框架

2020 年 4 月

声明

客户有责任对本文档中的信息进行独立评估。本文档：(a) 仅供参考，(b) 代表 AWS 当前的产品服务和实践，如有变更，恕不另行通知，以及 (c) 不构成 AWS 及其附属公司、供应商或授权商的任何承诺或保证。AWS 产品或服务均“按原样”提供，没有任何明示或暗示的担保、声明或条件。AWS 对其客户的责任和义务由 AWS 协议决定，本文档与 AWS 和客户之间签订的任何协议无关，亦不影响任何此类协议。

目录

摘要

本文档描述了适用于 AWS 架构完善的框架的*机器学习剖析*。本文档涵盖了常见的机器学习 (ML) 场景，并指明了确保工作负载架构设计符合最佳实践的关键元素。

简介

AWS 架构完善的框架能够帮助您认识到您在 AWS 上构建系统时所做决策的优缺点。通过使用此框架，您将了解在云中设计和运行可靠、安全、高效且具成本效益的系统的架构最佳实践。它提供了一种方法，使您能够根据最佳实践持续衡量架构，并确定需要改进的方面。我们相信，拥有架构完善的系统能够大大提高实现业务成功的可能性。

在本*机器学习剖析*中，我们重点介绍了如何在 AWS 云中对机器学习工作负载进行设计、部署和架构设计。我们会将本剖析添加到架构完善的框架中涵盖的最佳实践。为简洁起见，我们仅在本剖析中详细介绍专门针对机器学习 (ML) 的工作负载。在设计 ML 工作负载时，应使用 AWS 架构完善的框架白皮书中适用的最佳实践和问题。

本剖析面向的是技术人员，例如首席技术官 (CTO)、架构师、开发人员和运维团队成员。阅读本文后，您将了解在 AWS 上设计和运行 ML 工作负载时可以采用的最佳实践和策略。

定义

机器学习剖析以卓越运营、安全性、可靠性、性能效率和成本优化这五大支柱为基础。AWS 提供了适用于 ML 工作负载的多种核心组件，让您能够为 ML 应用程序设计出稳健的架构。

构建机器学习工作负载时，应对以下两个方面进行评估：

- 机器学习堆栈
- 机器学习工作负载的阶段

机器学习堆栈

在 AWS 中构建基于 ML 的工作负载时，可以选择不同的抽象化级别，以通过自定义级别和 ML 技能级别来平衡推向市场的速度：

- 人工智能 (AI) 服务
- ML 服务
- ML 框架和基础设施

AI 服务

AI 服务级别提供了完全托管的服务，支持您使用 API 调用将 ML 功能快速添加到工作负载中。因此，您可以构建功能强大的智能应用程序，并具有计算机视觉、语音、自然语言、Chatbot、预测和推荐等功能。此级别的服务基于预先训练或自动训练的机器学习和深度学习模型，因此您无需掌握 ML 知识即可使用它们。

AWS 提供了许多 AI 服务，您可以通过 API 调用将它们与您的应用程序集成。例如，您可以使用 Amazon Translate 翻译或本地化文本内容，使用 Amazon Polly 进行文本到语音的转换，以及使用 Amazon Lex 构建会话聊天机器人。

ML 服务

ML 服务级别为开发人员、数据科学家和研究人员提供了面向机器学习的托管服务和资源。这些类型的服务让您能够标记数据，构建、训练、部署和操作自定义 ML 模型，而无需担心底层基础设施的需求。基础设施的管理繁重而且千篇一律，由云供应商负责管理后，您的数据科学团队就可以专注于最擅长的领域了。

在 AWS 中， Amazon SageMaker 让开发人员和数据科学家能够快速轻松地构建、训练和部署任意规模的机器学习模型。例如，Amazon SageMaker Ground Truth 可帮助您快速构建高度精确的 ML 训练数据集；而借助 Amazon SageMaker Neo，开发人员对 ML 模型进行一次训练后，即可在云或边缘环境的任意位置运行它们。

ML 框架和基础设施

ML 框架和基础设施级别面向专家级机器学习从业人员。这些人员熟知如何设计自己的工具和工作流程来构建、训练、调优和部署模型，他们对在框架和基础设施级别工作习以为常。

在 AWS 中，您可以使用开源 ML 框架，例如 TensorFlow、PyTorch 和 Apache MXNet。此级别的深度学习 AMI 和深度学习容器预安装了多个已针对性能进行优化的 ML 框架。这种优化让它们随时可以在功能强大、经过 ML 优化的计算基础设施（例如 Amazon EC2 P3 和 P3dn 实例）上启动，从而提高机器学习工作负载的速度和效率。

组合级别

工作负载通常使用 ML 堆栈多个级别的服务。可以根据业务使用案例对不同级别的服务和基础设施进行组合，以满足多种需求和实现多个业务目标。例如，您可以使用 AI 服务对零售网站上的客户评论进行情绪分析，以及通过托管的 ML 服务利用自己的数据来构建自定义模型，预测未来的销售情况。

ML 工作负载的阶段

构建和运行典型的 ML 工作负载是一个迭代过程，其中包含多个阶段。我们以跨行业数据挖掘标准流程 (CRISP-DM) 的开放标准流程模型作为一般性指导原则，来大致标识这些阶段。之所以将

CRISP-DM 作为基准，是因为它是经过验证的业界公认的工具，而且不受应用程序限制，这使得它易于应用，可应用于各种 ML 管道和工作负载。

端对端机器学习流程包括以下阶段：

- 确定业务目标
- 制定 ML 问题框架
- 数据收集和集成
- 数据准备
- 数据可视化和分析
- 特征工程
- 模型训练
- 模型评估
- 业务评估
- 生产部署（模型部署和模型推理）

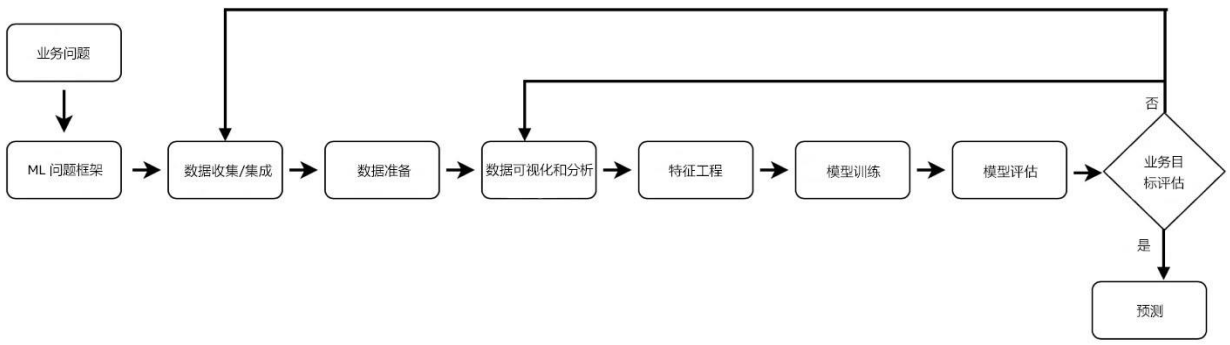

图 1 – 端到端机器学习流程

确定业务目标

确定业务目标是最重要的阶段。如果组织考虑使用 ML，它应该清楚地了解要解决的问题，以及通过使用 ML 解决该问题可以实现的商业价值。您必须能够根据特定的业务目标和成功标准来衡

量商业价值。尽管这对于任何技术解决方案都适用，但是在考虑 ML 解决方案时，这一步特别具有挑战性，因为 ML 是一项变革性技术。

确定成功标准后，请评估组织实际执行该目标的能力。应该将目标定在可实现范围之内，而且目标应该能够为生产指明清晰的道路。

您将需要验证 ML 是否为实现业务目标的适当方法。在确定方法时，请评估可用于实现目标的所有选项、所产生结果的准确性以及每种方法的成本和可扩展性。

要想基于 ML 的方法取得成功，拥有大量相关的、高质量的数据（这些数据适用于您要训练的算法）是必须可少的。仔细评估数据的可用性，以确保可以使用和访问正确的数据源。例如，您不仅需要训练数据来训练 ML 模型并对其进行基准测试，还需要获取业务数据来评估 ML 解决方案的价值。

应用以下最佳实践：

- 了解业务需求
- 提出业务问题
- 确定项目的 ML 可行性和数据要求
- 评估采集、训练、推理和错误预测数据的成本
- 查看相似领域中业经验证或已发布的作品（如果有）
- 确定关键性能指标（包括可接受的错误）
- 根据业务问题定义机器学习任务
- 确定必备的关键功能

制定 ML 问题框架

在这个阶段，将业务问题的框架设定为机器学习问题：可以看到的内容和应该预测的内容（称为标签或目标变量）。确定要预测的内容以及需要如何优化性能和错误指标是 ML 中的关键一步。

例如，设想这样一种情形：一家制造公司想要确定哪些产品将产生最大的利润。实现这一业务目标在一定程度上取决于确定要生产的合适的产品数量。在这种情形中，您希望根据过去和当前的

销售情况预测产品未来的销售情况。预测未来的销售情况成为要解决的问题，而解决该问题的其中一种方法就是使用 ML。

应用以下最佳实践：

- 定义项目取得成功的标准
- 为项目建立可观察和可量化的性能指标，例如精确度、预测延迟或最小化存货价值
- 根据输入、期望的输出和要优化的性能指标来阐述 ML 问题
- 评估 ML 是否可行以及是否是合适的方法
- 创建数据源和数据注释目标，以及实现该目标的策略
- 从易于解释的简单模型开始，这样将更易于管理调试

数据收集

在 ML 工作负载中，数据（输入和相应的期望输出）具有三个重要功能：

- 定义系统的目标：输出表示形式以及每个输出与每个输入之间的关系（通过输入/输出对实现）
- 训练将输入与输出相关联的算法
- 测量经过训练的模型的性能，并评估是否满足性能目标

第一步是确定 ML 模型所需的数据，并评估可用于收集数据以训练模型的各种方法。

随着组织收集和分析的数据量越来越大，用于数据存储、数据管理和分析的传统本地解决方案已无法满足需求。云端数据湖是一个集中式存储库，可让您存储任何规模的所有结构化和非结构化数据。您可以按原样存储数据（无需首先构建数据），并运行各种类型的分析（从控制面板和可视化对象到大数据处理、实时分析和 ML），来指导您做出更好的决策。

AWS 提供了很多方法来从静态资源或从动态生成的新资源（例如网站、移动应用程序和互联网连接的设备）中批量提取数据。例如，您可以使用 Amazon Simple Storage Service (Amazon S3) 来构建高度可扩展的数据湖。您可以使用 AWS Lake Formation 来轻松地设置数据湖。

您可以使用 AWS Direct Connect 将数据中心通过私有网络直接连接到 AWS 区域，来提取数据。您可以使用 AWS Snowball 批量物理传输 PB 级数据；或者，使用 AWS Snowmobile 传输 EB 级数据。您可以使用 AWS Storage Gateway 集成现有的本地存储，或使用 AWS Snowball Edge 添加云功能。您还可以使用 Amazon Kinesis Data Firehose 收集并提取多个流式传输数据源。

应用以下最佳实践：

- 详细介绍提取数据所需的各种源和步骤
- 确认数据的可用性，包括数量和质量
- 在准备供下游使用之前全面了解数据
- 定义数据监管：数据的所有者、可以访问数据的用户、数据的适用范围以及按需访问和删除特定数据的能力
- 跟踪数据沿袭，以便在进一步处理期间跟踪并了解位置和数据源
- 使用托管的 AWS 服务来进行数据收集和集成
- 使用集中式方法来存储数据，例如数据湖

数据准备

ML 模型的好坏取决于用于训练它们的数据。完成收集数据后，数据的集成、注释、准备和处理是至关重要的过程。判断训练数据是否适用的一个重要特征就是：它能否以最适合学习和归纳的方式提供。数据准备应从少量统计有效的样本开始，然后使用不同的数据准备策略反复进行改进，同时继续保持数据完整性。

AWS 提供了多种可大规模注释数据以及提取、转换和加载 (ETL) 数据的服务。

Amazon SageMaker 是一项完全托管的服务，涵盖整个 ML 工作流程，包括标记和准备数据，选择、训练、调整和优化算法以用于部署，以及做出预测。

Amazon SageMaker Ground Truth 可以方便地访问公开和私有的人工标签工具，并提供内置的工作流程和用户界面来处理标签任务。 它使用机器学习模型自动标记原始数据，来快速生成高质量的训练数据集，而其成本远远低于人工标记。只有在主动学习模型在标记数据没有把握时，才

会将数据路由到人工端。 该服务提供动态自定义工作流程、进行作业链接和作业跟踪，通过将以前的标记作业的输出用作新标记作业的输入，来节省后续 ML 标记作业的时间。

AWS Glue 是一项完全托管的提取、转换和加载 (ETL) 服务，可用于自动化 ETL 管道。AWS Glue 借助 Glue 数据目录自动发现和配置数据，推荐并生成 ETL 代码，以将源数据转换为目标 schema，并在完全托管的横向扩展 Apache Spark 环境中运行 ETL 作业，以将数据加载到其目的地。它还支持设置、协调和监控复杂的数据流。

Amazon EMR 提供一个托管的 Hadoop 框架，使您可以轻松、快速地处理跨动态可扩展的 Amazon EC2 实例的大量数据。您还可以运行 EMR 中其他常用的分布式框架（例如 Apache Spark、HBase、Presto 和 Flink），以及与其他 AWS 数据存储服务（例如 Amazon S3 和 Amazon DynamoDB）中的数据进行交互。

数据准备不仅适用于用以构建机器学习模型的训练数据，还适用于在模型部署后用于对模型进行推理的新业务数据。通常情况下，您应用于训练数据的数据处理步骤与应用于推理请求的数据处理步骤的顺序应相同。

Amazon SageMaker 推理管道部署了管道，以便您可以传递原始输入数据并对实时和批量推理请求执行预处理、预测和后处理。推理管道可以让您重复使用现有的数据处理功能。

应用以下最佳实践：

- 从少量统计有效的样本开始进行数据准备
- 反复尝试不同的数据准备策略
- 在数据清理流程中实施反馈循环，从而在数据准备步骤中提醒异常
- 持续加强数据完整性
- 利用托管的 ETL 服务

数据可视化和分析

了解数据的一个关键方面是识别模式。如果您只查看表中的数据，这些模式通常并不明显。正确的可视化工具可以帮助您快速且更深入地了解数据。在创建任何图表或图形之前，您必须先确定想要显示的内容。例如，图表可以传达关键绩效指标 (KPI)、关系、比较、分布或组成等信息。

AWS 提供了多种可用于大规模可视化和分析数据的服务。

Amazon SageMaker 提供了一个托管的 Jupyter 笔记本环境，可用于可视化和分析数据。Project Jupyter 是一个开源 Web 应用程序，支持您创建可视化对象和叙述文字，以及执行数据清理、数据转换、数值模拟、统计建模和数据可视化。

Amazon Athena 是一项完全托管的交互式查询服务。借助它，您可以使用 ANSI SQL 运算符和函数来查询 Amazon S3 中的数据。Amazon Athena 是一项无服务器服务，可以无缝扩展来满足您的查询需求。

Amazon Kinesis Data Analytics 通过分析流数据来获得可行性见解，从而提供实时分析功能。该服务可自动扩展，来满足传入数据的量和吞吐量需求。

Amazon QuickSight 是一项基于云的商业智能 (BI) 服务，可提供控制面板和可视化功能。该服务可自动扩展，以为数百用户提供支持，并为情节提要提供安全共享和协作功能。此外，该服务具有内置的 ML 功能，可提供立即使用的异常检测、预测和假设分析。

应用以下最佳实践：

- 配置数据（分类、序数以及定量可视化）
- 为您的使用案例（例如数据大小、数据复杂性以及实时与批处理）选择正确的工具或工具组合
- 监控数据分析管道
- 验证数据假设

特征工程

在通过可视化和分析探究并了解数据之后，就可以进行特征工程了。我们将数据的每个唯一属性都视作一个特征。例如，在设计用于预测客户流失问题的解决方案时，您可以先持续收集客户数据。客户数据可以捕获客户位置、年龄、收入水平和近期购买商品等特征（也称为属性）。

特征工程是指在使用机器学习或统计建模创建预测性模型时选择和转换变量的过程。特征工程通常包括特征创建、特征转换、特征提取和特征选择。

- **特征创建**可识别数据集中与当前问题相关的特征。

- **特征转换**可管理替换丢失的特征或无效的特征。其中包括形成特征的笛卡尔积、非线性转换（例如将数值变量归类）以及创建特定于域的特征等技术。
- **特征提取**是指从现有特征创建新特征的过程，通常是为了减少特征的维度。
- **特征选择**是指从数据集中过滤掉不相关或冗余特征的过程。通常情况下，需要通过观察方差或相关性阈值确定要删除的特征来完成这一过程。

Amazon SageMaker 为 Jupyter 笔记本环境提供了 Spark 和 scikit-learn 预处理器，可用于特征设计和数据转换。借助 Amazon SageMaker，您还可以使用 ETL 服务（例如 AWS Glue 或 Amazon EMR）来运行特征提取和转换作业。此外，您还可以使用 Amazon SageMaker 推理管道来重复使用现有的数据处理功能。

Amazon SageMaker Processing 可以在完全托管的环境中借助 Amazon SageMaker 提供的所有安全性和合规性功能大规模运行特征工程（和模型评估）的分析作业。借助 Amazon SageMaker Processing，您可以灵活地使用内置数据处理容器或自带容器，以及提交自定义作业在托管的基础设施上运行。提交后，Amazon SageMaker 会启动计算实例、处理输入数据并对其进行分析，并在完成后释放资源。

应用以下最佳实践：

- 请领域专家帮助评估特征的可行性和重要性
- 去除掉冗余和不相关的特征（以减少数据中的噪声并减少关联性）
- 首先从上下文中归纳特征
- 在构建模型时进行迭代（新特征、特征组合和新调整的目标）

模型训练

在本阶段，您首先选择适合解决您问题的机器学习算法，然后训练 ML 模型。在训练中，您需为算法提供用于学习的训练数据，并设置模型参数来优化训练过程。

通常情况下，训练算法会计算多个指标，例如训练误差和预测准确性。这些指标有助于确定模型的学习情况是否良好，以及是否可以进行很好地归纳，以对隐藏数据进行预测。算法报告的指标

