Unable to locate subtitle

AWS Well-Architected Framework

AWS Well-Architected Framework: ***Unable to locate subtitle***

Table of Contents

摘要和介绍 1
引言 1
定义 2
关于架构 3
一般设计原则 4
框架的支柱 6
卓越运营 6
设计原则 6
定义 7
最佳实践 7
资源 14
安全性 14
设计原则 15
定义 15
最佳实践 16
资源 21
可靠性 22
设计原则 22
定义 23
最佳实践 23
资源 27
性能效率 28
设计原则 28
定义 28
最佳实践 29
资源 34
成本优化 35
设计原则 35
定义 36
最佳实践 36
资源 41
可持续性 42
设计原则 42
定义 43

最佳实践 43
审查流程 49
总结 51
贡献者 52
延伸阅读 53
文档修订 54
附录：问题和最佳实践 56
卓越运营 56
组织 56
准备 74
运营 115
演进 142
安全性 153
安全基础知识 153
身份与权限管控 161
检测 180
基础设施保护 187
数据保护 200
事件响应 214
可靠性 226
基础 226
工作负载架构 246
变更管理 268
故障管理 292
性能效率 361
选择 362
审核 430
监控 434
权衡 443
成本优化 451
践行云财务管理 451
支出和使用情况意识 466
具有成本效益的资源 484
管理需求和供应资源 502
随着时间的推移不断优化 506
可持续性 509

区域选择 509
用户行为模式 510
软件和架构模式 516
数据模式 521
硬件模式 527
开发和部署流程 531
声明 536

AWS Well-Architected Framework

发布日期： 2022 年 10 月 20 日 (文档修订)

AWS Well-Architected Framework 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、高效且经济实惠的系统的架构最佳实践。

引言

AWS Well-Architected Framework 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。使用该框架有助于您了解在 AWS Cloud 中设计和运行安全、可靠、高效且经济实惠的可持续工作负载的架构最佳实践。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。审查架构的流程是关于架构决策的建设性对话，不是一种审核机制。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。

AWS 解决方案架构师拥有多年为各种垂直行业和使用案例设计解决方案的经验。我们也已经帮助成千上万客户对其 AWS 之上的架构进行设计与审查。从这些经验中，我们得以总结出在云中设计系统的最佳实践和核心策略。

AWS Well-Architected Framework 囊括了一系列基础性问题，来帮助您了解某种架构是否符合云最佳实践。该框架为您提供了一种一致的方法，来对标您所期望的现代云端系统能力，建立一整套质量评估体系，以及评估实现这样的质量需要采取的具体措施。随着 AWS 不断发展，我们将继续与客户协作并增进了解，同时将实际经验融入到 Well-Architected 定义的持续完善当中。

此框架面向各类技术性角色，例如首席技术官 (CTO)、架构师、开发人员和运维团队成员。它介绍了可在设计和运行云工作负载时使用的 AWS 最佳实践和策略，提供了进一步实施细节和架构模式的链接。有关更多信息，请参阅 AWS Well-Architected 主页。

AWS 还提供可用于审查您的工作负载的免费服务。如示例所示， AWS Well-Architected Tool （AWS WA Tool）是一种云服务，它提供统一的流程，可帮助您使用 AWS Well-Architected Framework 对自己的架构进行审核和衡量。借助 AWS WA Tool 提供的建议，可让您的工作负载变得更加可靠、安全、高效和经济实惠。

为了帮助您应用最佳实践，我们创建了 AWS Well-Architected 实验室，它可以为您提供代码和文档的存储库，让您亲自体验最佳实践的实施。我们还与 AWS 合作伙伴网络（APN，AWS Partner Network）中的精选合作伙伴开展合作，他们是 AWS Well-Architected 合作伙伴计划的成员.这些 AWS 合作伙伴拥有丰富的 AWS 知识，可以帮助您审查并改进工作负载。

定义

AWS 的专家每天都在帮助客户设计系统，以利用云中的最佳实践。在设计过程中，我们与您一起对架构进行权衡调整。当您在真实环境中部署这些系统时，我们将关注这些系统的运作状况，同时衡量上述调整的效果。

依托于实践经验，我们构建了 AWS Well-Architected Framework，它为客户和合作伙伴评估架构提供了一系列最佳实践，并提供了相应的可用于评估架构是否符合 AWS 最佳实践的问题。

AWS Well-Architected Framework 建立在六个支柱的基础上，它们分别是卓越运营、安全性、可靠性、性能效率、成本优化和可持续性。

表 1.AWS Well-Architected Framework 的支柱

姓名	描述
卓越运营	能够有效地支持发展和运行工作负载，获取对运营的洞察，以及不断改进支持流程和程序以实现商业价值。
安全性	安全性支柱描述了如何利用云技术来保护数据、系统和资产，以改善您的安全状况。
可靠性	可靠性支柱涵盖相关工作负载按照计划正确而稳定执行其预期功能的能力。它包括在其全部生命周期内运行和测试工作负载的能力。本白皮书深度介绍了有关在 AWS 中实施可靠工作负载的最佳实践指导。
性能效率	有效利用计算资源来满足系统要求，并随着需求变化和技术发展保持这种效率的能力。
成本优化	以最低价格运行系统来交付商业价值的能力。
可持续性	这是指以下能力：通过最大限度地提高所预置资源的收益并最大限度地减少所需的总资源，降低能源消耗并提高工作负载的所有组件的效率，从而持续改善可持续性影响。

在 AWS Well-Architected Framework 中，我们使用了以下术语：

- 此处的 组件 是指针对相关需求提供的代码、配置和 AWS 资源的组合。组件通常是技术处理单元，与其他组件分离。
- 术语 工作负载 指的是共同提供商业价值的组件集合。工作负载通常是业务和技术领导者沟通的细节层次。
- 我们将 架构 定义为组件在工作负载中协同工作的方式。架构图的重点通常是组件如何通信和交互。
- Milestones 将随着架构在整个产品生命周期内（设计、实施、测试、上线和生产）的演进记录架构中的关键变更。
- 组织内的 技术产品组合 是业务运营所需的工作负载集合。
- 如示例所示， 工作量 用于对执行任务所需的时间、精力和复杂性进行分类。每个组织都需要考虑团队的规模和专业知识以及工作负载的复杂性，了解更多背景信息，以便对组织的工作量进行正确分类。
  - 高： 这项工作可能需要数周或数月。它可以分解为若干案例、发布和任务。
  - 中： 这项工作可能需要数天或数周。它可以分解为若干发布和任务。
  - 低： 这项工作可能需要数小时或数天。它可以分解为若干任务。

在设计工作负载时，您会基于您的业务环境在各个支柱之间做出权衡。这些业务决策可以确定设计优先事项。在开发环境中，您可能会进行优化，牺牲一部分可靠性来改进可持续性影响并降低成本；而对于任务关键型解决方案，您可能会在成本和可持续性影响方面做出妥协，来提高可靠性。在电子商务解决方案中，性能可能会影响收入和客户的购买偏好。对于安全性和卓越运营，一般不会在它们和其他支柱之间进行权衡。

关于架构

在本地环境中，客户通常有一个技术架构中心团队，来监督其他产品或功能团队，从而确保他们遵循最佳实践。技术架构团队通常包含一组角色，比如技术架构师（基础设施）、解决方案架构师（软件）、数据架构师、网络架构师和安全架构师。这些团队一般将 TOGAF 或 Zachman Framework 用作企业架构能力的一部分。

在 AWS，我们倾向于将能力分配到多个团队，而不是只让一个核心团队具有这种能力。当您选择分配决策权限时，会存在一定的风险，例如，确保团队达到内部标准。我们两种方法降低这些风险。第一，我们 有一些实践 （即行为方式、流程、标准和公认的规范），专注于让每个团队都具有这种能力，并且我们通过设置一些专家来确保团队不断提高他们需要满足的标准。第二，我们实施了 各种机制， 来自动执行检查，以确保满足各项标准。

> “徒有良好的心愿没有用，需要良好的机制来实现它们”– 杰夫·贝索斯（Jeff Bezos）。

这意味着用机制（通常是自动的）来替代人类工作，检查是否遵守了规则或流程。这种分布式方法由亚马逊的领导力原则提供支持，在所有角色中建立一种 从客户出发 的工作文化。逆向工作是我们的创新过程的基本组成部分。我们从客户和客户需求出发，定义和指导我们的工作。只有以客户为中心的团队才能开发出真正满足客户需求的产品。

对于架构，这意味着我们希望每个团队都有能力创建架构并遵循最佳实践。为了帮助新团队获得这些能力或帮助现有团队提高其标准，我们创建了一个由首席工程师组成的虚拟社群，这些工程师可以检查现有团队的设计，帮助他们了解 AWS 最佳实践。首席工程师社群旨在让您能够接触和了解最佳实践。例如，通过午间谈话交流如何将最佳实践应用到实例中。这些谈话会被记录下来，用作新团队成员入门材料的一部分。

AWS 最佳实践源于我们在互联网规模上运行成千上万个系统的经验。我们倾向于使用数据定义最佳实践，同时我们还通过首席工程师等主题专家来设定最佳实践。当首席工程师发现新的最佳实践时，他们将以社群的形式确保所有团队遵循这些最佳实践。同时，这些最佳实践还会被正式纳入我们的内部审查流程以及强制性合规机制中。架构完善的框架是面向客户实施我们的内部审查流程，其中将我们在不同领域角色（例如解决方案架构和内部工程团队）中的主要设计思维编制成文。架构完善的框架是一种可扩展的机制，使您能够有效利用现有的经验。

通过首席工程师在社群内分散架构责任的方法，我们相信设计良好的企业架构是由客户的需求驱动的，并且可以付诸实现。通过让技术主管（例如首席信息官或开发经理）针对所有工作负载执行良好架构审查，您能够更好地了解技术栈存在的风险。以此方法，您可以确定不同团队间可以使用的主题，通过机制、培训或午间谈话等方式，让首席工程师可以与多个团队分享他们在特定领域的想法。

一般设计原则

架构完善的框架（Well-Architected Framework）定义了一系列一般性设计原则，以促进良好的云端设计：

- 停止猜测您的容量需求：如果您在部署工作负载时作出糟糕的容量决策，结果常常造成昂贵的资源闲置或因容量不足而影响性能。利用云计算，这些问题都不复存在。您可以按需使用容量，并自动对容量规模进行扩缩。
- 以生产规模进行系统测试：在云中，您可以根据需要创建一套生产规模等级的测试环境，完成测试，然后停用资源。由于测试环境只需在运行时付费，您模拟真实环境的成本仅为本地测试成本的一小部分。

- 实现自动化，使架构试验变得更容易：通过自动化操作，您可以低成本创建和复制工作负载，避免人力支出。您可以跟踪自动化变更，审核所产生的影响，并在必要时恢复到以前的参数。
- 支持实现架构演进：在传统环境中，架构决策通常作为静态的一次性事件实现，在其生命周期内包含几个重要的系统版本。随着业务及其环境继续演进，这些初始决策可能无法适应不断变化的业务能力需求。在云中，自动化和按需测试能力将显著降低设计变更所带来影响的风险。这使系统能够随时间推移不断演进，以便企业能够不断地发展创新。
- 利用数据驱动架构：在云中，您可以收集有关您的架构选择如何影响工作负载表现的数据。这使您能够基于事实做出如何改进工作负载的决策。您的云基础设施以代码形式存在，因此您可以随着时间的推移，基于这些数据做出明智的架构选择和改进。
- 通过实际演练不断改进：通过定期安排实际演练来模拟生产中的各种事件，测试架构和流程的性能。这将帮助您了解可以从哪些方面作出改进，并有助于培养组织处理各种事件的经验。

框架的支柱

构建软件系统与建楼很像。如果基础不牢固，结构问题将会破坏整栋大楼的完整性和功能。在设计技术解决方案时，如果忽视卓越运营、安全性、可靠性、性能效率、成本优化和可持续性这六大支柱，就很难构建一个能够满足您的期望和需求的系统。通过把这些支柱整合到架构中，您将能构建稳定而高效的系统，这将使您能够专注于设计的其他方面，例如功能性需求。

要素

- 卓越运营
- 安全性
- 可靠性
- 性能效率
- 成本优化
- 可持续性

卓越运营

卓越运营支柱能够有效地支持发展和运行工作负载，获取对运营的洞察，以及不断改进支持流程和程序以实现商业价值。

卓越运营支柱概述了各种设计原则、最佳实践和问题。如需有关具体实施的说明性指导，请参阅《卓越运营支柱》白皮书.

主题

- 设计原则
- 定义
- 最佳实践
- 资源

设计原则

在云中实现卓越运营有五个设计原则：

- 执行运营即代码：在云中，您可以将用于应用程序代码的工程规范应用于整个环境。您可以将整个工作负载（应用程序、基础设施）定义为代码，并使用该代码进行更新。您可以将运营流程写成代码

（脚本），并通过事件触发来自动执行这些脚本。通过以代码形式执行操作，您可以减少人为错误并实现对事件的一致响应。

- 频繁进行可逆的小规模更改：将工作负载设计为支持组件定期更新。以较小增量进行失败时可逆的更改（尽可能不影响客户）。
- 经常优化运营流程：在使用运营程序时，要寻找机会改进它们。在改进工作负载的同时，您也要适当改进一下流程。设置定期的实际演练，以检查并验证所有流程是否有效，以及团队是否熟悉这些流程。
- 预测故障：执行“故障演练”，找出潜在的问题，以便消除和缓解问题。测试您的故障场景，并确认您了解相应影响。测试您的响应流程以确保它们有效，并确保团队能够熟练执行。设置定期的实际演练，以测试工作负载和团队对模拟事件的响应。
- 从所有运营故障中吸取经验教训：从所有运营事件和故障中吸取的经验教训，推动改进。在多个团队乃至组织范围中分享经验教训。

定义

在云中实现卓越运营有四个领域的最佳实践：

- 组织
- 准备
- 运营
- 演进

您的组织领导层负责定义业务目标。您的组织必须了解各种要求和重点，并利用它们来组织和开展工作，从而为获得业务成果提供支持。您的工作负载必须发出所需信息以提供支持。采用多种服务来支持工作负载的集成、部署和交付，这将通过自动化重复流程，增加对生产的有益更改。

工作负载的运营可能存在固有风险。您必须了解这些风险并做出明智的生产决策。您的团队必须能够支持您的工作负载。从预期业务成果中得出的业务和运营指标将使您能够了解工作负载的运行状况、运营活动以及对事件的响应。您的重点将随着您的业务需求和业务环境的变化而变化。将这些作为反馈循环，持续推动组织和工作负载运营的改进。

