为成功做好规划 MLOps

AWS 规范性指导

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

AWS 规范性指导: 为成功做好规划 MLOps

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon 的商标和商业外观不得用于任何非 Amazon 的商品或服务，也不得以任何可能引起客户混淆、贬低或诋毁 Amazon 的方式使用。所有非 Amazon 拥有的其他商标均为各自所有者的财产，这些所有者可能附属于 Amazon、与 Amazon 有关联或由 Amazon 赞助，也可能不是如此。

Table of Contents

简介 1
目标业务成果 1
数据 2
标签 2
提供清晰的标签说明 2
使用多数票 2
拆分和数据泄露 2
将您的数据分成至少三组 3
使用分层拆分算法 3
考虑重复的样本 4
考虑可能不可用的功能 4
特色商店 4
使用时空旅行查询 4
使用 IAM 角色 4
使用单元测试 5
训练 6
创建基线模型 6
使用以数据为中心的方法和错误分析 7
设计您的模型以实现快速迭代 7
追踪您的机器学习实验 9
对训练作业进行故障排除 10
部署 11
自动化部署周期 11
选择部署策略 12
蓝/绿 12
金丝雀 12
影子 12
A/B 测试 12
考虑您的推理要求 13
实时推理 13
异步推理 13
批量转换 14
监控 15
后续步骤和资源 18

资源 18
文档历史记录 20
术语表 21
..... 21
A 21
B 24
C 25
D 28
E 31
F 33
G 34
H 35
我 36
L 38
M 39
O 43
P 45
Q 47
R 48
S 50
T 53
U 54
V 55
W 55
Z 56
..... lvii

为成功做好规划 MLOps

Bruno Klein，Amazon Web Services ()AWS

2021 年 12 月（文档历史记录）

在生产环境中部署机器学习 (ML) 解决方案会带来许多在标准软件开发项目中不会出现的挑战。机器学习解决方案一开始就更复杂、更棘手。它们还存在于通常不稳定的环境中，由于各种预期和意想不到的原因，数据分布会随着时间的推移而出现显著的偏差。

许多机器学习从业者不是来自软件工程背景，因此他们可能不熟悉该行业的最佳实践，例如编写可测试的代码、模块化组件和有效使用版本控制，这一事实进一步加剧了这些问题。这些挑战造成了技术债务，随着时间的推移，在复合效应的推动下，机器学习团队的解决方案变得更加复杂和难以维护。

本指南列举了有助于缓解机器学习项目和工作负载中这些挑战的机器学习操作 (MLOps) 最佳实践。

由于 MLOps 这是一个跨领域的问题，因此这些问题不仅会影响部署和监控流程，还会影响整个模型生命周期。在本指南中， MLOps 最佳实践分为四个主要领域：

- 数据
- 训练
- 部署
- 监控

目标业务成果

在生产环境中部署机器学习模型是一项需要持续努力和专门的团队来维护这些资源的整个生命周期（在某些情况下甚至是几年）。机器学习模型可以从业务数据中释放可观的价值，但它们的成本很高。为了最大限度地降低成本，企业应遵循软件开发和数据科学方面的良好实践。他们应该意识到机器学习系统的细微差别，例如数据漂移，这会使模型在一段时间后出人意料地运行。通过意识到这些问题，企业可以在短期和长期内安全、灵活地实现其业务目标。

机器学习模型有几种，它们所针对的行业有不同类型的机器学习任务和业务问题，因此您需要为每种模型和行业考虑不同的问题。本指南中列出的实践并非特定于模型或业务，而是适用于各种模型和行业，以缩短部署时间、提高生产率并建立更强的治理和安全性。

将模型投入生产是一项多学科任务，需要数据科学家、机器学习工程师、数据工程师和软件工程师。在组建机器学习团队时，我们建议您以这些技能和背景为目标。

数据

DevOps 是一种软件工程实践，涉及软件的操作化。的常见元素 DevOps 包括版本控制代码、持续集成和持续交付 (CI/CD) 管道、单元测试以及可重现的代码构建和部署，所有这些都涉及代码。机器学习模型是代码和数据的产物，因此数据必须符合与代码相同的标准。 MLOps 必须解决与数据有关的问题，例如如何保持数据质量、如何识别数据中的边缘情况、如何保护数据以及如何使数据更易于维护。

主题

- 标签
- 拆分和数据泄露
- 特色商店

标签

提供清晰的标签说明

数据集可能包含模棱两可的样本，从而导致整个数据集的标签不一致。例如，考虑为包含狗的图像添加标签的任务。有些样本可能只包含动物的一瞥。这些标签应该标上正面还是负面标签？ 这类问题可以通过向贴标商提供清晰客观的说明来解决。

使用多数票

现在考虑一个问题，即使用语音与其他词语相似或相同的单词来标记包含嘈杂音频 speech-to-text的数据集，例如 kno w and go、shoe and tw o、c r y and hig h 或 right and w rite 或 right and wr it e。在这种情况下，贴标员可能会给这些样本加上不一致的标签。

为了保持标签的高度正确性，一种常见的方法是使用多数投票，即向多名工作人员提供相同的数据样本，然后汇总他们的结果。博客文章中描述了这种方法及其更复杂的变体。在 M AWS achine Learning 博客上利用人群的智慧和 SageMaker Amazon AI Ground Truth 更准确地注释数据。

拆分和数据泄露

当您的模型在推理期间（模型投入生产并收到预测请求的那一刻）获得它不应该访问的数据时，就会发生数据泄露，例如用于训练的数据样本，或者在生产中部署模型时不可用的信息。

如果您的模型无意中在训练数据上进行了测试，则数据泄露可能会导致过度拟合。过度拟合意味着您的模型无法很好地推广到看不见的数据。本节提供了避免数据泄露和过度拟合的最佳实践。

将您的数据分成至少三组

数据泄露的一个常见来源是在训练期间不正确地划分（拆分）数据。例如，数据科学家可能有意或无意地根据用于测试的数据对模型进行了训练。在这种情况下，您可能会观察到由过度拟合导致的成功指标非常高。要解决这个问题，你应该将数据分成至少三组：trainingvalidation、和testing。

通过以这种方式拆分数据，您可以使用validation集合来选择和调整用于控制学习过程的参数（超参数）。当您达到预期的结果或达到改善的稳定状态时，请对testing现场进行评估。该testing集合的性能指标应与其他集合的指标相似。这表明集合之间没有分布不匹配，并且您的模型有望在生产中很好地进行概化。

使用分层拆分算法

当您将数据拆分为trainingvalidation、和testing对于小型数据集，或者处理高度不平衡的数据时，请务必使用分层拆分算法。分层可确保每个拆分包含的类别数量或分布大致相同。scikit-learn 机器学习库已经实现了分层，Apache Spark 也是如此。

对于样本量，请确保验证集和测试集有足够的评估数据，这样您就可以得出具有统计学意义的结论。例如，相对较小的数据集（少于 100 万个样本）的常见拆分大小为 70%、15% 和 15%，对于trainingvalidation、和testing。对于非常大的数据集（超过 100 万个样本），您可以使用 90%、5% 和 5% 来最大限度地利用可用的训练数据。

在某些用例中，将数据拆分为其他集合很有用，因为在收集生产数据期间，生产数据的分布可能发生了剧烈的突然变化。例如，考虑使用数据收集流程来构建杂货店商品的需求预测模型。如果数据科学团队在2019年收集training数据，收集2020年1月至2020年3月testing的数据，那么模型可能会在现场得分不错。testing但是，当该模型在生产中部署时，由于 COVID-19 疫情，某些物品的消费模式已经发生了重大变化，并且该模型将产生糟糕的结果。在这种情况下，添加另一组（例如recent_testing）作为模型批准的额外保障措施是有意义的。此添加可能会使您无法批准因分布不匹配而立即表现不佳的模型投入生产。

在某些情况下，您可能需要创建包含特定类型样本的额外validation或testing集合，例如与少数族裔群体相关的数据。这些数据样本对于正确处理很重要，但在整个数据集中可能无法很好地呈现。这些数据子集称为切片。

以用于信用分析的机器学习模型为例，该模型是根据整个国家的数据进行训练的，并且经过平衡以均衡地考虑目标变量的整个域。此外，请考虑此模型可能具有City特征。如果使用这种模式的银行将其业务扩展到特定城市，它可能会对该模型在该地区的表现感兴趣。因此，批准渠道不仅应根据整个国家的测试数据评估模型的质量，还应评估给定城市的测试数据。

当数据科学家研究新模型时，他们可以通过在模型的验证阶段整合代表性不足的切片来轻松评估模型的能力并考虑边缘情况。

进行随机拆分时要考虑重复的样本

另一个不太常见的泄漏源是可能包含过多重复样本的数据集。在这种情况下，即使将数据拆分为子集，不同的子集也可能有共同的样本。根据重复项的数量，过度拟合可能会被误认为是泛化。

考虑在生产环境中接收推论时可能不可用的功能

当模型使用生产中不可用的功能进行训练时，在调用推论的那一刻，也会发生数据泄露。由于模型通常基于历史数据构建，因此可能会使用在某个时间点不存在的其他列或值来丰富这些数据。以信用审批模式为例，该模型具有跟踪客户在过去六个月中向银行发放了多少贷款的功能。如果部署此模型并用于银行没有六个月历史记录的新客户的信贷审批，则存在数据泄露的风险。

Amazon SageMaker AI Feature Store 可以帮助解决这个问题。您可以使用时空旅行查询来更准确地测试模型，这些查询可用于查看特定时间点的数据。

特色商店

使用 SageMaker AI Feature Store 可以提高团队的工作效率，因为它可以分离组件边界（例如，存储与使用情况）。它还为组织内的不同数据科学团队提供了功能的可重用性。

使用时空旅行查询

Feature Store 中的时空旅行功能有助于重现模型构建并支持更强大的治理实践。当组织想要评估数据沿袭时，这可能很有用，类似于 Git 等版本控制工具评估代码的方式。时空旅行查询还可以帮助组织为合规性检查提供准确的数据。有关更多信息，请参阅 Machine Learn ing 博客上的 “ AWS 了解亚马逊 SageMaker AI 功能商店的关键功能”。

使用 IAM 角色

Feature Store 还有助于在不影响团队工作效率和创新的情况下提高安全性。您可以使用 AWS Identity and Access Management (IAM) 角色授予或限制特定用户或群组对特定功能的精细访问权限。

例如，以下策略限制了对功能商店中敏感功能的访问权限。

```
{
    "Version": "2012-10-17",
```

```
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Deny",
            "Action": "*",
            "Resource": "arn:aws:s3:::us-east-2-12345678910-features/12345678910/
sagemaker/us-east-2/offline-store/doctor-appointments"
        }
    ]
}
```

有关使用 Feature Store 进行数据安全和加密的更多信息，请参阅 SageMaker AI 文档中的安全和访问控制。

使用单元测试

当数据科学家根据某些数据创建模型时，他们通常会对数据的分布做出假设，或者他们进行彻底的分析以充分了解数据的属性。部署这些模型后，它们最终会过时。当数据集过时时，数据科学家、机器学习工程师和（在某些情况下）自动化系统会使用从在线或离线商店获取的新数据对模型进行重新训练。

但是，这些新数据的分布可能已经改变，这可能会影响当前算法的性能。检查这类问题的一种自动方法是借鉴软件工程中的单元测试概念。需要测试的常见内容包括缺失值的百分比、类别变量的基数，以及使用诸如假设检验统计量（t 检验）之类的框架，实值列是否符合某种预期分布。您可能还需要验证数据架构，以确保它没有发生变化，也不会静默生成无效的输入要素。