取决于业务问题以及您使用的 ML 技术。例如，分类算法是通过捕获真假正和真假负的*混淆矩阵*来衡量，而回归算法则是通过均方根差 (RMSE) 来衡量。

可以调整为控制 ML 算法的行为和结果模型架构的设置称为*超参数*。ML 算法中超参数的数量和类型依据各模型而定。以下是一些常用的超参数示例：学习速率、Epoch 数、隐藏层、隐藏单元和激活函数。超参数调整或优化是指选择最佳模型架构的过程。

Amazon SageMaker 提供了多种热门的内置算法，可以使用您已准备好并存储在 Amazon S3 中的训练数据来对其进行训练。您也可以自带自定义算法，在 Amazon SageMaker 上进行训练。应使用 Amazon ECS 和 Amazon ECR 对自定义算法进行容器化。

选择算法后，您可以通过 API 调用在 Amazon SageMaker 上启动训练。您可以选择在单个实例或分布式实例集群上进行训练。训练流程所需的基础设施管理由 Amazon SageMaker 管理，从而消除了无差别的繁重工作负担。

Amazon SageMaker 还支持通过超参数调优作业来实现模型自动调优。完成配置后，超参数调优作业会使用您指定的算法和超参数范围在数据集上运行诸多训练作业，从而找到最佳的模型版本。然后，它选择会产生最佳性能模型（根据所选指标进行衡量）的超参数值。您将 Amazon SageMaker 自动模型调优与内置算法、自定义算法和 Amazon SageMaker 预构建的 ML 框架容器结合使用。

Amazon SageMaker Debugger 通过监控、记录和分析定期捕获的训练作业状态的数据，帮助用户直观地了解 ML 训练流程。此外，它还可以在训练流程中对于捕获的数据执行交互式探索，并可针对在训练期间检测到的错误发出提醒。例如，它可以自动检测常见的错误并发出相应的提醒，例如梯度值变得太大或太小。

Amazon SageMaker Autopilot 通过自动处理数据预处理、算法选择和超参数调优简化 ML 训练流程。 通过它，仅以表格形式提供训练数据即可构建分类和回归模型。此功能通过数据预处理器、算法和算法参数设置的不同组合来探索多种 ML 解决方案，以找到最准确的模型。Amazon SageMaker Autopilot 从其原生支持的高性能算法列表中选择最佳算法。它还会自动尝试在这些算法中使用不同的参数设置，来获得最佳模型质量。然后，您可以直接将最佳模型部署到生产环境中，或者评估多个候选模型来权衡对比准确性、延迟性和模型大小等指标。

AWS Deep Learning AMI 和 **AWS Deep Learning Containers** 支持您使用多个开源 ML 框架来训练基础设施。AWS Deep Learning AMI 预安装了热门的深度学习框架和接口，例如

TensorFlow、PyTorch、Apache MXNet、Chainer、Gluon、Horovod 和 Keras。可以在对 ML 性能进行了优化、功能强大的基础设施上启动 AMI 或容器。

Amazon EMR 具有分布式集群功能，还可以用于根据存储在集群本地或 Amazon S3 中的数据运行训练作业。

应用以下最佳实践：

- 在训练模型之前生成模型测试计划
- 清楚了解您需要训练的算法类型
- 确保训练数据能够代表您的业务问题
- 使用托管服务进行训练部署
- 采用渐进式训练或转换学习策略
- 如果按照目标指标进行衡量后，发现结果没有得到显著改善，那么应尽早停止训练作业，以避免过拟合并降低成本
- 密切监控您的训练指标，因为模型性能可能会随着时间的推移而降低
- 利用托管服务进行自动模型调优

模型评估和业务评估

对模型进行训练之后，对其进行评估，以确定其性能和准确性是否能够帮助您实现业务目标。您可能希望使用不同的方法来生成多个模型，并评估每个模型的有效性。例如，您可以为每个模型应用不同的业务规则，然后进行各种衡量来确定每个模型的适用性。您可能还需要评估模型是需要更敏感还是更独特。对于多类模型，请分别评估每个类的错误率。

您可以使用历史数据（离线评估）或实时数据（在线评估）来评估模型。在离线评估中，使用留作*保留集*的部分数据集对经过训练的模型进行评估。这些保留数据从未用于模型训练或验证，仅用于评估最终模型中的错误。保留数据注释需要具有较高的准确性，这样评估才有意义。分配额外的资源来验证保留数据的准确性。

用于模型训练的 AWS 服务在此阶段也会发挥作用。可以使用 Amazon SageMaker、AWS Deep Learning AMI 或 Amazon EMR 来执行模型验证。

您可以根据评估结果对数据和/或算法进行微调。在微调数据时，将应用数据清理、准备和特征工程的概念。

应用以下最佳实践：

- 清晰地了解衡量成功的标准
- 根据项目的业务期望评估模型指标
- 规划和执行生产部署（模型部署和模型推理）

对模型进行训练、调整和测试之后，您可以将模型部署到生产环境中，并对模型进行推理（预测）。

Amazon SageMaker 为部署和推理提供了多种选择，同时也是托管生产 ML 模型的理想 AWS 服务。

与模型训练一样，您可以使用 API 调用在 Amazon SageMaker 上托管模型。您可以选择在单个实例上托管模型，也可以跨多个实例托管模型。同一 API 让您能够配置自动扩展，以满足 ML 模型上的各种推理需求。托管模型所需的基础设施管理完全由 Amazon SageMaker 来负责，从而摆脱了无差别的繁重工作的负担。

Amazon SageMaker 推理管道使您能够部署推理管道，以便您可以传递原始输入数据，并对实时和批量推理请求执行预处理、预测并完成后处理。推理管道可以由任何 ML 框架、内置算法或可在 Amazon SageMaker 上使用的自定义容器组成。您可以使用 Spark ML 和 Scikit-learn 框架容器中提供的一系列特征转换器构建特征数据处理和特征工程管道，并将它们部署为推理管道的一部分，以重用数据处理代码并简化 ML 流程管理。这些推理管道是完全托管的，可以将预处理、预测和后处理整合到数据科学过程中。

Amazon SageMaker Model Monitor 持续监控生产环境中的 ML 模型。在生产环境中部署 ML 模型后，实际数据可能会与用于训练模型的数据有所不同，从而导致模型质量出现偏差，并最终导致模型准确性降低。Model Monitor 可以检测到偏差（例如数据偏差）。随着时间的推移，这些偏差会降低模型性能，并提醒您采取补救措施。

Amazon SageMaker Neo 支持只对 ML 模型进行一次训练，即可在云和边缘的任何位置运行它们。Amazon SageMaker Neo 由一个编译器和一个运行时组成。编译 API 读取从各框架中导出的模型，将它们转换为与不依赖于框架的表现形式，并生成经过优化的二进制代码。然后，每个目标平台的运行时将加载并执行编译后的模型。

Amazon Elastic Inference 可以将低成本 GPU 驱动的加速连接到 Amazon EC2 和 Amazon SageMaker 实例，降低运行深度学习推理的成本。独立 GPU 实例专为模型训练而设计，对于推理通常容量过大。虽然训练作业可并行批量处理数百个数据样本，但大多数推理在单个输入中实时发生，而且仅占用少量 GPU 计算。Amazon Elastic Inference 可以解决这一问题，方法是：将适当数量的 GPU 驱动的推理加速连接到任何 Amazon EC2 或 Amazon SageMaker 实例类型，无需更改代码。

虽然 TensforFlow 和 Apache MXNet 等深度学习框架原生支持 Elastic Inference，不过您也可以使用 Open Neural Network Exchange (ONNX) 导出模型并将其导入 MXNet，将 Elastic Inference 与其他深度学习框架一起使用。

应用以下最佳实践：

- 监控生产环境中的模型性能并与业务期望相对比
- 监控模型在训练和生产环境中的性能差异
- 检测到模型性能发生改变时，重新训练模型。例如，由于出现新的竞争，销售预期和后续预测可能都会发生变化
- 如果要对整个数据集进行推理，可以使用批量转换来托管服务
- 利用生产变体通过 A/B 测试来测试新模型的变化

一般设计原则

架构完善的框架定义了一系列一般性设计原则，有助于实现优良的机器学习工作负载云端设计：

- **通过高数据质量数据集的可用性实现敏捷性**

  数据科学工作负载需要访问交付管道中所有阶段的实时数据或批量数据。实施相应机制，以允许访问通过数据验证和质量控制的数据。

- **从简单的模型入手，通过试验循序渐进**

  从小的特征集入手，可以避免由复杂模型入手导致的错误，也可以避免丢失特征影响跟踪。选择一个简单的模型，并在整个流程中执行一系列实验。

- **将模型训练和评估与模型托管分离开来**

  通过分离模型训练、模型评估和模型托管资源，来选择与数据科学生命周期中特定阶段最匹配的资源。

- **检测数据偏差**

  要管理随时间推移产生的数据偏差，需在模型投产后持续测量推理的准确性。ML 中使用的数据通常有多个数据源，数据的形态和含义会随上游系统和流程的变化而变化。制定适当的机制来检测这些变化，以便可以采取适当的措施。

- **自动化训练和评估管道**

  通过自动化，您可以触发自动训练模型和创建模型构件，然后将它们一致地部署到多个终端节点环境。应用自动化来触发模型重新训练活动，可以减少人工工作量和人为错误，并支持持续改进模型性能。

- **选择更高的抽象级别，加速成果实现**

  选择合适的 AI/ML 服务时，应首先评估更高级别的服务是否适用，然后作为一种机制来快速实现业务目标、摆脱无差别的繁重工作并降低开发成本。

场景

下面介绍了一些会影响 AWS 上的机器学习工作负载的设计和架构的常见场景。每个场景均涵盖常见的设计驱动因素以及向您展示如何实施每种场景的参考架构。

使用 AWS AI 服务构建智能应用程序

如果组织希望通过最低的开发工作量和较短的周期向现有或新应用程序添加 AI 功能，机器学习堆栈中的 AWS AI 服务层是不错的选择。该层中的服务提供了完全托管、随时可用的计算机视觉、语音、自然语言和 Chatbot 功能。

开发人员使用这些服务时，无需对 ML 流程的数据准备、数据分析、模型训练和评估阶段进行管理。而是，可以通过简单的 API 调用将这些功能集成到应用程序中。

Amazon Comprehend 是一种自然语言处理 (NLP) 服务，它利用 ML 帮助您发现非结构化文本数据中的见解和关系。首先，服务会识别文本的语言。然后，它提取关键短语、地点、人物、品牌和事件。它使用令牌化和部分语音来分析文本，由于该服务了解文本的正面或负面程度，因此可以根据主题自动组织文本文件集合。您也可以使用 Amazon Comprehend 中的 AutoML 功能来构建一组自定义的实体或文本分类模型，这些实体或文本分类模型可根据组织的需求量身定制。

Amazon Lex 是一项可在任何使用语音和文本的应用程序内构建对话接口的服务。Amazon Lex 可提供自动语音识别 (ASR) 的高级深度学习功能，用于将语音转化成文本；以及自然语言理解 (NLU)，用于识别文本的意图。借助这些功能，您能构建具有高度吸引力的客户体验和逼真对话互动的应用程序。

Amazon Polly 是一种可将文本转换为逼真语音的服务，可让您构建支持聊天的应用程序，并且打造全新类别的具有语音功能的产品。Amazon Polly 是一种*文本转语音*服务，它使用先进的深度学习技术来合成听起来像是人声的语音。

Amazon Rekognition 让您能够轻松地向应用程序中添加图像和视频分析。您向 Amazon Rekognition API 提供一个图像或视频后，该服务即可识别物体、人物、文本、场景和活动，并且检测任何不当内容。Amazon Rekognition 还可以根据您提供的图像和视频，提供高度准确的面部分析和面部识别。您可以在各种各样的用户验证、人数统计和公共安全使用案例中检测、分析和比较面部。

Amazon Transcribe 是一种自动语音识别 (ASR) 服务，您可以通过它轻松地为应用程序添加语音转文本功能。通过使用 Amazon Transcribe API，您可以分析存储在 Amazon S3 中的语音文件，并让该服务返回转录语音的文本文件。您还可以将实时音频流发送到 Amazon Transcribe 并实时接收转录流。

Amazon Translate 是一种神经机器翻译服务，可提供快速、优质且经济实惠的语言翻译。神经机器翻译是一种语言翻译自动化形式，使用深度学习模型来提供比基于统计数据和规则的传统机器翻译算法更加准确、自然的声音翻译。借助 Amazon Translate，您可以对内容（如网站和应用程序内容）进行本地化，以方便国际用户使用，并且可以轻松、高效地翻译大量文本。

AWS AI 服务的响应还包括*置信度分数*，它表示 AI 服务对特定结果的信任程度。本质上来讲，所有 ML 系统都是概率性的，因此可以使用置信度分数来衡量系统对其结果的信任程度。使用 AI 服务时，请确保设置适当的阈值，并确保该阈值适用于您的特定使用案例。对于多级模型，请使用每个类的阈值，基于类错误率进行设置。例如，使用 Amazon Rekognition 衡量某事件中人群的兴趣度可能需要的置信度得分阈值较低，而使用相同的服务来分析医学图像可能需要的阈值就比较高。如果特定领域的使用案例的结果会产生较大影响（如医学图像分析），那么它可能还需要由医学专家进行二级验证。

由于 AI 服务是无服务器服务并采用按使用量付费的模式，因此您可以根据业务情况增加服务，并且可降低初始阶段和非高峰时间的成本。AI 服务的无服务器性质使它们成为使用 AWS Lambda 的事件驱动型架构的理想选择。借助 AWS Lambda，您可以为几乎任何类型的应用程序或后端服务运行代码，而且无需任何管理。您只需按使用的计算时间付费，未运行代码时不会产生费用。

我们来看一个示例使用案例：为了改善用户体验并吸引客户，捕获和分析零售商店中的客户统计数据。在捕获和处理面部图像时，必须实施保护措施来保护这些数据，并在使用这些数据之前应用适当的置信度级别。图 2 中显示的参考架构显示了如下实施过程：使用 Amazon Rekognition 进行面部分析、使用 Amazon Athena 分析面部属性数据，以及使用 Amazon QuickSight 分析可视化对象。

使用面部分析技术必须遵守所有法律，包括保护公民权利的法律。AWS 客户有责任遵守所有有关技术使用的适用法律。AWS 可接受使用政策 (AUP) 禁止客户违法使用任何 AWS 服务（包括 Amazon Rekognition），违反我们 AUP 的客户将无法使用我们的服务。

参考架构

图 2 – 客户统计数据分析解决方案

此参考架构包括以下高级流程：

- 创建一个 Amazon S3 存储桶用于临时存储图像，并在存储桶上启用加密来保护图像安全。使用 AWS IAM 限制对 S3 存储桶的访问：为上传过程提供只写权限（无公共读取），为 AWS Lambda 函数提供只读权限。启用使用 CloudTrail 将 S3 存储桶的数据事件日志记录到单独的 S3 存储桶，以便您可以收集与存储桶有关的所有活动的日志。
- 在零售商店中捕获的客户图像已上传到 Amazon S3 存储桶，因此您需要制定生命周期策略，以确保在处理后自动删除图像。
- 上传到 Amazon S3 的每个图像都会触发一个 AWS Lambda 函数，您可以使用面部分析来了解年龄、性别和情绪等人口统计数据。Lambda 函数会调用 Amazon Rekognition 服务，以从图像中提取反映客户的年龄、性别和情绪（例如高兴、平静或生气）的面部属性。属性中还包括推理信息和置信度级别。
- 人口统计数据以 .csv 格式存储在第二个 Amazon S3 存储桶中。使用安全存储的唯一密钥对 .csv 文件进行加密。可以使用 AWS Key Management Service (AWS KMS) 等服务来管理和存储这些密钥。
- Amazon Athena 从 .csv 文件读取和加载人口统计数据以用于进行查询。Amazon Athena 支持源数据和查询结果的加密数据，例如，将 Amazon S3 与 AWS KMS 一起使用。为确保恰当地使用置信度级别，需使用 Amazon Athena 中的视图将搜索限制为仅对您的使用案例具有足够置信度的结果。

- 在 Amazon QuickSight 中构建客户见解控制面板。使用 AWS IAM 限制对 Amazon QuickSight 控制面板和 Amazon Athena 查询的访问，仅允许符合条件的人员访问，并安全记录所有访问信息。

在本示例中，关注的对象为图像，并使用 Amazon Rekognition 来分析图像。采取相应保护措施，以保护面部图像、自动删除图像以及使用和记录推理所用的置信度级别。通过过滤掉低置信度结果来正确使用置信度级别。该架构可用于通过适当的 AI 服务来分析各种类型的对象，例如文本或音频。例如，可以使用 Amazon Transcribe 转录音频文件，也可以使用 Amazon Comprehend 分析非结构化文本。

提升成熟度

虽然为了处理特定任务（如图像分析或转录)需要对单个 AI 服务进行预训练，但由于这些服务的无服务器特性，您可以使用 AWS Step Functions 编排多个服务来构建成熟的解决方案。例如，AWS 媒体分析解决方案可以帮助客户轻松分析、理解和构建一个可搜索现有媒体文件的目录。以下参考架构使用多个 AI 服务（Amazon Rekognition、Amazon Transcribe、Amazon Comprehend）来分析和提取媒体元数据。服务会为提取的元数据建立索引，并将其保留在 Amazon Elasticsearch 中，以使整个媒体内容都可搜索。