最佳实践

主题

- 组织

- 准备
- 运营
- 演进

组织

您的团队需要对整个工作负载、他们在其中的角色以及共同的业务目标有一致的理解，以便设置运营重点以实现业务成功。明确运营重点可以让您的工作效益最大化。评估内部和外部客户需求，让包括业务、开发和运营团队在内的主要利益相关方参与进来，以便确定工作重心。评估客户需求将确保您充分了解实现业务成果所需的支持。确保了解组织监管规定的指导原则或义务，以及监管合规性要求和行业标准等可能需要遵循或重视的外部因素。验证您是否具有确定内部监管和外部合规性要求更改的机制。如果未确定要求，请确保您已对此决定进行尽职调查。定期审查您的运营重点，以便在需求发生变化时对其进行更新。

评估业务面临的威胁（例如业务风险和负债以及信息安全威胁），并在风险注册表中维护这些信息。评估风险的影响，在有冲突的利益或替代方法之间做出权衡。例如，新功能的加速上市可能会比成本优化更重要，或者您可以为非关系数据选择关系数据库来简化系统迁移工作，而无需重构。管理收益和风险，以便在确定工作重心时做出明智的决策。有些风险或选择可能在一段时间内可以接受，这可能会降低相关风险，或者允许风险继续存在可能会令人无法接受，在这种情况下，您将采取措施来化解风险。

您的团队必须了解他们在实现业务成果方面所发挥的作用。团队需要了解自己在其他团队获得成功过程中所扮演的角色、其他团队在他们获得成功的过程中所扮演的角色，并设定共同的目标。了解责任分配、所有权归属、决策制定方式以及决策者将有助于集中精力，最大限度地发挥团队的优势。团队的需求将由其所支持的客户、所在组织、团队的组成以及工作负载的特征决定。期望单个运营模式能够支持组织中的所有团队及其工作负载是不合理的。

确保每个应用程序、工作负载、平台和基础设施组件都有确定的负责人，并且每个流程和程序都有确定的负责人负责其定义，有负责人负责其性能。

了解每个组件、流程和程序的商业价值，了解为什么要配置这些资源或为什么要执行这些活动，以及为什么要拥有该所有权，这些都有助于确定团队成员的行动。清晰定义团队成员的责任以便他们可以适当地采取行动，并制定相关机制，确定责任和所有权。制定用于请求添加、更改和例外的机制，以免限制创新。在团队之间定义协议，描述团队之间如何开展合作以相互支持以及您的业务成果。

为您的团队成员提供支持，以便他们可以更有效地采取行动并为您的业务成果提供支持。参与其中的高层领导应设定期望并衡量是否成功。他们应是采用最佳实践和组织发展的发起人、倡导者和推动者。授权团队成员在成果面临风险时采取行动以尽可能减少影响，并鼓励他们在认为存在风险时向决策者和利

益相关者上报，以便解决问题并避免事故。及时、清晰、可行地传达已知风险和计划内事件，以便团队成员可以及时采取适当行动。

鼓励进行试验，以加快学习速度，并使团队成员保持兴趣和参与热情。团队必须增强自己的技能组合，以采用新技术，并随需求和责任的变化继续提供支持。专门安排学习时间，以提供支持并鼓励参与其中。确保您的团队成员拥有取得成功所需的资源（包括工具和团队成员），并具有支持您的业务成果的规模。利用跨组织的多样性来寻求多种独特的见解。利用这种见解提高创新能力、对您的假设提出质疑，并降低确认偏差的风险。在团队内部提升包容性、多样性和可达性有助于获取有益的见解。

如果存在适用于您组织的外部法规或合规性要求，则应使用 AWS 云合规性 提供的资源来帮助培训您的团队，以便他们能够确定运营重点会受到的影响。架构完善的框架强调学习、衡量和改进。它为您提供了一种一致的方法来评估架构，并实施将随着时间推移而扩展的设计。AWS 提供了 AWS Well-Architected Tool，可帮助您在开发之前审查方法，在生产之前审查工作负载状态，以及在生产过程中审查工作负载状态。您可以将其与最新的 AWS 架构最佳实践进行比较，监控整体状态，并深入了解潜在风险。AWS Trusted Advisor 是一种工具，让您可以访问一组核心检查，这些检查会提出优化建议，帮助确定您的运维重点。商业支持和企业支持客户可以访问其他检查，这些检查重点关注安全性、可靠性、性能和成本优化，可进一步帮助他们帮助确定运营重点。

AWS 可以帮您向团队介绍 AWS 及其服务，让他们深入了解自己的选择会如何影响工作负载。您应该使用由 AWS Support（AWS 知识中心、AWS 开发论坛和 AWS Support 中心）和 AWS 文档提供的资源来培训您的团队。请通过 AWS Support 中心联系 AWS Support，获取与 AWS 问题有关的帮助。AWS 还分享了我们通过在 Amazon Builders' Library 中的 AWS 运维学到的最佳实践和模式。您可以通过 AWS Blog 和 AWS 官方播客，获得各种其他有用信息。AWS Training and Certification 提供了一些免费培训，可以通过自定进度的数字课程，学习 AWS 的基础知识。您还可以报名参加讲师指导培训，进一步帮助培养您团队的 AWS 技能。

您应使用能够跨 AWS Organizations 等账户集中监管环境的工具或服务帮助管理运营模式。AWS Control Tower 等服务扩展了这一管理功能，使您能够定义账户设置的蓝图（支持您的运营模式），使用 AWS Organizations 进行持续监管以及自动预置新账户。托管服务提供商（如 AWS Managed Services、AWS Managed Services 合作伙伴或 AWS 合作伙伴网络中的托管服务提供商）会提供实施云环境的专业知识，并为您的安全性和合规性要求以及业务目标提供支持。将托管服务添加到您的运营模式可以节省您的时间和资源，并使您的内部团队保持精干，专注于凸显业务优势的战略成果，而不是开发新的技能和功能。

以下问题主要针对卓越运营的准备阶段。（有关卓越运营问题的列表和最佳实践，请参阅 附录）。

OPS 1：您如何确定自己的重点？

每个人都需要了解自己在业务成功中扮演的角色。设置共同的目标，以便为资源设定重点。这可以让您的工作效益最大化。

OPS 2：如何构建组织结构来为业务成果提供支持？

您的团队必须了解他们在实现业务成果方面所发挥的作用。团队需要了解自己在其他团队获得成功过程中所扮演的角色、其他团队在他们获得成功的过程中所扮演的角色，并设定共同的目标。了解责任分配、所有权归属、决策制定方式以及决策者将有助于集中精力，最大限度地发挥团队的优势。

OPS 3：组织文化如何为业务成果提供支持？

为您的团队成员提供支持，以便他们可以更有效地采取行动并为您的业务成果提供支持。

您可能会发现，您需要在某个时间点侧重于一小部分运营重点。长期使用平衡的方法来确保所需能力的发展和风险管理。定期回顾运营重点，并根据需求变化进行更新。当责任和所有权不确定或未知时，您将面临以下风险：没有及时执行必要的活动，以及在处理这些需求时可能出现工作冗余和潜在冲突。组织文化会直接影响团队成员的工作满意度和保留率。增强团队成员的参与度和能力，助力业务成功。创新必须进行试验，才能将创意转化为成果。应认识到，取得非预期结果也算试验成功，因为这种试验发现了无法实现成功的途径。

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。然后，您需要能够针对它们进行设计，以提供对其状态的洞察并构建程序以提供支持。

将工作负载设计成能够提供必要的信息，以便您了解其所有组件的内部状态（例如指标、日志、事件和跟踪信息），为可观测性和调查问题提供支持。迭代开发必要的遥测技术，以监控工作负载的运行状况，确定结果何时面临风险并做出有效响应。在检测工作负载时，请捕获一组广泛的信息以启用情景感知（例如，状态变化、用户活动、特权访问和利用率计数器等的变更），因为您可以随时间变化筛选最有用的信息。

采用改进生产调整流程并支持重构、快速质量反馈和错误修复的方法。这些方法可以加快有益更改进入生产环境的速度、减少产生的问题，并能够快速识别和修复通过部署活动引入的问题或在环境中发现的问题。

采用提供快速质量反馈，并且若更改没有达到目标成效，则支持快速恢复的方法。使用这些实践可以减轻因部署更改而产生的问题的影响。制定计划以防更改不成功，这样在必要时能够更快速的响应，并测试和验证所做的更改。了解环境中的计划活动，以便管理更改风险，避免影响计划活动。强调频繁、小规模、可逆更改，以限制更改范围。这样可以简化故障排除工作、加快修复速度，并支持回滚更改。此外，还意味着能够更频繁地从有价值的更改中获益。

评估工作负载、流程和程序以及工作人员的运营准备就绪情况，以了解与工作负载相关的运营风险。您应该使用一致的流程（包括手动或自动化检查清单）来了解何时可运营工作负载或进行更改。这也使您能够发现需要制定计划予以解决的任何问题。准备好记录日常活动的运行手册和指导问题解决流程的行动手册。了解收益和风险，以便做出明智的决策，从而使更改应用到生产环境。

AWS 使您能够将整个工作负载（应用程序、基础设施、策略、监管和运维）视为代码。这意味着，您可以将用于应用程序代码的工程规范应用于堆栈的每个元素，并在团队或组织之间共享，提高开发工作的效益。使用云中的运营即代码功能和安全测试功能开发工作负载、运营流程和故障演练。使用 AWS CloudFormation，您可以实现一致的模板化沙箱开发、测试和生产环境，提高运营管理水平。

以下问题主要针对卓越运营的准备阶段。

OPS 4：如何设计工作负载以便自己了解其状态？

将工作负载设计成能够提供所有组件（例如指标、日志和跟踪信息）的必要信息，以便您了解其内部状态。这让您能够在适当的时候提供有效的响应。

OPS 5：如何减少缺陷、简化修复和改进生产流程？

支持在生产时调整改进流程并支持重构、快速质量反馈和错误修复方法。这些方法可以加快有益更改进入生产环境的速度、减少产生的问题，并能够快速识别和修复通过部署活动引入的问题。

OPS 6：您如何缓解部署风险？

采用提供快速质量反馈，并且若更改没有达到目标成效，则支持快速恢复的方法。使用这些实践可以减轻因部署更改而产生的问题的影响。

OPS 7：如何知道您已经准备好支持某种工作负载？

评估工作负载、流程及程序和工作人员的操作准备就绪情况，以便了解与工作负载相关的操作风险。

对代码化运营进行投资，以最大限度地提高运营人员的工作效率，最大限度地降低错误率，并实现自动响应。使用“故障演练”来预测故障，并根据需要创建程序。使用资源标签和 AWS Resource Groups，按照一致的标记策略应用元数据，以标识您的资源。标记您的资源，以便进行整理、成本核算、访问控制并有针对性地自动执行操作活动。利用云的弹性特点结合相应部署实践，来推动开发活动和系统的预部署，以加快部署速度。当您对用于评估工作负载的检查清单进行更改时，请计划要对不再符合条件的活动系统执行哪些操作。

运营

工作负载运营是否成功通过业务成果和客户结果的实现情况加以衡量。定义预期结果、确定成功的衡量方式，并确定将在这些计算中使用的指标，以确定工作负载和运营是否成功。运营状况包括工作负载的运行状况，以及为支持工作负载而执行的操作的运行状况和成败（例如，部署和事件响应）。设立改进、调查和介入的指标基线，收集和分析您的指标，然后验证您对运营成功的理解及其随时间变化的规律。使用收集的指标来确定您是否可以满足客户需求和业务需求，并确定需要改进的领域。

要实现卓越运营，您需要进行有效且高效的运营事件管理。这适用于计划内和计划外的运营事件。使用已确定的运行手册解决易于理解的事件，并使用行动手册来帮助调查和解决问题。您需要根据事件对业务和客户的影响排定其优先级。务必确保在出现事件警报时，会有指定负责人启动相关流程。事先定义解决事件所需的人员，并配备一个上报触发器，以便根据紧急程度和影响在必要时引入额外人员。确定并引入有权决定行动方案的人员，这些行动方案将对之前未解决的事件响应产生业务影响。

通过为目标受众（例如，客户、业务人员、开发人员、运营人员）定制的控制面板和通知来发布工作负载的运行状态，以便他们可以采取相应措施、管理预期，并在恢复正常运营时收到通知。

在 AWS 中，您可以为收集的工作负载指标和 AWS 自带指标生成控制面板视图。您可以利用 CloudWatch 或第三方应用程序来汇总和呈现运维活动的业务、工作负载和运营级别视图。AWS 通过日志记录功能（包括 AWS X-Ray、CloudWatch、CloudTrail 和 VPC 流日志）提供工作负载洞察，从而帮助识别工作负载问题，以支持根本原因分析和修复。

以下问题主要针对卓越运营的准备阶段。

OPS 8：您如何了解工作负载的运行状况？

定义、记录和分析指标以便了解工作负载事件，从而采取适当的措施。

OPS 9：您如何了解自己的运营状况？

定义、记录和分析运营指标以便了解运营事件，从而采取适当的措施。

OPS 10：您如何应对工作负载事件和运营事件？

制定和验证用于响应事件的程序，以便尽可能减少其对工作负载的干扰。

您收集的所有指标都应该与业务需求及其支持的结果相符。为充分理解的事件开发脚本式响应，并自动执行响应以识别事件。

演进

必须学习、分享和不断改进，以保持卓越运营。专注于工作周期，以持续进行渐进式改进。对影响客户的所有事件执行事件后分析。确定导致这些事件的因素和预防措施，以限制或防止再次发生。根据需要与受影响的团体沟通导致这些事件的因素。定期评估并优先处理改进机会（例如，功能请求、问题修复和合规性要求），包括工作负载和运营程序。

将反馈周期纳入您的流程，以快速确定需要改进的领域，并从运营执行中获取经验教训。

在团队中分享得到的经验教训，并从中受益。分析经验教训中的趋势，并对运营指标进行跨团队回顾性分析，以确定改进的机会和方法。实施改进措施，并评估结果以确定是否成功。

在 AWS 上，您可以将日志数据导出到 Amazon S3 或将日志直接发送到 Amazon S3 以便长期存储。使用 AWS Glue，您可以在 Amazon S3 中发现并准备您的日志数据以供分析，并将相关元数据存储在 AWS Glue Data Catalog 中。然后，通过 Amazon Athena 与 AWS Glue 的原生集成，可以对您的日志数据进行分析，并使用标准 SQL 查询日志数据。使用像 Amazon QuickSight 这样的商业智能工具，您可以直观显示、浏览和分析您的数据。发现可能推动改进的相关趋势和活动。

以下问题主要针对卓越运营方面的注意事项。

OPS 11：如何改进运营？

分配专用的时间和资源用于持续增量改进，以便提高运营的有效性和效率。

运营的成功演进建立在以下基础上：频繁的小规模改进；提供安全的环境和时间来试验、开发和测试改进；以及鼓励人们从失败中获取经验教训的整体氛围。随着运营控制水平的提高，对于沙箱、开发、测试和生产环境的运营支持促进了开发，并提高了对生产环境中部署的变更结果成功与否的可预测性。

资源

请参阅以下资源，详细了解卓越运营的最佳实践。

文档

- DevOps 和 AWS

白皮书

- 卓越运营支柱

视频

- Amazon DecOps

安全性

安全性支柱包括保护数据、系统和资产以利用云技术来改善安全性的能力。

安全性支柱概述了设计原则、最佳实践和问题。如需有关具体实施的说明性指导，请参阅《安全性支柱》白皮书.