单元测试需要了解数据及其域，以便您可以计划在机器学习项目中执行的确切断言。有关更多信息，请参阅 AWS 大数据博客 PyDeequ上的大规模测试数据质量。

训练

MLOps 关注机器学习生命周期的运营化。因此，它必须促进数据科学家和数据工程师的工作，以创建务实的模型，这些模型可以满足业务需求并长期运行良好，而不会产生技术债务。

遵循本节中的最佳实践，以帮助解决模型训练难题。

主题

- 创建基线模型
- 使用以数据为中心的方法和错误分析
- 设计您的模型以实现快速迭代
- 追踪您的机器学习实验
- 对训练作业进行故障排除

创建基线模型

当从业者面临机器学习解决方案的业务问题时，他们通常首先倾向于使用该 state-of-the-art算法。这种做法是有风险的，因为该 state-of-the-art算法可能还没有经过时间考验。此外，该 state-of-the-art算法通常更复杂且不太为人所知，因此与更简单的替代模型相比，它可能只会带来微不足道的改进。更好的做法是创建一个基准模型，该模型的验证和部署速度相对较快，并且可以赢得项目利益相关者的信任。

创建基线时，我们建议您尽可能评估其指标性能。将基准模型的性能与其他自动化或手动系统进行比较，以保证其成功，并确保模型实施或项目可以在中长期内交付。

应与机器学习工程师一起进一步验证基准模型，以确认该模型可以满足为项目确定的非功能性要求，例如推理时间、数据预计移动分布的频率、在这些情况下是否可以轻松地对模型进行再训练，以及如何部署，这将影响解决方案的成本。获取有关这些问题的多学科观点，以增加开发成功且长期运行的模型的机会。

数据科学家可能倾向于在基线模型中添加尽可能多的特征。尽管这提高了模型预测预期结果的能力，但其中一些功能可能只会增加指标改进。许多功能，尤其是那些高度相关的功能，可能是多余的。添加太多功能会增加成本，因为它需要更多的计算资源和调整。特征过多也会影响模型的 day-to-day操作，因为数据漂移的可能性更大，或者发生得更快。

假设一个模型，其中两个输入特征高度相关，但只有一个特征具有因果关系。例如，预测贷款是否会违约的模型可能具有客户年龄和收入等输入特征，这些特征可能高度相关，但只能使用收入来发放或拒绝

贷款。基于这两个特征训练的模型可能依赖于没有因果关系的特征（例如年龄）来生成预测输出。如果模型在投入生产后，收到的客户年龄大于或小于训练集所含平均年龄的推理请求，则其性能可能会开始不佳。

此外，每个特征在生产过程中都可能出现分布偏移，从而导致模型出现意外行为。出于这些原因，模型的特征越多，它在漂移和陈旧方面就越脆弱。

数据科学家应使用相关性度量和 Shapley 值来衡量哪些特征为预测增加了足够的价值，哪些特征应该保留。拥有如此复杂的模型会增加出现反馈回路的机会，在这种回路中，模型会改变建模的环境。一个例子是推荐系统，在这种系统中，消费者行为可能会因为模型的推荐而改变。跨模型作用的反馈回路不太常见。例如，考虑一个推荐电影的推荐系统，以及另一个推荐书籍的系统。如果两种模式都针对同一组消费者，它们就会相互影响。

对于您开发的每个模型，请考虑哪些因素可能影响这些动态，以便您知道在生产中要监控哪些指标。

使用以数据为中心的方法和错误分析

如果您使用简单的模型，您的机器学习团队可以专注于改进数据本身，并采用以数据为中心的方法而不是以模型为中心的方法。如果您的项目使用非结构化数据，例如图像、文本、音频和其他可以由人类评估的格式（与结构化数据相比，结构化数据可能更难有效地映射到标签），则要获得更好的模型性能，一个好的做法是进行错误分析。

错误分析包括评估验证集上的模型并检查最常见的错误。这有助于识别模型可能难以正确处理的潜在相似数据样本组。要执行误差分析，您可以列出预测误差更高的推论，或者将一个类别的样本预测为来自另一个类别的样本的误差进行排名。

设计您的模型以实现快速迭代

当数据科学家遵循最佳实践时，他们可以在概念验证甚至再训练期间轻松快速地尝试新算法或混合和匹配不同的功能。这项实验有助于成功生产。一个好的做法是在基线模型的基础上构建，采用稍微复杂一点的算法，迭代添加新功能，同时监控训练和验证集的性能，将实际行为与预期行为进行比较。该训练框架可以提供预测能力的最佳平衡，并有助于使模型尽可能简单，同时减少技术债务足迹。

为了实现快速迭代，数据科学家必须交换不同的模型实现，以确定用于特定数据的最佳模型。如果您的团队规模庞大，截止日期短，并且还有其他与项目管理相关的后勤工作，那么如果没有适当的方法，则很难快速迭代。

在软件工程中，Liskov替换原理是一种设计软件组件之间交互的机制。该原则指出，您应该能够在不破坏客户端应用程序或实现的情况下将接口的一个实现替换为另一种实现。当你为机器学习系统编写训练

代码时，你可以利用这个原理来建立边界并封装代码，这样你就可以轻松地替换算法，更有效地尝试新的算法。

例如，在以下代码中，您可以通过添加新的类实现来添加新的实验。

```
from abc import ABC, abstractmethod

from pandas import DataFrame


class ExperimentRunner(object):

    def __init__(self, *experiments):
        self.experiments = experiments

    def run(self, df: DataFrame) -> None:
        for experiment in self.experiments:
            result = experiment.run(df)
            print(f'Experiment "{experiment.name}" gave result {result}')


class Experiment(ABC):

    @abstractmethod
    def run(self, df: DataFrame) -> float:
        pass

    @property
    @abstractmethod
    def name(self) -> str:
        pass


class Experiment1(Experiment):

    def run(self, df: DataFrame) -> float:
        print('performing experiment 1')
        return 0

    def name(self) -> str:
        return 'experiment 1'


class Experiment2(Experiment):
```

```
    def run(self, df: DataFrame) -> float:
        print('performing experiment 2')
        return 0

    def name(self) -> str:
        return 'experiment 2'


class Experiment3(Experiment):

    def run(self, df: DataFrame) -> float:
        print('performing experiment 3')
        return 0

    def name(self) -> str:
        return 'experiment 3'


if __name__ == '__main__':
    runner = ExperimentRunner(*[
        Experiment1(),
        Experiment2(),
        Experiment3()
    ])
    df = ...
    runner.run(df)
```

追踪您的机器学习实验

当你进行大量实验时，重要的是要衡量你观察到的改进是实施的变化还是偶然的产物。您可以使用 Amazon SageMaker AI Ex periments 轻松创建实验，并将元数据与实验关联起来，以便跟踪、比较和评估。

降低模型构建过程的随机性对于调试、故障排除和改善治理非常有用，因为在给定相同的代码和数据的情况下，您可以更确定地预测输出模型的推断。

由于随机权重初始化、并行计算同步性、GPU 内部复杂性以及类似的非确定性因素，通常不可能使训练代码完全可复制。但是，正确设置随机种子，确保每次训练都从同一点开始并且行为相似，可以显著提高结果的可预测性。

对训练作业进行故障排除

在某些情况下，即使是非常简单的基线模型，数据科学家也可能很难拟合。在这种情况下，他们可能会决定需要一种能够更好地拟合复杂函数的算法。一个好的测试方法是使用数据集中很小一部分（例如，大约 10 个样本）的基线来确保算法过度适合此样本。这有助于排除数据或代码问题。

另一个用于调试复杂场景的有用工具是 Amazon SageMaker AI Debugger，它可以捕获与算法正确性和基础设施相关的问题，例如最佳计算使用率。

部署

在软件工程中，将代码投入生产需要尽职调查，因为代码可能会出现意外行为，不可预见的用户行为可能会破坏软件，并且会发现意想不到的边缘情况。软件工程师和 DevOps 工程师通常采用单元测试和回滚策略来降低这些风险。使用机器学习，将模型投入生产需要更多的规划，因为实际环境预计会发生变化，而且在许多情况下，模型是根据指标进行验证的，这些指标代表了他们想要改进的实际业务指标。

请遵循本节中的最佳实践，以帮助应对这些挑战。

主题

- 自动化部署周期
- 选择部署策略
- 考虑您的推理要求

自动化部署周期

培训和部署过程应完全自动化，以防止人为错误并确保始终如一地运行构建检查。用户不应拥有生产环境的写入权限。

Amazon SageMaker AI Pipelines 和AWS CodePipeline帮助创建CI/CD pipelines for ML projects. One of the advantages of using a CI/CD管道是，所有用于采集数据、训练模型和执行监控的代码都可以使用 Git 等工具进行版本控制。有时，你必须使用相同的算法和超参数，但数据不同，来重新训练模型。验证你使用的算法版本是否正确，唯一的方法是使用源代码控制和标签。您可以使用 A SageMaker I 提供的默认项目模板作为 MLOps 练习的起点。

在创建 CI/CD 管道来部署模型时，请务必使用构建标识符、代码版本或提交以及数据版本来标记构建工件。此练习可帮助您解决任何部署问题。在高度监管的领域进行预测的模型有时也需要标记。能够向后工作并识别与机器学习模型相关的确切数据、代码、构建、检查和批准，可以帮助显著改善治理。

CI/CD 管道的部分工作是对其正在构建的内容进行测试。尽管预计数据单元测试将在feature store摄取数据之前进行，但管道仍负责对给定模型的输入和输出执行测试并检查关键指标。此类检查的一个例子是在固定验证集上验证新模型，并使用既定阈值确认其性能与以前的模型相似。如果性能明显低于预期，则构建应该失败，模型不应投入生产。

CI/CD 管道的广泛使用还支持拉取请求，这有助于防止人为错误。使用拉取请求时，每项代码更改都必须经过至少一名其他团队成员的审核和批准，然后才能投入生产。拉取请求对于识别不符合业务规则的代码以及在团队中传播知识也很有用。

选择部署策略

MLOps 部署策略包括blue/green, canary, shadow, and A/B测试。

蓝/绿

Blue/green deployments are very common in software development. In this mode, two systems are kept running during development: blue is the old environment (in this case, the model that is being replaced) and green is the newly released model that is going to production. Changes can easily be rolled back with minimum downtime, because the old system is kept alive. For more in-depth information about blue/green在的背景下部署 SageMaker，请参阅 M AWS achine Learning 博客 AWS CodeDeploy上的 “安全部署和监控 SageMaker Amazon AI 终端节点” 的博客文章。 AWS CodePipeline

金丝雀

Canary 部署与blue/green deployments in that both keep two models running together. However, in canary deployments, the new model is rolled out to users incrementally, until all traffic eventually shifts over to the new model. As in blue/green部署类似，风险可以降低，因为新的（可能存在故障）模型在初始部署期间会受到密切监控，并且可以在出现问题时回滚。在 SageMaker AI 中，您可以使用 InitialVariantWeightAPI 指定初始流量分布。

影子

您可以使用影子部署将模型安全地投入生产。在此模式下，新模型可与旧模型或业务流程配合使用，并且在不影响任何决策的情况下进行推断。在将模型升级到生产之前，此模式可用作最终检查或更高保真度的实验。

当你不需要任何用户推理反馈时，暗影模式非常有用。您可以通过执行误差分析并将新模型与旧模型进行比较来评估预测的质量，也可以监控输出分布以验证其是否符合预期。要了解如何使用 SageMaker AI 进行影子部署，请参阅 Machine Learning 博客上的博客文章 “在 Amazon SageMaker AI 中部署影子 AWS 机器学习模型”。