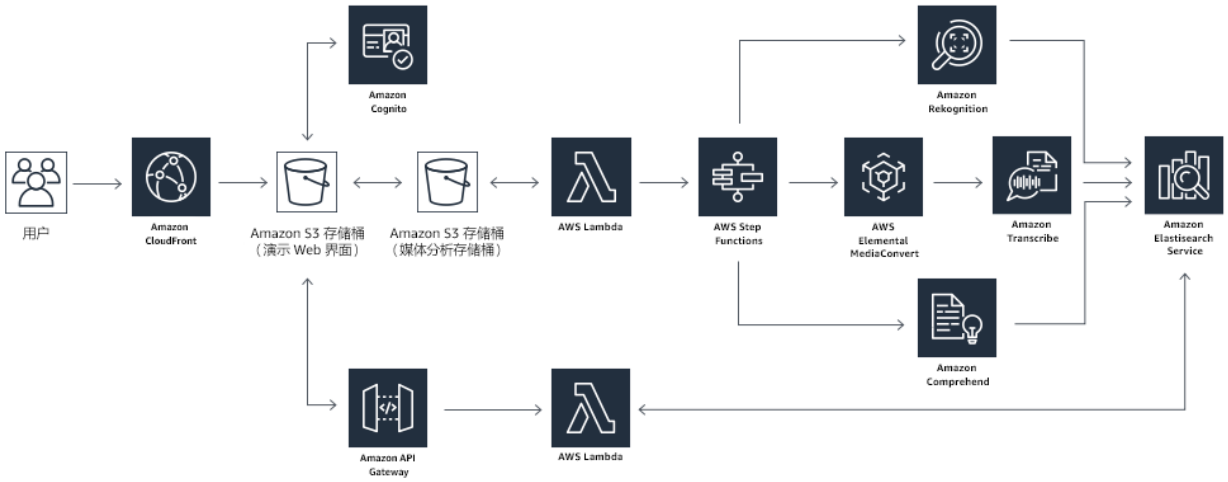

图 3 – 媒体分析解决方案参考架构

此参考架构包括以下高级流程：

- 在 Amazon S3 存储桶中部署 Web 界面，这样，您将可以立即使用简单的 Web 界面来分析小型媒体文件。使用 Amazon CloudFront 来限制对 Amazon S3 存储桶内容的访问。
- 上传的媒体文件会流经 Amazon API Gateway RESTful API、处理 API 请求的 AWS Lambda 函数以及 Amazon Cognito 用户池，实现与媒体文件的安全交互。
- AWS Step Functions 状态机可以协调媒体分析流程。第二个 Lambda 函数使用托管的 AI 服务（例如 Amazon Rekognition、Amazon Transcribe 和 Amazon Comprehend）来执行分析和元数据提取。
- 上传 MP4 视频文件后，AWS Elemental MediaConvert 会提取音频以供 Amazon Transcribe 和 Amazon Comprehend 进行分析。
- 系统会将元数据结果存储在 S3 存储桶中，并在 Amazon Elasticsearch Service（Amazon ES）集群中为其建立索引。

AI 服务可满足图像分析、语言翻译、转录等特定使用案例的需求，让您无需掌握大量机器学习和深度学习知识即可构建功能强大的智能功能。最终实现的结果就是可对您的业务目标进行快速试验和评估，从而缩短销售就绪时间。在本示例中，错误造成的影响很小，因此对于所有 ML 方法，您都可以使用较低的置信度级别。

将您的数据应用于 AI 服务

虽然我们之前讨论的 AI 服务都以预训练模型为基础，但是 AWS 也会提供能返回使用您的数据训练的 ML 模型的 AI 服务。

Amazon Personalize 是一项完全托管的服务。借助它，您可以根据您自己的用户-项目交互数据为应用程序创建私有的自定义个性化推荐。 无论是在应用程序内部进行及时的视频推荐，还是在恰当的时机发送个性化的通知电子邮件，这些基于您数据的个性化体验通常会为客户提供更多相关的体验，并带来更高的业务回报。

Amazon Forecast 是一项完全托管的服务，可根据您提供的历史数据生成高度准确的预测。 该服务使用深度学习从多个数据集学习，并能够自动尝试不同的算法，从而找到最适合您数据算法。它可用于多种使用案例，例如估计产品需求、云计算使用量、财务计划或供应链管理系统中的资源计划。

使用托管的 ML 服务构建自定义 ML 模型

您可以使用托管服务方法基于您自己的数据构建和部署 ML 模型，来创建商业价值的预测模型和指导性模型。使用托管的 ML 服务时，您的开发和数据科学团队将负责管理端到端 ML 流程的数据准备、数据分析、模型训练、模型评估和模型托管阶段。

Amazon SageMaker 是一种完全托管的服务，涵盖整个 ML 工作流程，包括标记和准备数据、选择算法、训练模型、为部署而对算法进行调整和优化、做出预测以及采取行动。为了让开发人员和数据科学家在构建 ML 模型时能够摆脱无差别的基础设施管理的负担，Amazon SageMaker 提供了以下功能：

- **收集和准备训练数据**

  使用 Amazon SageMaker Ground Truth 标记数据，并利用多个预构建的笔记本解决诸多常见的 ML 问题。

- **支持机器学习算法**

  从多种内置的高性能算法中进行选择、自带算法，或者通过 AWS Marketplace 找到适合您使用案例的算法。

- **模型训练**

  使用您自己的数据通过 API 调用（API 调用可设置、管理和终止高性能训练集群）训练 ML 模型。配置训练以使用单个实例，或选择多个实例来支持分布式训练。Amazon SageMaker Debugger 能够在训练期间自动捕获和分析数据，让用户实时了解训练过程。

- **模型优化**

  使用 Amazon SageMaker Neo，只需在 Amazon SageMaker 上对模型训练一次，即可针对其他 ML 框架对它进行优化。

- 在生产环境中部署模型

  使用 API 调用，将经过训练的模型部署到您选择的具有 Auto Scaling 功能的基础设施上。

- 监控已部署的模型

持续监控生产环境中的 ML 模型，以检测可能会降低模型性能的偏差（如数据偏差）并自动执行补救措施。

AWS Lambda 工具可支持事件驱动型架构，并将 ML 流程的多个阶段（从数据摄取到做出预测）连接在一起。

参考架构

以下参考架构 (图 4) 介绍了如何使用 Amazon SageMaker、Amazon Kinesis Data Streams、Amazon S3 和 AWS Lambda 来自动执行端到端 ML 流程，该参考架构适用于使用 *Amazon SageMaker 和 Data Lake on AWS* 解决方案的*预测数据科学*。

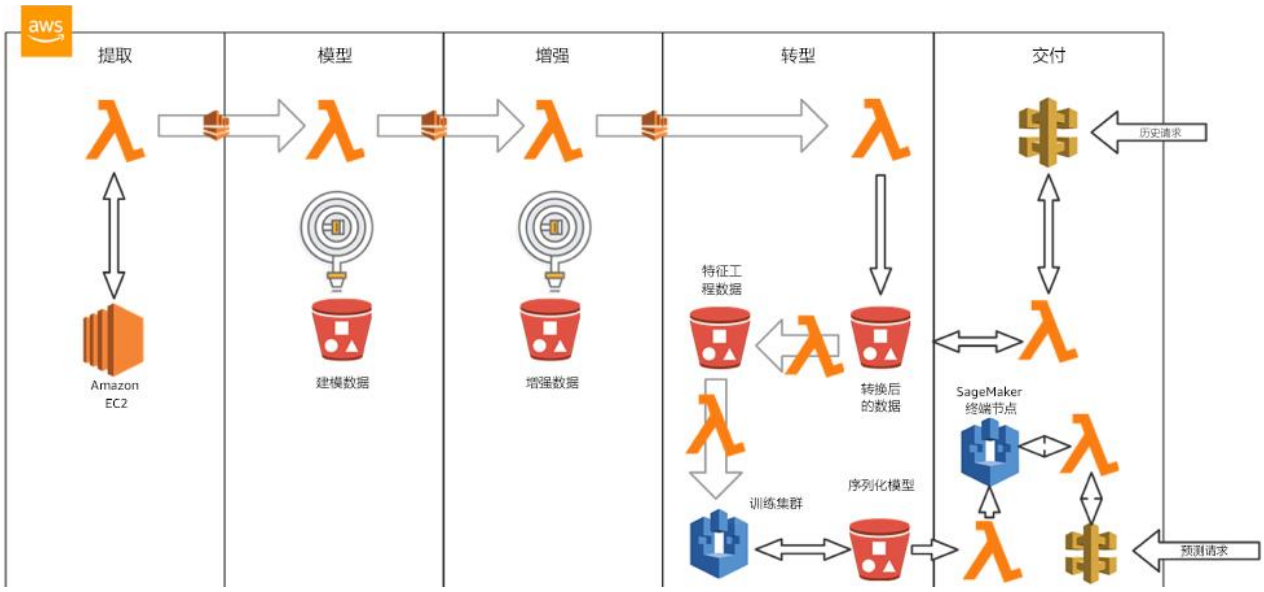

图 4 – 使用 Amazon SageMaker 和 Data Lake on AWS 的预测数据科学

此参考架构包括以下高级元素：

- Amazon S3 用作数据湖，存储原始、建模、增强和转换的数据。
- Amazon Kinesis Data Streams 可在提取、建模、增强和转换等各个阶段实时处理新数据。
- 将数据转换代码托管在 AWS Lambda 上，以准备原始数据（供使用和 ML 模型训练）并转换数据输入和输出。

- AWS Lambda 可以根据计划或者通过数据湖中的数据变化触发，来自动执行 Amazon SageMaker API 调用为新模型构建、管理和创建 REST 终端节点。

此架构可使用客户数据自动、连续训练 ML 模型，并对其进行改进，而且摆脱了无差别的繁重基础设施管理工作。

数据转换代码托管在 AWS Lambda 上。也可以在 Amazon SageMaker 笔记本实例上执行数据转换代码。但是，这些选项并非在所有情况下都是正确选择，尤其对于大规模数据转换场景。

利用托管 ETL 服务进行数据处理

数据处理活动（如清理、发现和大规模特征工程）非常适合使用 Apache Spark（为数据发现提供 SQL 支持）等工具以及其他有用的实用程序进行处理。在 AWS 上，Amazon EMR 不仅简化了 Spark 集群的管理，而且支持弹性扩展等功能，同时通过 Spot 实例定价最大程度地降低成本。

Amazon SageMaker Notebooks 支持连接到外部 Amazon EMR 集群，从而能够使用 Apache Spark 在可弹性扩展的集群上进行数据处理。之后，可以使用 Amazon SageMaker 训练和部署 API 训练和部署模型。

例如，假设这样一个业务使用案例：深入了解消费者行为，之后对消费者进行有针对性的营销。Amazon Pinpoint 是一项托管服务，可以通过多种交互渠道（如电子邮件、文本和 SMS）向消费者发送有针对性的消息。以下是一些有针对性的营销活动示例：促销提醒和客户维系活动，以及交易性消息，如订单确认和密码重置消息。但是，确定要向其发送消息的正确客户或客户群是一个关键组成部分。您可以使用 ML 根据历史消费者购买模式预测未来的购买行为。然后，可以使用预测的购买行为，通过 Amazon Pinpoint 推出有针对性的活动。

参考架构

此参考架构展示了如何在 ML 的不同阶段使用 Amazon EMR、Apache Spark 和 Amazon SageMaker，以及如何使用 Amazon Pinpoint 发送有针对性的营销信息。

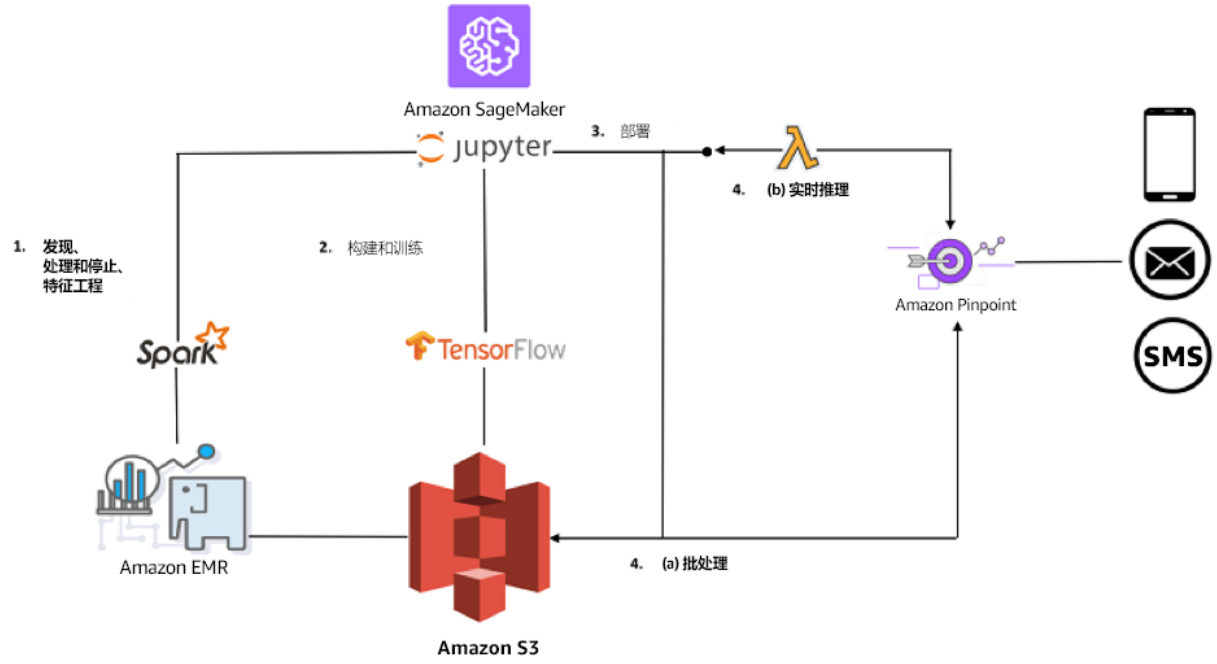

图 5 – Amazon SageMaker 上的 ML 助力 Amazon Pinpoint 推出营销活动

此参考架构包括以下高级元素：

- 使用 Amazon S3 作为数据湖来存储大量数据。
- 配置 Amazon SageMaker 笔记本以针对外部 Amazon EMR 集群运行。数据清理、处理、发现和特征工程都在 EMR 集群上使用 Apache Spark 完成。转换后的数据存储在 Amazon S3 中。
- 使用 Amazon SageMaker 基于转换后的数据和分布式训练功能训练自定义模型。
- 使用 Amazon SageMaker 为经过训练的模型创建 Auto Scaling API 终端节点。
- 使用 API 终端节点进行批处理和实时推理。
- 批量处理预测，并在数据湖中进行编目。然后，市场营销团队可以将数据导入到 Amazon Pinpoint，推出营销活动。

边缘和多个平台上的机器学习

训练 ML 模型需要在云中具备强大的计算基础设施。然而，对这些模型进行推理需要的计算能力通常要少得多。在某些情况下，例如使用边缘设备时，即使与云的连接受限或没有连接，也需要进行推理。挖掘领域就是这类使用案例的一个示例。要确保边缘设备能够快速响应本地事件，关键的一点是能够以低延迟获得推理结果。

AWS IoT Greengrass 实现了在边缘设备上进行机器学习推理。使用 AWS IoT Greengrass，可以使用在云中创建、训练和优化的模型，轻松在设备本地执行 ML 推理。使用 Amazon SageMaker、AWS Deep Learning AMI 或 AWS Deep Learning Containers 构建并持久存储在 Amazon S3 中的 ML 模型部署在边缘设备上。

图 6 展示了 AWS IoT Greengrass 与 AWS 云中 ML 模型训练之间的交互。

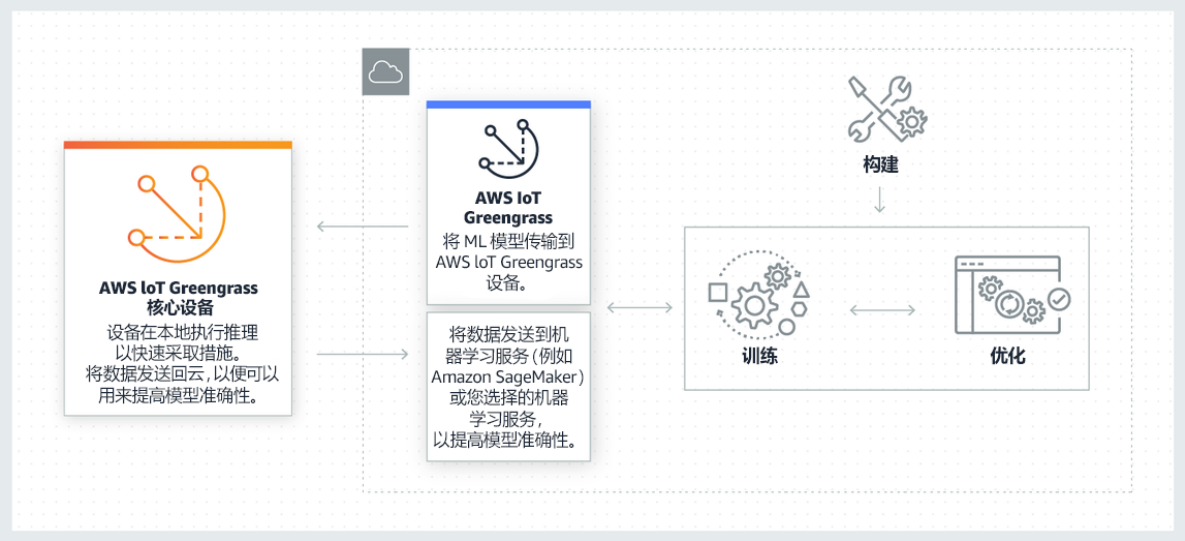

图 6 – AWS IoT Greengrass 和云中的 ML 模型

在运行 AWS IoT Greengrass 的互联设备上本地执行推理可以缩短延迟并降低成本。您可以直接在设备上运行推理，而不必将所有设备数据发送到云，执行 ML 推理并进行预测。当在这些边缘设备上进行预测时，您可以捕获并分析结果以检测异常值。然后，分析后的数据可发送回云中的 Amazon SageMaker，可以在其中对数据进行重新分类和标记以改进 ML 模型。

您可以使用在云中构建、训练和优化的 ML 模型，并在设备上本地运行推理。例如，可以在 Amazon SageMaker 中构建预测性模型以进行场景检测分析，进行优化以在任何相机中运行，然

后部署以预测可疑活动并发送警报。从在 AWS IoT Greengrass 上运行的推理收集到的数据可以发送回 Amazon SageMaker，然后可以在其中对数据进行标记并用于持续改进 ML 模型的质量。

参考架构

图 7 中展示了[在边缘设备上识别鸟类]使用案例的参考架构。在该架构中，在 Amazon SageMaker 上训练对象检测模型，然后部署到边缘设备上。自定义对象检测已经成为许多行业和使用案例的重要推动力，例如从核磁共振成像中发现肿瘤、发现病变作物和监控铁路站台。在此使用案例使用的边缘设备是 AWS DeepLens，它是一台实现了深度学习的摄像机。