主题

- 设计原则
- 定义
- 最佳实践

- 资源

设计原则

在云中实现安全性有七个设计原则：

- 健壮的身份验证体系：实施最小权限原则，并通过对每一次与 AWS 资源之间的交互进行适当授权来强制执行职责分离。集中进行身份管理，并努力消除对长期静态凭证的依赖。
- 实现可追溯性：实时监控和审计对环境执行的操作和更改并发送警报。为系统集成日志和指标收集功能，以自动调查并采取措施。
- 在所有层面应用安全措施：利用多种安全控制措施实现深度防御。应用到所有层面（例如网络边缘、VPC、负载均衡、每个实例和计算服务、操作系统、应用程序和代码）。
- 自动实施安全最佳实践：借助基于软件的自动化安全机制，您能够以更为快速且更具成本效益的方式实现安全扩展。创建安全架构，包括实施可在版本控制模板中以代码形式定义和管理的控制措施。
- 保护动态数据和静态数据：将您的数据按敏感程度进行分类，并采用加密、令牌和访问控制等机制（如适用）。
- 限制对数据的访问：使用相关机制和工具来减少和消除直接访问或人工处理数据的需求。这样可以降低处理敏感数据时数据处理不当、被修改以及人为错误的风险。
- 做好应对安全性事件的准备：制定符合您组织要求的事件管理和调查策略和流程，做好应对事件的准备工作。开展事件响应模拟演练并使用具有自动化功能的工具来提高检测、调查和恢复的速度。

定义

在云中实现安全性包括六个方面的最佳实践：

- 安全性
- 身份和权限管理
- 检测
- 基础设施保护
- 数据保护
- 事件响应

在为任何工作负载设计架构之前，您需要确定可能影响安全性的实践。您需要控制谁可以执行什么操作。另外，您希望能够识别安全事件、保护您的系统和服务，并通过数据保护机制来保持数据的机密性

和完整性。您应该具备一个定义明确且经过实践的流程来响应安全事件。这些工具和方法非常重要，因为它们有助于实现诸如避免财务损失或遵循法律与合规性要求等一系列目标。

借助 AWS 责任共担模式，组织能够利用云服务实现其安全性和合规性目标。AWS 负责保护用于支持云服务的基础设施，作为 AWS 的客户，您能够专注于使用云上的各种服务来实现您的目标。还可通过 AWS Cloud 更好地访问安全数据，并以自动化方式响应安全性事件。

最佳实践

主题

- 安全性
- 身份与权限管控
- 检测
- 基础设施保护
- 数据保护
- 事件响应

安全性

为了安全地操作您的工作负载，您必须对安全性的各个方面应用总体最佳实践。采用您在组织和工作负载层面的卓越运营中定义的要求和流程，并将它们应用到各个方面。

及时了解最新的 AWS、行业建议以及威胁情报信息可帮助您改进您的威胁模型和控制目标。实现安全流程、测试和验证的自动化可扩展您的安全运营。

以下问题主要针对安全方面的注意事项。（有关安全性问题的列表和最佳实践，请参阅 附录）。

SEC 1：如何安全地操作您的工作负载？

为了安全地操作您的工作负载，您必须对安全性的各个方面应用总体最佳实践。采用您在组织和工作负载层面的卓越运营中定义的要求和流程，并将它们应用到各个方面。及时了解来自 AWS 的建议、行业资源以及威胁情报信息可帮助您改进您的威胁模型和控制目标。实现安全流程、测试和验证的自动化可扩展您的安全运营。

在 AWS 中，建议根据账户的功能和合规性或数据敏感性要求分离不同的工作负载。

身份与权限管控

身份识别与访问管理是信息安全计划的关键部分，可以确保只有经过授权和通过身份验证的用户和组件才能访问您的资源，并且只能以您要求的方式进行访问。例如，您需要定义一些主体（即可以在您的账户中执行操作的账户、用户、角色和服务）、创建与这些主体相匹配的策略，并实施严格的凭证管理。这些权限管理元素构成了身份验证和授权的核心。

在 AWS 中，权限管理主要通过 AWS Identity and Access Management（IAM）服务来实现，您可以使用该服务控制对 AWS 服务和资源的用户和编程访问。您需要应用细粒度的策略向用户、组、角色或资源分配权限。您还可以应用强密码原则（例如复杂程度）来避免重复使用并强制执行多重验证 (MFA)。您可以将联合身份验证与现有的目录服务配合使用。对于需要系统接入 AWS 的工作负载，IAM 可以通过角色、实例配置文件、身份联合验证和临时凭证进行安全访问。

以下问题主要针对安全方面的注意事项。

SEC 2：如何管理人员和机器的身份？

在访问和运行安全的 AWS 工作负载时，您需要管理两种类型的身份。了解管理和授予访问权限所需的身份类型，这有助于确保正确的身份能够在正确的条件下访问正确的资源。

人员身份：您的管理员、开发人员、操作员和最终用户需要确定身份才能访问您的 AWS 环境和应用程序。这些是您的组织成员或您与之协作的外部用户，以及通过 Web 浏览器、客户端应用程序或交互式命令行工具与您的 AWS 资源交互的用户。

机器身份：您的服务应用程序、操作工具和工作负载需要一个身份来向 AWS 服务发出请求，例如，读取数据。这些身份包括在 AWS 环境中运行的机器，例如 Amazon EC2 实例或 AWS Lambda 函数。您还可以管理需要访问权限的外部各方的机器身份。此外，您可能还有需要访问您 AWS 环境的 AWS 之外的机器。

SEC 3：如何管理人员和机器的权限？

管理权限以控制对需要访问 AWS 和您的工作负载的人员和机器身份的访问。权限用于控制哪些人可以在什么条件下访问哪些内容。

凭证不得与任何用户或系统共享。应使用最小权限原则授予用户访问权限，并采用密码规则和强制执行 MFA 等最佳实践。应使用临时凭证和有限权限凭证（例如 AWS Security Token Service 发放的凭证）来执行程序访问（包括对 AWS 服务的 API 调用）。

AWS 提供了能够帮助您使用 Identity and Access Management 的资源。为了帮助学习最佳实践，请探索我们的 管理凭证和身份验证， 控制人员访问和 控制程序访问的相关动手实验.

检测

您可以使用检测控制来识别潜在的安全威胁或事件。检测控制是管理框架的重要组成部分，并且可以用于支持质量流程、法律或合规，还可以用于威胁识别和响应工作。检测控制分为多种不同类型。例如，编制资产清单及其详细属性有助于更有效地做出决策（以及进行生命周期管理），从而有助于建立运营基准。您可以通过内部审计（是指对信息系统相关的控制措施进行的检查）来确保实践符合策略和要求，并确保您已根据定义的条件设置了正确的自动告警通知。这些控制措施都是重要的响应手段，可以帮助您的组织识别和了解异常活动的范围。

在 AWS 中，您可以通过处理可用于审计、自动化分析和触发警报的日志、事件以及监控来实施检测控制。CloudTrail 日志、AWS API 调用和 CloudWatch 可以提供对指标进行监控以及报警的功能，AWS Config 可以提供配置历史记录。Amazon GuardDuty 是一种托管的威胁检测服务，可以持续监控恶意或未经授权的行为，从而帮助您保护您的 AWS 账户和工作负载。您还可以使用服务级别日志，例如，您可以使用 Amazon Simple Storage Service (Amazon S3) 来记录访问请求。

以下问题主要针对安全方面的注意事项。

SEC 4：您如何检测和调查安全事件？
通过日志和指标来记录和分析事件，以便了解信息。针对安全事件和潜在的威胁采取措施，以便保护您的工作负载。

日志管理对于架构完善的工作负载至关重要，这其中原因众多，包括安全性或取证、法律或法规要求。分析日志并相应地做出响应至关重要，这样您能够识别潜在的安全事件。借助 AWS 提供的功能，您能够定义数据保留生命周期或定义数据保存、存档或最终删除的位置，从而更轻松地管理日志。这样，您就能够以更为简单且更具成本效益的方式进行可预测且可靠的数据处理。

基础设施保护

基础设施保护包括满足最佳实践和组织、法律及监管义务所必需的控制方法（例如深度防御）。使用这些方法对于在云中或本地持续成功运营是至关重要的。

在 AWS 中，您可以通过使用 AWS 原生技术或使用 AWS Marketplace 提供的合作伙伴产品和服务来进行有状态和无状态数据包检查。您还可使用 Amazon Virtual Private Cloud (Amazon VPC) 创建一个安全且可扩展的私有环境，您可以在其中定义拓扑结构，包括网关、路由表以及公有子网和私有子网。

以下问题主要针对安全方面的注意事项。

SEC 5：如何保护您的网络资源？

任何以某种形式连接至网络的工作负载（互联网或私有网络）都需要多层防御，以帮助防御基于外部和内部网络的威胁。

SEC 6：如何保护计算资源？

工作负载内的计算资源需要采用多层防御，才有助于免受内部和外部威胁。计算资源包括 EC2 实例、容器、AWS Lambda 函数、数据库服务、IoT 设备等。

在任何类型的环境，我们都建议使用多层防御。在基础设施保护方面，许多概念和方法在跨云和本地模型中都有效。实施边界保护、监控入站点和出站点以及建立全面的日志记录、监控和告警机制对于制定有效的信息安全计划至关重要。

AWS 客户能够定制或加强 Amazon Elastic Compute Cloud（Amazon EC2）、Amazon Elastic Container Service（Amazon ECS）容器或 AWS Elastic Beanstalk 实例的配置，并将配置保存到不可变的亚马逊云机器镜像（AMI，Amazon Machine Image）。之后，无论是由 Auto Scaling 触发还是手动启动，使用此 AMI 启动的所有新虚拟服务器（实例）都会收到上述加强的配置。

数据保护

在为任何系统设计架构之前，您应确定可能影响安全性的基本实践。例如，数据分级提供了一种基于敏感程度对组织数据进行分类的方法，加密通过让未经授权的用户无法获知数据的真正内容来保护数据。这些工具和方法非常重要，因为它们有助于实现诸如避免财务损失或遵循法律与合规性要求等一系列目标。

在 AWS 中，以下实践有助于保护数据：

- 作为 AWS 客户，您拥有对自己的数据的完全控制权。
- AWS 可帮助您更轻松地加密数据和管理密钥（包括定期密钥轮换），这些操作可以由 AWS 轻松自动执行，也可由您执行。
- 我们还提供包含文件访问和更改等重要内容的详细日志记录。

- AWS 设计的存储系统具有优异的弹性。例如，Amazon S3 标准、S3 标准 – IA、S3 单区 – IA 和 Amazon Glacier 都设计为可以在一年内实现 99.999999999% 的对象持久性。这一持久性级别相当于平均每年有 0.000000001% 的数据对象丢失。
- 作为较大规模数据生命周期管理流程中的一部分，版本控制可以防止意外覆盖、删除数据和类似损害。
- AWS 永远不会主动在区域之间移动数据。除非您明确启用相关功能或利用提供该功能的服务移动数据，否则放置在某个区域中的内容将保留在该区域中。

以下问题主要针对安全方面的注意事项。

SEC 7：如何对数据进行分类？

分类提供了一种基于关键性和敏感度对数据进行分类的方法，以帮助您确定适当的保护和保留控制措施。

SEC 8：如何保护静态数据？

通过实施多个控制措施来保护静态数据，以降低未经授权的访问或处理不当带来的风险。

SEC 9：如何保护传输中的数据？

通过实施多个控制措施来保护传输中的数据，以降低未经授权的访问或数据丢失所带来的风险。

AWS 提供了多种加密静态数据和传输中数据的方法。我们将这些功能内置在我们的服务中，这样您可以更轻松地加密数据。例如，我们为 Amazon S3 实施了服务器端加密 (SSE)，这样您可以更轻松地以加密的方式存储数据。您还可以将整个 HTTPS 加密和解密过程（通常称为 SSL 终端）交给 Elastic Load Balancing (ELB) 来完成。

事件响应

即使采用极为成熟的预防和检测控制机制，您的组织仍应制定相关流程来响应安全事件并缓解安全事件可能带来的影响。工作负载的架构会极大地影响团队在事件发生期间采取行动、隔离或约束系统并将运行状态恢复到已知的良好状态的能力。在安全事件发生之前确保相关工具部署到位，而后定期进行响应演练，将有助于确保您的架构有能力及时进行调查和恢复。

在 AWS 中，以下实践有助于做出有效的事故响应：

- 我们提供包含文件访问和更改等重要内容的详细日志记录。
- 事件可以自动处理，并且会触发通过使用 AWS API 自动做出响应的工具。
- 您可以使用 AWS CloudFormation 预先配置工具和一个“清洁屋”。这样您就可以在安全且隔离的环境中进行取证。

以下问题主要针对安全方面的注意事项。

SEC 10：如何预测、响应事件以及从事件中恢复？

准备工作对于及时有效地调查、响应安全事件以及从安全事件中恢复至关重要，可以尽可能减少对组织的破坏。

确保您能够快速授予安全团队访问权限，而且系统可以自动隔离实例并自动捕捉数据与状态信息用于取证。

资源

请参阅以下资源，详细了解安全方面的最佳实践。

文档

- AWS Cloud 安全
- AWS 合规性
- AWS 安全性博客

白皮书

- 安全性支柱
- AWS 安全性概述
- AWS 风险和合规性

视频

- AWS 安全中心

- 责任共担模式概述

可靠性

可靠性支柱涵盖相关工作负载按照计划正确而稳定执行其预期功能的能力。它包括在其全部生命周期内运行和测试工作负载的能力。本白皮书深度介绍了有关在 AWS 中实施可靠工作负载的最佳实践指导。

可靠性支柱概述了设计原则、最佳实践和问题。如需有关具体实施的说明性指导，请参阅 《可靠性支柱》白皮书.