A/B 测试

当机器学习从业者在其环境中开发模型时，他们优化的指标通常是真正重要的业务指标的代名词。这使得很难确定新模式是否真的会改善业务成果，例如收入和点击率，并减少用户投诉的数量。

以电子商务网站为例，该网站的业务目标是销售尽可能多的产品。审核小组知道，销售和客户满意度与内容丰富且准确的评论直接相关。团队成员可能会提出一种新的评论排名算法来提高销量。通过使用A/

B测试，他们可以将新旧算法推广到不同但相似的用户组，并监控结果，以查看从新模型中获得预测的用户是否更有可能进行购买。

A/B 测试还有助于衡量模型过时和漂移对业务的影响。团队可以将新模型投入生产并重复使用，对每个模型进行 A/B 测试，并创建年龄与性能对比图表。这将有助于团队了解其生产数据中的数据漂移波动性。

有关如何使用 SageMaker AI 执行 A/B 测试的更多信息，请参阅 Machine Learning 博客上的博客文章 “使用 Amazon A SageMaker I 在生产环境中测试 ML 模型 A/B 测试 ML 模型” AWS 。

考虑您的推理要求

借 SageMaker 助 AI，您可以选择底层基础架构，以不同的方式部署模型。这些推理调用功能支持不同的用例和成本概况。您的选项包括实时推理、异步推理和批量转换，如以下各节所述。

实时推理

实时推理非常适合需要实时、交互式、低延迟的推理工作负载。您可以将模型部署到 SageMaker AI 托管服务，并获得可用于推理的终端节点。这些终端节点是完全托管的，支持自动扩展（参见自动扩展 SageMaker Amazon AI 模型），并且可以部署在多个可用区中。

如果你有使用 Apache MXNet、或构建的深度学习模型 PyTorch TensorFlow，你也可以使用 A mazon A SageMaker I Elastic Inference (EI)。借助 EI，您可以将分数附加 GPUs 到任何 SageMaker AI 实例以加快推理。您可以选择客户端实例来运行您的应用程序，并连接 EI 加速器以使用正确的 GPU 加速量来满足您的推理需求。

另一种选择是使用多模型端点，它为部署大量模型提供了一种可扩展且经济实惠的解决方案。这些端点使用可托管多个模型的共享服务容器。与使用单模型端点相比，多模型端点通过提高端点利用率来降低托管成本。它们还可以减少部署开销，因为 SageMaker AI 可以管理在内存中加载模型并根据流量模式对其进行扩展。

有关在 SageMaker AI 中部署 ML 模型的其他最佳实践，请参阅 A SageMaker I 文档中的部署最佳实践。

异步推理

Amazon SageMaker AI 异步推理是 SageMaker AI 中的一项功能，用于对传入的请求进行排队并异步处理这些请求。此选项非常适合负载大小高达 1 GB、处理时间长、延迟要求接近实时的请求。异步推断使您能够在没有请求要处理时自动将实例数缩放到零，从而节省成本，因此您只需在终端节点处理请求时才付费。

批量转换

要执行以下操作时，请使用批量转换：

- 预处理数据集以从数据集中删除可能干扰训练或推理的噪声或偏差。
- 从大型数据集获取推理。
- 当您不需要持续性终端节点时运行推理。
- 将输入记录与推理相关联，以帮助解释结果。

监控

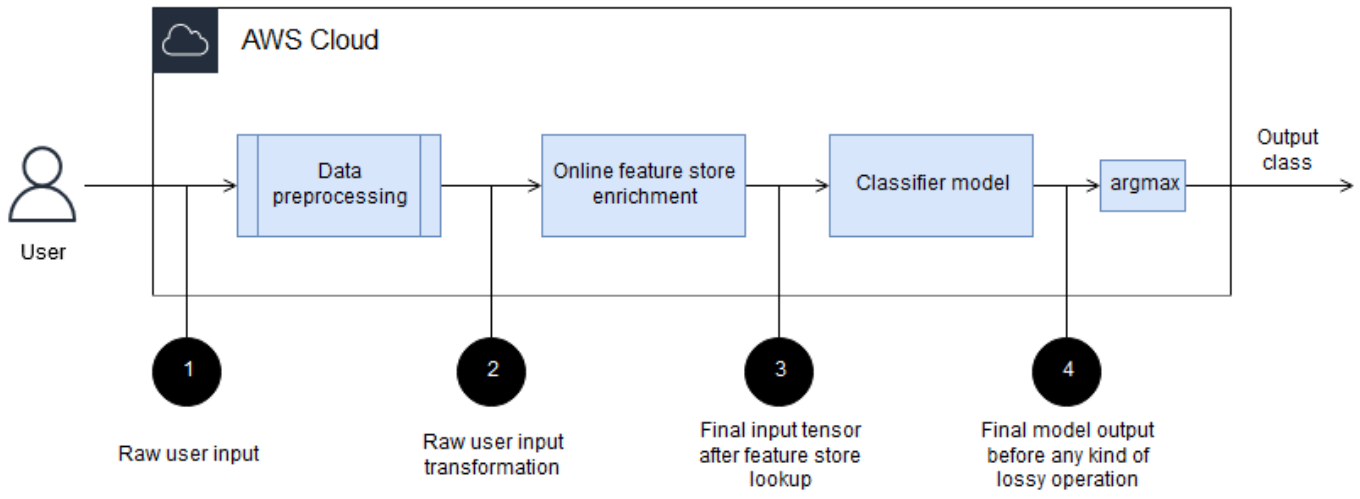
当模型已经投入生产并提供商业价值时，请进行持续检查，以确定何时必须对模型进行再训练或采取行动。

您的监控团队应采取主动行动，而不是被动行动，以更好地了解环境的数据行为，并识别数据漂移的频率、速率和突发性。团队应识别数据中可能在训练集、验证集和其他边缘案例切片中代表性不足的新边缘案例。他们应存储服务质量 (QoS) 指标，使用警报在出现问题时立即采取行动，并定义提取和修改当前数据集的策略。这些实践首先记录模型的请求和响应，为故障排除或其他见解提供参考。

理想情况下，数据转换应在处理过程中的几个关键阶段进行记录：

- 在进行任何类型的预处理之前
- 在进行任何形式的 feature store 充实之后
- 在模型的所有主要阶段之后
- 在模型输出上出现任何类型的有损函数之前，例如 `argmax`

下图说明了这些阶段。



您可以使用 SageMaker AI 模型监控器自动捕获输入和输出数据，并将其存储在亚马逊简单存储服务 (Amazon S3) Simple Service 中。您可以通过向自定义服务容器添加日志来实现其他类型的中间日志记录。

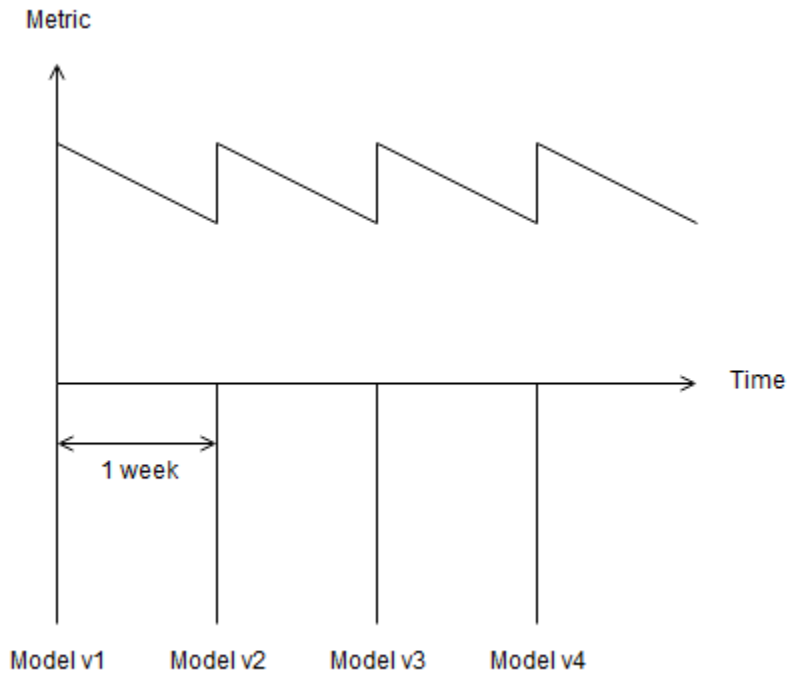
记录模型中的数据后，您可以监控分布偏差。在某些情况下，你可以在推断后不久获得真实情况（正确标记的数据）。一个常见的例子是预测要向用户展示的最相关的广告的模型。用户离开页面后，您就可以确定他们是否点击了广告。如果用户点击了广告，则可以记录该信息。在这个简单的示例中，您可以使用可以在训练和部署中衡量的指标（例如准确度或 F1）轻松量化模型的成功程度。有关标记数据的这些场景的更多信息，请参阅 SageMaker AI 文档中的监控模型质量。但是，这些简单的场景并不常见，因为模型通常是为了优化仅代表实际业务结果的数学上便捷的指标而设计的。在这种情况下，最佳做法是在生产中部署模型时监控业务结果。

以评论排名模型为例。如果机器学习模型的定义业务结果是在网页顶部显示最相关和最有用的评论，则可以通过添加诸如 “这有用吗？” 之类的按钮来衡量模型的成功 对于每条评论。衡量此按钮的点击率可能是一种衡量业务结果的衡量标准，可以帮助你衡量模型在生产中的表现。

要监控 SageMaker AI 中输入或输出标签的偏差，可以使用 A SageMaker I Model Monitor 的数据质量功能，该功能可以监控输入和输出。您还可以通过构建自定义容器来实现自己的 SageMaker AI 模型监视器逻辑。

监控模型在开发时间和运行时收到的数据至关重要。工程师不仅应监控数据是否存在架构更改，还应监控分布不匹配情况。检测架构更改更容易，并且可以通过一组规则来实现，但是分布不匹配通常更棘手，特别是因为它需要您定义一个阈值来量化何时发出警报。在已知受监控分布的情况下，最简单的方法通常是监视分布的参数。如果是正态分布，则为均值和标准差。其他关键指标，例如缺失值的百分比、最大值和最小值，也很有用。

您还可以创建持续监控作业，对训练数据和推理数据进行采样并比较它们的分布。您可以为模型输入和模型输出创建这些作业，并根据时间绘制数据，以可视化任何突然或逐渐的漂移。下图说明了这一点。



为了更好地了解数据的偏差特征，例如数据分布发生显著变化的频率、变化速度或突然性，我们建议您持续部署新的模型版本并监控其性能。例如，如果您的团队每周都部署一个新模型，并且发现每次模型的性能都显著提高，则他们可以确定至少应在不到一周的时间内交付新模型。

后续步骤和资源

本指南将引导您在规划要投入生产的机器学习模型的生命周期时需要考虑的一些注意事项。它讨论了数据、培训、部署和监控这四个领域的挑战和最佳实践，并包括其他相关资源。

AWS 提供 Well-Architected Framework，可帮助云架构师为各种应用程序、工作负载和技术领域构建安全、高性能、弹性和高效的基础架构。欲了解更多内容，请参阅 Well-Architecte AWS d 提供的机器学习镜头。

资源

亚马逊 SageMaker AI 文档

- 亚马逊 SageMaker AI 专题商店
- 功能库安全和访问控制
- Shapley 的价值观
- 亚马逊 A SageMaker I 调试器
- 亚马逊 SageMaker AI 管道
- 亚马逊 SageMaker AI 默认项目模板
- SageMaker AI 实时推理
- 自动缩放 Amazon SageMaker 人工智能模型
- 亚马逊 SageMaker AI 异步推理
- SageMaker AI 模型监视器