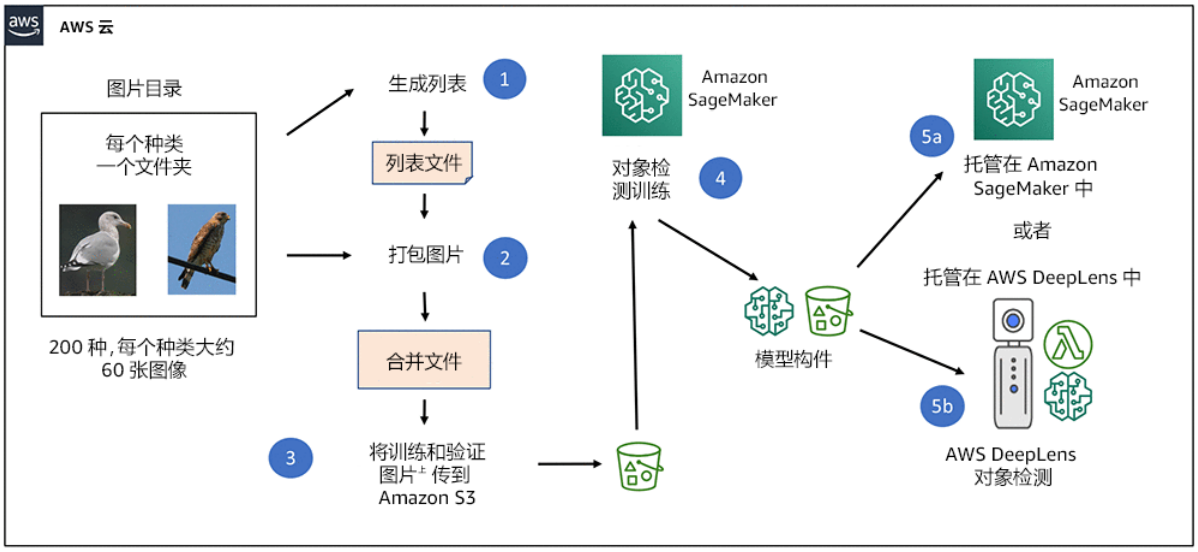

图 7 – 在边缘架构上识别鸟类

此参考架构包含以下元素：

- 收集、了解并准备鸟类图像数据集
- 使用 Amazon SageMaker 内置算法训练对象检测模型
- 使用 Amazon SageMaker 终端节点托管模型
- 将模型部署到 AWS DeepLens 边缘设备：
    - 在部署到 AWS DeepLens 之前转换模型构件

    - 通过 AWS DeepLens 上的 AWS Lambda 函数优化模型
- 在 AWS DeepLens 上执行模型推理和鸟类识别

AWS DeepLens 是在上述架构中使用的边缘设备之一。尽管可以在边缘和云中将 ML 模型部署到多个硬件平台（如 Intel 或 NVIDIA），但这并不总是切实可行，因为 ML 模型与用于对其进行训练的框架（如 MXNet、Tensor 或 PyTorch）紧密耦合。如果要将 ML 模型部署到的平台并非您为之训练该模型的平台，您必须先优化模型。随着 ML 框架和平台数量的增加，针对更多平台优化模型所需的工作也会增加，并且可能会非常耗时。

Amazon SageMaker Neo 包含可解决此问题的两个组件：编译器和运行时。编译器将模型转换为一种有效的通用格式，然后通过一种紧凑的运行时在设备上执行该格式，这种运行时使用的资源不到通用框架通传统用量的百分之一。Amazon SageMaker Neo 运行时针对底层硬件进行了优化，并使用特定的指令集，可帮助加速 ML 推理。模型经过优化，内存占用空间不足十分之一，因此可以在资源受限的设备上运行，例如家庭监控摄像头和执行器。

模型部署方法

经过训练的 ML 模型应该以一种使用者可以轻松调用并从中获得预测的方式进行托管。ML 模型的使用者可以是组织的外部使用者，也可以是组织的内部使用者。ML 模型的使用者通常不了解 ML 流程，他们只想要一个简单的 API，可以实时或以批处理模式提供预测。

Amazon SageMaker 为模型部署提供模型托管服务，并提供一个 HTTPS 终端节点，可在其中使用 ML 模型进行推理。

使用 Amazon SageMaker 托管服务部署模型分为三个步骤完成：

1. 在 Amazon SageMaker 中创建模型。

   应用最佳实践，确保模型满足业务要求，然后再继续。

2. 为 HTTPS 终端节点创建终端节点配置。

指定生产变体中一个或多个模型的名称，并指定希望 Amazon SageMaker 启动以托管每个生产变体的 ML 计算实例。通过终端节点配置，您可以将多个模型附加到同一个终端节点，并采用不同的权重和实例配置（生产变体）。在终端节点的生命周期内，可以随时更新配置。

3. 创建 HTTPS 终端节点。

   Amazon SageMaker 启动 ML 计算实例并按照在终端节点配置详细信息中指定的方式部署一个或多个模型，同时提供 HTTPS 终端节点。然后，模型使用者可以使用该终端节点进行推理。

利用 Amazon SageMaker 模型终端节点配置*生产变体*功能，可以将多个 ML 模型托管在不同的基础设施上，每个模型可处理部分或全部推理请求。您可以利用生产变体最大程度地降低部署风险。

对于所有变体，都在模型终端节点响应中包含一个模型版本。当模型推理出现问题时，或者在需要模型可解释性的情况下，了解特定的模型版本可以帮助您追溯到变更的源头。

标准部署

在标准模型部署中，Amazon SageMaker 终端节点配置有单个生产变体。生产变体配置指定了用于托管模型的实例的类型和数量。所有推理流量都由终端节点上托管的单个模型进行处理。

以下是一个标准部署的生产变体配置示例。

```
ProductionVariants=[{
  'InstanceType':'ml.m4.xlarge',
  'InitialInstanceCount':1,
  'ModelName':model_name,
  'VariantName':'AllTraffic'
}])
```

蓝/绿部署

蓝/绿部署技术提供了两个相同的生产环境。当需要将模型的新版本部署到生产环境中时，可以使用这项技术。

如 图 8 所示，此技术需要两个相同的环境：

- 实时生产环境（蓝色），运行**版本 n**，
- 此环境完全相同的副本（绿色），运行**版本 n+1**。

当蓝色环境（版本 n）处理实时流量时，可以使用合成流量在绿色环境中测试下一个版本（版本 n+1）。测试应包括验证新模型是否同时满足技术指标和业务指标。如果在绿色环境中版本 n+1 的所有测试都成功，则实时流量将切换到绿色环境。然后在绿色环境中再次验证这些指标，这次使用的是实时流量。如果在此测试中发现任何问题，则将流量切换回蓝色环境。如果在一段时间内未发现问题，可以删除蓝色环境。

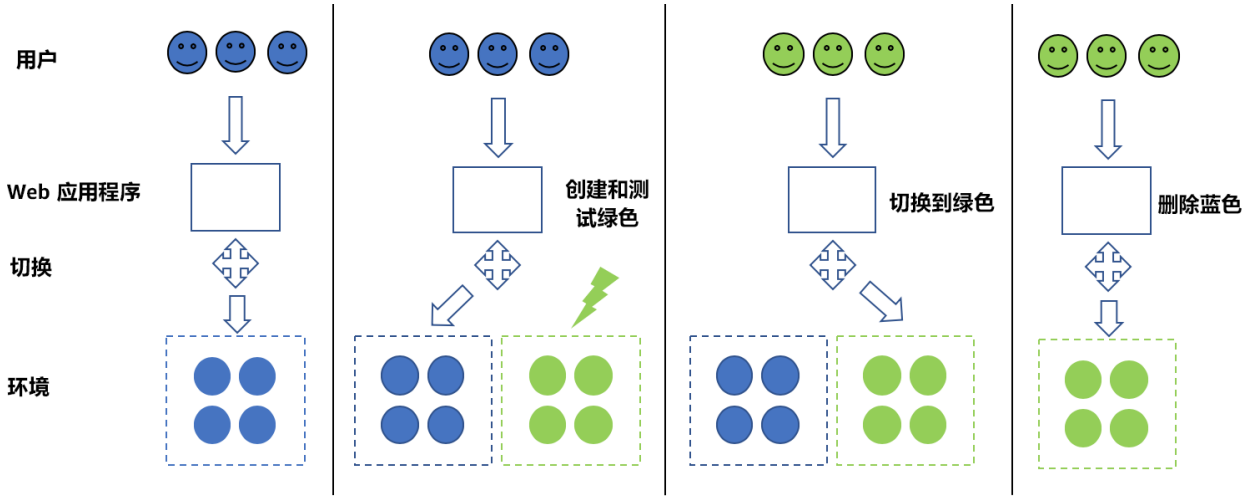

图 8 – 蓝/绿部署技术

在 Amazon SageMaker 上实施蓝/绿部署包括以下步骤：

1. 创建新的终端节点配置，且对现有实时模型和新模型使用相同的生产变体。
2. 使用新的终端节点配置更新现有的实时终端节点。
3. Amazon SageMaker 将为新的生产变体创建所需的基础设施并更新权重，而无需停机。
4. 通过 API 调用将流量切换到新模型。
5. 创建仅具有新生产变体的新终端节点配置，并将其应用于终端节点。

   Amazon SageMaker 将终止前一个生产变体的基础设施。

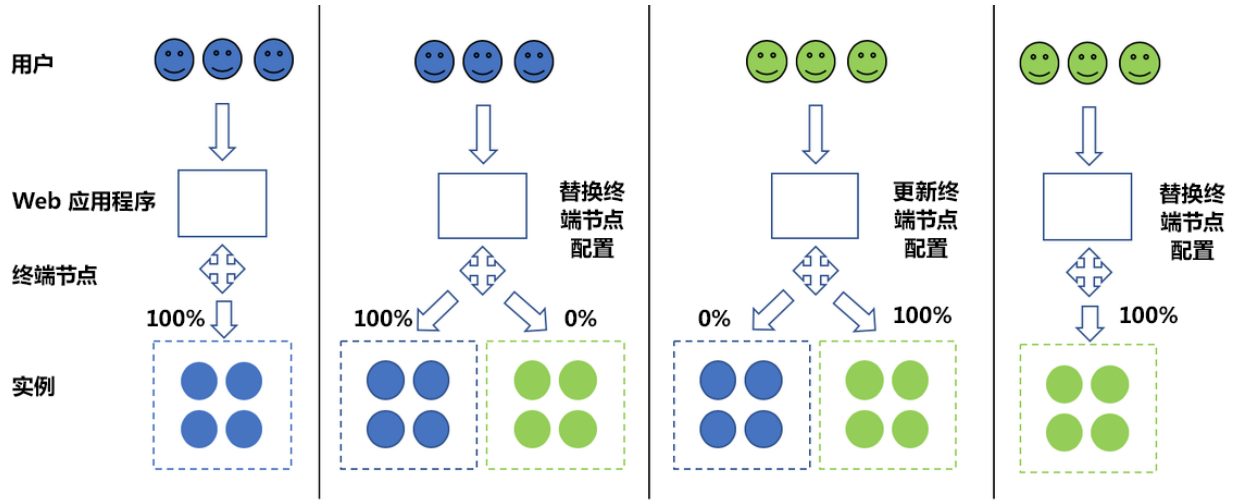

图 9 – 采用 Amazon SageMaker 生产变体的蓝/绿部署

Canary 部署

利用 Canary 部署，可以先将新版本部署给一小组用户，这样可以将验证风险降到最低。其他用户继续使用以前的版本，直到您对新版本满意为止。然后，可以逐步将新版本部署给所有用户。

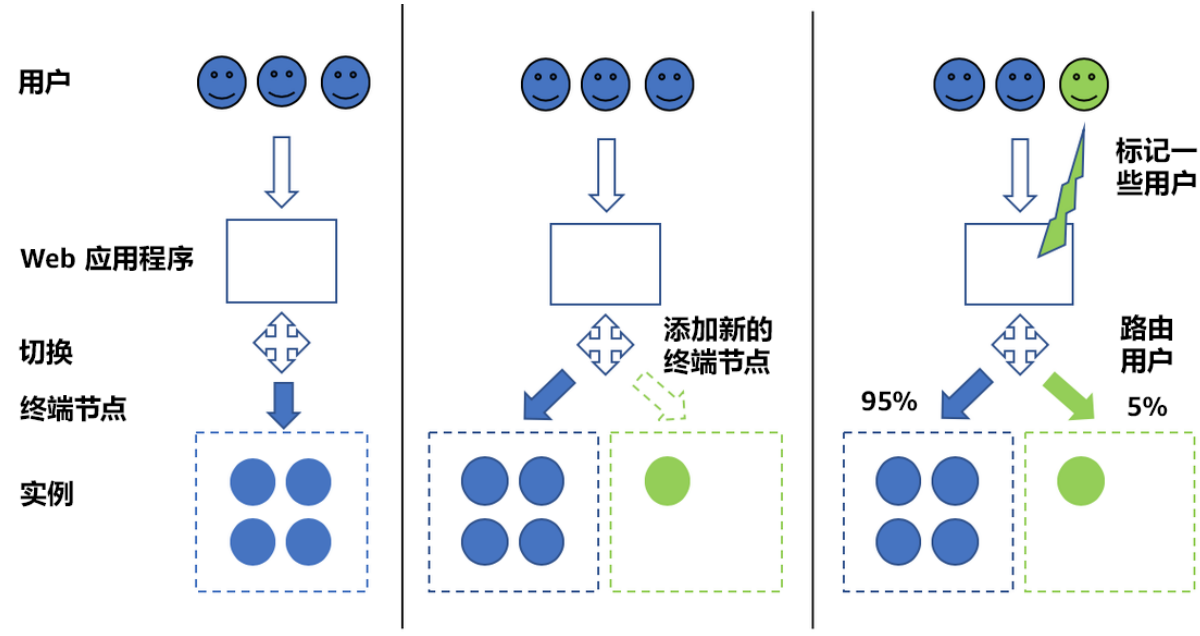

图 10 – 采用 Amazon SageMaker 生产变体的 Canary 部署：初始部署

确认新模型的性能符合预期之后，可以逐步将其部署给所有用户，并相应地扩展和缩减终端节点。

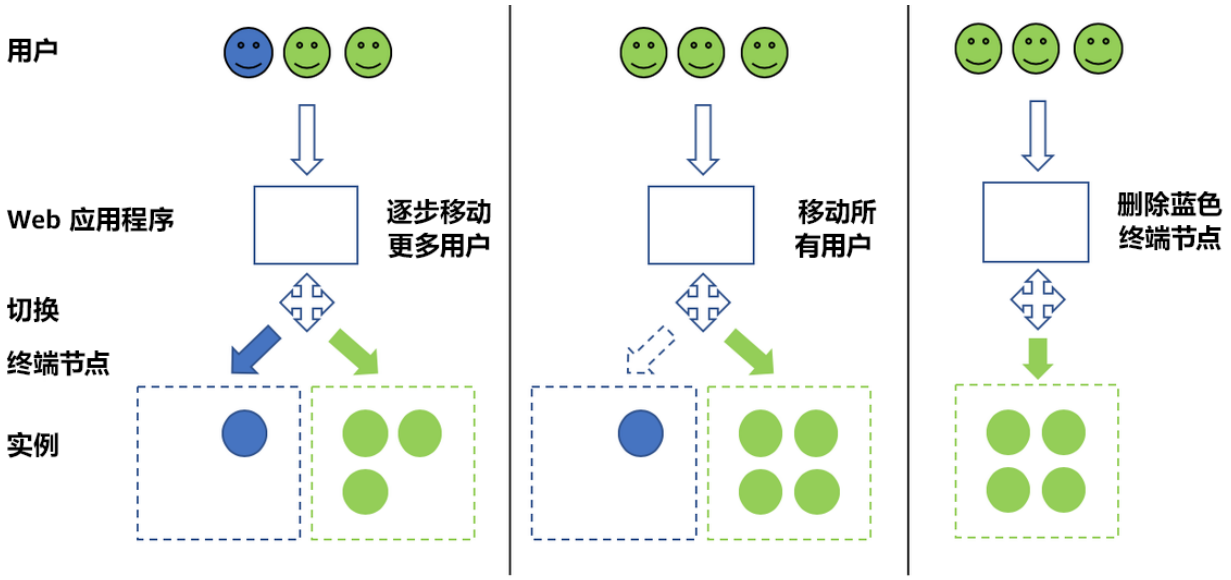

图 11 – 采用 Amazon SageMaker 生产变体的 Canary 部署：完成部署

A/B 测试

A/B 测试是一种技术，可用于比较相同特征的不同版本的性能，同时监控高级指标，例如点击率或转化率。在此背景下，这意味着对不同的用户使用不同的模型进行推理，然后分析结果。不同模型使用相同的算法（内置的 Amazon SageMaker 算法或您的自定义算法）进行构建，但使用两种不同的超参数设置。

A/B 测试类似于 Canary 测试，但具有更大的用户组和更长的时间范围，通常是数天甚至数周。对于这种类型的测试，Amazon SageMaker 终端节点配置使用两个生产变体：一个用于模型 A，另一个用于模型 B。首先，配置两个模型的设置，以均衡模型之间的流量 (50/50)，并确保两个模型具有相同的实例配置。在初始权重设置相同的情况下监控两个模型的性能后，可以逐步更改流量权重，使模型失去平衡（60/40、80/20 等），也可以一步更改权重，继续操作直到一个模型处理所有实时流量。

以下是一个 A/B 测试的生产变体配置示例。

```
ProductionVariants=[
  {
    'InstanceType':'ml.m4.xlarge',
    'InitialInstanceCount':1,
    'ModelName':'model_name_a',
    'VariantName':'Model-A',
    'InitialVariantWeight':1
  },
  {
    'InstanceType':'ml.m4.xlarge',
    'InitialInstanceCount':1,
    'ModelName':'model_name_b',
    'VariantName':'Model-B',
    'InitialVariantWeight':1
  }
])
```

架构完善的框架的支柱

以下每个支柱都非常重要，可帮助您实现架构完善的机器学习工作负载解决方案。对于每个支柱，我们只介绍特定于机器学习剖析的详细信息，包括定义、最佳实践、问题、注意事项，以及特定于 ML 工作负载的关键 AWS 服务。

在设计 ML 工作负载时，同时要确保使用 AWS 架构完善的框架白皮书中适用的最佳实践和问题。

卓越运营支柱

卓越运营支柱包括运行、监控和洞察系统以创造商业价值并持续改善支持流程和程序的能力。

设计原则

在云中，有许多原则可帮助您加强优化 ML 工作负载运营方面的能力。拥有实施这些工作负载的能力对于将 ML 工作负载快速推向市场至关重要。

AWS 卓越运营最佳实践旨在确保 ML 工作负载在云中高效运营。有关适用于所有 AWS 工作负载的标准卓越运营实践，请参阅卓越运营支柱：AWS 架构完善的框架白皮书。优化 ML 工作负载卓越运营的设计原则包括：