主题

- 设计原则
- 定义
- 最佳实践
- 资源

设计原则

在云中实现可靠性有五个设计原则：

- 自动从故障中恢复：通过监控工作负载的关键绩效指标 (KPI)，您可以在指标超过阈值时触发自动化功能。这些 KPI 应该是对商业价值（而不是服务运营的技术方面）的一种度量。这包括自动发送故障通知和跟踪故障，以及启动解决或修复故障的自动恢复流程。借助更高级的自动化功能，您可以在故障发生之前预测和修复故障。
- 测试恢复过程：在本地环境中，经常会通过执行测试来证明工作负载能够在特定场景中正常运作。通常不会利用测试来验证恢复策略。在云中，您可以测试工作负载的故障情况，并验证您的恢复程序。您可以采用自动化方式来模拟不同的故障，也可以重新建立之前导致故障的场景。此方式可以在实际的故障发生以前揭示您可以测试与修复的故障路径，从而降低风险。
- 横向扩展以提高聚合工作负载的可用性：使用多个小型资源替换一个大型资源，以降低单个故障对整个工作负载的影响。跨多个较小的资源分配请求，以确保它们不共用常见故障点。
- 无需再预估容量：本地工作负载出现故障的常见原因是资源饱和，即对工作负载的需求超过该工作负载的容量（这通常是拒绝服务攻击的目标）。在云中，您可以监控需求和工作负载利用率，并自动添加或删除资源，以保持最佳水平来满足需求，而不会出现超额预置或预置不足的问题。虽然还有很多限制，但有些配额是可控的，其他配额也可以管理（请参阅“管理Service Quotas与限制”）。

- 管理自动化变更：应利用自动化功能对基础设施进行更改。需要管理的变更包括，对自动化的变更，可对其进行跟踪与审查。

定义

在云中实现可靠性包括四个方面的最佳实践：

- 基础
- 工作负载架构
- 变更管理
- 故障管理

要实现可靠性，您必须从基础入手，而基础是服务配额和网络拓扑适应工作负载的环境。在设计时，分布式系统的工作负载架构必须能够预防与减少故障。工作负载必须处理需求或要求的变化，而且它的设计必须能够检测故障，并自动加以修复。

最佳实践

主题

- 基础
- 工作负载架构
- 变更管理
- 故障管理

基础

基础要求是指其范围超出单个工作负载或项目的因素。在为任何系统设计架构之前，您应确定影响可靠性的基本要求。例如，您必须为数据中心提供足够的网络带宽。

在您使用 AWS 时，这些基础要求中的大部分已经包含在内，并且可以根据需要进行处理。云环境在设计层面拥有几乎无限的资源，因此 AWS 要负责满足对足够联网和计算容量的需求，让您可以根据需求随意更改资源大小和分配。

以下问题主要针对可靠性的注意事项。（有关可靠性问题的列表和最佳实践，请参阅 附录）。

REL 1：如何管理服务配额和限制？

基于云的工作负载架构存在服务配额（也被称作服务限制）。存在这些配额是为了防止意外预置超过您所需的资源，并对 API 操作的请求速率进行限制，以保护服务不会遭到滥用。还存在资源限制，例如，将比特推入光缆的速率，或物理磁盘上的存储量。

REL 2：如何规划网络拓扑？

工作负载通常存在于多个环境中。其中包括多个云环境（可公开访问云和私有云），可能还包括现有数据中心基础设施。相关计划必须涵盖网络注意事项，如系统内部和系统间连接、公有 IP 地址管理、私有 IP 地址管理，以及域名解析。

基于云的工作负载架构存在服务配额（也被称作服务限制）。这些配额的存在目的在于，防止您意外预置超出必要量的资源，限制 API 操作的请求速率，从而避免服务遭到滥用。工作负载通常存在于多个环境中。您必须为所有工作负载环境监控和管理这些配额。其中包括多个云环境（可公开访问的云和私有云），可能还包括您的现有数据中心基础设施。相关计划必须涵盖网络注意事项，如系统内部和系统间连接、公有 IP 地址管理、私有 IP 地址管理以及域名解析。

工作负载架构

可靠的工作负载始于前期的软件和基础设施设计决策。您的架构选择将影响所有 Well-Architected 支柱的工作负载行为。针对可靠性，您必须遵循特定的模式。

使用 AWS 时，工作负载开发人员可以选择要使用的语言和技术。AWS 开发工具包通过为 AWS 服务提供特定于语言的 API，省去了复杂的代码编写过程。通过这些开发工具包，以及语言选择，开发人员可以实现此处列出的可靠性最佳实践。开发人员还可以通过以下资料库阅读并了解 Amazon 构建和运营软件的方法： Amazon Builders' Library.

以下问题主要针对可靠性的注意事项。

REL 3：如何设计工作负载服务架构？

使用面向服务的架构 (SOA) 或微服务架构构建高度可扩展的可靠工作负载。面向服务的架构 (SOA) 可通过服务接口使软件组件可重复使用。微服务架构则进一步让组件变得更小、更简单。

REL 4：您如何在分布式系统中设计交互以预防发生故障？

分布式系统依赖于通信网络实现组件（例如服务器或服务）的互联。尽管这些网络中存在数据丢失或延迟，但是您的工作负载必须可靠运行。分布式系统组件的运行方式不得对其他组件或工作负载产生负面影响。这些最佳实践能够预防故障，并改善平均故障间隔时间（MTBF）。

REL 5：您如何在分布式系统中进行交互设计，从而缓解或经受住故障影响？

分布式系统依赖于通信网络以便使组件互相连接（如服务器或服务等）。尽管这些网络中存在数据丢失或延迟，但是您的工作负载必须可靠运行。分布式系统组件的运行方式不得对其他组件或工作负载产生负面影响。这些最佳实践使工作负载能够承受压力或故障，从中更快地恢复，并且降低此类伤害的影响。其结果是缩短平均恢复时间（MTTR）。

变更管理

您必须提前为工作负载或其环境的更改做好准备，从而实现工作负载的可靠操作。此类更改包括，外部因素施加到工作负载上的更改（如，需求高峰），以及内部更改（如功能部署和安全补丁）。

使用 AWS，您可以监控工作负载的行为并自动对 KPI 做出响应。例如，您的工作负载可以在某项工作负载用户增加时，添加更多服务器。您可以控制谁有权进行工作负载变更并审核这些变更的历史记录。

以下问题主要针对可靠性的注意事项。

REL 6：如何监控工作负载资源？

日志和指标是用于了解工作负载运行状况的强大工具。您可以配置工作负载以监控日志和指标，并在超出阈值或发生重大事件时发送通知。监控让您的工作负载可以发现超出低性能阈值和发生故障的情形，从而在响应中自动恢复。

REL 7：您如何设计工作负载，以适应不断变化的需求？

可扩展工作负载具有自动添加或移除资源的弹性，因此确保在任何时间点都能准确满足当前的需求。

REL 8：如何实施更改？

要部署新功能，必须对更改加以控制，以确保工作负载和操作环境正在运行已知的软件，并以可预测的方式进行修补和替换。如果此类更改不受控制，您将难以预测这些更改的影响，或难以处理由它们引发的问题。

当您构建工作负载来根据需求变化自动添加和删除资源时，这不仅可以提高可靠性，还可以确保业务成功不至于带来额外负担。借助既有的监控功能，当 KPI 偏离预期标准时，系统会自动向您的团队发送提醒。通过自动记录环境变更，您可以审核并快速识别可能影响可靠性的操作。对变更管理的控制确保您可以实施可提供所需的可靠性的规则。

故障管理

在任何具备一定复杂度的系统中，发生故障在意料之中。可靠性要求您的工作负载知晓故障的发生，并采取相应措施以避免对可用性产生影响。工作负载必须既能承受故障，又能自动解决问题。

使用 AWS，您可以利用自动化机制对监控数据做出响应。例如，当特定指标超过阈值时，您可以触发自动操作来解决问题。此外，与其尝试诊断并修复作为生产环境一部分的失败资源，您可以将其替换为新的资源，并对被替换的旧有资源进行故障排查。由于云使您能够以低成本构建整个系统的临时版本，您可以使用自动化测试来验证完整的恢复流程。

以下问题主要针对可靠性的注意事项。

REL 9：如何备份数据？

备份数据、应用程序和配置，以满足恢复时间目标（RTO）和恢复点目标（RPO）的要求。

REL 10：如何使用故障隔离来保护您的工作负载？

故障隔离边界可将一个工作负载内的故障影响限制于有限数量的组件。边界以外的组件不会受到故障的影响。使用多个故障隔离边界，您可以限制作用于您的工作负载的影响。

REL 11：如何将您的工作负载设计为可承受组件故障的影响？

在设计具有高可用性和较短平均恢复时间 (MTTR) 要求的工作负载时必须考虑到弹性。

REL 12：如何测试可靠性？

在为您的工作负载采用弹性设计以应对生产压力以后，测试是确保其按设计预期运行，并且提供您所预期弹性的唯一方式。

REL 13：如何规划灾难恢复 (DR)？

拥有适当的备份和冗余工作负载组件是您的 DR 策略的开始。RTO 和 RPO 是您恢复工作负载的目标。根据业务需求设置这些目标。通过实施策略来实现这些目标，同时考虑工作负载资源和数据的位置和功能。中断概率和恢复成本也是关键因素，有助于了解为工作负载提供灾难恢复的商业价值。

请定期备份数据并测试备份文件，确保您可以从逻辑和物理错误中恢复。管理故障的关键在于自动且频繁地测试工作负载以致其出现故障，然后观察它们如何恢复。请定期执行此操作，并确保在工作负载发生重大变更后也会触发此测试。主动跟踪 KPI（以及恢复时间目标（RTO，Recovery Time Objective）和恢复点目标（RPO，Recovery Point Objective））以评估工作负载的弹性（特别是在故障测试场景中）。跟踪 KPI 将有助于您识别和减少单点故障。充分测试您的工作负载恢复流程，确保可以恢复所有数据并继续为您的客户提供服务，即使面对持续存在的问题也是如此。您的恢复流程应该与您的标准生产流程一样完备而有效。

资源

请参阅以下资源，详细了解可靠性的最佳实践。

文档

- AWS 文档
- AWS 全球基础设施
- AWS Auto Scaling：扩展计划的工作原理
- 什么是 AWS Backup？

白皮书

- 可靠性支柱：AWS Well-Architected
- 在 AWS 上实施微服务

性能效率

性能效率要素包括有效地使用计算资源以满足系统要求的能力以及在需求变化和技术改进时保持此效率的能力。

性能效率支柱概述了设计原则、最佳实践和问题。如需有关具体实施的说明性指导，请参阅《性能效率支柱》白皮书.

主题

- 设计原则
- 定义
- 最佳实践
- 资源

设计原则

在云中实现性能效率包括五个方面的最佳实践：

- 普及先进技术：通过将复杂的任务委派给云供应商，降低您的团队实施高级技术的难度。与要求您的 IT 团队学习有关托管和运行新技术的知识相比，考虑将新技术作为服务使用是一种更好的选择。例如，NoSQL 数据库、媒体转码和机器学习都是需要专业知识才能使用的技术。在云中，这些技术会转变为团队可以使用的服务，让团队能够专注于产品开发，而不是资源预置和管理。
- 数分钟内实现全球化部署：您可以在全球多个 AWS 区域中部署工作负载，从而以最低的成本为客户提供更低的延迟和更好的体验。
- 使用无服务器架构：借助无服务器架构，您无需运行和维护物理服务器即可执行传统计算活动。例如，无服务器存储服务可以充当静态网站（从而无需再使用 Web 服务器），事件服务则可以实现代码托管。这不仅能够消除管理物理服务器产生的运行负担，还可以借由以云规模运行的托管服务来降低业务成本。
- 提升试验频率：利用虚拟和可自动化的资源，您可以快速利用各种类型的实例、存储或配置执行对比测试。
- 考虑软硬件协同编程：了解如何使用云服务，并始终使用最适合您工作负载目标的技术方法。例如，在选择数据库或存储方法时考虑数据访问模式。

定义

在云中实现性能效率包括四个方面的最佳实践：

- 选择
- 审核
- 监控
- 权衡

采用数据驱动型方法来构建高性能架构。收集架构各方面的数据，涵盖从高级设计到资源类型的选择与配置等。

定期审核您的选择，确保充分利用不断发展的 AWS 云的优势。监控可以确保您随时发现与预期性能的偏差。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

主题

- 选择
- 审核
- 监控
- 权衡

选择

针对特定工作负载的最佳解决方案各不相同，而且解决方案通常会结合多种方法。架构完善的工作负载会使用多种解决方案，并且启用各种不同的功能来提高性能。

我们提供多种类型和配置的 AWS 资源，可让您更轻松地找到最能满足您工作负载需求的方法。此外，我们还提供了无法使用本地基础设施轻松实现的选项。例如，Amazon DynamoDB 之类的托管服务可以提供完全托管的 NoSQL 数据库，确保在任何规模下都只会有几毫秒的延迟。

以下问题主要针对性能效率方面的注意事项。（有关性能效率问题的列表和最佳实践，请参阅 附录）。

PERF 1：如何选择性能最好的架构？

一个工作负载通常需要采用多种方法才能实现最佳性能。架构完善的系统会使用多种解决方案和功能来提高性能。

使用数据驱动型方法来为您的架构选择模式和实施方式，获得经济高效的解决方案。AWS 解决方案架构师、AWS 参考架构和 AWS 合作伙伴网络（APN，AWS Partner Network）合作伙伴可以根据自身的专业知识帮助您选择合适的架构，不过需要使用通过基准测试或负载测试提取的数据来优化您的架构。

您的架构可能会结合多种不同的架构方法（例如事件驱动、ETL 或管道）。在架构的实施中，将使用各种专门用于优化架构性能的 AWS 服务。在接下来的章节中，我们会介绍您应该考虑的四种主要资源类型：计算、存储、数据库和网络。

计算

选择满足您的要求和性能需求并具有出色成本效益的计算资源，将使您能够利用同等数量的资源获取更多收益。在评估计算选项时，请注意您的工作负载性能需求和成本要求，并以此做出明智的决策。

在 AWS 中，计算资源有三种形式：实例、容器和函数：

- 实例 是虚拟化服务器，因此您只需通过一个按钮或一次 API 调用即可对其功能进行调整。因为云中的资源决策不是固定不变的，所以您可以尝试使用不同的服务器类型。在 AWS 中，这些虚拟服务器实例具有不同的系列和大小，并且可以提供各种功能，包括固态硬盘（SSD，Solid-State Drive）和图形处理单元（GPU，Graphics Processing Unit）。
- 容器 是一种操作系统虚拟化方法，允许您在资源隔离的进程中运行应用程序及其依赖项。AWS Fargate 是适用于容器的无服务器计算引擎。如果您需要控制计算环境的安装、配置和管理，则可以使用 Amazon EC2。此外，您还可以从多个容器编排平台中进行选择：Amazon Elastic Container Service (ECS) 或 Amazon Elastic Kubernetes Service (EKS)。
- 函数 从您要执行的代码中抽象出执行环境。例如，AWS Lambda 让您可以在不运行实例的情况下执行代码。

以下问题主要针对性能效率方面的注意事项。

PERF 2：如何选择计算解决方案？

适合工作负载的最佳计算解决方案会根据应用程序设计、使用模式和配置设置而有所不同。架构可以使用不同的计算解决方案来支持各种组件，并且可以实现各种不同的功能来提高性能。为架构选择错误的计算解决方案可能会降低性能效率。

在设计如何使用计算资源时，您应该利用弹性机制来确保自己具有充足的容量，以便在需求发生变化时保持性能水平。

存储

云存储是云计算的关键组成部分，它存储着工作负载所使用的信息。云存储通常比传统的本地存储系统更加安全可靠且可扩展。从对象、数据块和文件存储服务以及您工作负载的云数据迁移选项中进行选择。