AWS 开发者工具

- AWS CodePipeline

AWS 博客文章

- 了解 Amazon A SageMaker I Feature Store 的关键功能
- 使用以下方法大规模测试数据质量 PyDeequ
- 亚马逊 SageMaker AI 实验
- 使用和安全部署和监控 Amazon SageMaker 终端 CodePipeline 节点 AWS CodeDeploy

- 在 Amazon A SageMaker I 中部署影子机器学习模型
- A/B 使用 Amazon A SageMaker I 在生产环境中测试 ML 模型

文档历史记录

下表介绍了本指南的一些重要更改。如果您希望收到有关未来更新的通知，可以订阅 RSS 源。

变更	说明	日期
初次发布	—	2021 年 12 月 20 日

AWS 规范性指导词汇表

以下是 AWS 规范性指导提供的策略、指南和模式中的常用术语。若要推荐词条，请使用术语表末尾的提供反馈链接。

数字

7 R

将应用程序迁移到云中的 7 种常见迁移策略。这些策略以 Gartner 于 2011 年确定的 5 R 为基础，包括以下内容：

- 重构/重新架构 - 充分利用云原生功能来提高敏捷性、性能和可扩展性，以迁移应用程序并修改其架构。这通常涉及到移植操作系统和数据库。示例：将您的本地 Oracle 数据库迁移到兼容 Amazon Aurora PostgreSQL 的版本。
- 更换平台 - 将应用程序迁移到云中，并进行一定程度的优化，以利用云功能。示例：在中将您的本地 Oracle 数据库迁移到适用于 Oracle 的亚马逊关系数据库服务 (Amazon RDS) AWS Cloud。
- 重新购买 - 转换到其他产品，通常是从传统许可转向 SaaS 模式。示例：将您的客户关系管理 (CRM) 系统迁移到 Salesforce.com。
- 更换主机（直接迁移）- 将应用程序迁移到云中，无需进行任何更改即可利用云功能。示例：在中的 EC2 实例上将您的本地 Oracle 数据库迁移到 Oracle AWS Cloud。
- 重新定位（虚拟机监控器级直接迁移）：将基础设施迁移到云中，无需购买新硬件、重写应用程序或修改现有操作。您可以将服务器从本地平台迁移到同一平台的云服务。示例：将Microsoft Hyper-V应用程序迁移到 AWS。
- 保留（重访）- 将应用程序保留在源环境中。其中可能包括需要进行重大重构的应用程序，并且您希望将工作推迟到以后，以及您希望保留的遗留应用程序，因为迁移它们没有商业上的理由。
- 停用 - 停用或删除源环境中不再需要的应用程序。

A

ABAC

请参阅基于属性的访问控制。

抽象服务

参见托管服务。

ACID

参见原子性、一致性、隔离性、耐久性。

主动-主动迁移

一种数据库迁移方法，在这种方法中，源数据库和目标数据库保持同步（通过使用双向复制工具或双写操作），两个数据库都在迁移期间处理来自连接应用程序的事务。这种方法支持小批量、可控的迁移，而不需要一次性割接。与主动-被动迁移相比，它更灵活，但需要更多的工作。

主动-被动迁移

一种数据库迁移方法，在这种方法中，源数据库和目标数据库保持同步，但在将数据复制到目标数据库时，只有源数据库处理来自连接应用程序的事务。目标数据库在迁移期间不接受任何事务。

聚合函数

一个 SQL 函数，它对一组行进行操作并计算该组的单个返回值。聚合函数的示例包括`SUM`和`MAX`。

AI

参见人工智能。

AIOps

参见人工智能操作。

匿名化

永久删除数据集中个人信息的过程。匿名化可以帮助保护个人隐私。匿名化数据不再被视为个人数据。

反模式

一种用于解决反复出现的问题的常用解决方案，而在这类问题中，此解决方案适得其反、无效或不如替代方案有效。

应用程序控制

一种安全方法，仅允许使用经批准的应用程序，以帮助保护系统免受恶意软件的侵害。

应用程序组合

有关组织使用的每个应用程序的详细信息的集合，包括构建和维护该应用程序的成本及其业务价值。这些信息是产品组合发现和分析过程的关键，有助于识别需要进行迁移、现代化和优化的应用程序并确定其优先级。

人工智能（AI）

计算机科学领域致力于使用计算技术执行通常与人类相关的认知功能，例如学习、解决问题和识别模式。有关更多信息，请参阅什么是人工智能？

人工智能操作 (AIOps)

使用机器学习技术解决运营问题、减少运营事故和人为干预以及提高服务质量的过程。有关如何在 AIOps AWS 迁移策略中使用的更多信息，请参阅操作集成指南。

非对称加密

一种加密算法，使用一对密钥，一个公钥用于加密，一个私钥用于解密。您可以共享公钥，因为它不用于解密，但对私钥的访问应受到严格限制。

原子性、一致性、隔离性、持久性（ACID）

一组软件属性，即使在出现错误、电源故障或其他问题的情况下，也能保证数据库的数据有效性和操作可靠性。

基于属性的访问权限控制（ABAC）

根据用户属性（如部门、工作角色和团队名称）创建精细访问权限的做法。有关更多信息，请参阅 AWS Identity and Access Management (I AM) 文档 AWS中的 AB AC。

权威数据源

存储主要数据版本的位置，被认为是最可靠的信息源。您可以将数据从权威数据源复制到其他位置，以便处理或修改数据，例如对数据进行匿名化、编辑或假名化。

可用区

中的一个不同位置 AWS 区域 ，不受其他可用区域故障的影响，并向同一区域中的其他可用区提供低成本、低延迟的网络连接。

AWS 云采用框架 (AWS CAF)

该框架包含指导方针和最佳实践 AWS ，可帮助组织制定高效且有效的计划，以成功迁移到云端。AWS CAF将指导分为六个重点领域，称为视角：业务、人员、治理、平台、安全和运营。业务、人员和治理角度侧重于业务技能和流程；平台、安全和运营角度侧重于技术技能和流程。例如，人员角度针对的是负责人力资源（HR）、人员配置职能和人员管理的利益相关者。从这个角度来看，AWS CAF 为人员发展、培训和沟通提供了指导，以帮助组织为成功采用云做好准备。有关更多信息，请参阅 AWS CAF 网站和 AWS CAF 白皮书。

AWS 工作负载资格框架 (AWS WQF)

一种评估数据库迁移工作负载、推荐迁移策略和提供工作估算的工具。 AWS WQF 包含在 AWS Schema Conversion Tool (AWS SCT) 中。它用来分析数据库架构和代码对象、应用程序代码、依赖关系和性能特征，并提供评测报告。

B

坏机器人

旨在破坏个人或组织或对其造成伤害的机器人。

BCP

参见业务连续性计划。

行为图

一段时间内资源行为和交互的统一交互式视图。您可以使用 Amazon Detective 的行为图来检查失败的登录尝试、可疑的 API 调用和类似的操作。有关更多信息，请参阅 Detective 文档中的行为图中的数据。

大端序系统

一个先存储最高有效字节的系统。另请参见字节顺序。

二进制分类

一种预测二进制结果（两个可能的类别之一）的过程。例如，您的 ML 模型可能需要预测诸如“该电子邮件是否为垃圾邮件？” 或“这个产品是书还是汽车？”之类的问题

bloom 筛选条件

一种概率性、内存高效的数据结构，用于测试元素是否为集合的成员。

蓝/绿部署

一种部署策略，您可以创建两个独立但完全相同的环境。在一个环境中运行当前的应用程序版本（蓝色），在另一个环境中运行新的应用程序版本（绿色）。此策略可帮助您在影响最小的情况下快速回滚。

自动程序

一种通过互联网运行自动任务并模拟人类活动或互动的软件应用程序。有些机器人是有用或有益的，例如在互联网上索引信息的网络爬虫。其他一些被称为恶意机器人的机器人旨在破坏个人或组织或对其造成伤害。

僵尸网络

被恶意软件感染并受单方（称为机器人牧民或机器人操作员）控制的机器人网络。僵尸网络是最著名的扩展机器人及其影响力的机制。

分支

代码存储库的一个包含区域。在存储库中创建的第一个分支是主分支。您可以从现有分支创建新分支，然后在新分支中开发功能或修复错误。为构建功能而创建的分支通常称为功能分支。当功能可以发布时，将功能分支合并回主分支。有关更多信息，请参阅关于分支（GitHub 文档）。

破碎的玻璃通道

在特殊情况下，通过批准的流程，用户 AWS 账户 可以快速访问他们通常没有访问权限的内容。有关更多信息，请参阅 Well -Architected 指南中的 “实施破碎玻璃程序” 指示 AWS 器。

棕地策略

您环境中的现有基础设施。在为系统架构采用棕地策略时，您需要围绕当前系统和基础设施的限制来设计架构。如果您正在扩展现有基础设施，则可以将棕地策略和全新策略混合。

缓冲区缓存

存储最常访问的数据的内存区域。

业务能力

企业如何创造价值（例如，销售、客户服务或营销）。微服务架构和开发决策可以由业务能力驱动。有关更多信息，请参阅在 AWS上运行容器化微服务白皮书中的围绕业务能力进行组织部分。

业务连续性计划（BCP）

一项计划，旨在应对大规模迁移等破坏性事件对运营的潜在影响，并使企业能够快速恢复运营。

C

CAF

参见AWS 云采用框架。

金丝雀部署

向最终用户缓慢而渐进地发布版本。当你有信心时，你可以部署新版本并全部替换当前版本。

CCoE

参见云卓越中心。

CDC

请参阅变更数据捕获。

更改数据捕获（CDC）

跟踪数据来源（如数据库表）的更改并记录有关更改的元数据的过程。您可以将 CDC 用于各种目的，例如审计或复制目标系统中的更改以保持同步。

混沌工程

故意引入故障或破坏性事件来测试系统的弹性。您可以使用 AWS Fault Injection Service (AWS FIS) 来执行实验，对您的 AWS 工作负载施加压力并评估其响应。

CI/CD

查看持续集成和持续交付。

分类

一种有助于生成预测的分类流程。分类问题的 ML 模型预测离散值。离散值始终彼此不同。例如，一个模型可能需要评估图像中是否有汽车。

客户端加密

在目标 AWS 服务 收到数据之前，对数据进行本地加密。

云卓越中心 (CCoE)

一个多学科团队，负责推动整个组织的云采用工作，包括开发云最佳实践、调动资源、制定迁移时间表、领导组织完成大规模转型。有关更多信息，请参阅 AWS Cloud 企业战略博客上的 CCoE 帖子。

云计算

通常用于远程数据存储和 IoT 设备管理的云技术。云计算通常与边缘计算技术相关。

云运营模型

在 IT 组织中，一种用于构建、完善和优化一个或多个云环境的运营模型。有关更多信息，请参阅构建您的云运营模型。

云采用阶段

组织迁移到以下阶段时通常会经历四个阶段 AWS Cloud：

- 项目 - 出于概念验证和学习目的，开展一些与云相关的项目
- 基础 — 进行基础投资以扩大云采用率（例如，创建着陆区、定义 CCo E、建立运营模型）
- 迁移 - 迁移单个应用程序
- 重塑 - 优化产品和服务，在云中创新

Stephen Orban在 AWS Cloud 企业战略博客的博客文章《云优先之旅和采用阶段》中定义了这些阶段。有关它们与 AWS 迁移策略的关系的信息，请参阅迁移准备指南。