- **建立跨职能团队**：为确保 ML 工作负载有通向生产的途径，项目团队中需具有跨职能和领域专家。还需包含开发、部署和支持 ML 工作负载所需的所有利益相关者。
- **尽早确定端到端架构和运营模型**：在 ML 开发生命周期的早期，确定端到端架构和运营模型，以进行模型训练和托管。这样，可尽快确定开发、部署、管理和集成 ML 工作负载所需的架构和运营注意事项。
- **持续监控并衡量 ML 工作负载**：确定并定期收集与训练、托管和根据模型所做预测相关的关键指标。这样可确保您能够跨关键评估标准（如系统指标、模型延迟或数据偏差检测）持续监控已部署模型的运行状况。
- **建立模型重新训练策略**：已部署模型的性能和有效性可能会随时间而变化。确定指示模型版本的性能和有效性何时满足业务目标的指标，并创建阈值警报，指示需要对模型进行重新训练以触发这些活动。警报可以触发一些活动，例如使当前模型失效、恢复到旧模型版本、基于新的基本事实数据重新训练新模型、或者数据科学团队改进您的模型重新训练策略。
- **记录机器学习发现活动和发现结果**：数据科学发现和探究任务为机器学习模型的创建和演进提供了背景和见解。将这些活动记录在一个代码包中，以便可在源代码控制中进行管理和版本控制。
- **对机器学习输入和构件进行版本控制**：对输入和构件进行版本控制后，可以为 ML 工作负载以前的版本重新创建构件。对使用的输入进行版本控制以创建模型，除了模型构件之外，还包括训练数据和训练源代码。此外，还对使用的算法、特征工程源代码、托管配置、推理代码和数据以及后处理源代码进行版本控制。
- **自动化机器学习部署管道**：尽可能减少 ML 部署管道中的人为接触点，以确保使用定义模型从开发进入生产的方式的管道一致且反复部署 ML 模型。确定并实施满足使用案例要求以及解决业务问题的部署策略。如果需要，在您的管道中设立人工质量检验关，人工评估模型是否已准备好部署到目标环境。

定义

在云中实现卓越运营有三个领域的最佳实践：

- 准备
- 运营
- 演进

最佳实践

准备

要为卓越运营做好准备，您必须了解您的工作负载及其预期行为。要做好 ML 工作负载运营准备，您需要评估：

- 运营重点
- 运营设计
- 运营准备

> MLOPS 01：如何使团队做好准备以运营和支持机器学习工作负载？

从支持的角度来看，ML 工作负载通常各不相同，因为负责执行集成和部署 ML 模型的团队可能并不熟悉 ML 工作负载的各个运营方面。确保 ML 模型有效集成到生产环境并满足业务目标的最佳实践包括：确保团队之间能够跨团推协作，对负责支持和维护机器学习工作负载的资源进行培训，使之达到基本熟练水平。

对于数据科学家可能无法适应的 ML 工作负载，通常必须考虑一些运营要求，例如扩展或建模延迟的能力。相比之下，还需要捕获一些运营人员可能无法评估其度量的特定模型行为，例如模型的持续有效性。

在考虑如何让团队做好准备来集成和运营 ML 工作负载的方法时，关键实践包括：

- 在开发模型和 API 的团队与提供支持或负责审计 ML 工作负载的团队之间开展高级跨团队培训。

- 建立跨职能团队，以确保模型和 API 能够有效地集成到生产解决方案。这样可消除通常会阻止部署 ML 工作负载并将其与生产解决方案集成的障碍。

MLOPS 02：如何记录模型创建活动?

ML 模型开发生命周期与应用程序开发生命周期有很大不同，部分原因在于确定模型的最终版本前需要进行大量试验。为了更清晰地支持和使用模型版本，请记录模型创建流程，特别是与所做的假设、模型所需的数据预处理和后处理，以及将系统或应用程序与模型版本集成有关的流程。

将此流程记录成档有助于负责集成和支持模型的其他利益相关者能够更透明地了解模型。将此文档存储在安全、经过版本控制的位置（如源代码控制存储库），还可以捕获与模型创建和演进相关的知识产权。

在 AWS 中，Amazon SageMaker Notebooks 和 Amazon SageMaker Studio 提供托管笔记本环境，数据科学家可以在其中记录他们的开发流程和试验。这些笔记本可以与源代码控制系统集成，并成为针对每个已部署模型创建的文档的标准部分。

MLOPS 03：如何跟踪模型沿袭?

使用不同的算法以及为每种算法使用不同的超参数以迭代方式开发 ML 模型时，会进行许多模型训练试验和产生许多模型版本。跟踪这些模型并跟踪任何给定模型的沿袭不仅对审计和确保合规性非常重要，对针对模型性能下降问题执行根本原因分析同样重要。

此外，将模型沿袭与数据沿袭同步也非常重要，因为随着数据处理代码版本和模型版本的生成，训练每个模型版本的完整数据管道都需要记录下来，以解决调试模型错误和进行合规性审计。

在 AWS 中，您可以使用 Amazon SageMaker Experiments 整理和跟踪 ML 模型的迭代。Amazon SageMaker Experiments 会自动捕获每个模型的输入参数、配置和输出构件，并将其存储为试验。这样，无需手动跟踪或构建自定义跟踪解决方案来管理为每次模型开发迭代创建和使用的输入与输出构件的多种版本。利用这种方法，团队可以轻松地选择和部署模型，并从多次试验中获得最佳结果。

MLOPS 04：如何自动化 ML 工作负载的开发和部署管道?

创建一个运营架构，定义如何将 ML 工作负载作为其设计的一部分进行部署、更新和运营。结合基础设施即代码 (IaC) 和配置即代码 (CaC) 的常用实践，可确保部署一致性，而且能够可靠地跨环境重新创建资源。此外，确保建立一个自动化机制，以受控的方式协调 ML 工作负载在各个阶段和目标环境之间的移动，从而降低更新工作负载时的风险。

将持续集成和持续交付 (CI/CD) 实践整合到 ML 工作负载 (MLOps)，确保自动化包括可追溯性和质量检验关。例如，CI/CD 管道从源代码和构件版本控制开始，不仅支持标准变更管理活动，而且提高了调试活动的置信度。将源代码、数据和构件版本控制实践应用于 ML 工作负载，能够追溯到部署的版本，从而改进运营调试活动。此外，利用版本控制，还能够在更改失败后或新模型无法提供所需功能时回滚到特定的已知可正常运行的版本。

在整个 CI/CD 管道中实施日志记录和监控也为插入质量检验关、允许或拒绝部署到更高级环境奠定了基础。最佳实践包括标准的质量检验关，例如检查容器是否存在软件包漏洞，以及确保在管道中包括专门针对 ML 的质量检验关。这些质量检验关应该使用特定于业务使用案例的已确定指标评估模型，其中可包括评估精确率、召回率、F1 或准确率等指标。插入质量检验关有助于确保在发现的状况指示存在运营问题（如安全漏洞或模型性能或准确性指标下降）时，模型的新版本不会替换当前部署的模型。

在 AWS 中，Amazon Polly 等 AI 服务通过 API 终端节点提供。因此，在此方面没有独特的最佳实践，因为模型已经过训练且已部署。与 API 终端节点通信的代码和系统的相关开发和部署自动化工作应该遵循标准的 AWS 最佳实践。有些 AWS AI 服务（如 Amazon Personalize）会基于您提供的训练数据来训练模型。在这些服务中创建或更新模型时，请遵循本白皮书中所述的最佳实践来保护您的数据安全。

在 AWS 中构建和训练您自己的 ML 模型时，可以结合使用 AWS 服务和第三方集成来实现开发和部署管道自动化。确定正确的服务或工具来为模型部署创建自动化管道，需要确定部署策略、模型特征和模型训练策略。

每个 ML 工作负载都会因所使用的 AWS ML 服务不同而有所不同。但是，创建管道的一般准则包括将业务流程层（例如，AWS CodePipeline）与负责在管道中执行这些阶段的逻辑结合使用。由于基于函数的逻辑不需要管理服务器，所以运营开销较低，可以使用 AWS Lambda 来创建和执

行此逻辑。下图显示了在 AWS 上部署的参考管道。但是，此部署会因之前讨论的各项因素而有所不同。

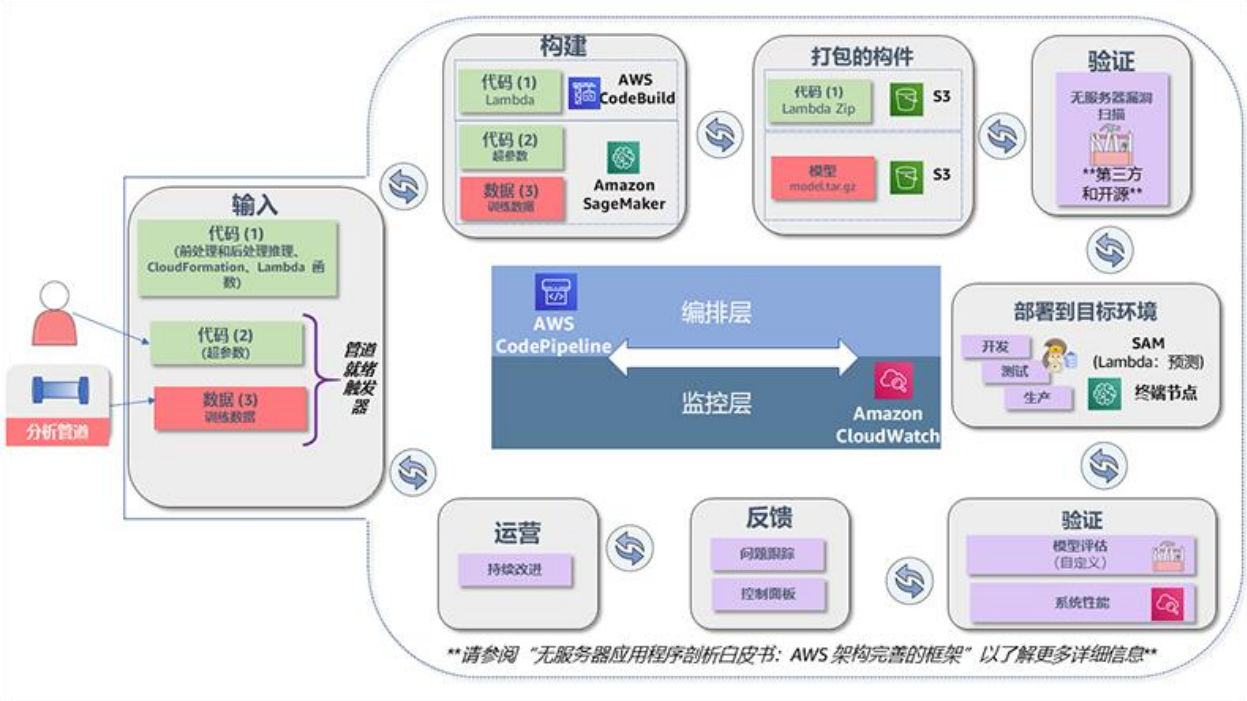

图 12 – AWS 上的机器学习参考 MLOps CI/CD 管道

运营

> MLOPS 05：如何监控和记录模型托管活动？

当托管预测的模型终端节点时，该终端节点应设置监控和提醒，以识别并响应任何潜在问题或改进机会。模型终端节点应包括对指标进行监控和提醒，这些指标衡量托管终端节点和监控终端节点响应运行状况的底层计算资源的运营状况。

在 AWS 中，管理终端节点计算资源的运营状况的标准做法应包括已经在 AWS 架构完善：卓越运营白皮书中定义的做法。Amazon SageMaker 可自动监控核心系统指标，它还具有为您托管的模型设置自动扩展能力的功能，以便能够根据需要动态调整支持终端节点的底层计算。此功能可确保您的终端节点可以动态支持需求，同时还能减少运营开销。

除了监控计算资源以支持自动扩展之外，Amazon SageMaker 还可以输出终端结点指标，用于对终端节点的使用情况和运营情况进行监控。Amazon SageMaker Model Monitor 提供了在生产环境中监控 ML 模型的能力，并且可以在出现数据质量问题时发出提醒。最佳实践包括创建一种使用服务（如 Amazon Elasticsearch）聚合和分析模型预测终端节点指标的机制，并内置对 Kibana 的支持以使用仪表板和可视化。此外，Amazon SageMaker Model Monitor 还能确保将托管指标追溯回带有版本的输入，从而能够对可能会影响当前运营性能的更改进行分析。

演进

> MLOPS 06：如何知道什么时候使用新数据或更新的数据对 ML 模型重新训练？

ML 工作负载最初可以提供高价值的预测，但随着时间的推移，同一模型预测的准确性可能会下降。这通常是由称为偏移的这一概念导致的，有很多因素可能会导致出现偏移，其中包括基础真实数据随着时间的推移发生了变化。由于模型预测结果会集成到业务决策中，因此，这可能会间接影响现有模型的性能。例如，一个预测与特定装运相关的风险的零售场景，其训练数据包括过去损坏的装运。随着企业开始使用该模型来制定业务决策，这会间接影响相关数据，因为损坏产品的实例将会减少。

经常需要使用新数据或更新的数据来对模型进行重新训练，以确保模型能够根据可用的最新数据进行有效学习和预测。为了能够将更多数据有效整合到 ML 模型中，必须实施一种机制来根据所定义的指标对现有模型性能进行分析，并在模型差异达到特定阈值时触发警报或重新训练事件，或者随着时间的推移根据新的已知数据主动对模型进行重新训练。

考虑集成更多数据的最佳实践包括：

- 定义能够指示模型性能和准确性的指标
- 确保建立一个机制来定期捕获这些指标，以便根据指标阈值进行分析并发出提醒。例如，可能需要建立一个系统来识别、捕获下游结果，并跟踪结果返回到特定模型预测，以便能够随着时间的推移计算指标（如错误率）。

- 评估是否适合对模型进行重新训练。确定是否有其他基本真实数据或是否可以获取这些数据，或者是否需要对其他数据进行标记。根据已知的工作负载特性决定初步重新训练策略，例如，定期使用新数据制定训练计划，将新数据作为触发重新训练的条件，或者根据指标阈值对重新训练进行评估。该策略应对更改的数量、重新训练的成本，以及将新模型投入生产环境的潜在价值进行权衡评估。根据制定的策略来建立自动化重新训练。

在 AWS 中，像 Amazon Translate 这样的 AI 服务会在新数据上自动完成训练，以便您能够利用 AWS 更新的模型不断地提高模型性能。

在 AWS 上使用 ML 服务构建和训练您自己的模型时，AWS 提供了多个功能来支持持续使用新数据对模型进行重新训练。将准备用于训练的数据存储在 Amazon S3 中。包括以下重新训练场景，并且应根据工作负载特性考虑这些场景：

- **模型偏移（指标驱动的重新训练）：**对于对变化比较敏感的 ML 工作负载（例如，当分发的数据与原始训练数据相差较大时，或者样本外数据数量增加时），建立一个自动化机制来根据所定义的指标或出现新的已准备数据时触发对模型的重新训练。在 AWS 中，一个识别数据偏移的机制包括利用 Amazon SageMaker Model Monitor 来检测什么时候所分发的数据发生了变化。可以通过 AWS CloudWatch 指标来检测偏移，这些指标可用于自动触发重新训练作业。
- **其他训练数据：**AWS 支持根据输入到 Amazon S3 存储桶中的新数据来自动触发重新训练的机制。启动受控执行模型重新训练的首选方法是建立一个 ML 管道，其中包括根据对源 Amazon S3 存储桶的更改来触发事件。要检测 S3 存储桶中是否存在新训练数据，将 CloudTrail 与 CloudWatch Events 结合使用可以触发 AWS Lambda 函数或 AWS Step Functions 工作流程来启动您的训练管道中的重新训练任务。下图显示了将 AWS CodePipeline 与 ML 服务结合使用的实践：

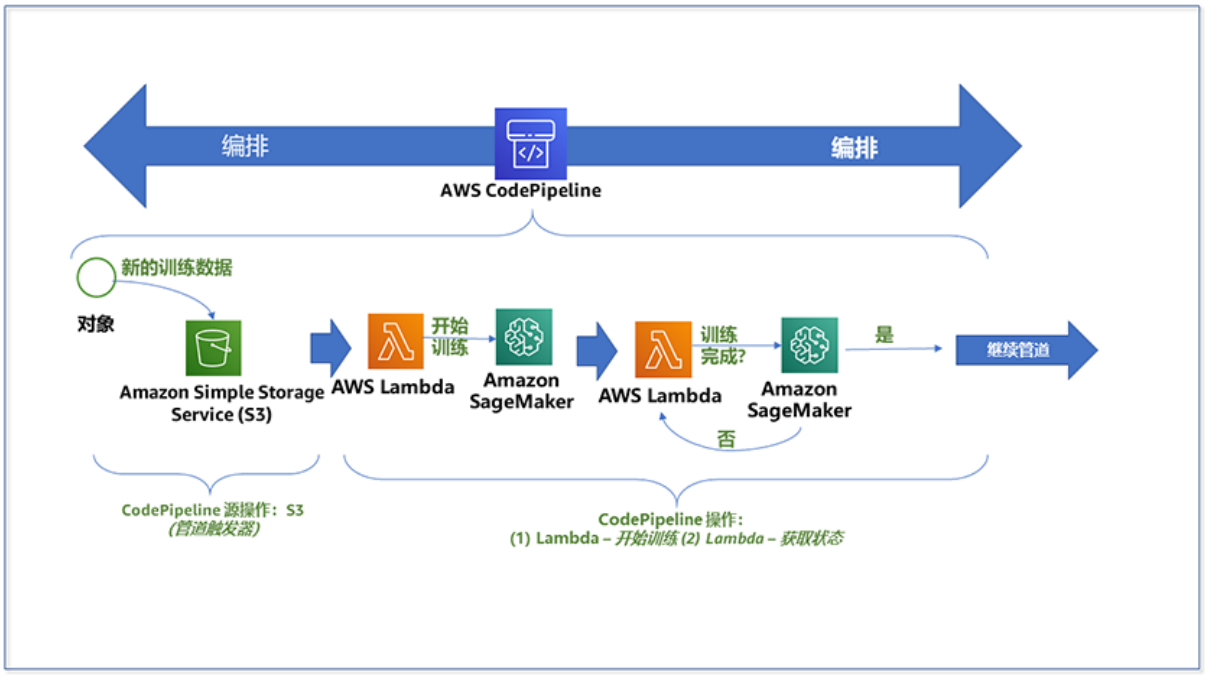

图 13 – 触发 ML 服务新训练数据的示例事件

或者，您也可以使用与 AWS 服务 API 集成的第三方部署业务流程工具（例如 Jenkins），在新数据可用时，自动完成模型重新训练。

定义将更多数据集成到模型中的策略时，请确保策略支持模型版本控制，以便能够按原始格式保留所有之前的训练数据，或者可以轻松再现以前版本的训练数据。这可确保如果模型构件被意外删除时，可以使用用于创建版本化构件的所有组件的组合版本，重新创建相同的模型构件。