在 AWS 中，存储有三种形式：对象、数据块和文件：

- 对象存储 提供了一个可扩展的耐用平台，允许从任何互联网位置访问数据，适用于用户生成的内容、活跃的存档、无服务器计算、大数据存储或备份，以及恢复。Amazon Simple Storage Service (Amazon S3) 是一种对象存储服务，提供行业领先的可扩展性、数据可用性、安全性和性能。Amazon S3 的耐用性可达到 99.999999999%（11 个 9），为全球各地的公司存储数百万个应用程序的数据。
- 数据块存储 可为每个虚拟主机提供具有高可用性、一致性且低延迟的数据块存储，类似于直连式存储 (DAS) 或存储区域网络 (SAN)。Amazon Elastic Block Store (Amazon EBS) 旨在满足需要持久性存储的工作负载的需求，此类持久性存储可通过 EC2 实例访问，可帮助您根据适合的存储容量、性能和成本对应用程序进行微调。
- 文件存储 可以跨多个系统提供对共享文件系统的访问。Amazon Elastic File System（EFS）等文件存储解决方案非常适合大型内容存储库、开发环境、媒体存储或用户主目录等使用案例。Amazon FSx 让您可以轻松且经济高效地启动和运行常用文件系统，因此您可以利用应用广泛的开源和商业许可文件系统的丰富功能集和高速性能。

以下问题主要针对性能效率方面的注意事项。

PERF 3：如何选择存储解决方案？

针对特定系统的最佳存储解决方案往往取决于访问类型（块、文件或者对象存储）、访问模式（随机或者连续）、数据吞吐量要求、访问频率（在线、离线、归档）、更新频度（WORM、动态）以及可用性与持久性限制等因素。架构良好的系统使用多种解决方案，并且可以实现各种不同的功能来提高性能。

选择存储解决方案时，确保它与您的访问模式保持一致对于实现预期性能至关重要。

数据库

云可以提供专用数据库服务，解决您的工作负载所带来的各种问题。您可以从许多专用数据库引擎（包括关系、键值、文档、内存、图形、时间序列和分类账数据库）中进行选择。通过选择最佳数据库来解

决特定问题或一组问题，您可以摆脱限制性的“一刀切”整体式数据库，并专注于构建应用程序以满足客户的性能需求。

在 AWS 中，您可以从多个专用数据库引擎（包括关系、键值、文档、内存、图形、时间序列和分类账数据库）中进行选择。有了 AWS 数据库，您再也无需担心数据库管理任务，例如数据库预置、修补、设置、配置、备份或恢复。AWS 会通过自修复存储和自动扩展功能持续监控您的集群，使您的工作负载保持正常运行，这样您就可以专注于更高价值的应用程序开发。

以下问题主要针对性能效率方面的注意事项。

PERF 4：如何选择数据库解决方案？

针对特定系统的最优数据库解决方案取决于您的具体需求，包括可用性、一致性、分区容错性、延迟、持久性、可扩展性以及查询能力等等。许多系统会使用多种不同的数据库解决方案满足其各子系统的实际需要，并启用不同的功能来提高性能。为系统选择错误的数据库解决方案和功能可能会导致性能效率降低。

工作负载的数据库方法对性能效率具有重大影响。它通常是根据组织默认设置（而不是通过数据驱动型方法）选择的区域。如同考虑存储问题时一样，请务必要考虑工作负载的访问模式，另外还要考虑其他非数据库解决方案是否可以更高效地解决问题（例如图形、时间序列或内存存储数据库）。

网络

由于网络位于所有工作负载组件之间，因此可能会对工作负载性能和行为产生巨大的正面和负面影响。还有一些严重依赖网络性能的工作负载，例如，对于高性能计算 (HPC)，深入了解网络对于提高群集性能很重要。您必须确定带宽、延迟、抖动和吞吐量方面的工作负载要求。

在 AWS 中，网络资源以虚拟化形式存在，而且支持多种类型和配置。这让您可以更轻松地找到贴合您需求的网络方案。AWS 提供多种产品功能（例如增强联网、Amazon EBS 优化实例、Amazon S3 Transfer Acceleration 和动态 Amazon CloudFront）来优化网络流量。AWS 还可以提供多种联网功能（例如 Amazon Route 53 中的基于延迟的路由、Amazon VPC 端点、AWS Direct Connect 和 AWS Global Accelerator）来减少网络距离或抖动。

以下问题主要针对性能效率方面的注意事项。

PERF 5：如何配置联网解决方案？

适合某个工作负载的最佳网络解决方案会因延迟、吞吐量要求、抖动和带宽而有所不同。物理限制（例如用户资源或本地资源）决定位置选项。这些限制可以通过边缘站点或资源置放来抵消。

您必须考虑部署网络的位置。您可以选择将资源放置在靠近使用地点的位置，以缩短距离。使用网络指标来随着工作负载的发展对网络配置进行更改。利用区域、置放群组和边缘服务，您可以显著提高性能。基于云的网络可以快速重建或修改，因此有必要随着时间的推移改进网络架构，以保持性能效率。

审核

云技术的发展日新月异，因此您必须确保工作负载组件使用的是最新的技术和方法，以持续提高性能。您必须不断评估工作负载组件并考虑对其进行更改，以确保您能够满足其性能和成本目标。机器学习和人工智能 (AI) 等新技术可以让您重塑客户体验，并对所有业务工作负载进行创新。

利用由客户需求驱动的 AWS 持续创新。我们会定期发布新的区域、边缘站点、服务和功能。这些发布内容都可以明显提高架构的性能效率。

以下问题主要针对性能效率方面的注意事项。

PERF 6：如何改进工作负载以便利用新的版本？

在最初构建解决方案时，您可能会从有限的方案选项中进行选择。但是随着时间的推移，可提升工作负载性能的新技术和方法会不断涌现。

通常，不存在或不完整的性能审核流程会导致架构性能不佳。如果您的架构性能不佳，请实施性能审核流程，以便应用戴明的计划-执行-检查-处理 (PDCA) 循环来驱动迭代改进。

监控

实施工作负载后，必须监控其性能，以便在问题对客户造成影响之前进行补救。您应该使用监控指标，确保系统在指标超出阈值时发出告警。

Amazon CloudWatch 是一项监控和可观测性服务，可为您提供相关数据和切实见解，以监控工作负载、响应系统范围的性能变化、优化资源利用率，并在统一视图中查看运行状况。CloudWatch 以日志、指标和事件的形式从在 AWS 和本地服务器上运行的工作负载中收集监控和运维数据。AWS X-Ray 可以帮助开发人员分析和调试分布式生产应用程序。借助 AWS X-Ray，您可以了解应用程序的执行情况，发现根本原因并确定性能瓶颈。使用这些分析结果，您可以快速做出反应，保证工作负载顺畅运行。

以下问题主要针对性能效率方面的注意事项。

PERF 7：如何监控资源以确保其性能？

系统性能会随着时间的推移而降低。监控系统性能，以发现性能降低的情况，并针对内部或外部因素（例如操作系统或应用程序负载）采取修复措施。

有效监控解决方案的关键是确保不会看到误报。自动触发器可以避免人为错误，并且可以缩短解决问题的用时。请安排时间在生产环境中执行模拟以测试告警解决方案，确保它可以正确识别各种问题。

权衡

在架构解决方案时，需要权衡各种因素才能确保获得最佳方案。根据具体情况，您可以在一致性、持久性和空间与时间或延迟之间进行权衡，以便实现更高的性能。

使用 AWS，您可以在几分钟内实现全球化部署，并可在世界范围内的多个位置部署资源，从而缩短与最终用户的距离。您还可以将只读副本动态添加到信息存储系统（例如数据库系统），以减少主数据库的负载。

以下问题主要针对性能效率方面的注意事项。

PERF 8：如何使用权衡机制来提高性能？

在构建解决方案时，确定权衡机制可以帮助您选出最佳方法。通常，您可以牺牲一致性、持久性和空间来换取缩短时间和延迟，从而提高性能。

对工作负载进行更改时，需要收集并评估各项指标，以确定更改产生的影响。衡量对系统和最终用户的影响，以便了解权衡机制如何影响工作负载。使用负载测试等系统的方法来确定权衡机制是否可以提高性能。

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

- Amazon S3 性能优化
- Amazon EBS 卷性能

白皮书

- 性能效率支柱

视频

- AWS re:Invent 2019：Amazon EC2 基础（CMP211-R2）
- AWS re:Invent 2019：领导会议：联邦存储现状（STG201-L）
- AWS re:Invent 2019：领导会议：AWS 专用数据库（DAT209-L）
- AWS re:Invent 2019：连接 AWS 和混合 AWS 网络架构（NET317-R1）
- AWS re:Invent 2019：推动下一代 Amazon EC2 的发展：深入了解 Nitro 系统（CMP303-R2）
- AWS re:Invent 2019：扩展到第一个 1000 万用户（ARC211-R）

成本优化

成本优化支柱包括以最低价格运行系统来交付商业价值的能力。

成本优化支柱概述了设计原则、最佳实践和问题。如需有关具体实施的说明性指导，请参阅 部分.

主题

- 设计原则
- 定义
- 最佳实践
- 资源

设计原则

在云中实现成本优化有五个设计原则：

- 践行云财务管理：为获得财务上的成功并加速在云中实现商业价值，需要投资云财务管理/成本优化。您的组织需要投入时间和资源增强自身在这个新的技术和使用情况管理领域中的能力。与安全性或卓越运营能力类似，您的组织需要通过知识构建、计划、资源和流程来培养能力，从而成为一家具有成本效益的组织。

- 采用消费模型：仅为所需计算资源付费，并可根据业务需求而非复杂的预测增加或减少使用量。例如，开发和测试环境通常只需要在每个工作日运行八个小时。您可以在不需要时停用这些资源，从而实现 75% 的潜在成本节约（40 小时对比 168 小时）。
- 衡量整体效率：衡量工作负载的业务产出及这些产出的实现成本。使用这种衡量方式了解您通过提高产出和降低成本获得的收益。
- 不再将资金投入到无差别的繁重任务上：AWS 会负责繁重的数据中心运维任务，例如服务器的安装、堆叠和供电。它还消除了使用托管服务管理操作系统和应用程序的运营负担。因此，您可以集中精力处理客户和业务项目而非 IT 基础设施。
- 对支出进行分析和归因：使用云，您可以更轻松地确定系统的准确使用量和成本，从而将 IT 成本透明地分摊到各个工作负载拥有者。这有助于衡量投资回报率 (ROI)，并让工作负载拥有者能够据此优化资源和降低成本。

定义

在云中实现成本优化包括五个方面的最佳实践：

- 践行云财务管理
- 支出和使用情况意识
- 具有成本效益的资源
- 管理需求和供应资源
- 随着时间的推移不断优化

与良好架构框架中的其他支柱一样，成本优化支柱也需要权衡各种因素，例如，是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、交付新功能或只是为了按时完成任务，而不是优化预付成本。设计决策有时是在仓促中而非是由数据决定的，并且人们总是倾向于过度补偿“以防万一”，而不是花时间进行基准测试以获得成本最优的部署。这可能会导致过度预置和优化不足的部署。但是，当您需要将资源从本地环境“直接迁移”到云，然后再进行优化时，这是一个合理的选择。通过预先在成本优化策略中投入适量的精力，您可以确保始终如一地遵守最佳实践，避免不必要的过度预置，从而更轻松地实现云的经济优势。以下部分介绍了一些技巧和最佳实践，可帮助您开始并持续实施工作负载的云财务管理和成本优化。

最佳实践

主题

- 践行云财务管理

- 支出和使用情况意识
- 具有成本效益的资源
- 管理需求和供应资源
- 随着时间的推移不断优化

践行云财务管理

采用云后，由于缩短了审批、采购和基础设施部署周期，技术团队的创新速度会更快。要实现商业价值和财务成功，需要实施一种在云中管理财务的新方法。这种方法便是云财务管理，通过实施组织范围的知识构建、计划、资源和流程，在整个组织内培养能力。

许多组织由许多不同的单位构成，而这些单位又具有不同的要务。若能让组织遵循一组商定的财务目标并为组织提供实现这些目标的机制，将会打造一个更高效的组织。一个有能力的组织的创新和构建速度更快，更敏捷，并能够适应任何内部或外部因素。

在 AWS 中，您可以使用 Cost Explorer，也可以选择使用 Amazon Athena 和 Amazon QuickSight 查看成本和使用情况报告（CUR，Cost and Usage Report），从而了解整个组织的成本和使用情况。AWS Budgets 可主动发出成本和使用情况通知。AWS 博客提供有关新服务和新功能的信息，以确保您及时了解新发布的服务。

以下问题主要针对成本优化方面的注意事项。（有关成本优化问题的列表和最佳实践，请参阅 附录）。

COST 1：如何实施云财务管理？

实施云财务管理后，组织可以在 AWS 上优化成本和使用情况并进行扩展，从而实现商业价值和财务成功。

在组建成本优化部门时，需要包括成员并为团队配备 CFM 和成本优化方面的专家。现有的团队成员将了解组织的当前运作方式以及如何快速实施改进。此外，还可以考虑配备拥有辅助或专业技能组合的人员，例如具备分析和项目管理能力的人员。

在组织中树立成本意识时，需要改进现有计划和流程或基于现有计划和流程进行构建。与构建新流程和计划相比，向现有流程和计划增添内容要快得多。这样将更快地取得成果。

支出和使用情况意识

通过云，您可以获得更大的灵活性和敏捷性，从而支持创新以及快速的开发和部署。这样便节省了自建本地基础设施所需的人工环节和时间，包括确定硬件规格、协商报价、管理购买订单、安排发货和部署资源。然而，要实现这种易用性并利用近乎无限的按需容量，我们需要以新方式考虑支出。

很多企业有多个由不同团队运行的系统。将资源成本分摊到各个组织或产品拥有者可以推动更高效的资源使用模式，减少浪费。准确的成本分摊能够帮助您了解哪些产品是真正盈利的，让您能够做出更明智的预算分配决策。

在 AWS 中，您可以使用 AWS Organizations 或 AWS Control Tower 创建账户结构，这种方式不仅实现了分离，而且有助于对成本和使用进行分配。此外，也可以通过资源标记在使用情况和成本中标注业务和组织信息。使用 AWS Cost Explorer 查看您的成本和使用情况，或者使用 Amazon Athena 和 Amazon QuickSight. 创建自定义控制面板和分析。成本和使用情况控制通过 AWS Budgets 的通知来实现，并使用 AWS Identity and Access Management（IAM）和 Service Quotas 进行控制。