CMDB

参见配置管理数据库。

代码存储库

通过版本控制过程存储和更新源代码和其他资产（如文档、示例和脚本）的位置。常见的云存储库包括GitHub或Bitbucket Cloud。每个版本的代码都称为一个分支。在微服务结构中，每个存储库都专门用于一个功能。单个 CI/CD 管道可以使用多个存储库。

冷缓存

一种空的、填充不足或包含过时或不相关数据的缓冲区缓存。这会影响性能，因为数据库实例必须从主内存或磁盘读取，这比从缓冲区缓存读取要慢。

冷数据

很少访问的数据，且通常是历史数据。查询此类数据时，通常可以接受慢速查询。将这些数据转移到性能较低且成本更低的存储层或类别可以降低成本。

计算机视觉 (CV)

人工智能领域，使用机器学习来分析和提取数字图像和视频等视觉格式的信息。例如，Amazon SageMaker AI 为 CV 提供了图像处理算法。

配置偏差

对于工作负载，配置会从预期状态发生变化。这可能会导致工作负载变得不合规，而且通常是渐进的，不是故意的。

配置管理数据库（CMDB）

一种存储库，用于存储和管理有关数据库及其 IT 环境的信息，包括硬件和软件组件及其配置。您通常在迁移的产品组合发现和分析阶段使用来自 CMDB 的数据。

合规性包

一系列 AWS Config 规则和补救措施，您可以汇编这些规则和补救措施，以自定义合规性和安全性检查。您可以使用 YAML 模板将一致性包作为单个实体部署在 AWS 账户 和区域或整个组织中。有关更多信息，请参阅 AWS Config 文档中的一致性包。

持续集成和持续交付（CI/CD）

自动执行软件发布过程的源代码、构建、测试、暂存和生产阶段的过程。 CI/CD is commonly described as a pipeline. CI/CD可以帮助您实现流程自动化、提高生产力、提高代码质量和更快地交付。有关更多信息，请参阅持续交付的优势。CD 也可以表示持续部署。有关更多信息，请参阅持续交付与持续部署。

CV

参见计算机视觉。

D

静态数据

网络中静止的数据，例如存储中的数据。

数据分类

根据网络中数据的关键性和敏感性对其进行识别和分类的过程。它是任何网络安全风险管理策略的关键组成部分，因为它可以帮助您确定对数据的适当保护和保留控制。数据分类是 Well-Architecte AWS d Framework 中安全支柱的一个组成部分。有关详细信息，请参阅数据分类。

数据漂移

生产数据与用来训练机器学习模型的数据之间的有意义差异，或者输入数据随时间推移的有意义变化。数据漂移可能降低机器学习模型预测的整体质量、准确性和公平性。

传输中数据

在网络中主动移动的数据，例如在网络资源之间移动的数据。

数据网格

一种架构框架，可提供分布式、去中心化的数据所有权以及集中式管理和治理。

数据最少化

仅收集并处理绝对必要数据的原则。在中进行数据最小化 AWS Cloud 可以降低隐私风险、成本和分析碳足迹。

数据边界

AWS 环境中的一组预防性防护措施，可帮助确保只有可信身份才能访问来自预期网络的可信资源。有关更多信息，请参阅在上构建数据边界。 AWS

数据预处理

将原始数据转换为 ML 模型易于解析的格式。预处理数据可能意味着删除某些列或行，并处理缺失、不一致或重复的值。

数据溯源

在数据的整个生命周期跟踪其来源和历史的过程，例如数据如何生成、传输和存储。

数据主体

正在收集和处理其数据的个人。

数据仓库

一种支持商业智能（例如分析）的数据管理系统。数据仓库通常包含大量历史数据，通常用于查询和分析。

数据库定义语言（DDL）

在数据库中创建或修改表和对象结构的语句或命令。

数据库操作语言（DML）

在数据库中修改（插入、更新和删除）信息的语句或命令。

DDL

参见数据库定义语言。

深度融合

组合多个深度学习模型进行预测。您可以使用深度融合来获得更准确的预测或估算预测中的不确定性。

深度学习

一个 ML 子字段使用多层人工神经网络来识别输入数据和感兴趣的目标变量之间的映射。

defense-in-depth

一种信息安全方法，经过深思熟虑，在整个计算机网络中分层实施一系列安全机制和控制措施，以保护网络及其中数据的机密性、完整性和可用性。当你采用这种策略时 AWS，你会在 AWS

Organizations 结构的不同层面添加多个控件来帮助保护资源。例如，一种 defense-in-depth方法可以结合多因素身份验证、网络分段和加密。

委托管理员

在中 AWS Organizations，兼容的服务可以注册 AWS 成员帐户来管理组织的帐户并管理该服务的权限。此账户被称为该服务的委托管理员。有关更多信息和兼容服务列表，请参阅 AWS Organizations 文档中使用 AWS Organizations的服务。

后

使应用程序、新功能或代码修复在目标环境中可用的过程。部署涉及在代码库中实现更改，然后在应用程序的环境中构建和运行该代码库。

开发环境

参见环境。

侦测性控制

一种安全控制，在事件发生后进行检测、记录日志和发出警报。这些控制是第二道防线，提醒您注意绕过现有预防性控制的安全事件。有关更多信息，请参阅在 AWS上实施安全控制中的侦测性控制。

开发价值流映射 (DVSM)

用于识别对软件开发生命周期中的速度和质量产生不利影响的限制因素并确定其优先级的流程。DVSM 扩展了最初为精益生产实践设计的价值流映射流程。其重点关注在软件开发过程中创造和转移价值所需的步骤和团队。

数字孪生

真实世界系统的虚拟再现，如建筑物、工厂、工业设备或生产线。数字孪生支持预测性维护、远程监控和生产优化。

维度表

在星型架构中，一种较小的表，其中包含事实表中有关定量数据的数据属性。维度表属性通常是文本字段或行为类似于文本的离散数字。这些属性通常用于查询约束、筛选和结果集标注。

灾难

阻止工作负载或系统在其主要部署位置实现其业务目标的事件。这些事件可能是自然灾害、技术故障或人为操作的结果，例如无意的配置错误或恶意软件攻击。

灾难恢复 (DR)

您用来最大限度地减少灾难造成的停机时间和数据丢失的策略和流程。有关更多信息，请参阅 Well-Architected Fr ame AWS work 中的 “工作负载灾难恢复：云端 AWS 恢复”。

DML

参见数据库操作语言。

领域驱动设计

一种开发复杂软件系统的方法，通过将其组件连接到每个组件所服务的不断发展的领域或核心业务目标。Eric Evans 在其著作领域驱动设计：软件核心复杂性应对之道（Boston: Addison-Wesley Professional, 2003）中介绍了这一概念。有关如何将领域驱动设计与 strangler fig 模式结合使用的信息，请参阅使用容器和 Amazon API Gateway 逐步将原有的 Microsoft ASP.NET（ASMX）Web 服务现代化。

DR

参见灾难恢复。

漂移检测

跟踪与基准配置的偏差。例如，您可以使用 AWS CloudFormation 来检测系统资源中的偏差，也可以使用 AWS Control Tower 来检测着陆区中可能影响监管要求合规性的变化。

DVSM

参见开发价值流映射。

E

EDA

参见探索性数据分析。

EDI

参见电子数据交换。

边缘计算

该技术可提高位于 IoT 网络边缘的智能设备的计算能力。与云计算相比，边缘计算可以减少通信延迟并缩短响应时间。

电子数据交换 (EDI)

组织之间自动交换业务文档。有关更多信息，请参阅什么是电子数据交换。

加密

一种将人类可读的纯文本数据转换为密文的计算过程。

加密密钥

由加密算法生成的随机位的加密字符串。密钥的长度可能有所不同，而且每个密钥都设计为不可预测且唯一。

字节顺序

字节在计算机内存中的存储顺序。大端序系统先存储最高有效字节。小端序系统先存储最低有效字节。

端点

参见服务端点。

端点服务

一种可以在虚拟私有云（VPC）中托管，与其他用户共享的服务。您可以使用其他 AWS 账户 或 AWS Identity and Access Management (IAM) 委托人创建终端节点服务， AWS PrivateLink 并向其授予权限。这些账户或主体可通过创建接口 VPC 端点来私密地连接到您的端点服务。有关更多信息，请参阅 Amazon Virtual Private Cloud（Amazon VPC）文档中的创建端点服务。

企业资源规划 (ERP)

一种自动化和管理企业关键业务流程（例如会计、MES 和项目管理）的系统。

信封加密

用另一个加密密钥对加密密钥进行加密的过程。有关更多信息，请参阅 AWS Key Management Service (AWS KMS) 文档中的信封加密。

环境

正在运行的应用程序的实例。以下是云计算中常见的环境类型：

- 开发环境 — 正在运行的应用程序的实例，只有负责维护应用程序的核心团队才能使用。开发环境用于测试更改，然后再将其提升到上层环境。这类环境有时称为测试环境。
- 下层环境 — 应用程序的所有开发环境，比如用于初始构建和测试的环境。

- 生产环境 — 最终用户可以访问的正在运行的应用程序的实例。在 CI/CD 管道中，生产环境是最后一个部署环境。
- 上层环境 — 除核心开发团队以外的用户可以访问的所有环境。这可能包括生产环境、预生产环境和用户验收测试环境。

epic

在敏捷方法学中，有助于组织工作和确定优先级的功能类别。epics 提供了对需求和实施任务的总体描述。例如， AWS CAF 安全史诗包括身份和访问管理、侦探控制、基础设施安全、数据保护和事件响应。有关 AWS 迁移策略中 epics 的更多信息，请参阅计划实施指南。

ERP

参见企业资源规划。

探索性数据分析（EDA）

分析数据集以了解其主要特征的过程。您收集或汇总数据，并进行初步调查，以发现模式、检测异常并检查假定情况。EDA 通过计算汇总统计数据和创建数据可视化得以执行。

F

事实表

星形架构中的中心表。它存储有关业务运营的定量数据。通常，事实表包含两种类型的列：包含度量的列和包含维度表外键的列。

失败得很快

一种使用频繁和增量测试来缩短开发生命周期的理念。这是敏捷方法的关键部分。

故障隔离边界

在中 AWS Cloud，诸如可用区 AWS 区域、控制平面或数据平面之类的边界，它限制了故障的影响并有助于提高工作负载的弹性。有关更多信息，请参阅AWS 故障隔离边界。

功能分支

参见分支。

特征

您用来进行预测的输入数据。例如，在制造环境中，特征可能是定期从生产线捕获的图像。

特征重要性

特征对于模型预测的重要性。这通常表示为数值分数，可以通过各种技术进行计算，例如 Shapley 加法解释（SHAP）和积分梯度。有关更多信息，请参阅使用机器学习模型的可解释性 AWS。

功能转换

为 ML 流程优化数据，包括使用其他来源丰富数据、扩展值或从单个数据字段中提取多组信息。这使得 ML 模型能从数据中获益。例如，如果您将“2021-05-27 00:15:37”日期分解为“2021”、“五月”、“星期四”和“15”，则可以帮助学习与不同数据成分相关的算法学习精细模式。

少量提示

在要求法学硕士执行类似任务之前，向其提供少量示例，以演示该任务和所需的输出。这种技术是情境学习的应用，模型可以从提示中嵌入的示例（镜头）中学习。对于需要特定格式、推理或领域知识的任务，Few-shot 提示可能非常有效。另请参见零镜头提示。

FGAC

请参阅精细的访问控制。

精细访问控制 (FGAC)