MLOPS 07：如何整合模型开发迭代、模型训练和模型托管之间的学习？

要整合学习，关键是要建立一个持续反馈机制，以便能够共享和交流成功的开发试验、分析失败原因和运营活动。这有助于持续改进 ML 工作负载的未来迭代。

关于学习的重要考虑因素应包括在以下几方面对模型进行评估：

- **业务评估**：要验证模型相对于业务目标是否成功，您必须确保存在基本业务指标，以及持续收集和监控相关信息的机制。例如，如果您的业务目标是通过针对特定客户开展广告宣传活动来提高产品销售量，则需要建立基本业务指标和运营机制来持续衡量成功的关键绩效指标 (KPI)，例如，产品的销售量、目标客户和购买产品的客户。
- **模型评估**：要根据您确定的 ML 问题验证模型是否成功，您必须通过端到端管道捕获与模型性能相关的关键指标。这包括训练指标（例如训练或验证错误），以及托管模型的持续指标（例如预测准确度）。应根据使用案例和业务 KPI 来选择具体指标。
- **系统评估**：要对用于支持 ML 工作负载各个阶段的系统级资源进行验证，关键是要持续收集并监控系统级资源，例如，计算、内存和网络。ML 工作负载的要求在不同阶段会有所变化。例如，训练作业对内存要求较高，而推理作业则对计算要求较高。

  在 AWS 中，除了这一领域的标准实践之外，您还可以利用 SageMaker 笔记本实例来捕获提供模型开发生命周期的相关文档和详细阐述的数据科学探索活动。这之所以十分重要，不仅在于它使您能够成功为生产环境中的模型提供支持，而且还能在模型的发展演进过程中，查看并跟踪多个数据科学家和开发人员的活动。此外，通过提供对所收集的重要运营指标的集中可见性，还使团队能够持续对您一段时间内的运营情况进行审核并进行回顾性分析。

资源

请参阅以下资源，详细了解卓越运营的最佳实践。

文档和博客

- Build end-to-end machine learning workflows with Amazon SageMaker and Apache Airflow
- Automated and continuous deployment of Amazon SageMaker models with AWS Step Functions
- 使用 Step Functions 管理 Amazon SageMaker
- 使用 AWS CodePipeline 和 AWS Lambda 创建管道

白皮书

- 卓越运营支柱 – AWS 架构完善的框架
- 无服务器应用程序剖析 – AWS 架构完善的框架

安全性支柱

安全性支柱包括通过风险评估和缓解策略在提供业务价值的同时保护信息、系统和资产的能力。

设计原则

除了总体架构完善的安全设计原则外，另有针对 ML 安全性的特定设计原则：

- **限制对 ML 系统的访问权限：**在设计 ML 系统时，应考虑对该系统的访问级别。应对用于对 ML 模型进行训练的 ML 模型和数据集的访问权限进行限制，以避免损坏数据和模型。应确保推理终端节点的安全，以便只有获得授权的相关方才能根据 ML 模型进行推理。
- **确保数据管理：**用于 ML 的数据可以从多个源进行收集，并且需要可供组织中的多个不同团队使用。由于不仅数据科学开发活动需要使用生产数据，而且训练模型也需要使用这些数据，因此确保团队具有对高质量数据集的端到端访问权限就需要建立数据管理策略，以确保数据集的完整性、安全性和可用性。实施具有管理和访问控制功能的数据湖解决方案可确保开发人员和数据科学家能够对高质量的数据进行受控访问，以便用于探索活动和训练模型。还必须对数据进行保护，以防止外泄或更改。控制组织中不同团队可以对数据执行哪些操作，以及他们可以将数据发送到哪里。
- **强制数据沿袭：**由于在 ML 过程的不同阶段会使用来自各种来源的数据，因此，需要不断监控并跟踪数据的来源和转换。数据沿袭可实现可见性，并简化跟踪数据处理和机器学习错误以找到根本原因的过程。严格控制哪些人可以访问数据，以及他们可以对数据执行哪些操作。需要对数据进行预防性控制、审计和监控，以证明数据在其生命周期内的受控情况。

- **强制实施法规遵从性：**与 ML 系统有关的监管事宜包括隐私注意事项，例如，在 HIPAA 或 GDPR 中规定的隐私条款，ML 系统必须遵守在这些框架中规定的监管准则。还可能包括财务风险管理事宜，例如，美联储的 SR 11-7 准则。与算法保持静态的传统模型不同，利用 ML/AI 算法的模型会随着时间的推移而变化，因此，要确保符合监管机构的要求，就需要时刻小心谨慎。

定义

在云中实现安全性有五个最佳实践领域：

- Identity and Access Management
- 检测性控制
- 基础设施保护
- 数据保护
- 事件响应

最佳实践

Identity and Access Management

> MLSEC 01：如何控制对 ML 工作负载的访问权限？

通常情况下，会有多个团队参与构建 ML 工作负载，每个团队负责一个或多个 ML 阶段。 必须采用最低访问权限原则，对 ML 过程的各个阶段中使用的所有资源（包括数据、算法、超参数、训练过的模型构件和基础设施）的访问进行严格控制。 例如，一个负责功能设计的团队可能不负责训练或部署模型，因此，这个团队不应拥有访问训练或部署资源的权限。 同样，一个负责将模型部署到生产中的运营团队不应拥有访问或修改训练数据的权限。 有些工作负载可能需要团队成员在 ML 工作负载的多个阶段中承担重叠的责任，因此需要具有相应的权限来执行角色职责。

在 AWS 中，可通过 AWS IAM 来控制对各种资源和服务的访问权限。虽然身份用于身份验证、精细控制可以访问数据的人员和过程，但是，修改数据和算法，启动训练作业，以及部署模型是使用 IAM 用户、组、角色和策略来实施的。

限制只有既定合法客户才能访问已部署的模型。对于位于您的 AWS 环境中或有办法检索临时 IAM 凭证以访问您的环境的模型客户，请使用具有最低权限的 IAM 角色来调用所部署的模型终端节点。 对于您的环境之外的客户，请结合使用 API 网关和托管的模型终端节点，通过一个安全 API 来提供访问权限。

检测性控制

请参阅“AWS 架构完善的框架”白皮书，了解关于适用于 ML 的安全性的检测性控制方面的最佳实践。

基础设施保护

请参阅“AWS 架构完善的框架”白皮书，了解关于适用于 ML 的安全性的基础设施保护方面的最佳实践。

数据保护

> MLSEC 02：如何保护和监控对 ML 工作负载中使用的敏感数据的访问权限？

在 ML 过程中，数据在所有阶段中使用。在项目的早期阶段，在确定业务目标后，您需要评估各种数据源的可访问性和可用性，并与可用数据进行交互。通常需要已经存在或事先构建一个集中数据湖，然后才能开始项目的 ML 部分。对数据湖中的静态数据，以及在您的 ML 过程的不同阶段中移动的动态数据进行安全保护。您组织中的所有团队都不需要访问所有数据。 对数据进行分类，为不同部分的数据实施基于最低权限的精细访问控制，并持续监控对这些数据的访问情况。

在 AWS 中，使用 AWS Lake Formation 在 Amazon S3 上实施了一个集中数据湖。保护和监控 Amazon S3 上的数据湖是通过结合使用各种服务和功能，来对动态和静态数据进行加密并监控访问权限来实现的，包括详细的 AWS IAM 策略、S3 存储桶策略、S3 访问日志、Amazon CloudWatch 和 AWS CloudTrail。 构建大数据存储解决方案（数据湖）以实现最大灵活性对使用这些不同功能来构建安全的数据湖进行了探讨。

除了通过 AWS IAM 实施访问控制之外，还可以使用 Amazon Macie 来保护 Amazon S3 中的数据并对这些数据进行分类。Amazon Macie 是一项完全托管的安全服务，该服务使用机器学习来自动发现和保护 AWS 中的敏感数据并为其分类。此服务可以识别个人可识别信息 (PII) 或知识产权之类的敏感数据，并让您了解此类数据的访问或移动方式。Amazon Macie 会持续监控数据访问活动的异常情况，并会在检测到未经授权的访问或意外数据泄露风险时生成详细的警报。

在数据从数据湖移动到计算实例时，无论是用于探索还是训练，都应确保也要对目标计算实例的访问进行严格控制。 再次强调，要对计算基础设施上的动态和静态数据进行加密。

在数据准备和功能设计阶段，有多个方案可实现在 AWS 上进行安全的数据探索。可以在由 Amazon SageMaker 托管的笔记本环境中或在 Amazon EMR 笔记本上探索数据。 您还可以使用托管服务（例如 Amazon Athena 和 AWS Glue）来探索数据，而无需将数据从 Amazon S3 上的数据湖中移出。 还可以结合使用这两种方法。 可以使用 Amazon SageMaker 笔记本实例上托管的 Jupyter 笔记本来对一小部分数据进行探索、可视化和特征工程，然后使用托管的 ETL 服务（例如，Amazon EMR 或 AWS Glue）来扩展特征工程。

使用 Amazon SageMaker 笔记本实例上托管的 Jupyter 笔记本在 Amazon VPC 中部署笔记本实例时，可以使用网络级控件来限制与托管笔记本的通信。此外，还可以在 VPC 流日志中捕获网络与笔记本实例之间的调入和调出，以实现网络级的更多可见性和控制。通过在 VPC 中部署笔记本，您还能查询数据源并从 VPC 内访问系统，例如，Amazon RDS 上的关系数据库或 Amazon Redshift 数据仓库。使用 IAM，您可以进一步限制访问基于 Web 的笔记本实例 UI，以便只允许从 VPC 内访问这些 UI。

要从 VPC 中的笔记本实例与存储在 Amazon S3 中的数据湖中的数据进行通信，请使用 VPC 接口终端节点连接。这可以确保您的笔记本实例和 Amazon S3 之间的通信完全在 AWS 网络中安全地进行。通过使用 AWS KMS 托管的密钥对连接到 Amazon SageMaker 笔记本实例的 EBS 卷进行加密，从而对您笔记本实例上的静态数据进行加密。

Jupyter 笔记本服务器提供对底层操作系统的基于 Web 的访问权限，从而使开发人员和数据科学家能够安装更多软件包或 Jupyter 内核来对环境进行自定义。默认情况下，用户有权拥有本地根权限，从而使他们能够完全控制底层 EC2 实例。可以将此访问权限限制为只允许删除用户拥有根权限，但仍给予他们对其本地用户环境的完全控制权限。

除了限制对根权限的访问权限之外，还可以使用生命周期配置来管理 Jupyter 笔记本实例。生命周期配置是一些 shell 脚本，第一次创建笔记本实例或开始执行笔记本实例时，这些脚本以根用户身份运行。通过这些脚本，您可以安装自定义工具、程序包或进行监控。可以更改生命周期配置并在多个笔记本实例中重复使用，以便您能够在更改一次之后，通过重新启动托管的笔记本实例将新的配置应用于这些实例。这样，IT 团队、运营团队和安全团队就能在为开发人员和数据科学家提供需求支持时，获得他们所需要的控制权。

对模型进行训练时，通常需要较高的计算能力，这是单个笔记本实例所无法提供的。在 AWS 中，您可以使用 Amazon SageMaker 对模型训练一组训练实例。Amazon SageMaker 可预置用于执行您的训练作业的底层基础设施，并对数据运行算法以生成经过训练的模型。

通过在 VPC 中启动一组训练实例，您可以对训练实例应用网络级控制，并授予通过 VPC 终端节点访问 AWS 服务（包括 Amazon S3 和 AWS ECR）的权限。可以使用安全组来限制训练作业对未托管在 VPC 中 AWS 服务上的数据源的访问权限。还可以使用代理服务器和安全组来控制 VPC 之外的网络访问权限。使用 KMS 托管的加密密钥来加密训练节点的 EBS 卷上的数据，以便在训练期间为敏感数据提供进一步保护。可以使用 Amazon SageMaker 控制平面 VPC 终端节点来实现 VPC 和 Amazon SageMaker 控制平面之间的专用通信，以便对训练作业进行管理和监控。

对模型训练一组实例时，还要对此过程中由算法交换的信息进行管理和监控。 作为分布式训练作业的一部分，这种情况很常见，因为像 TensorFlow 这样的框架需要共享诸如系数这样的信息。这些不是您的训练数据，而是算法之间需要保持同步的信息。默认情况下，不会始终对这些数据进行加密。作为分布式训练作业的一部分，请将 Amazon SageMaker 配置为对您的训练作业的节点间通信进行加密。然后，在这些节点间传输的数据会被动态加密。

除了保护托管的 Jupyter 笔记本环境和训练群集安全之外，还务必要保护 ML 算法实施的安全。Amazon SageMaker 使用容器技术来训练和托管算法与模型。这就使 Amazon SageMaker 和其他 ML 合作伙伴能够将算法和模型打包为容器，然后您可以在您的 ML 项目中使用这些容器。此外，您还可以打包任何技术、语言或框架，以便与 Amazon SageMaker 结合使用。创建您自己的容器时，将这些容器发布到 AWS Elastic Container Repository (ECR) 上托管的专用容器注册表，并使用 KMS 管理的密钥对 AWS ECR 上托管的静态容器进行加密。

在训练期间，Amazon SageMaker 会从 AWS ECR 检索您指定的容器，并准备好容器以便在训练实例上执行。对于较小的数据集，Amazon SageMaker 支持“文件”训练模式，这种模式可将训练数据从 S3 存储桶下载到连接到训练实例的 EBS 卷。这种模式允许算法从本地文件系统读取其训练数据，而无需直接与 Amazon S3 进行集成。 通过使用容器和从 Amazon S3 复制对象，Amazon SageMaker 可实现在训练和托管期间对算法和模型的网络进行隔离。

但是，如果您的训练数据集较大，在开始训练作业之前将数据集复制到本地文件系统效率较低。 对于这种情况，请使用 Amazon SageMaker 的“管道”模式，这种模式可将数据直接从

Amazon S3 流式传输到训练实例。这意味着您的训练作业能够更早启动，更快完成，并且需要较少的磁盘空间，从而降低了在 Amazon SageMaker 上对 ML 模型进行训练的总成本。

在训练期间由 Amazon SageMaker 生成的日志记录在 AWS CloudWatch Logs 中。使用 AWS KMS 托管的加密密钥来加密 AWS CloudWatch Logs 引入的日志数据。

不仅保护训练数据十分重要，而且保护用于推理的生产/实时数据也同样重要。例如，对 AWS AI 服务或 Amazon 上托管的模型终端节点的推理 API 调用。 应对针对这些 API 调用的 HTTPS 请求进行签名，以便能够对请求者标识进行验证，并在传输中对请求负载数据进行保护，以避免潜在的重放攻击。 当您使用 AWS 命令行界面 (AWS CLI) 或其中一个 AWS 开发工具包进行 API 调用时，这些工具会自动使用您在配置它们时指定的访问密钥为您对这些请求进行签名。但是，如果您通过编写自定义代码来将 HTTPS 请求发送到 AWS，则需要实施此功能来对请求进行签名。

此外，AWS AI 服务（例如 Amazon Translate 和 Amazon Comprehend）的预置可将您的数据用于持续开发和改进 AWS，并关联 ML 和 AI 技术。您可以通过联系 AWS Support 选择不将您的数据用于这些用途。您收到您的账户已退出此用途的确认并按照所提供的任何说明进行操作之后，您的内容将不再存储，也不会再用于开发或改进 AWS AI 服务或任何 Amazon ML/AI 技术。

MLSEC 03：如何保护经过训练的 ML 模型？

除了保护用于训练 ML 模型的数据之外，还要保护对训练过程生成的模型构件的访问。托管您的模型，以便模型的使用者能够安全地根据该模型进行推理。 ML 模型的使用者可以是内部或外部应用程序或用户，通常会通过单个终端节点或可以提供预测的 API 与该模型进行集成。

在 AWS 中，在训练阶段最后生成的 ML 模型通常存储在 Amazon S3 中。使用专用 VPC 终端节点将 VPC 内经过训练的模型上传到 Amazon S3。这将确保模型在 AWS 网络中安全地传输到 Amazon S3。使用 Amazon SageMaker 对模型进行训练后，此服务会对模型构件以及其他动态和静态系统构件进行加密。

Amazon SageMaker 会在一组推理计算节点上部署并托管经过训练的模型，并提供终端节点 (HTTPS URL) 来执行推理。Amazon SageMaker 托管的终端节点既支持实时推理，又支持批量转换预测。在这两种情况下，托管的终端节点能够实现相同的基于 VPC 的网络保护，对托管模型的容器进行网络隔离，以及对推理节点的 EBS 卷进行加密。

Amazon SageMaker 托管的终端节点使用 IAM 向您的模型和调用提供额外的安全保护。这使您能够控制哪些 IAM 用户、IAM 角色、源 VPC 或 IP 能够对您的模型执行推理。此外，您还可以使用 AWS PrivateLink 将您的模型以服务的形式安全地分享给其他使用者。

就像在训练过程中捕获的日志那样，Amazon SageMaker 会将模型推理活动记录到 AWS CloudWatch Logs。同样，还要确保使用 KMS 管理的加密密钥对 AWS CloudWatch Logs 引入的日志进行加密。这可以为您提供模型在推理过程中的活动日志，使您能够提供所需的详细信息来满足安全和审计要求。

ML 模型的使用者通常会通过托管模型的环境外部的应用程序来对模型进行预测，例如，可能会通过一个 Web 应用程序来对面向互联网的终端节点进行推理。下图显示了一个无服务器架构，用于访问托管在 Amazon SageMaker 上的模型。在此架构中，最终用户直接访问 API Gateway，而 AWS Lambda 和 Amazon SageMaker 模型终端节点在一个受保护的专用网络中运营。