以下问题主要针对成本优化方面的注意事项。

COST 2：您如何管理使用情况？

制定各种策略和机制，确保花费适当的成本来达到目标。采用制约与平衡方法，您可以在不超支的情况下进行创新。

COST 3：如何监控使用情况和成本？

建立策略和程序以便监控并适当分配您的成本。这让您能够衡量和改进工作负载的成本效益。

COST 4：您如何停用资源？

在从项目开始到结束的过程中实施变更控制和资源管理。这可以确保您关闭或终止未使用的资源，以便减少浪费。

您可以使用成本分配标签对 AWS 使用情况和成本进行分类并跟踪。当您对 AWS 资源（例如 EC2 实例或 S3 存储桶）应用标签后，AWS 将通过使用情况和成本标签生成成本和使用情况报告。您可以使用代表组织类别的标签（例如成本中心、工作负载名称或拥有者）整理您的多个服务的成本。

确保在成本和使用情况报告和监控中使用正确的详细级别和粒度。要获得大概见解和趋势，请在 AWS Cost Explorer 中使用每日粒度。要更深入地进行分析和检查，请在 AWS Cost Explorer 中使用每小时粒度，或者在 Amazon Athena 和 Amazon QuickSight 中以每小时为粒度查看成本和使用情况报告 (CUR)。

结合标记资源和实体生命周期跟踪（员工、项目），您可以确定无法再为组织创造价值而应停用的孤立资源或项目。您可以设置账单提醒，以在预计超支时通知您。

具有成本效益的资源

为工作负载使用合适的实例和资源是节约成本的关键。例如，在小型服务器上运行某个报告需要五个小时，而在另一个两倍成本的大型服务器上运行只需要一个小时。虽然两个服务器提供同样的结果，但小型服务器随着时间推移会产生更多成本。

良好架构的工作负载会使用最具有成本效益的资源，这样可以产生巨大而积极的经济效益。您还可以使用托管服务降低成本。例如，您可以使用按电子邮件收费的服务，而无需自己维护电子邮件服务器。

AWS 提供各种灵活且具有成本效益的定价选项，以最符合您需求的方式从 Amazon EC2 和其他服务获取实例。按需 实例 允许按小时支付计算容量的费用，且无需最低使用承诺。 Savings Plans 和预留实例 与按需定价相比最高可节约 75% 的成本。使用 Spot 实例，您可以利用未使用的 Amazon EC2 容量，并且与按需定价相比最高可节约 90% 的成本。Spot 实例 适用于以下情况：系统可以容忍使用服务器队列，其中单个服务器可以动态装卸（例如无状态 Web 服务器）、批处理或使用 HPC 和大数据。

选择合适的服务还可以减少使用量和降低成本；例如，使用 CloudFront 可以最大限度地减少数据传输成本；例如，使用 Amazon Aurora on RDS 可以消除昂贵的数据库许可成本。

以下问题主要针对成本优化方面的注意事项。

COST 5：您在选择服务时如何评估成本？

Amazon EC2、Amazon EBS 和 Amazon S3 属于构建块 AWS 服务。托管服务（如 Amazon RDS 和 Amazon DynamoDB）属于更高级别或应用程序级别的 AWS 服务。通过选择适当的基础服务和托管服务，您可以优化工作负载，从而降低成本。例如，使用托管服务，您可以节省或消除大部分管理和运营开销，从而使您有精力从事应用程序和业务相关活动。

COST 6：在选择资源类型、规模和数量时，如何实现成本目标？

确保选择适合当前任务的资源规模和资源数量。选择最经济实惠的资源类型、规模和数量可以尽可能减少浪费。

COST 7：您如何使用定价模式来降低成本？

使用最适合的资源定价模式可以尽可能减少支出。

COST 8：您如何规划数据传输费用？

务必要监控和规划您的数据传输费用，以便制定架构决策，尽可能降低成本。持续以小步迭代的方式进行架构优化可以实现运营成本的大幅降低。

通过在选择服务时考虑成本因素，并使用 Cost Explorer 和 AWS Trusted Advisor 等工具定期检查 AWS 使用情况，您可以主动监控利用率并相应地调整部署。

管理需求和供应资源

在您迁移到云时，您仅为所需内容付费。您可以在需要时供应与工作负载需求匹配的资源，从而消除昂贵且浪费的过度预置需求。还可以通过限流、缓冲区或队列来修改需求，以满足需求并以更少的资源达成目标，从而降低成本，或者在以后使用批处理服务处理需求。

在 AWS 中，您可以自动预置资源来满足工作负载需求。通过使用基于需求或时间的方法进行 Auto Scaling，您可根据需要添加和删除资源。如果您可以预测需求变化，便可以节省更多资金并确保资源与工作负载需求匹配。您可以使用 Amazon API Gateway 实施限流，也可以使用 Amazon SQS 在工作负载中实施队列。这两种方法都允许您修改工作负载组件的需求。

以下问题主要针对成本优化方面的注意事项。

COST 9：如何管理需求和供应资源？

为了工作负载的性能与支出实现平衡，请确保您支付过费用的所有资源都得到利用，并避免出现资源利用率过低的情况。无论是从运维成本（由于过度使用导致性能下降）还是从浪费 AWS 支出（由于超额配置）的角度衡量，利用率指标过高或过低都会对您的组织产生负面影响。

当进行修改需求和供应资源的设计时，请主动考虑资源使用模式、预置新资源所需要耗费的时间，以及需求模式的可预测性。当管理需求时，确保您具有大小正确的队列或缓冲区，并在所需的时间内响应工作负载需求。

随着时间的推移不断优化

AWS 不断发布新服务和功能，因此您最好不断审视现有架构决策，以便确保其始终最具成本效益。当您的需求发生变化时，请主动停用不再需要的资源、整体服务和系统。

实施新功能或资源类型可以逐步优化您的工作负载，同时最大程度地减少实施变更所需的工作量。这样可不断提高效率，并确保您始终使用最新的技术，从而降低运营成本。您还可以使用新服务替换或向工作负载中添加新组件。这可以显著提高效率，因此必须定期审查您的工作负载，并实施新服务和新功能。

以下问题主要针对成本优化方面的注意事项。

COST 10：如何评估新服务?

AWS 不断发布新服务和功能，因此您最好不断审视现有架构决策，以便确保其始终最具成本效益。

定期审查部署时，评估更新的服务如何帮助您节省成本。例如，Amazon Aurora on RDS 可以降低关系数据库的成本。使用无服务器（例如 Lambda）服务，无需操作和管理实例即可运行代码。

资源

请参阅以下资源，详细了解成本优化的最佳实践。

文档

- AWS 文档

白皮书

- 成本优化支柱

可持续性

可持续性支柱侧重于环境影响，尤其是能源消耗和效率，因为它们是架构师在直接采取行动以减少资源使用时依据的重要杠杆。如需有关具体实施的说明性指导，请参阅 《可持续性支柱》白皮书.

主题

- 设计原则
- 定义
- 最佳实践

设计原则

在云中实现可持续性有六个设计原则：

- 了解您的影响： 衡量您的云工作负载的影响并为您的工作负载的未来影响建模。包括所有影响来源，例如客户使用您的产品所产生的影响，以及产品最终淘汰和停用所产生的影响。通过查看每个工作单元所需的资源和排放量，将生产性输出与云工作负载的总体影响进行比较。使用这些数据来建立关键绩效指标（KPI），评估在降低影响的同时提高生产力的方法，并估计提议的更改随时间的推移所产生的影响。
- 设定可持续性目标： 对于每个云工作负载，建立长期可持续性目标，例如减少每个事务所需的计算和存储资源。针对现有工作负载的可持续性改进的投资回报进行建模，并为负责人提供投资于可持续性目标所需的资源。规划增长并构建您的工作负载，以便增长可降低影响强度（以适当的单位衡量，例如每用户或每事务）。目标可帮助您支持您的企业或组织更广泛的可持续发展目标、识别回归并确定潜在改进领域的优先级。
- 实现利用率最大化： 适当调整工作负载规模并实施高效设计，以确保高利用率并最大限度地提高底层硬件的能源效率。由于每台主机的基准功耗，两台以 30% 利用率运行的主机的效率低于一台以 60% 利用率运行的主机。同时，消除或尽可能减少空闲资源、处理和存储，以减少支持工作负载所需的总能源。
- 预测并采用更高效的新硬件和软件产品/服务： 支持您的合作伙伴和供应商进行上游改进，以帮助您减少云工作负载的影响。持续监控和评估更高效的新硬件和软件产品。设计灵活性以允许快速采用高效的新技术。
- 使用托管服务： 在庞大的客户群中共享服务有助于更充分地利用资源，从而减少支持云工作负载所需的基础设施数量。例如，客户可以通过将工作负载迁移到 AWS Cloud 并采用托管服务（例如用于无服务器容器的 AWS Fargate，AWS 在其中大规模运行并负责其高效运行）来分散电力和网络等常见数据中心组件的影响。使用有助于将影响降至最低的托管服务，例如使用 Amazon S3 生命周期配

置将不经常访问的数据自动移动到冷存储，或使用 Amazon EC2 Auto Scaling 来调整容量以满足需求。

- 减少云工作负载的下游影响： 减少使用您的服务所需的能源或资源量。减少或消除客户为了使用您的服务而升级其设备的需求。使用设备场进行测试以了解预期影响，并对客户进行测试以了解使用您服务的实际影响。

定义

在云中实现可持续性包括六个方面的最佳实践：

- 区域选择
- 用户行为模式
- 软件和架构模式
- 数据模式
- 硬件模式
- 开发和部署流程

云中的可持续性是一项持续的工作，主要关注工作负载的所有组件的节能和效率，通过从预置的资源中获得最大收益，并最大限度地减少所需的总资源来达成此目标。这项工作范围很广，包括一开始就选择高效的编程语言、采用现代算法、使用高效的数据存储技术、部署到适当规模的高效计算基础设施，以及最大限度地减少对高功耗最终用户硬件的需求。

最佳实践

主题

- 区域选择
- 用户行为模式
- 软件和架构模式
- 数据模式
- 硬件模式
- 开发和部署模式
- 资源

区域选择

根据您的业务需求和可持续发展目标，选择您将在其中实施工作负载的区域。

以下问题主要针对可持续性方面的注意事项。（有关可持续性问题和最佳实践的列表，请参阅 附录。）

SUS 1：如何选择区域来支持您的可持续发展目标？

选择亚马逊可再生能源项目附近的区域和其电网公布的碳强度低于其他位置（或区域）的区域。

用户行为模式

用户使用您的工作负载和其他资源的方式可以帮助您确定改进措施，以实现可持续性目标。扩展基础设施以持续匹配用户负载，并确保仅部署支持用户所需的最少资源。使服务水平与客户需求保持一致。定位资源以限制用户使用它们所需的网络。移除现有的未使用资产。识别已创建但未使用的资产并停止生成它们。为您的团队成员提供满足其需求的设备，同时最大限度地减少对可持续性的影响。

以下问题主要针对可持续性方面的注意事项：

SUS 2：您如何利用用户行为模式来支持您的可持续发展目标？

用户使用您的工作负载和其他资源的方式可以帮助您确定改进措施，以实现可持续性目标。扩展基础设施以持续匹配用户负载，并确保仅部署支持用户所需的最少资源。使服务水平与客户需求保持一致。定位资源以限制用户使用它们所需的网络。移除现有的未使用资产。识别已创建但未使用的资产并停止生成它们。为您的团队成员提供满足其需求的设备，同时最大限度地减少对可持续性的影响。

扩缩基础设施以匹配用户负载：确定利用率低或利用率为零的时段，缩减资源以消除过剩容量并提高效率。

使 SLA 与可持续发展目标保持一致：定义和更新服务等级协议（SLA，Service Level Agreement），例如可用性或数据留存期，以最大限度地减少支持工作负载所需的资源数量，同时继续满足业务需求。

消除创建和维护未使用资产的需求：分析应用程序资产（例如预编制的报告、数据集和静态图像）和资产访问模式，以识别冗余、利用率低下的情况和潜在的淘汰目标。整合具有冗余内容的已生成资产（例如，具有重叠或公用数据集和输出的月度报告），以消除重复输出时消耗的资源。淘汰未使用的资产（例如，已停售产品的图片）以释放消耗的资源，并减少用于支持工作负载的资源数量。

针对用户位置优化工作负载的地理位置：分析网络访问模式以识别您的客户建立连接的地理位置。选择可减少网络流量必须传输的距离的区域和服务，以减少支持您的工作负载所需的总网络资源。

针对执行的活动优化团队成员资源：优化提供给团队成员的资源，在支持其需求的同时最大程度地降低对可持续性的影响。例如，在利用率高的共享云桌面上，而不是在利用率不高的强力单用户系统上，执行渲染和编译等复杂的操作。

软件和架构模式

实施用于执行负载平滑和保持已部署资源始终如一的高利用率的模式，以最大限度地减少资源消耗。由于用户行为会随着时间的推移而发生变化，因此组件可能会因缺乏使用而变得空闲。修改模式和架构以整合未充分利用的组件，从而提高整体利用率。停用不再需要的组件。了解工作负载组件的性能，并优化消耗资源最多的组件。注意客户用来访问您服务的设备，并实施相应的模式以最大限度地减少设备升级需要。

以下问题主要针对可持续性的注意事项：

SUS 3：您如何利用软件和架构模式来支持您的可持续发展目标？

实施用于执行负载平滑和保持已部署资源始终如一的高利用率的模式，以最大限度地减少资源消耗。由于用户行为会随着时间的推移而发生变化，因此组件可能会因缺乏使用而变得空闲。修改模式和架构以整合未充分利用的组件，从而提高整体利用率。停用不再需要的组件。了解工作负载组件的性能，并优化消耗资源最多的组件。注意客户用来访问您服务的设备，并实施相应的模式以最大限度地减少设备升级需要。

针对异步和计划作业优化软件和架构：使用高效的软件设计和架构来尽可能减少每个工作单元所需的平均资源。实施可促成均匀的组件利用率的机制，以减少任务之间的空闲资源并最大限度地减少负载峰值的影响。

删除或重构很少或没有使用的工作负载组件：监控工作负载活动以识别各个组件的利用率随时间的变化。移除未使用且不再需要的组件，并重构利用率低的组件，以限制资源浪费。

优化消耗最多时间或资源的代码区域：监控工作负载活动以识别消耗最多资源的应用程序组件。优化在这些组件中运行的代码，以最大限度地减少资源使用和提高性能。

优化对客户设备的影响：了解客户用来使用您服务的设备、它们的预期生命周期，以及更换这些组件对财务和可持续性的影响。实施软件模式和架构，以最大限度地减少客户更换和升级设备的需求。例如，使用与旧硬件和操作系统版本向后兼容的代码来实施新功能，或管理有效负载的大小，使其不超过目标设备的存储容量。

使用最能支持数据访问和存储模式的软件模式和架构：了解数据在工作负载中的使用方式、用户使用数据的方式，以及数据的传输和存储方式。选择相应的技术以最大限度地减少数据处理和存储要求。

数据模式

实施用于执行负载平滑和保持已部署资源始终如一的高利用率的模式，以最大限度地减少资源消耗。由于用户行为会随着时间的推移而发生变化，因此组件可能会因缺乏使用而变得空闲。修改模式和架构以整合未充分利用的组件，从而提高整体利用率。停用不再需要的组件。了解工作负载组件的性能，并优化消耗资源最多的组件。注意客户用来访问您服务的设备，并实施相应的模式以最大限度地减少设备升级需要。