使用多个条件允许或拒绝访问请求。

快闪迁移

一种数据库迁移方法，它使用连续的数据复制，通过更改数据捕获在尽可能短的时间内迁移数据，而不是使用分阶段的方法。目标是将停机时间降至最低。

FM

参见基础模型。

基础模型 (FM)

一个大型深度学习神经网络，一直在广义和未标记数据的大量数据集上进行训练。 FMs 能够执行各种各样的一般任务，例如理解语言、生成文本和图像以及用自然语言进行对话。有关更多信息，请参阅什么是基础模型。

G

生成式人工智能

人工智能模型的一个子集，这些模型已经过大量数据训练，可以使用简单的文本提示来创建新的内容和工件，例如图像、视频、文本和音频。有关更多信息，请参阅什么是生成式 AI。

地理封锁

请参阅地理限制。

地理限制（地理阻止）

在 Amazon 中 CloudFront，一种阻止特定国家/地区的用户访问内容分发的选项。您可以使用允许列表或阻止列表来指定已批准和已禁止的国家/地区。有关更多信息，请参阅 CloudFront 文档中的限制内容的地理分布。

GitFlow 工作流程

一种方法，在这种方法中，下层和上层环境在源代码存储库中使用不同的分支。Gitflow 工作流程被认为是传统的，而基于主干的工作流程是现代的首选方法。

金色影像

系统或软件的快照，用作部署该系统或软件的新实例的模板。例如，在制造业中，黄金映像可用于在多个设备上配置软件，并有助于提高设备制造运营的速度、可扩展性和生产力。

全新策略

在新环境中缺少现有基础设施。在对系统架构采用全新策略时，您可以选择所有新技术，而不受对现有基础设施（也称为棕地）兼容性的限制。如果您正在扩展现有基础设施，则可以将棕地策略和全新策略混合。

防护机制

一项高级规则，可帮助管理各组织单位的资源、策略和合规性 (OUs)。预防性防护机制会执行策略以确保符合合规性标准。它们是使用服务控制策略和 IAM 权限边界实现的。侦测性防护机制会检测策略违规和合规性问题，并生成警报以进行修复。它们通过使用 AWS Config、、Amazon、 AWS Security Hub GuardDuty AWS Trusted Advisor、Amazon Inspector 和自定义 AWS Lambda 支票来实现。

H

HA

参见高可用性。

异构数据库迁移

将源数据库迁移到使用不同数据库引擎的目标数据库（例如，从 Oracle 迁移到 Amazon Aurora）。异构迁移通常是重新架构工作的一部分，而转换架构可能是一项复杂的任务。AWS 提供了 AWS SCT 来帮助实现架构转换。

高可用性 (HA)

在遇到挑战或灾难时，工作负载无需干预即可连续运行的能力。HA 系统旨在自动进行故障转移、持续提供良好性能，并以最小的性能影响处理不同负载和故障。

历史数据库现代化

一种用于实现运营技术 (OT) 系统现代化和升级以更好满足制造业需求的方法。历史数据库是一种用于收集和存储工厂中各种来源数据的数据库。

抵制数据

从用于训练机器学习模型的数据集中扣留的一部分带有标签的历史数据。通过将模型预测与抵制数据进行比较，您可以使用抵制数据来评估模型性能。

同构数据库迁移

将源数据库迁移到共享同一数据库引擎的目标数据库（例如，从 Microsoft SQL Server 迁移到 Amazon RDS for SQL Server）。同构迁移通常是更换主机或更换平台工作的一部分。您可以使用本机数据库实用程序来迁移架构。

热数据

经常访问的数据，例如实时数据或近期的转化数据。这些数据通常需要高性能存储层或存储类别才能提供快速的查询响应。

修补程序

针对生产环境中关键问题的紧急修复。由于其紧迫性，修补程序通常是在典型的 DevOps 发布工作流程之外进行的。

hypercare 周期

割接之后，迁移团队立即管理和监控云中迁移的应用程序以解决任何问题的时间段。通常，这个周期持续 1-4 天。在 hypercare 周期结束时，迁移团队通常会将应用程序的责任移交给云运营团队。

我

IaC

参见基础设施即代码。

基于身份的策略

附加到一个或多个 IAM 委托人的策略，用于定义他们在 AWS Cloud 环境中的权限。

空闲应用程序

90 天内平均 CPU 和内存使用率在 5% 到 20% 之间的应用程序。在迁移项目中，通常会停用这些应用程序或将其保留在本地。

IIoT

参见工业物联网。

不可变的基础架构

一种为生产工作负载部署新基础架构，而不是更新、修补或修改现有基础架构的模型。不可变基础架构本质上比可变基础架构更一致、更可靠、更可预测。有关更多信息，请参阅 Well-Architected Framework 中的使用不可变基础架构 AWS 部署最佳实践。

入站（入口）VPC

在 AWS 多账户架构中，一种接受、检查和路由来自应用程序外部的网络连接的 VPC。AWS 安全参考架构建议设置您的网络帐户，包括入站、出站和检查， VPCs 以保护您的应用程序与更广泛的互联网之间的双向接口。

增量迁移

一种割接策略，在这种策略中，您可以将应用程序分成小部分进行迁移，而不是一次性完整割接。例如，您最初可能只将几个微服务或用户迁移到新系统。在确认一切正常后，您可以逐步迁移其他微服务或用户，直到停用遗留系统。这种策略降低了大规模迁移带来的风险。

工业 4.0

该术语由克劳斯·施瓦布（Klaus Schwab）于2016年推出，指的是通过连接、实时数据、自动化、分析和人工智能/机器学习的进步实现制造流程的现代化。

基础设施

应用程序环境中包含的所有资源和资产。

基础设施即代码（IaC）

通过一组配置文件预置和管理应用程序基础设施的过程。IaC 旨在帮助您集中管理基础设施、实现资源标准化和快速扩展，使新环境具有可重复性、可靠性和一致性。

工业物联网 (IIoT)

在工业领域使用联网的传感器和设备，例如制造业、能源、汽车、医疗保健、生命科学和农业。有关更多信息，请参阅制定工业物联网 (IIoT) 数字化转型战略。

检查 VPC

在 AWS 多账户架构中，一种集中式 VPC，用于管理对 VPCs（相同或不同 AWS 区域）、互联网和本地网络之间的网络流量的检查。AWS 安全参考架构建议设置您的网络帐户，包括入站、出站和检查， VPCs 以保护您的应用程序与更广泛的互联网之间的双向接口。

物联网（IoT）

由带有嵌入式传感器或处理器的连接物理对象组成的网络，这些传感器或处理器通过互联网或本地通信网络与其他设备和系统进行通信。有关更多信息，请参阅什么是 IoT？

可解释性

它是机器学习模型的一种特征，描述了人类可以理解模型的预测如何取决于其输入的程度。有关更多信息，请参阅使用机器学习模型的可解释性 AWS。

IoT

参见物联网。

IT 信息库（ITIL）

提供 IT 服务并使这些服务符合业务要求的一套最佳实践。ITIL 是 ITSM 的基础。

IT 服务管理（ITSM）

为组织设计、实施、管理和支持 IT 服务的相关活动。有关将云运营与 ITSM 工具集成的信息，请参阅运营集成指南。

ITIL

请参阅 IT 信息库。

ITSM

请参阅 IT 服务管理。

L

基于标签的访问控制 (LBAC)

强制访问控制 (MAC) 的一种实施方式，其中明确为用户和数据本身分配了安全标签值。用户安全标签和数据安全标签之间的交集决定了用户可以看到哪些行和列。

登录区

landing zone 是一个架构精良的多账户 AWS 环境，具有可扩展性和安全性。这是一个起点，您的组织可以从这里放心地在安全和基础设施环境中快速启动和部署工作负载和应用程序。有关登录区的更多信息，请参阅设置安全且可扩展的多账户 AWS 环境。

大型语言模型 (LLM)

一种基于大量数据进行预训练的深度学习 AI 模型。法学硕士可以执行多项任务，例如回答问题、总结文档、将文本翻译成其他语言以及完成句子。有关更多信息，请参阅什么是 LLMs。

大规模迁移

迁移 300 台或更多服务器。

LBAC

请参阅基于标签的访问控制。

最低权限

授予执行任务所需的最低权限的最佳安全实践。有关更多信息，请参阅 IAM 文档中的应用最低权限许可。

直接迁移

见 7 R。

小端序系统

一个先存储最低有效字节的系统。另请参见字节顺序。

LLM

参见大型语言模型。

下层环境

参见环境。

M

机器学习（ML）

一种使用算法和技术进行模式识别和学习的人工智能。ML 对记录的数据（例如物联网（IoT）数据）进行分析和学习，以生成基于模式的统计模型。有关更多信息，请参阅机器学习。

主分支

参见分支。

恶意软件

旨在危害计算机安全或隐私的软件。恶意软件可能会破坏计算机系统、泄露敏感信息或获得未经授权的访问。恶意软件的示例包括病毒、蠕虫、勒索软件、特洛伊木马、间谍软件和键盘记录器。

托管服务

AWS 服务 它 AWS 运行基础设施层、操作系统和平台，您可以访问端点来存储和检索数据。亚马逊简单存储服务 (Amazon S3) Service 和 Amazon DynamoDB 就是托管服务的示例。这些服务也称为抽象服务。

制造执行系统 (MES)

一种软件系统，用于跟踪、监控、记录和控制在车间将原材料转化为成品的生产过程。

MAP

参见迁移加速计划。

机制

一个完整的过程，在此过程中，您可以创建工具，推动工具的采用，然后检查结果以进行调整。机制是一种在运行过程中自我增强和改进的循环。有关更多信息，请参阅在 Well-Architect AWS ed 框架中构建机制。

成员账户

AWS 账户 除属于组织中的管理账户之外的所有账户 AWS Organizations。一个账户一次只能是一个组织的成员。

MES

参见制造执行系统。

消息队列遥测传输 (MQTT)

一种基于发布/订阅模式的轻量级 machine-to-machine (M2M) 通信协议，适用于资源受限的物联网设备。

微服务

一种小型的独立服务，通过明确的定义进行通信 APIs ，通常由小型的独立团队拥有。例如，保险系统可能包括映射到业务能力（如销售或营销）或子域（如购买、理赔或分析）的微服务。微服务

的好处包括敏捷、灵活扩展、易于部署、可重复使用的代码和恢复能力。有关更多信息，请参阅使用 AWS 无服务器服务集成微服务。

微服务架构

一种使用独立组件构建应用程序的方法，这些组件将每个应用程序进程作为微服务运行。这些微服务使用轻量级通过定义明确的接口进行通信。 APIs该架构中的每个微服务都可以更新、部署和扩展，以满足对应用程序特定功能的需求。有关更多信息，请参阅在上实现微服务。 AWS

迁移加速计划（MAP）

AWS 该计划提供咨询支持、培训和服务，以帮助组织为迁移到云奠定坚实的运营基础，并帮助抵消迁移的初始成本。MAP 提供了一种以系统的方式执行遗留迁移的迁移方法，以及一套用于自动执行和加速常见迁移场景的工具。

大规模迁移

将大部分应用程序组合分波迁移到云中的过程，在每一波中以更快的速度迁移更多应用程序。本阶段使用从早期阶段获得的最佳实践和经验教训，实施由团队、工具和流程组成的迁移工厂，通过自动化和敏捷交付简化工作负载的迁移。这是 AWS 迁移策略的第三阶段。

迁移工厂

跨职能团队，通过自动化、敏捷的方法简化工作负载迁移。迁移工厂团队通常包括运营、业务分析师和所有者、迁移工程师、开发 DevOps 人员和冲刺专业人员。20% 到 50% 的企业应用程序组合由可通过工厂方法优化的重复模式组成。有关更多信息，请参阅本内容集中有关迁移工厂的讨论和云迁移工厂指南。