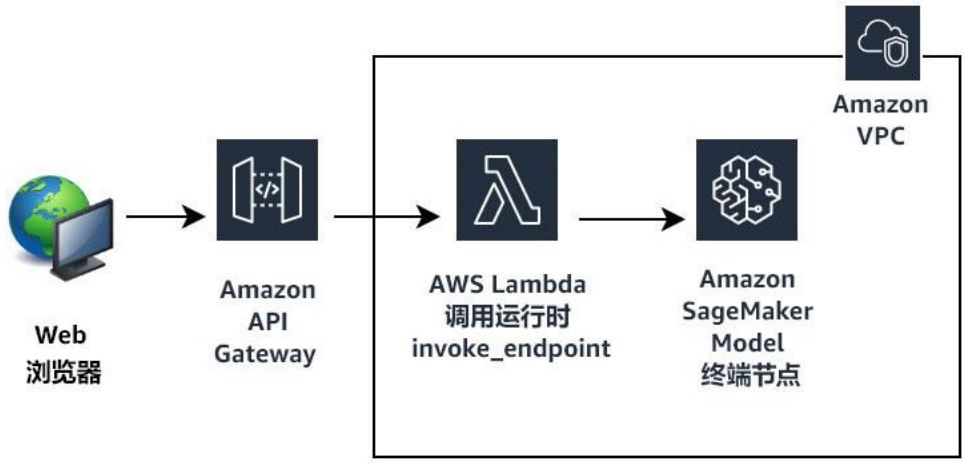

图 14 – 用于推理的无服务器框架。

此架构中的高级别步骤包括：

1. 使用者应用程序使用请求参数值调用 API Gateway API。
2. API Gateway 将参数值传递到 Lambda 函数。Lambda 函数解析此值，然后将其发送到 Amazon SageMaker 模型终端节点。

3. 此模型将执行预测并将预测值返回到 AWS Lambda。Lambda 函数会对返回的值进行解析，并将其发送回 API Gateway。
4. API Gateway 会使用推理值来响应客户端。

有关此架构服务的完整使用案例，请参阅 Call an Amazon SageMaker model endpoint using Amazon API Gateway and AWS Lambda。

事件响应

请参阅“AWS 架构完善的框架”白皮书，了解关于适用于 ML 的安全性事件响应方面的最佳实践。

关键 AWS 服务

AWS 上用于数据安全和监控的关键 AWS 服务包括：

- AWS IAM
- Amazon Virtual Private Cloud (Amazon VPC) 和 VPC 终端节点
- Amazon SageMaker

资源

请参阅以下资源，了解关于 AWS 上安全性最佳实践的更多信息。

白皮书

- AWS 安全性最佳实践
- 构建大数据存储解决方案（数据湖）以实现最大灵活性

文档和博客

- OWASP 安全编码最佳实践
- Amazon SageMaker 的身份验证和访问控制
- Call an Amazon SageMaker model endpoint using Amazon API Gateway and AWS Lambda
- Build a serverless frontend for an Amazon SageMaker endpoint

可靠性支柱

可靠性支柱包含系统从基础设施中断或服务中断恢复、动态获取计算资源以满足需求以及减少中断（如错误配置或暂时性网络问题）的能力。

设计原则

在云中，有许多原则可帮助您增强系统可靠性。有关标准做法，请参阅可靠性支柱：AWS 架构完善的框架白皮书。还有其他一些原则旨在专门帮助您提高 ML 工作负载的可靠性：

- **通过自动化管理对模型输入的更改**：ML 工作负载对管理用于训练模型的数据更改有更多要求，以便能够在出现故障或人为错误时重新创建模型完全相同的版本。通过自动化管理版本和更改可提供可靠且一致的恢复方法。
- **一次训练可以在多个环境中进行部署**：在多个账户或环境中部署相同版本的 ML 模型时，应为模型训练应用曾应用于应用程序代码的相同构建实践。模型的特定版本只应训练一次，并且应利用输出模型构件在多个环境中进行部署，以避免对多个环境中的模型引入任何意外更改。

定义

就确保云中的可靠性而言，有三个方面的最佳实践：

- 基础
- 变更管理
- 故障管理

最佳实践

基础

对于属于这一子部分的 ML 工作负载，没有特定的基础性实践。应使用可靠性支柱：AWS 架构完善的框架白皮书中确定的实践来确保基础功能。

变更管理

MLREL 01：如何管理对机器学习模型和预测终端节点的更改？

对于 ML 工作负载，务必要创建一个机制来跟踪对模型和对预测终端节点的更改。如果新的模型没有按预期工作，这样可以加快故障诊断速度并恢复到之前的模型版本。模型的部署版本应该可以追溯到特定版本的模型构件，该构件在构件存储库中受到保护，并且对有限资源具有只读访问权限。将模型构件保留企业所定义的保留期限。

要减少开销和手动干预，应通过包含与企业要求的任何更改管理跟踪系统集成的管道，自动完成对模型或终端节点的更改。利用包含通过版本化管道输入和构件实现的跟踪功能，您可以对更改进行跟踪，并在更改失败后自动回滚。

要部署对模型的更改，建议您使用标准 A/B 测试策略，在这些测试策略中，流量的指定部分被定向到新模型，而其余部分的流量则定向到旧模型。在这种情况下，回滚包括将 DNS 更改定向回到旧版本。要有效地确定何时需要回滚或向前滚动，必须实施评估模型性能的指标，以提醒什么时候需要回滚或向前滚动操作。在设计回滚或向前滚动的架构时，务必要评估每个模型的以下方面：

- 将模型构件存储在哪里？
- 是否对模型构件进行版本化？
- 每个版本中包含哪些更改？
- 对于部署的终端节点，部署的是模型的哪个版本？

构建跟踪机制以追溯资源并自动部署模型，可提供可靠的回滚和恢复功能。此外，要确保模型能够可靠地部署到多个环境，应使用一次训练策略来减少部署过程中发生的任何意外变化。

在 AWS 上创建模型时，建议您使用现有服务功能，并实施可确保能够将 ML 模型还原回以前版本的标准。

对于 AWS 上的 AI 服务（例如，Amazon Transcribe），AWS 会对用于进行终端节点预测的已部署终端节点进行版本控制。AWS 负责对与托管终端节点作为服务有关的更改进行管理。

关于 AWS 上的 ML 服务和 ML 框架以及 AWS 上的接口提供更改管理跟踪功能，以及确保向前滚动和回滚功能，有一些常用标准。将模型构件作为版本化对象存储在 Amazon S3 中可确保模型的持久性和可访问性。确保用于模型训练和模型托管的容器镜像存储在持久且安全的镜像存储库中，例如，AWS Elastic Container Registry (ECR)。此外，还要通过使用基于 IAM 角色的访问来限制对模型构件的访问权限，以及对适用于资源的策略实施最低特权原则，来保护模型构件和容器镜像的完整性。将用于创建构件的所有配置以代码形式存储在一个托管的源控制系统中，例如，AWS CodeCommit。

此外，具有对构件的跟踪能力还使您能够向前滚动或回滚到特定版本。创建并维护一个模型构件版本历史清单，突出显示所部署的模型构件版本之间的变化。可以通过将更改清单数据存储在一个持久性存储中来完成此操作，最好是按照后面***故障管理***部分中所述，通过一个可控制端到端模型开发和部署的自动化部署管道来完成此操作。当可以对多个模型中的变化进行评估并快速做出响应时，通过生产变体使用 SageMaker 原生的 A/B 测试功能。

对于 ML 框架和 AWS 上的接口，提供了多项创建可重用设计标准的功能，以 AWS CloudFormation 模板和 AWS 开发人员工具的形式提高您的工作负载的自动化和可恢复性功能。按照可靠性支柱：AWS 架构完善的框架白皮书中的说明，创建一个支持回滚和向前滚动功能的部署策略，以便对多个模型中的更改进行评估并快速做出响应。

通过实施自动回滚和向前滚动功能，您可以从失败的更改、系统失败或模型性能下降中恢复。此功能要求制定一个明确定义的版本控制策略，并建立一个跟踪和还原机制，在检测到问题时能够将更改进行还原。确保对所有重要模型评估指标进行定义和收集。收集监控系统（例如，Amazon CloudWatch）中的指标，并设置警报，当模型版本未按预期运行时触发回滚事件。

MLREL 02：如何更改已在整个工作负载中协调的 ML 模型？

要确保所引入的 ML 模型更改对现有工作负载功能的干扰最小或无干扰，务必要在设计接口应用程序时考虑到从属系统和功能将如何集成。灵活的应用程序和 API 设计有助于将更改与接口应用程序分离。此外，还务必要制定一个关于如何将更改通知给从属系统和/或应用程序并与它们进行协调的策略。按照已制定的更改管理策略来引入更改，将更改通知给受影响的团队，并实现对这些更改的跟踪功能。

采用与管理和控制应用程序级别变更相同的方法，来管理对新模型版本的部署。ML 模型的更改管理策略必须考虑到如何通知和部署更改，以避免服务中断以及模型性能下降和准确性降低。您的新模型版本部署策略还必须包含要在部署到目标环境之前和之后执行的验证活动。

要确保更改管理得到一致且正确的执行，最佳实践是使用遵守最低特权原则的访问控制在整个 CI/CD 管道中执行所有更改，以强制实施您的部署过程。通过将自动化与手动或自动质量检验关相结合来控制部署，可确保更改在部署之前使用从属系统进行有效地验证。

> MLREL 03：如何缩放托管预测模型的终端节点？

关键是实施允许自动缩放模型终端节点的功能。这可以确保您能够可靠地处理预测以满足不断变化的工作负载需求。要缩放您的终端节点，您必须对终端节点进行监控，以确定触发添加或删除支持当前需求的资源的阈值。收到触发缩放的信号后，必须实施相应解决方案来缩放支持该终端节点的后端资源。对终端节点执行负载测试，以便能够验证它们有效扩展和为预测提供可靠服务的能力。

在 AWS 中，终端节点的缩放和在这方面的职责依赖于所利用的 AI/ML 服务。对于包括 Amazon Comprehend、Amazon Polly 和 Amazon Translate 在内的 AWS AI 服务，终端节点由 AWS 自动管理和扩展。对于像 Amazon SageMaker 这样的 AWS ML 服务，可以在此服务中配置多个可用区中的自动缩放功能。要实现高可用性，请为所有生产变体配置跨多个可用区的自动水平缩放功能。配置完成后，务必要执行故障测试，以确保您的终端节点能够从故障中恢复并支持可用性要求。

对于 AWS ML 框架和接口，无论模型是托管在 EC2 实例上，还是使用 EC2 实例或 AWS Fargate 上托管的 ECS 或 EKS 上的容器，都要设置自动缩放的负载平衡功能。自动缩放功能还可以通过自动替换故障实例用于自行修复 EC2 实例。请参阅 AWS 架构完善的框架中的可靠性支柱，了解关于这方面的标准最佳实践。

故障管理

> MLREL 04：如何从故障或意外丢失经过训练的 ML 模型中恢复？

经过训练的 ML 模型是一个构件包，它必须能够在发生故障或丢失时恢复。资源故障或丢失可能是由于从系统故障到人为错误的多个事件导致的。对于 ML 模型，应考虑以下故障场景并与您的恢复目标进行比较，以确保部署合适的策略。如果由于人为错误或底层存储变得不可用而导致模型构件被意外删除，您可以轻松恢复或重新创建该构件吗？

保护模型构件不被意外删除的功能可通过以下方式实现：通过只允许使用构件所需的最低特权，为特权用户删除实施其他一些机制（例如 MFA），以及按照您制定的灾难恢复策略的要求存储构件的次要副本，确保对该模型构件进行安全保护。同时实施构件版本控制策略可以恢复特定版本的构件。其次，具备重新创建特定版本模型构件的能力可提供更多失败或丢失防护。对包括数据和训练代码在内的模型输入应用相同的保护机制和版本控制策略。

在 AWS 中，与模型故障和恢复能力相关的最佳实践因所使用的 AWS 服务而异。AWS AI 服务（例如 Amazon Polly）使用由 AWS 保护和管理的内置模型。AWS ML 服务（例如 Amazon SageMaker）可在 Amazon S3 中创建并存储模型构件。在这种情况下，您可以利用 AWS IAM 提供的访问控件来保护模型输入和构件以便保护资源。此外，您还可以将某个机制（例如 Amazon S3 版本控制）与对象标记结合使用，以便对模型构件进行版本控制和跟踪，从而能够在出现故障时进行恢复。

MLREL 05：如何恢复故障或意外丢失的托管资源的模型？

具备恢复 ML 工作负载中任何组件的能力，可确保解决方案能够经受住资源故障或丢失的考验。资源故障或丢失可能是由于从系统故障到人为错误的多个事件导致的。对于 ML 工作负载，应考虑以下故障场景并与您的恢复目标进行比较，以确保部署合适的策略。如果模型终端节点被意外删除，能够重新创建该终端节点以将终端节点恢复到特定版本吗？

在 AWS 中，与故障管理相关的最佳实践因所使用的 AWS 服务而异。由于 AWS AI 服务（例如 Amazon Polly）是由 AWS 托管、缩放和管理的，因此，您不负责对终端节点进行恢复。对于 AWS ML 服务以及 AWS ML 基础设施和框架，最佳实践是确保负责托管模型预测的终端节点能够完全恢复到特定版本或您的业务所确定的时间点。要具备恢复模型终端节点的能力，需要将用于创建该终端节点的所有组件和配置都包含在一个管理的版本控制策略中，才能在挂起任何不可用的组件的同时实现完全恢复。

例如，在 SageMaker 中重新创建一个终端节点需要在 SageMaker 中包含多个组件的相关版本，包括模型构件、容器镜像和终端节点配置。要重新创建特定的模型构件版本，您还需要知道用于创建该模型构件的训练数据和算法的版本控制策略。要确保您能够在发生故障时在管道中重新创建任何组件，还要对所有依赖资源进行版本控制。除了版本控制之外，还务必要确保所有版本化构件都包含在记录部署过程的清单中，并且使用“安全支柱”中规定的最低特权原则对所有版本化构件进行安全保护。

资源

请参阅以下资源，了解关于可靠性最佳实践的更多信息：

白皮书

- 可靠性支柱 – AWS 架构完善的框架

性能效率支柱

性能效率支柱专注于有效利用计算资源来满足需求的能力，以及如何在需求发生变化和技术不断演进的情况下保持这种效率。

设计原则

在云中，有许多原则可以帮助您提高系统的性能效率。有关标准做法，请参阅性能效率支柱：AWS 架构完善的框架白皮书。另外，还有一些原则旨在专门帮助您提高 ML 工作负载的性能效率：

- **优化 ML 工作负载的计算：**大部分 ML 工作负载的计算密集性程度较高，这是因为需要对大量数据和参数执行大量矢量乘法和矢量加法。尤其是在深度学习中，需要对芯片组进行缩放以提供更深的队列深度、更高的算术逻辑单元和寄存器计数，以允许大量并行处理。因此，GPU 是训练深度学习模型的首选处理器类型。我们将在下面的 MLPER 01 部分中详细讨论如何选择适当的计算资源。

- **为模型定义延迟和网络带宽性能要求：**某些 ML 应用程序可能需要近乎瞬时的推理结果才能满足业务要求。要提供尽可能最短的延迟，可能需要消除与最近的 API 终端节点之间成本高昂的往返路线。通过直接在设备本身上运行推理可以减少这种延迟。这称为“边缘的机器学习”。此类要求的一个常见使用案例是工厂中的预测性维护。这种低延迟形式和边缘近乎实时的推理可以在故障实际发生之前早期指示故障，从而可能缓解成本高昂的设备维修。
- **持续监控和衡量系统性能：**确定并定期收集与构建、训练、托管模型和对模型运行预测相关的重要指标的做法，可确保您能够持续监控根据各项重要评估标准的总体成功与否。要对用于支持各阶段 ML 工作负载的系统级资源进行验证，重要的是要持续收集并监控系统级资源，例如，计算、内存和网络。在前面两个设计原则中已经讨论过，在训练作业不同阶段中 ML 工作负载更改要求更多的是内存密集型，而推理作业要求更多的是计算密集型。

定义

在云中实现性能效率包括四个方面的最佳实践：

- 选择（计算、存储、数据库和网络）
- 审核
- 监控
- 权衡

采用数据驱动型方法来选择高性能架构。收集架构各方面的数据，从总体设计到资源类型的选择与配置都包括在内。通过周期性审查您的选择，您可以确保充分利用 AWS 服务持续演进所带来的优势。监控可以确保您随时发现与预期性能的任何偏差，并采取针对性措施。最后，您可以对您的架构作出权衡以便提高性能，例如使用压缩或缓存，或放宽一致性要求。

最佳实践

选择

> MLPER 01：如何为训练和托管模型选择最合适的实例类型？

典型的机器学习管道由一系列步骤组成。此过程从收集训练和测试数据开始，然后是进行特征工程和对所收集的数据进行转换。在初始数据准备阶段之后，对 ML 模型进行训练、评估和调整，然后进入部署、服务和监控模型整个生命周期性能的最终阶段。由于每个阶段的计算需求不同，因此，应对 ML 工作负载每个阶段的性能进行仔细考量。 例如，尽管您可能需要一组功能强大的 GPU 实例用于模型训练，但是在进行推理时，一组可以自动缩放的 CPU 实例也许就能满足您的性能要求。

数据大小、数据类型和所选择的算法可能会对哪个配置最有效产生巨大影响。重复对同一模型进行训练时，强烈建议对范围广泛的实例类型执行初步测试，以发现既性能好又成本低的配置。一般原则是，对于大部分深度学习目的，建议使用 GPU 实例，因为相对于 CPU 实例，在 GPU 实例上训练新模型速度更快。如果有多 GPU 实例，或者如果在多个 GPU 实例中使用分布式训练，可以子线性方式进行缩放。但是，请务必注意，在 GPU 上训练最有效的算法可能不一定需要 GPU 才能进行有效推理。

AWS 提供了一系列经过优化的实例类型，以适应机器学习 (ML) 不同使用案例的需求。实例类型包括 CPU、GPU、FPGA、内存、存储和网络容量的不同组合。此外，您可以通过 Amazon Elastic Inference 将 GPU 驱动的推理加速器附加到 Amazon EC2 或 Amazon SageMaker 实例，或使用由 AWS Inferentia（由 AWS 定制设计的高性能 ML 推理芯片）提供支持的 Amazon EC2 实例。这样能让您灵活地选择经过优化的合适资源组合，以满足您的 ML 使用案例需求，无论您是训练模型还是针对训练后的模型进行推理，皆可使用。每种实例类型都包含一个或多个实例大小，从而使您可以扩展资源以满足目标工作负载的要求。

最后，某些算法（例如 XGBoost）实现了针对 CPU 计算进行优化的开源算法，而在 AWS Deep Learning AMI (DLAMI) 上，某些框架（例如 Caffe）仅在 GPU 支持下工作，无法在 CPU 模式中运行。

无论您选择哪种实例进行训练和托管，都需要对实例或 Amazon SageMaker 终端节点进行负载测试，以确定实例或终端节点可以支持的峰值负载以及随着并发性增加而产生的请求延迟。