以下问题主要针对可持续性方面的注意事项：

SUS 4：您如何利用数据访问模式和使用模式来支持您的可持续发展目标？

实施数据管理实践以减少支持工作负载所需的预置存储，以及使用存储所需的资源。了解您的数据，并使用最能支持数据的商业价值及其使用方式的存储技术和配置。当需求减少时，将数据移到更高效、性能更低的存储中，并删除不再需要的数据。

实施数据分类策略：对数据进行分类以了解其对业务成果的重要性。使用此信息来确定何时可以将数据移动到更节能的存储，或者何时可以安全删除数据。

使用支持您的数据访问模式和存储模式的技术：使用最能支持您的数据访问和存储方式的存储，以在支持您的工作负载的同时最大限度地减少预置的资源。例如，固态硬盘（SSD，Solid State Device）比磁性驱动器更耗能，应该仅用于活跃的数据使用场景。对不常访问的数据使用节能的存档级存储。

使用生命周期策略删除不必要的数据：管理所有数据的生命周期并自动执行删除时间表，以最大限度地减少工作负载的总存储需求。

最大限度地减少数据块存储中的过度预置：要尽可能减少总预置存储，请创建大小分配适合工作负载的数据块存储。随着数据的增长，使用弹性卷扩展存储，而无需调整附加到计算资源的存储大小。定期检查弹性卷并缩小过度配置的卷，以适应当前数据大小。

删除不需要或多余的数据：仅在必要时复制数据，以最大程度地减少消耗的总存储空间。使用备份技术在文件和数据块级别进行重复数据删除。限制使用独立驱动器冗余阵列（RAID）配置，除非需要满足SLA。

使用共享文件系统或对象存储来访问公用数据：采用共享存储和单一事实来源，以避免重复数据删除并降低工作负载的总存储需求。仅在需要时从共享存储中获取数据。分离未使用的卷以释放资源。最大限

度地减少跨网络的数据移动：使用共享存储和访问区域数据存储中的数据，以最大限度地减少支持工作负载数据移动所需的总网络资源。

仅在难以重新创建时备份数据：为了最大限度地减少存储消耗，仅备份具有商业价值或满足合规性要求所必需的数据。检查备份策略并在恢复方案中排除没有价值的临时存储。

硬件模式

寻找机会，通过更改硬件管理实践来降低工作负载可持续性影响。最大限度地减少预置和部署所需的硬件数量，并为您的各项工作负载选择最高效的硬件。

以下问题主要针对可持续性方面的注意事项：

SUS 5：您的硬件管理和使用实践如何支持您的可持续发展目标？

寻找机会，通过更改硬件管理实践来降低工作负载可持续性影响。最大限度地减少预置和部署所需的硬件数量，并为您的各项工作负载选择最高效的硬件。

使用最少的硬件来满足您的需求：通过使用云的功能，您可以对工作负载实施进行频繁更改。在需求变化时更新已部署的组件。

使用影响最小的实例类型：持续监控新实例类型的发布并利用能效改进，包括那些旨在支持特定工作负载（例如机器学习训练和推理以及视频转码）的实例类型。

使用托管服务：托管服务将维持已部署硬件的高平均利用率和可持续性优化的责任转移给 AWS。使用托管服务将服务的可持续性影响分散到服务的所有租户，从而减少您的个人份额。

优化您对 GPU 的使用：图形处理单元（GPU，Graphics Processing Unit）可能是高功耗的来源，许多 GPU 工作负载是高度可变的，例如渲染、转码以及机器学习训练和建模。仅在需要时运行 GPU 实例，并在不需要时自动停用它们，以最大限度地减少资源消耗。

开发和部署模式

寻找机会，通过对开发、测试和部署实践进行更改来降低可持续性影响。

以下问题主要针对可持续性方面的注意事项：

SUS 6：您的开发和部署流程如何支持您的可持续发展目标？

寻找机会，通过对开发、测试和部署实践进行更改来降低可持续性影响。

采用可以快速引入可持续性改进的方法：在将潜在改进部署到生产环境之前，先进行测试和验证。在计算改进的潜在未来收益时，考虑测试成本。开发低成本的测试方法，以实现细微的改进。

让您的工作负载保持最新：最新的操作系统、库和应用程序可以提高工作负载效率，并简化更高效技术的采用。最新的软件可能还包括更准确地衡量工作负载对可持续性的影响的功能，因为供应商提供的功能是为了满足其自身的可持续性目标。

提高构建环境的利用率：使用自动化功能和基础设施即代码，在需要时启动预生产环境，并在不使用时将其关闭。一种常见模式是安排与开发团队成员的工作时间相吻合的可用时段。休眠是一个有用的工具，它可以保存状态，并且只在需要时才快速将实例上线。使用具有突增容量的实例类型、Spot 实例、弹性数据库服务、容器和其他技术，使开发和测试能力与使用相一致。

使用托管 Device Farm 进行测试：托管式设备场将硬件制造和资源使用的可持续性影响分散到多个租户。托管式设备场提供多种设备类型，使您能够支持不太受欢迎的较旧硬件，并避免不必要的设备升级对客户可持续性的影响。

资源

请参阅以下资源，详细了解可持续性的最佳实践。

白皮书

- 可持续性支柱

视频

- The Climate Pledge

审查流程

需要持续不断对架构进行审查，同时要允许试错，建立良好的研究探索氛围。架构审查本身应该是一个简单流程（数小时，而不是几天），是一种对话，而不是审核。审查架构的目的是找出任何需要解决的关键问题或可以改进之处。审查后应采取一些措施，以改善客户体验。

正如在“关于架构”部分讨论的那样，团队中的每位成员都应该对架构质量负责。我们建议负责架构的团队利用架构完善的框架持续检视架构，而不仅仅只是召开一个正式的审查会议。持续的审查使团队成员能够随着架构的演进不断获得知识体系与对架构认识的更新，并在您推出新功能时改进架构。

AWS Well-Architected Framework 高度借鉴了 AWS 在内部审查系统和服务的方式，并与之保持一致。它基于一套可以影响架构方法的设计原则，并确保不会忽略常见于根因分析 (RCA) 中的那些因素。当内部系统、AWS 服务或客户遇到严重问题时，我们会查看 RCA，寻求改进所用审查流程的可能性。

应在产品生命周期内的关键里程碑阶段和设计阶段早期进行多次审查， 以避免单向决策， 因为它们很难进行更改，然后在正式投入使用之前再次审查。（许多决策都是可逆的，是双向的。这些决策可以采用简单流程。单向决策很难，不可逆，所以在做出决策之前需要更加全面的检查。）进入生产阶段后，您的工作负载会随着您不断添加新功能和更改技术实施而继续演进。架构也会随之演进。您需要遵循良好的架构实践，避免出现架构退化。当架构面临重大变更时，您应遵循一套系统健康规范与流程，包括执行 Well-Architected 审查。

如果您想把审查用作一次性快照或独立的衡量方法，则需要确保让所有相关人员参与对话。我们经常发现，通过审查，团队才第一次真正了解他们实施了什么。在审查其他团队的工作负载时，一种有效的方法是围绕架构展开一系列非正式的对话，在此过程中您可以收集到大多数问题的答案。然后通过一两次会议进行跟进，来帮助您理清思路，或深入了解不明确的方面或已感知的风险。

下面建议了一些召开会议需要准备的事项：

- 配有白板的会议室
- 任何图表或设计说明的打印件
- 需要带外研究来获取答案的问题（例如，“是否已启用加密？”）

完成审查后，您应有一个问题清单，并基于您的业务环境来确定这些问题的优先级。您还需要考虑这些问题对团队日常工作的影响。如能及早解决这些问题，您就可以腾出时间开展创造商业价值的工作，而不是解决重复出现的问题。在解决问题时，您可以反复进行审查，来确认架构的改进效果。

虽然在完成审查后，审查的价值显而易见，但您可能发现新团队在开始时可能会对审查抱有抵触情绪。可以通过与团队沟通审查的益处来解决下列异议：

- “我们实在太忙了！”（一般会在团队准备重大发布时这么说。）
  - 如果您正在为重大发布做准备，您会希望一切进展顺利。审查能够帮助您发现您可能错过的任何问题。
  - 我们建议您在产品生命周期早期执行审查，以及时发现风险，并制定与功能交付路线图一致的规避计划。
- “我们已经没有时间了，结果已成定局！”（一般会在他们面临无法改变的事件 [比如超级碗] 时这么说。）
  - 这些事件是无法改变的。您真的想在不了解架构风险的情况下将它投入使用吗？ 即使不能解决所有问题，您仍然可以编制一个潜在问题处理手册。
- “我们不希望其他人知道我们实施解决方案的秘诀！”
  - 如果您向团队指出 Well-Architected Framework 中的问题，他们不会在这些问题中发现任何商业或技术专有信息。

在您与团队进行多次审核后，您可能会发现一些问题。例如，您可能发现一些团队在某个支柱或主题方面出现较多问题。建议您以全局眼光看待所有审查，找出能够帮助解决这些问题的任何机制、培训或首席工程师会谈方案。

总结

AWS Well-Architected Framework 涵盖了六大支柱，强调了在云中设计和运行可靠、安全、高效、经济实惠且可持续的系统的架构最佳实践。该框架提供了一系列问题清单，来帮助您审查现有或将要实现的架构。它还为每个支柱提供了一组 AWS 最佳实践。在架构中应用该框架将能帮助您打造稳定且高效的系统，从而使您能够将主要精力集中在功能需求上。

贡献者

以下是对本文做出贡献的个人和组织：

- Brian Carlson，Amazon Web Services“架构完善”的操作主管
- Ben Potter，Amazon Web Services Well-Architected 安全主管
- Seth Eliot，Amazon Web Services Well-Architected 可靠性主管
- Eric Pullen，Amazon Web Services 高级解决方案架构师
- Rodney Lester，Amazon Web Services 首席解决方案架构师
- Jon Steele，Amazon Web Services 高级技术客户经理
- Max Ramsay，Amazon Web Services 首席安全解决方案架构师
- Callum Hughes，Amazon Web Services 解决方案架构师
- Aden Leirer，Amazon Web Services Well-Architected 内容计划经理

延伸阅读

AWS Architecture Center

AWS 云合规性

AWS Well-Architected 合作伙伴计划

AWS Well-Architected Tool

AWS Well-Architected 主页

《卓越运营支柱》白皮书

《安全性支柱》白皮书

《可靠性支柱》白皮书

《性能效率支柱》白皮书

部分

《可持续性支柱》白皮书

Amazon Builders' Library

文档修订

要获得有关此白皮书的更新通知，请订阅 RSS 源。

变更	说明	日期
次要更新	在附录中增加了工作量定义和更新的最佳实践。	October 20, 2022
已更新白皮书	增加了可持续性支柱并更新了链接。	December 2, 2021
主要更新	可持续性支柱已添加到框架。	November 20, 2021
次要更新	删除了非包容性用语。	April 22, 2021
次要更新	修复了许多链接。	March 10, 2021
次要更新	贯穿全文的次要编辑更改。	July 15, 2020
新框架的更新	审核并重写大多数问题和答案。	July 8, 2020
已更新白皮书	增加了 AWS Well-Architected Tool 以及指向 AWS Well-Architected 实验室和 AWS Well-Architected 合作伙伴的链接，进行了小的修复以支持框架的多语言版本。	July 1, 2019
已更新白皮书	审核并重写大多数问题和答案，确保问题一次集中在一个主题上。这导致某些之前的问题被拆分成多个问题。向定义中添加了常见术语（工作负载、组件等）。更改了正文中问题的表达以包含描述性文本。	November 1, 2018

已更新白皮书	进行了更新，以简化问题文本、实现答案的标准化和提高可读性。	June 1, 2018
已更新白皮书	将卓越运营移至第一个支柱，并进行重新编写，以此引出其他支柱。更新了其他支柱以反映 AWS 的发展。	November 1, 2017
已更新白皮书	更新了框架，添加了卓越运营支柱，修订并更新了其他支柱，以减少重复并整合从成千上万的客户审查实践中吸取的经验。	November 1, 2016
次要更新	为附录更新了当前的 Amazon CloudWatch Logs 信息。	November 1, 2015
原始版本	发布了 AWS Well-Architected Framework。	October 1, 2015

附录：问题和最佳实践

主题

- 卓越运营
- 安全性
- 可靠性
- 性能效率
- 成本优化
- 可持续性

卓越运营

主题

- 组织
- 准备
- 运营
- 演进

组织

问题

- OPS 1 您如何确定自己的重点？
- OPS 2 如何构建组织结构来为业务成果提供支持？
- OPS 3 组织文化如何为业务成果提供支持？

OPS 1 您如何确定自己的重点？

每个人都需要了解自己在业务成功中扮演的角色。设置共同的目标，以便为资源设定重点。这可以让您的工作效益最大化。

最佳实践

- OPS01-BP01 评估外部客户需求

- OPS01-BP02 评估内部客户需求
- OPS01-BP03 评估监管要求
- OPS01-BP04 评估合规性要求
- OPS01-BP05 评估威胁形势
- OPS01-BP06 评估权衡
- OPS01-BP07 管理收益和风险

OPS01-BP01 评估外部客户需求

让包括业务、开发和运维团队在内的主要利益相关方参与进来，以便确定将工作重心放在哪里来满足外部客户的需求。这可以确保您充分了解实现您期望的业务成果所需的运营支持。

常见反模式：

- 您决定核心业务时间之外不再提供客户支持，但是您还没有查看历史支持请求数据。您不知道这是否会对客户产生影响。
- 您正在开发一项新功能，但尚未与客户沟通，不了解客户是否需要；如果需要，以什么形式提供；并且尚未通过试验来验证交付需求和方法。

建立此最佳实践的好处： 需求得到满足的客户流失的可能性更小。评估和了解外部客户需求将为您提供相关信息，告知您如何通过安排工作的优先级来实现商业价值。

未建立此最佳实践暴露的风险等级： 高

实施指导

- 了解业务需求：包括业务、开发和运营团队在内的利益相关方需要有共同的目标和共同的理解，才能实现业务成功。
  - 审查外部客户的业务目标、需求和重点：让包括业务、开发和运营团队在内的主要利益相关方参与进来，讨论外部客户的目标、需求和重点。这可以确保您充分了解实现业务成果和客户成果所需的运营支持。
  - 建立共识：建立共识，确定工作负载的业务功能、每个团队在运行工作负载方面的角色，以及这些因素如何支持内部和外部客户共同的业务目标。

资源

相关文档：

- AWS Well-Architected Framework 概念 – 反馈循环

OPS01-BP02 评估内部客户需求

让包括业务、开发和运营团队在内的主要利益相关方参与进来，以便确定怎样将工作重心放在内部客户的需求上。这可以确保您充分了解实现业务成果所需的运营支持。

使用这些已明确的重点，将改进工作集中部署在能发挥最大影响（例如，开发团队技能、提高工作负载性能、降低成本、自动化运行手册或增强监控）的方面。要随着需求的变化更新重点。