迁移元数据

有关完成迁移所需的应用程序和服务器的信息。每种迁移模式都需要一套不同的迁移元数据。迁移元数据的示例包括目标子网、安全组和 AWS 账户。

迁移模式

一种可重复的迁移任务，详细列出了迁移策略、迁移目标以及所使用的迁移应用程序或服务。示例： EC2 使用 AWS 应用程序迁移服务重新托管向 Amazon 的迁移。

迁移组合评测（MPA）

一种在线工具，可提供信息，用于验证迁移到的业务案例。 AWS Cloud MPA 提供了详细的组合评测（服务器规模调整、定价、TCO 比较、迁移成本分析）以及迁移计划（应用程序数据分析和数据收集、应用程序分组、迁移优先级排序和波次规划）。所有 AWS 顾问和 APN 合作伙伴顾问均可免费使用 MPA 工具（需要登录）。

迁移准备情况评测（MRA）

使用 AWS CAF 深入了解组织的云就绪状态、确定优势和劣势以及制定行动计划以缩小已发现差距的过程。有关更多信息，请参阅迁移准备指南。MRA 是 AWS 迁移策略的第一阶段。

迁移策略

用于将工作负载迁移到的方法 AWS Cloud。有关更多信息，请参阅此词汇表中的 7 R 条目和动员组织以加快大规模迁移。

ML

参见机器学习。

现代化

将过时的（原有的或单体）应用程序及其基础设施转变为云中敏捷、弹性和高度可用的系统，以降低成本、提高效率和利用创新。有关更多信息，请参阅中的应用程序现代化策略。 AWS Cloud

现代化准备情况评估

一种评估方式，有助于确定组织应用程序的现代化准备情况；确定收益、风险和依赖关系；确定组织能够在多大程度上支持这些应用程序的未来状态。评估结果是目标架构的蓝图、详细说明现代化进程发展阶段和里程碑的路线图以及解决已发现差距的行动计划。有关更多信息，请参阅中的评估应用程序的现代化准备情况 AWS Cloud。

单体应用程序（单体式）

作为具有紧密耦合进程的单个服务运行的应用程序。单体应用程序有几个缺点。如果某个应用程序功能的需求激增，则必须扩展整个架构。随着代码库的增长，添加或改进单体应用程序的功能也会变得更加复杂。若要解决这些问题，可以使用微服务架构。有关更多信息，请参阅将单体分解为微服务。

MPA

参见迁移组合评估。

MQTT

请参阅消息队列遥测传输。

多分类器

一种帮助为多个类别生成预测（预测两个以上结果之一）的过程。例如，ML 模型可能会询问“这个产品是书、汽车还是手机？”或“此客户最感兴趣什么类别的产品？”

可变基础架构

一种用于更新和修改现有生产工作负载基础架构的模型。为了提高一致性、可靠性和可预测性，Well-Architect AWS ed Framework 建议使用不可变基础设施作为最佳实践。

O

OAC

请参阅源站访问控制。

OAI

参见源访问身份。

OCM

参见组织变更管理。

离线迁移

一种迁移方法，在这种方法中，源工作负载会在迁移过程中停止运行。这种方法会延长停机时间，通常用于小型非关键工作负载。

OI

参见运营集成。

OLA

参见运营层协议。

在线迁移

一种迁移方法，在这种方法中，源工作负载无需离线即可复制到目标系统。在迁移过程中，连接工作负载的应用程序可以继续运行。这种方法的停机时间为零或最短，通常用于关键生产工作负载。

OPC-UA

参见开放流程通信-统一架构。

开放流程通信-统一架构 (OPC-UA)

一种用于工业自动化的 machine-to-machine（M2M）通信协议。OPC-UA 提供了数据加密、身份验证和授权方案的互操作性标准。

运营级别协议（OLA）

一项协议，阐明了 IT 职能部门承诺相互交付的内容，以支持服务水平协议（SLA）。

运营准备情况审查 (ORR)

一份问题清单和相关的最佳实践，可帮助您理解、评估、预防或缩小事件和可能的故障的范围。有关更多信息，请参阅 Well-Architecte AWS d Frame work 中的运营准备情况评估 (ORR)。

操作技术 (OT)

与物理环境配合使用以控制工业运营、设备和基础设施的硬件和软件系统。在制造业中，OT 和信息技术 (IT) 系统的集成是工业 4.0 转型的重点。

运营整合（OI）

在云中实现运营现代化的过程，包括就绪计划、自动化和集成。有关更多信息，请参阅运营整合指南。

组织跟踪

由此创建的跟踪 AWS CloudTrail ，用于记录组织 AWS 账户 中所有人的所有事件 AWS Organizations。该跟踪是在每个 AWS 账户 中创建的，属于组织的一部分，并跟踪每个账户的活动。有关更多信息，请参阅 CloudTrail文档中的为组织创建跟踪。

组织变革管理（OCM）

一个从人员、文化和领导力角度管理重大、颠覆性业务转型的框架。OCM 通过加快变革采用、解决过渡问题以及推动文化和组织变革，帮助组织为新系统和战略做好准备和过渡。在 AWS 迁移策略中，该框架被称为人员加速，因为云采用项目需要变更的速度。有关更多信息，请参阅 OCM 指南。

来源访问控制（OAC）

在中 CloudFront，一个增强的选项，用于限制访问以保护您的亚马逊简单存储服务 (Amazon S3) 内容。OAC 全部支持所有 S3 存储桶 AWS 区域、使用 AWS KMS (SSE-KMS) 进行服务器端加密，以及对 S3 存储桶的动态`PUT`和`DELETE`请求。

来源访问身份（OAI）

在中 CloudFront，一个用于限制访问权限以保护您的 Amazon S3 内容的选项。当您使用 OAI 时，CloudFront 会创建一个 Amazon S3 可以对其进行身份验证的委托人。经过身份验证的委托人只能通过特定 CloudFront 分配访问 S3 存储桶中的内容。另请参阅 OAC，其中提供了更精细和增强的访问控制。

ORR

参见运营准备情况审查。

OT

参见运营技术。

出站（出口）VPC

在 AWS 多账户架构中，一种处理从应用程序内部启动的网络连接的 VPC。AWS 安全参考架构建议设置您的网络帐户，包括入站、出站和检查， VPCs 以保护您的应用程序与更广泛的互联网之间的双向接口。

P

权限边界

附加到 IAM 主体的 IAM 管理策略，用于设置用户或角色可以拥有的最大权限。有关更多信息，请参阅 IAM 文档中的权限边界。

个人身份信息 (PII)

直接查看其他相关数据或与之配对时可用于合理推断个人身份的信息。PII 的示例包括姓名、地址和联系信息。

PII

查看个人身份信息。

playbook

一套预定义的步骤，用于捕获与迁移相关的工作，例如在云中交付核心运营功能。playbook 可以采用脚本、自动化运行手册的形式，也可以是操作现代化环境所需的流程或步骤的摘要。

PLC

参见可编程逻辑控制器。

PLM

参见产品生命周期管理。

policy

一个对象，可以在中定义权限（参见基于身份的策略）、指定访问条件（参见基于资源的策略）或定义组织中所有账户的最大权限 AWS Organizations （参见服务控制策略）。

多语言持久性

根据数据访问模式和其他要求，独立选择微服务的数据存储技术。如果您的微服务采用相同的数据存储技术，它们可能会遇到实现难题或性能不佳。如果微服务使用最适合其需求的数据存储，则可以更轻松地实现微服务，并获得更好的性能和可扩展性。有关更多信息，请参阅在微服务中实现数据持久性。

组合评测

一个发现、分析和确定应用程序组合优先级以规划迁移的过程。有关更多信息，请参阅评估迁移准备情况。

谓词

返回`true`或的查询条件`false`，通常位于子`WHERE`句中。

谓词下推

一种数据库查询优化技术，可在传输前筛选查询中的数据。这减少了必须从关系数据库检索和处理的数据量，并提高了查询性能。

预防性控制

一种安全控制，旨在防止事件发生。这些控制是第一道防线，帮助防止未经授权的访问或对网络的意外更改。有关更多信息，请参阅在 AWS上实施安全控制中的预防性控制。

主体

中 AWS 可以执行操作和访问资源的实体。此实体通常是 IAM 角色的根用户或用户。 AWS 账户有关更多信息，请参阅 IAM 文档中角色术语和概念中的主体。

通过设计保护隐私

一种在整个开发过程中考虑隐私的系统工程方法。

私有托管区

一个容器，其中包含有关您希望 Amazon Route 53 如何响应针对一个或多个 VPCs域名及其子域名的 DNS 查询的信息。有关更多信息，请参阅 Route 53 文档中的私有托管区的使用。

主动控制

一种安全控制措施，旨在防止部署不合规的资源。这些控件会在资源配置之前对其进行扫描。如果资源与控件不兼容，则不会对其进行配置。有关更多信息，请参阅 AWS Control Tower 文档中的控制参考指南，并参见在上实施安全控制中的主动控制 AWS。

产品生命周期管理 (PLM)

在产品的整个生命周期中，从设计、开发和上市，到成长和成熟，再到衰落和移除，对产品进行数据和流程的管理。

生产环境

参见环境。

可编程逻辑控制器 (PLC)

在制造业中，一种高度可靠、适应性强的计算机，用于监控机器并实现制造过程自动化。

提示链接

使用一个 LLM 提示的输出作为下一个提示的输入，以生成更好的响应。该技术用于将复杂的任务分解为子任务，或者迭代地完善或扩展初步响应。它有助于提高模型响应的准确性和相关性，并允许获得更精细的个性化结果。

假名化

用占位符值替换数据集中个人标识符的过程。假名化可以帮助保护个人隐私。假名化数据仍被视为个人数据。

publish/subscribe (pub/sub)

一种支持微服务间异步通信的模式，以提高可扩展性和响应能力。例如，在基于微服务的 MES 中，微服务可以将事件消息发布到其他微服务可以订阅的频道。系统可以在不更改发布服务的情况下添加新的微服务。

Q

查询计划

一系列步骤，例如指令，用于访问 SQL 关系数据库系统中的数据。

查询计划回归

当数据库服务优化程序选择的最佳计划不如数据库环境发生特定变化之前时。这可能是由统计数据、约束、环境设置、查询参数绑定更改和数据库引擎更新造成的。

R

RACI 矩阵

参见 “负责任、负责、咨询、知情”（RACI）。

RAG

请参见检索增强生成。

勒索软件

一种恶意软件，旨在阻止对计算机系统或数据的访问，直到付款为止。

RASCI 矩阵

参见 “负责任、负责、咨询、知情”（RACI）。

RCAC

请参阅行和列访问控制。

只读副本

用于只读目的的数据库副本。您可以将查询路由到只读副本，以减轻主数据库的负载。

重新架构师

见 7 R。

恢复点目标 (RPO)

自上一个数据恢复点以来可接受的最长时间。这决定了从上一个恢复点到服务中断之间可接受的数据丢失情况。

恢复时间目标 (RTO)

服务中断和服务恢复之间可接受的最大延迟。

重构

见 7 R。

区域

地理区域内的 AWS 资源集合。每一个 AWS 区域 都相互隔离，彼此独立，以提供容错、稳定性和弹性。有关更多信息，请参阅指定 AWS 区域 您的账户可以使用的账户。