MLPER 02：如何在保持最佳性能的同时扩展 ML 工作负载？

在考虑如何扩展 ML 架构以增加需求和实现最佳性能时，区分通过托管服务体验部署模型与自行部署和管理 ML 模型的不同非常重要。

在 AWS 上，虽然 Amazon SageMaker 提供了托管 ML 体验，但您通常会在 EC2 实例上使用深度学习 AMI (DLAMI)，它提供 MXNet、TensorFlow、Caffe、Chainer、Theano、PyTorch 和 CNTK 框架，便于您自行管理模型。本节讨论了这两个使用案例，同时还简要讨论了使用 AWS AI 服务的第三个使用案例。

Amazon SageMaker 代表您管理生产计算基础设施，以执行运行状况检查、应用安全补丁以及执行其他日常维护，所有这些均通过内置的 Amazon CloudWatch 监控和日志记录功能来完成。Amazon SageMaker Model Monitor 持续监控生产环境中的 ML 模型、检测会不断降低模型性能的偏差（如数据偏移），并提醒您采取补救措施

此外，Amazon SageMaker 托管服务使用 Application Auto Scaling 自动扩展至应用程序所需的性能。通过使用 Application Auto Scaling，您可以自动调整推理容量，以低成本维持可预测的性能。此外，您可以通过修改终端节点配置来手动更改 EC2 实例的数量和类型，而不会导致停机。

对于深度学习工作负载，您可以选择使用 Amazon Elastic Inference (EI) 进行扩展，以增加吞吐量并减少针对深度学习模型的实时推理的延迟。Amazon Elastic Inference 使您可以将 GPU 驱动的推理加速附加到任何 Amazon EC2 实例。此功能也可用于 Amazon SageMaker 笔记本实例和终端节点，以更低的成本实现内置算法和深度学习环境提速。

深度学习神经网络非常适合利用多个处理器，在不同类型和数量的处理器之间无缝高效地分发工作负载。借助可通过云获得的大量按需资源，您可以部署几乎无限制的资源来处理任何规模的深度学习模型。通过利用分布式网络，云中的深度学习可让您更快地设计、开发和训练深度学习应用程序。

适用于 Ubuntu 和 Amazon Linux 的 AWS Deep Learning AMI 支持具有接近线性扩展效率的 TensorFlow 深度学习模型的分布式训练。AWS Deep Learning AMI 预先构建了 TensorFlow 的增

强版本，该增强版本与 Horovod 分布式训练框架的优化版本集成在一起。这种优化带来了高性能的实施，从而允许节点彼此直接通信，而无需使用 ring-allreduce 算法通过集中式节点和平均梯度进行通信。除了提到的 Horovod 分发之外，还有许多其他框架支持分布式训练，例如使用 Chainer 或 Keras。

使用 AWS AI 服务可以使您通过对预先训练的服务的 API 调用向应用程序添加智能，而不必开发和训练自己的模型。扩展使用 AWS AI 服务的架构涉及监控资源和速率限制，例如 API 请求速率。您可以在一般架构完善的框架的可靠性部分中找到有关如何管理服务限制的详细信息。

资源

请参阅以下资源，详细了解提升性能效率的最佳实践。

文档和博客

- Scalable multi-node training with TensorFlow
- Amazon Elastic Inference
- Load test and optimize an Amazon SageMaker endpoint using automatic scaling
- 选择 DLAMI 的实例类型

白皮书

- 性能支柱 – AWS 架构完善的框架

成本优化支柱

成本优化支柱包括系统在整个生命周期中不断完善和改进的过程。从最开始的概念验证的初始设计到生产工作负载的持续运营，您都可以采用本文中的实践来构建和运营具有成本意识的系统，从而在实现业务成果的同时尽可能降低成本，使您的企业能够最大限度地提高投资回报率。

设计原则

在云中，有许多原则可帮助您改善系统的成本优化。有关标准做法，请参阅 AWS 架构完善的框架白皮书。另外，还有一些原则旨在专门帮助您提高 ML 工作负载的成本优化：

- **使用托管服务降低拥有成本**：在 ML 工作负载的每个阶段采用适当的托管服务，以充分利用“仅按您的使用量付费”模型。例如，模型调优通常是一个计算和时间密集型过程。为避免产生任何不必要的费用，请使用托管服务来创建分布式训练集群、执行训练作业以调优模型、保留生成的模型，并在完成训练后自动停用集群。
- **使用小型数据集实验**：虽然 ML 工作负载受益于大型高质量的训练数据集，但首先从小型计算实例（或您的本地系统）上的较小数据集开始，以实现低成本的快速迭代。 实验期过后，进行扩展以使用分布式计算集群上可用的完整数据集进行训练。 在使用完整数据集进行训练时，请选择将数据流式传输到集群中，而不是将数据存储在集群节点上。
- **适当大小的训练和模型托管实例**：对于模型训练和托管，请进行实验以确定所需的最佳计算容量。建议您从小型实例开始，先横向扩展，然后再纵向扩展。还要测量训练和托管期间 CPU 与 GPU 需求之间的差异。 虽然某些 ML 模型需要高功率的 GPU 实例进行训练，但针对已部署模型的推理可能并不需要全功率的 GPU。
- **考虑基于消费模式的推理架构**：某些模型（例如电子商务欺诈检测）需要持续可用以进行实时预测，而其他模型（例如电子商务预测模型）可能只需要定期可用。在第一种情况下，采用 24 X 7 全天候托管模型的成本是合理的，但在第二种情况下，可以通过按需部署模型、执行预测然后停用模型来实现显著的成本节省。
- **定义总体 ROI 和机会成本**：权衡采用 ML 的成本与不依赖 ML 转换的机会成本。 专用资源，例如数据科学家时间或模型上市时间，可能是您最昂贵和最受约束的资源。 如果最具成本效益的硬件选择限制了实验和开发速度，则可能不会优化成本。

定义

云中的成本优化包括四个领域的最佳实践：

- 资源成本效益
- 供需匹配
- 支出认知
- 持续优化

与其他支柱一样，也需要评估权衡各种因素。例如，您想优化上市速度还是优化成本？ 在某些情况下，最好优化上市速度以便快速上市、交付新功能或仅仅只是为了按时完成任务，而不是优化预付成本。由于人们总是倾向于以过度补偿的方式来“以防万一”，而不是花时间进行基准测试，逐渐找出成本最优的部署，导致设计决策有时会过于仓促，缺乏对经验数据的参考。这通常会导致明显的过度预置和优化不足的部署。以下最佳实践介绍了一些技巧和战略性指导，以帮助您实现初始部署和持续成本优化。

最佳实践

资源成本效益

> MLCOST 01：如何优化数据标记成本？

构建 ML 模型需要开发人员和数据科学家准备用于训练 ML 模型的数据集。在开发人员选择他们的算法、构建并部署模型以进行预测之前，人工注释器手动查看数千个示例并添加训练 ML 模型所需的标签。这个过程耗时且成本高昂。

在 AWS 中，Amazon SageMaker Ground Truth 使用人工注释器通过 Amazon Mechanical Turk、第三方供应商或他们自己的员工简化了数据标记任务。Amazon SageMaker Ground Truth 实时学习这些人工注释，并应用主动学习来自动标记大部分剩余数据集，从而减少了所需的人工检查工作。与单独的人工注释相比，人工和 ML 功能的结合使 Amazon SageMaker Ground Truth 可以创建高度准确的训练数据集、节省时间、降低复杂性，并节约成本。

> MLCOST 02：如何在 ML 实验中优化成本？

笔记本是探索和使用少量数据进行实验的一种流行方法。在机器学习中，通常会在本地对数据集的一小部分样本进行迭代，然后进行扩展以对整个数据集进行分布式训练。

在 AWS 中，Amazon SageMaker 笔记本实例提供了托管的 Jupyter 环境，可用于探索小型数据样本。当您不主动使用笔记本实例时，请停止它们。在可行的情况下，提交您的工作，停止这些实例，然后在再次需要它们时重新启动它们。存储持久进行，您可以使用生命周期配置来自动执行软件包安装或存储库同步。

在进行训练模型实验时，请使用 Amazon SageMaker 笔记本“本地”模式在笔记本实例本身上而不是在单独的托管训练集群上训练模型。您可以迭代和测试您的工作，而不必每次都等待构建新的训练或托管集群。这样可以节省与创建托管训练集群相关的时间和成本。实验也可以在笔记本外部进行，例如在本地计算机上进行。在本地计算机上，您可以使用 SageMaker SDK 在 AWS 上训练和部署模型。

在进行实验时，还请查看面向机器学习的 AWS Marketplace，其中提供了不断增加的机器学习算法和模型目录。来自 AWS Marketplace 的模型直接部署到 Amazon SageMaker，并允许您快速构建 ML 应用程序。这样可以节省与模型开发相关的成本和时间。

此外，面向机器学习的 AWS Marketplace 让您能够出售自己开发的模型，从而为您提供额外的收入来源来利用内部模型创收。您可以将自定义模型提供给其他客户，同时保护您的知识产权。Amazon SageMaker 允许通过安全终端节点访问您的模型，而无需公开底层模型。

MLCOST 03：如何选择最节省成本的资源进行 ML 训练？

当您准备好使用完整的训练数据训练 ML 模型时，除非数据集很小，否则请避免在笔记本实例上以本地模式运行训练作业。相反，请使用一个或多个计算实例启动训练集群以进行分布式训练。根据工作负载在训练集群中正确调整计算实例的大小。

在 AWS 中，使用 Amazon SageMaker Training API 创建托管实例的集群。在训练集群中使用多个实例可以实现分布式训练，从而缩短了训练时间。训练完成后，训练集群中的所有实例都会自动终止。

尽管可以使用具有不同容量配置的多种实例类型进行训练，但是根据所使用的 ML 算法正确调整训练实例的大小非常重要。请注意，简单模型可能无法在大型实例上更快地训练，因为它们可能无法从增强的硬件并行机制中受益。由于较高的 GPU 通信开销，它们甚至可能训练速度更慢。建议您从小型实例开始，先横向扩展，然后再纵向扩展。此外，如果您选择的 ML 算法支持检查点，则可以使用托管现场训练和 Amazon SageMaker 进行评估，以节省成本。

除了选择用于训练的优化实例类型外，选择用于更快训练的 ML 框架的优化版本也很重要。AWS 提供了优化的框架版本，例如 TensorFlow、Chainer、Keras 和 Theano，其中包括针对整个 Amazon EC2 实例系列的高性能训练的优化。

在处理大量训练数据时，Amazon SageMaker “管道”模式比 Amazon SageMaker “文件”模式提供明显更好的读取吞吐量。在开始模型训练之前，“文件”模式将数据下载到本地 Amazon EBS 卷，而在训练 ML 模型时，“管道”模式将数据从 Amazon S3 流式传输到 Amazon SageMaker。这意味着您的训练作业将更快开始、更快完成，并且需要的磁盘空间更少，从而降低了在 Amazon SageMaker 上训练 ML 模型的总体成本。

为 ML 模型确定正确的超参数集可能成本很高。此过程通常需要诸如网格搜索或随机搜索之类的技术，这些技术涉及训练数百种不同的模型。Amazon SageMaker 自动进行模型调优（也称为超参数调优），通过使用您指定的算法和超参数范围在数据集中运行许多训练作业来查找模型的最佳版本。然后，它选择使模型表现最佳的超参数值，由您选择的指标进行性能衡量。超参数调优使用 ML 技术，可以使用有限数量的训练作业快速有效地确定最佳参数集。

此外，超参数调优作业的热启动可以加速调优过程并降低调优模型的成本。热启动超参数调优作业无需再从头开始调优作业。 相反，您可以基于选定的父级作业创建一个新的超参数调优作业，以便可以将在这些父级作业中进行的训练作业作为先验知识重复使用，从而降低与模型调优相关的成本。

最后，评估可自动分析数据并构建模型的 Amazon SageMaker AutoPilot，从而节省您的时间和成本。Autopilot 会从高性能算法列表中选择最佳算法，并自动在这些算法上尝试不同的参数设置，以获得最佳模型质量。

MLCOST 04：如何优化 ML Inference 成本?

根据模型训练，了解哪种实例类型最适合您的工作负载非常重要。首先要考虑延迟、吞吐量和成本。再次建议您从较小的实例开始，先横向扩展，然后再纵向扩展。

除了使用 ML 计算实例自动扩展以节省成本外，还要衡量 CPU 与 GPU 之间的差异。虽然深度学习 ML 模型需要高功率的 GPU 实例进行训练，但针对深度学习模型的推理通常并不需要全功率的 GPU。因此，在全功率的 GPU 上托管这些深度学习可能会导致利用率低下和产生不必要的成本。此外，请考虑您的 ML 模型所需的推理架构。也就是说，确定是否可以根据批量推理请求需要按需部署模型，或者是否可以确保模型 24 X 7 全天候可用以进行实时预测。

在 AWS 上，Amazon SageMaker 终端节点支持自动扩展，使您能够匹配资源的供求。通过 Amazon SageMaker 中的自动扩展，您可以通过选择正确的指标来扩展推理终端节点，从而确保模型的弹性和可用性，并优化成本。

在自动扩展增加或减少终端节点后面的实例数量以匹配推理请求量的同时，您还可以通过将部分 GPU 计算容量连接到实例来提高托管实例的计算能力。借助 Amazon Elastic Inference，可以将低成本 GPU 驱动的加速连接到 Amazon EC2 和 Amazon SageMaker 实例，从而减少运行深度学习推理的成本。

尽管独立的 GPU 实例非常适合需要并行处理数百个数据样本的模型训练活动，但通常它们的大小过大，无法进行推理，这会占用少量 GPU 资源。不同的模型需要不同数量的 GPU、CPU 和内存资源。选择 GPU 实例类型以满足最苛刻资源的要求，通常会导致其他资源利用不足和花费不必要的成本。

借助 Amazon Elastic Inference，您可以选择最适合模型的总体 CPU 和内存需求的 Amazon EC2 实例类型，然后单独配置所需的推理加速量级，有效使用资源和降低推理运行成本。

一些应用程序不需要在线/实时预测，并且更适合于周期性批量预测。在这种情况下，不需要 24 X 7 全天候托管终端节点。这些应用程序非常适合 Amazon SageMaker 批量转换。批量转换不是为每个请求进行单个推理，而是为整个数据集生成推理。批量转换管理执行推理所需的所有计算资源。这包括启动实例并在批量转换作业完成后终止它们。

有时对给定数据执行推理需要对数据进行预处理、后处理或进行这两种处理。这可能涉及在最终模型可以生成所需推理之前，将来自多个中间模型的推断链接在一起。在这种情况下，请使用 Amazon SageMaker Inference Pipelines，而不是在多个终端节点上部署中间模型。

*推理管道*是由一系列线性容器组成的 Amazon SageMaker 模型，该管道处理数据推理请求。这些推理管道是完全托管的，可以进行预处理、预测和后处理操作。管道模型调用按一系列 HTTPS 请求处理。通过将所有相关步骤部署到同一终端节点，您可以节省成本并减少推理延迟。

当生产中有多个模型且每个模型都部署在不同的终端节点上时，推理成本将随模型数量成比例增加。但是，如果您可以通过共享的服务容器提供大量相似的模型，并且不需要同时访问所有模型，请使用 Amazon SageMaker 多模型终端节点功能。这使您可以将多个训练有素的模型部署到终端节点，并使用单个容器为其提供服务。通过在预测请求中将目标模型名称指定为参数，可以

轻松调用特定模型。如果不经常访问 ML 模型出现长尾，则使用一个多模型终端节点可以有效地为推理流量提供服务，并节省大量成本。

供需匹配

有关适用于 ML 工作负载的成本优化方法中供需匹配领域的最佳实践，请参阅“AWS 架构完善的框架”白皮书。

支出认知

有关适用于 ML 工作负载的成本优化方法中支出认知领域的最佳实践，请参阅“AWS 架构完善的框架”白皮书。

持续优化

有关适用于 ML 工作负载的成本优化方法中持续优化领域的最佳实践，请参阅“AWS 架构完善的框架”白皮书。

资源

请参阅以下资源，详细了解成本优化的最佳实践。

文档和博客

- Amazon SageMaker 定价
- Use the Amazon SageMaker local mode to train on your notebook instance
- 充分利用 Amazon SageMaker 机器学习预算
- Lowering total cost of ownership for machine learning and increasing productivity with Amazon SageMaker

总结

机器学习为组织实现自动化、高效率和创新提供了无与伦比的机会。在本文中，我们通过 ML 的视角重新审视了架构完善的框架的五大支柱，以提供用于在 AWS 云中构建和操作可靠、安全、高效且极具成本效益的 ML 工作负载的架构最佳实践。在使用 AWS 上的 AI 服务、托管 ML 服务

和 ML 框架的背景下讨论了最佳实践，从而为您提供使用最合适的选项来实现业务目标的选择。使用由一系列问题组成的这些最佳实践来审视您现有的或建议的 ML 工作负载。

在运行 ML 工作负载时，请确保跨职能团队的参与和端到端管道的自动化准备就绪。为了提高可靠性，请利用自动扩展的自我修复功能并通过版本控制跟踪所有构件，以确保对系统故障做出可靠响应，从而可以自动重建功能系统。

同时应使用身份验证和授权控件来保护 AWS 上 ML 工作负载的安全，该控件严格控制谁可以访问各种 ML 构件以及可以访问哪些内容。安全的应用程序能够保护您组织的敏感信息资产，同时能够满足各层的合规性要求。要启用高性能的 ML 工作负载，您可以选择经过优化的几个实例类型，以适应不同的 ML 要求。采用数据驱动的方法来选择 CPU 与 GPU 实例，同时牢记 CPU、GPU、FPGA、内存、存储和可用网络容量的各种组合。对于成本优化，请充分利用“仅按您的使用量付费”，并根据数据、训练和推理需求调整资源大小，以减少不必要的浪费。

随着工具和流程的生态系统日趋成熟，机器学习的格局正在不断发展壮大。届时，我们会继续更新本文档，帮助您确保 ML 应用程序保持完善的架构。

参与者

本文档的参与者包括：

- Sireesha Muppala，Amazon Web Services 的 AI/ML 专家 SA
- Shelbee Eigenbrode，Amazon Web Services 的 AI/ML 解决方案架构师
- Christian Williams，Amazon Web Services 的机器学习专家 SA
- Bardia Nikpourian，Amazon Web Services 的 AI/ML 专家 TAM
- Ryan King，Amazon Web Services 的 AWS Managed Cloud 高级 TPM

延伸阅读

有关更多信息，请参阅以下内容：

- 管理机器学习项目（AWS 白皮书）

文档修订

日期	描述
2020 年 4 月	首次发布