回归

一种预测数值的 ML 技术。例如，要解决“这套房子的售价是多少？”的问题 ML 模型可以使用线性回归模型，根据房屋的已知事实（如建筑面积）来预测房屋的销售价格。

重新托管

见 7 R。

版本

在部署过程中，推动生产环境变更的行为。

搬迁

见 7 R。

更换平台

见 7 R。

回购

见 7 R。

故障恢复能力

应用程序抵御中断或从中断中恢复的能力。在中规划弹性时，高可用性和灾难恢复是常见的考虑因素。 AWS Cloud有关更多信息，请参阅AWS Cloud 弹性。

基于资源的策略

一种附加到资源的策略，例如 AmazonS3 存储桶、端点或加密密钥。此类策略指定了允许哪些主体访问、支持的操作以及必须满足的任何其他条件。

责任、问责、咨询和知情（RACI）矩阵

定义参与迁移活动和云运营的所有各方的角色和责任的矩阵。矩阵名称源自矩阵中定义的责任类型：负责 (R)、问责 (A)、咨询 (C) 和知情 (I)。支持 (S) 类型是可选的。如果包括支持，则该矩阵称为 RASCI 矩阵，如果将其排除在外，则称为 RACI 矩阵。

响应性控制

一种安全控制，旨在推动对不良事件或偏离安全基线的情况进行修复。有关更多信息，请参阅在AWS上实施安全控制中的响应性控制。

保留

见 7 R。

退休

见 7 R。

检索增强生成（RAG）

一种生成式人工智能技术，其中法学硕士在生成响应之前引用其训练数据源之外的权威数据源。例如，RAG 模型可以对组织的知识库或自定义数据执行语义搜索。有关更多信息，请参阅什么是 RAG。

轮换

定期更新密钥以使攻击者更难访问凭据的过程。

行列访问控制 (RCAC)

使用已定义访问规则的基本、灵活的 SQL 表达式。RCAC 由行权限和列掩码组成。

RPO

参见恢复点目标。

RTO

参见恢复时间目标。

运行手册

执行特定任务所需的一套手动或自动程序。它们通常是为了简化重复性操作或高错误率的程序而设计的。

S

SAML 2.0

许多身份提供商 (IdPs) 使用的开放标准。此功能支持联合单点登录 (SSO)，因此用户无需在 IAM 中为组织中的所有人创建用户即可登录 AWS Management Console 或调用 AWS API 操作。有关基于 SAML 2.0 的联合身份验证的更多信息，请参阅 IAM 文档中的关于基于 SAML 2.0 的联合身份验证。

SCADA

参见监督控制和数据采集。

SCP

参见服务控制政策。

secret

在中 AWS Secrets Manager，您以加密形式存储的机密或受限信息，例如密码或用户凭证。它由密钥值及其元数据组成。密钥值可以是二进制、单个字符串或多个字符串。有关更多信息，请参阅 Secret s Manager 密钥中有什么？ 在 Secrets Manager 文档中。

安全性源于设计

一种在整个开发过程中考虑安全性的系统工程方法。

安全控制

一种技术或管理防护机制，可防止、检测或降低威胁行为体利用安全漏洞的能力。安全控制主要有四种类型：预防性、侦测、响应式和主动式。

安全加固

缩小攻击面，使其更能抵御攻击的过程。这可能包括删除不再需要的资源、实施授予最低权限的最佳安全实践或停用配置文件中不必要的功能等操作。

安全信息和事件管理（SIEM）系统

结合了安全信息管理（SIM）和安全事件管理（SEM）系统的工具和服务。SIEM 系统会收集、监控和分析来自服务器、网络、设备和其他来源的数据，以检测威胁和安全漏洞，并生成警报。

安全响应自动化

一种预定义和编程的操作，旨在自动响应或修复安全事件。这些自动化可作为侦探或响应式安全控制措施，帮助您实施 AWS 安全最佳实践。自动响应操作的示例包括修改 VPC 安全组、修补 Amazon EC2 实例或轮换证书。

服务器端加密

在目的地对数据进行加密，由接收方 AWS 服务 进行加密。

服务控制策略（SCP）

一种策略，用于集中控制组织中所有账户的权限 AWS Organizations。 SCPs 定义防护措施或限制管理员可以委托给用户或角色的操作。您可以使用 SCPs 允许列表或拒绝列表来指定允许或禁止哪些服务或操作。有关更多信息，请参阅 AWS Organizations 文档中的服务控制策略。

服务端点

的入口点的 URL AWS 服务。您可以使用端点，通过编程方式连接到目标服务。有关更多信息，请参阅 AWS 一般参考 中的 AWS 服务 端点。

服务水平协议（SLA）

一份协议，阐明了 IT 团队承诺向客户交付的内容，比如服务正常运行时间和性能。

服务级别指示器 (SLI)

对服务性能方面的衡量，例如其错误率、可用性或吞吐量。

服务级别目标 (SLO)

代表服务运行状况的目标指标，由服务级别指标衡量。

责任共担模式

描述您在云安全与合规方面共同承担 AWS 的责任的模型。 AWS 负责云的安全，而您则负责云中的安全。有关更多信息，请参阅责任共担模式。

SIEM

参见安全信息和事件管理系统。

单点故障 (SPOF)

应用程序的单个关键组件出现故障，可能会中断系统。

SLA

参见服务级别协议。

SLI

参见服务级别指标。

SLO

参见服务级别目标。

split-and-seed 模型

一种扩展和加速现代化项目的模式。随着新功能和产品发布的定义，核心团队会拆分以创建新的产品团队。这有助于扩展组织的能力和服务，提高开发人员的工作效率，支持快速创新。有关更多信息，请参阅中的分阶段实现应用程序现代化的方法。 AWS Cloud

恶作剧

参见单点故障。

星型架构

一种数据库组织结构，它使用一个大型事实表来存储交易数据或测量数据，并使用一个或多个较小的维度表来存储数据属性。此结构专为在数据仓库中使用或用于商业智能目的而设计。

strangler fig 模式

一种通过逐步重写和替换系统功能直至可以停用原有的系统来实现单体系统现代化的方法。这种模式用无花果藤作为类比，这种藤蔓成长为一棵树，最终战胜并取代了宿主。该模式是由 Martin Fowler 提出的，作为重写单体系统时管理风险的一种方法。有关如何应用此模式的示例，请参阅使用容器和 Amazon API Gateway 逐步将原有的 Microsoft ASP.NET（ASMX）Web 服务现代化。

子网

您的 VPC 内的一个 IP 地址范围。子网必须位于单个可用区中。

监控和数据采集 (SCADA)

在制造业中，一种使用硬件和软件来监控有形资产和生产操作的系统。

对称加密

一种加密算法，它使用相同的密钥来加密和解密数据。

综合测试

以模拟用户交互的方式测试系统，以检测潜在问题或监控性能。你可以使用 Amazon S CloudWatch ynthetics 来创建这些测试。

系统提示符

一种向法学硕士提供上下文、说明或指导方针以指导其行为的技术。系统提示有助于设置上下文并制定与用户交互的规则。

T

tags

键值对，充当用于组织资源的元数据。 AWS 标签可帮助您管理、识别、组织、搜索和筛选资源。有关更多信息，请参阅标记您的 AWS 资源。

目标变量

您在监督式 ML 中尝试预测的值。这也被称为结果变量。例如，在制造环境中，目标变量可能是产品缺陷。

任务列表

一种通过运行手册用于跟踪进度的工具。任务列表包含运行手册的概述和要完成的常规任务列表。对于每项常规任务，它包括预计所需时间、所有者和进度。

测试环境

参见环境。

训练

为您的 ML 模型提供学习数据。训练数据必须包含正确答案。学习算法在训练数据中查找将输入数据属性映射到目标（您希望预测的答案）的模式。然后输出捕获这些模式的 ML 模型。然后，您可以使用 ML 模型对不知道目标的新数据进行预测。

中转网关

一个网络传输中心，可用于将您的网络 VPCs 和本地网络互连。有关更多信息，请参阅 AWS Transit Gateway 文档中的什么是公交网关。

基于中继的工作流程

一种方法，开发人员在功能分支中本地构建和测试功能，然后将这些更改合并到主分支中。然后，按顺序将主分支构建到开发、预生产和生产环境。

可信访问权限

向您指定的服务授予权限，该服务可代表您在其账户中执行任务。 AWS Organizations 当需要服务相关的角色时，受信任的服务会在每个账户中创建一个角色，为您执行管理任务。有关更多信息，请参阅 AWS Organizations 文档中的AWS Organizations 与其他 AWS 服务一起使用。

优化

更改训练过程的各个方面，以提高 ML 模型的准确性。例如，您可以通过生成标签集、添加标签，并在不同的设置下多次重复这些步骤来优化模型，从而训练 ML 模型。

双披萨团队

一个小 DevOps 团队，你可以用两个披萨来喂食。双披萨团队的规模可确保在软件开发过程中充分协作。

U

不确定性

这一概念指的是不精确、不完整或未知的信息，这些信息可能会破坏预测式 ML 模型的可靠性。不确定性有两种类型：认知不确定性是由有限的、不完整的数据造成的，而偶然不确定性是由数据中固有的噪声和随机性导致的。有关更多信息，请参阅量化深度学习系统中的不确定性指南。

无差别任务

也称为繁重工作，即创建和运行应用程序所必需的工作，但不能为最终用户提供直接价值或竞争优势。无差别任务的示例包括采购、维护和容量规划。

上层环境

参见环境。

V

vacuum 操作

一种数据库维护操作，包括在增量更新后进行清理，以回收存储空间并提高性能。

版本控制

跟踪更改的过程和工具，例如存储库中源代码的更改。

VPC 对等连接

两者之间的连接 VPCs ，允许您使用私有 IP 地址路由流量。有关更多信息，请参阅 Amazon VPC 文档中的什么是 VPC 对等连接。

漏洞

损害系统安全的软件缺陷或硬件缺陷。

W

热缓存

一种包含经常访问的当前相关数据的缓冲区缓存。数据库实例可以从缓冲区缓存读取，这比从主内存或磁盘读取要快。

暖数据

不常访问的数据。查询此类数据时，通常可以接受中速查询。

窗口函数

一个 SQL 函数，用于对一组以某种方式与当前记录相关的行进行计算。窗口函数对于处理任务很有用，例如计算移动平均线或根据当前行的相对位置访问行的值。

工作负载

一系列资源和代码，它们可以提供商业价值，如面向客户的应用程序或后端过程。

工作流

迁移项目中负责一组特定任务的职能小组。每个工作流都是独立的，但支持项目中的其他工作流。例如，组合工作流负责确定应用程序的优先级、波次规划和收集迁移元数据。组合工作流将这些资产交付给迁移工作流，然后迁移服务器和应用程序。

蠕虫

参见一次写入，多读。

WQF

参见AWS 工作负载资格框架。

一次写入，多次读取 (WORM)

一种存储模型，它可以一次写入数据并防止数据被删除或修改。授权用户可以根据需要多次读取数据，但他们无法对其进行更改。这种数据存储基础架构被认为是不可变的。

Z

零日漏洞利用

一种利用未修补漏洞的攻击，通常是恶意软件。

零日漏洞

生产系统中不可避免的缺陷或漏洞。威胁主体可能利用这种类型的漏洞攻击系统。开发人员经常因攻击而意识到该漏洞。

零镜头提示

向法学硕士提供执行任务的说明，但没有示例（镜头）可以帮助指导任务。法学硕士必须使用其预先训练的知识来处理任务。零镜头提示的有效性取决于任务的复杂性和提示的质量。另请参阅 few-shot 提示。

僵尸应用程序

平均 CPU 和内存使用率低于 5% 的应用程序。在迁移项目中，通常会停用这些应用程序。

本文属于机器翻译版本。若本译文内容与英语原文存在差异，则一律以英文原文为准。