常见反模式：

- 您决定更改产品团队的 IP 地址分配（没有与他们商议），以便更轻松地管理网络。您不知道这是否会对您的产品团队产生影响。
- 您正在采用一种新的开发工具，但尚未与内部客户沟通，不了解他们是否需要，或者是否与他们的现有实践兼容。
- 您正在实施一个新的监控系统，但尚未与内部客户沟通，不了解他们是否有监控或报告需求需要考虑。

建立此最佳实践的好处： 评估和了解内部客户需求将为您提供相关信息，告知您通过安排工作的优先级来实现商业价值。

未建立此最佳实践暴露的风险等级： 高

实施指导

- 了解业务需求：包括业务、开发和运营团队在内的利益相关方需要有共同的目标和共同的理解，才能实现业务成功。
  - 分析内部客户的业务目标、需求和重点：让包括业务、开发和运营团队在内的主要利益相关方参与进来，讨论内部客户的目标、需求和重点。这可以确保您充分了解实现业务成果和客户成果所需的运营支持。
  - 建立共识：建立共识，确定工作负载的业务功能、每个团队在运行工作负载方面的角色，以及这些因素如何支持内部和外部客户共同的业务目标。

资源

相关文档：

- AWS Well-Architected Framework 概念 – 反馈循环

OPS01-BP03 评估监管要求

确保您了解组织确定的指导方针或义务，它们可能会要求或强调特定的重点。评估内部因素，例如组织策略、标准和要求。验证您是否制定了相应的机制，来识别监管变化。如果未确定监管要求，请确保您已对此决定进行尽职调查。

常见反模式：

- 您正在接受审核，并需要提供内部监管的合规性证明。您不知道自己是否合规，因为您从未评估过合规性要求。
- 您遭受了经济损失。您发现，本可以弥补经济损失的保险取决于您实施的特定安全控制措施，而这些措施并未到位，而且亦非监管所需。
- 您的管理账户被盗用，导致公司网站损坏，并损害了客户的信任。您的内部监管要求使用多重身份验证（MFA，Multifactor Authentication）来保护管理账户。您未使用 MFA 保护管理账户，并受到处罚。

建立此最佳实践的好处： 评估和了解组织对工作负载施行的监管要求将为您提供相关信息，告知您如何通过安排工作的优先级来实现商业价值。

未建立这种最佳实践的情况下暴露的风险等级： 高

实施指导

- 了解监管要求：评估内部监管因素，例如计划或组织策略、计划策略、问题或系统特定策略、标准、程序、基准和准则。验证您是否制定了相应的机制，来识别监管变化。如果未确定监管要求，请确保您已对此决定进行尽职调查。

资源

相关文档：

- AWS Cloud 合规性

OPS01-BP04 评估合规性要求

评估监管合规性要求和行业标准等外部因素，确保您了解自己可能需要遵循或重视的指导原则或义务。如果未确定合规性要求，请确保您已对此决定进行尽职调查。

常见反模式：

- 您正在接受审核，并需要提供行业法规的合规性证明。您不知道自己是否合规，因为您从未评估过合规性要求。
- 您的管理账户被盗用，导致客户数据被下载，并损害了客户的信任。您的行业最佳实践要求使用 MFA 来保护管理账户。您未使用 MFA 保护管理账户，并遭到客户投诉。

建立此最佳实践的好处： 评估和了解适用于工作负载的合规性要求将为您提供相关信息，告知您如何通过安排工作的优先级来实现商业价值。

未建立这种最佳实践的情况下暴露的风险等级： 高

实施指导

- 了解合规性要求：评估监管合规性要求和行业标准等外部因素，确保您了解自己可能需要遵循或重视的指导原则或义务。如果未确定合规性要求，请确保您已对此决定进行尽职调查。
  - 了解监管合规性要求：确定您在法律上有义务满足的监管合规性要求。根据这些要求来确定工作重心。例如，隐私和数据保护法案中规定的义务。
    - AWS 合规性
    - AWS 合规性计划
    - AWS 合规性最新新闻
  - 了解行业标准和最佳实践：确定适用于工作负载的行业标准和最佳实践要求，如支付卡行业数据安全标准（PCI DSS，Payment Card Industry Data Security Standard）。根据这些要求来确定工作重心。
    - AWS 合规性计划
  - 了解内部合规性要求：确定组织制定的合规性要求和最佳实践。根据这些要求来确定工作重心。例如，信息安全策略和数据分类标准。

资源

相关文档：

- AWS Cloud 合规性
- AWS 合规性
- AWS 合规性最新新闻
- AWS 合规性计划

OPS01-BP05 评估威胁形势

评估对业务的威胁（例如竞争、业务风险和负债、运营风险和信息安全威胁），并在风险注册表中维护当前信息。在确定工作重心时，将风险的影响考虑在内。

如示例所示， Well-Architected Framework 强调学习、衡量和改进。它为您提供了一种一致的方法来评估架构，并实施将随着时间推移而扩展的设计。AWS 提供 AWS Well-Architected Tool ，可帮助您在开发之前查看方法、生产前的工作负载状态以及生产中的工作负载状态。您可以将其与最新的 AWS 架构最佳实践进行比较，监控工作负载的整体状态，并深入了解潜在风险。

AWS 客户可以使用针对任务关键型工作负载的指导式 Well-Architected 审核，以 根据 AWS 最佳实践来衡量其架构。企业支持客户可以使用 运营审核，该审核旨在帮助他们找出云中的运营方法所存在的漏洞。

这些审核需要跨团队参与，可帮助各团队在工作负载以及他们在实现成功中的角色方面达成一致的理解。通过审查所确定的需求可以帮助确定您的运营重点。

AWS Trusted Advisor 是一种工具，让您可以访问一组核心检查，这些检查会提出优化建议，帮助确定您的运营重点。 商业和企业支持客户 可以访问其他检查，这些检查重点关注安全性、可靠性、性能和成本优化，可进一步帮助他们帮助确定运营重点。

常见反模式：

- 您在产品中使用的是旧版软件库。对于可能会对工作负载产生意外影响的问题，需要对库进行安全更新，而您忽略了这一点。
- 您的竞争对手刚刚发布了新的产品版本，可以解决许多客户对您产品的投诉。您没有优先解决这些已知问题。
- 监管机构一直在追查像您这样的不符合法律法规要求的公司。您没有优先处理任何未解决的合规性要求。

建立此最佳实践的好处： 发现并了解对组织和工作负载的威胁后，您可以确定要解决的威胁、需解决的威胁的优先级以及执行操作所需的资源。

未建立这种最佳实践的情况下暴露的风险等级： 中

实施指导

- 评估威胁形势：评估对业务的威胁（例如竞争、业务风险和负债、运营风险和信息安全威胁），以便您在确定工作重心时可以将其影响考虑在内。

- AWS 最新安全公告
- AWS Trusted Advisor
- 维护威胁模型：建立并维护威胁模型，确定潜在威胁、计划内和已实施的缓解措施及其优先级。审核威胁酿成意外事件的可能性、从意外事件恢复的成本和预期造成的危害，以及防止这些意外事件发生的成本。根据威胁模型内容的更改修订优先级。

资源

相关文档：

- AWS Cloud 合规性
- AWS 最新安全公告
- AWS Trusted Advisor

OPS01-BP06 评估权衡

在有冲突的利益或替代方法之间做出权衡并评估其影响，以便在确定工作重心或选择行动方案时做出明智的决策。例如，加快新功能上市的速度可能会比成本优化更重要，或者您可以为非关系数据选择关系数据库，以简化迁移系统的工作，而不是迁移到针对您的数据类型优化的数据库和更新您的应用程序。

AWS 可以帮您向团队介绍 AWS 及其服务，让他们深入了解自己的选择会如何影响工作负载。您应该使用由 AWS Support（AWS 知识中心，AWS 开发论坛和 AWS Support 中心）和 AWS 文档提供的资源 来培训您的团队。请通过 AWS Support 中心联系 AWS Support，获取与 AWS 问题相关的帮助。

AWS 还在 Amazon Builders' Library 中分享了我们通过 AWS 运营 学到的最佳实践和模式。您可以通过 AWS Blog 和 AWS 官方播客，获得各种其他有用信息。

常见反模式：

- 您正在使用关系数据库管理时间序列和非关系数据。有一些数据库选项经过优化后可以支持您正在使用的数据类型，但是您没有意识到这些好处，因为您尚未评估解决方案之间的权衡。
- 您的投资者要求您证明符合支付卡行业数据安全标准 (PCI DSS)。您没有在满足他们的要求和继续进行当前的开发工作之间进行权衡，而是在没有证明合规性的情况下继续进行开发工作。投资者出于对平台安全性和投资的担忧停止了对公司的支持。

建立此最佳实践的好处： 了解您的选择产生的影响和后果可以帮助您排定选择的优先顺序。

未建立此最佳实践暴露的风险等级： 中

实施指导

- 评估权衡：在利益互有冲突的目标之间做出权衡并评估其影响，以便在确定工作重心时做出明智的决策。例如，加快新功能上市的速度可能比成本优化更重要。
- AWS 可以帮您向团队介绍 AWS 及其服务，让他们深入了解自己的选择会如何影响工作负载。您应该使用由 AWS Support（AWS 知识中心、AWS 开发论坛和 AWS Support 中心）和 AWS 文档提供的资源来培训您的团队。请通过 AWS Support 中心联系 AWS Support，获取与 AWS 问题相关的帮助。
- AWS 还在 Amazon Builders' Library 中分享了我们通过 AWS 运营学到的最佳实践和模式。您可以通过 AWS Blog 和 AWS 官方播客，获得各种其他有用信息。

资源

相关文档：

- AWS Blog
- AWS Cloud 合规性
- AWS 开发论坛
- AWS 文档
- AWS 知识中心
- AWS Support
- AWS Support 中心
- Amazon Builders' Library
- AWS 官方播客

OPS01-BP07 管理收益和风险

管理收益和风险，以便在确定工作重心时做出明智的决策。例如，为了向客户提供重要的新功能，部署仍存在未决问题的工作负载是可以接受的。这可能会降低相关风险，或者允许风险继续存在可能会令人无法接受，在这种情况下，您将采取措施来化解风险。

您可能会发现，您需要在某个时间点侧重于一小部分运营重点。长期使用平衡的方法来确保所需能力的发展和风险管理。要随着需求的变化更新重点

常见反模式：

- 您决定设置一个库，这是您的一位开发人员在互联网上找到的万能库。您尚未评估从未知来源采用此库的风险，也不知道它是否包含漏洞或恶意代码。
- 您决定开发和部署新功能，而不是修复现有问题。您尚未评估在部署功能之前将问题继续留存的风险，也无从知晓会对客户产生哪些影响。
- 由于合规团队提出了未指明的顾虑，您决定不部署客户频繁请求的功能。

建立此最佳实践的好处： 确定选择可以带来的收益并了解组织所面临的风险有助于您做出明智的决定。

未建立此最佳实践暴露的风险等级： 低

实施指导

- 管理收益和风险：在决策的收益与涉及的风险之间取得平衡。
  - 确定收益：根据业务目标、需求和优先事项来确定收益。例如上市时间、安全性、可靠性、性能和成本等。
  - 确定风险：根据业务目标、需求和优先事项来确定风险。例如上市时间、安全性、可靠性、性能和成本等。
  - 对照风险评估收益并做出明智决策：根据包括业务、开发和运营团队在内的主要利益相关方的目标、需求和优先事项，确定收益和风险的影响。对照发生风险的可能性及其影响产生的成本来评估收益的价值。例如，强调上市速度而不是可靠性可能会带来竞争优势。但是如果出现可靠性问题，就可能会导致正常运行时间缩短。

OPS 2 如何构建组织结构来为业务成果提供支持？

您的团队必须了解他们在实现业务成果方面所发挥的作用。团队需要了解自己在其他团队获得成功过程中所扮演的角色、其他团队在他们获得成功的过程中所扮演的角色，并设定共同的目标。了解责任分配、所有权归属、决策制定方式以及决策者将有助于集中精力，最大限度地发挥团队的优势。

最佳实践

- OPS02-BP01 确定资源所有者
- OPS02-BP02 确定流程和程序所有者
- OPS02-BP03 确定对运营活动绩效负责的所有者
- OPS02-BP04 团队成员知道自己的责任
- OPS02-BP05 制定用于确定责任和所有权的机制
- OPS02-BP06 制定用于请求添加、更改和例外的机制

- OPS02-BP07 预先定义或协商团队间的职责

OPS02-BP01 确定资源所有者

了解对每个应用程序、工作负载、平台和基础设施组件拥有所有权的人员，各组件提供了哪些商业价值，以及为什么具有这种所有权。了解这些独立组件的商业价值以及它们如何支持业务成果将为它们应用的流程和程序提供信息。

建立此最佳实践的好处： 了解所有权可以确定谁有权批准改进和/或实施改进。

未建立这种最佳实践的情况下暴露的风险等级： 高

实施指导

- 确定资源所有者：定义所有权对于环境中的资源使用案例的意义。指定并记录资源所有者，至少包括名称、联系信息、组织和团队。使用元数据（例如标签或资源组）将资源所有权信息与资源存储在一起。使用 AWS Organizations 构建账户并实施策略，确保捕获所有权和联系信息。
  - 定义所有权形式及其分配方式：在您的组织中，不同的使用案例对所有权的定义可能也有所不同。您可能希望将工作负载所有者定义为承担工作负载操作的风险和责任，并最终有权对工作负载做出决策的个人。您可能希望根据财务或行政责任来定义所有权，这样所有权将转移到上级组织。开发人员可以是开发环境的所有者，并对运营引发的事件负责。他们的产品负责人可能要对与开发环境运营相关的财务成本负责。
  - 定义组织、账户、资源集合或单个组件的所有者：在适于访问、易于发现的位置定义和记录所有权。及时更新定义和所有权的详细信息。
  - 在资源元数据中捕获所有权：使用元数据（例如标签或资源组）捕获资源所有权，详细说明所有权和联系信息。使用 AWS Organizations 确定账户结构，并确保捕获所有权和联系信息。

OPS02-BP02 确定流程和程序所有者

了解谁对各个流程和程序的定义拥有所有权、为何使用这些特定的流程和程序，以及为什么存在这种所有权。了解使用特定流程和程序的原因将有助于发现改进机会。

建立此最佳实践的好处： 了解所有权可以确定谁有权批准改进和/或实施改进。

未建立这种最佳实践的情况下暴露的风险等级： 高

实施指导

- 确定负责定义流程和程序的所有者：捕获环境中使用的流程和程序，以及负责其定义的个人或团队。

- 确定流程和程序：确定为支持工作负载而开展的运营活动。将这些活动记录在易于发现的位置。
- 确定谁负责定义流程或程序：唯一标识负责活动规范的个人或团队。他们负责确保由技能娴熟且具有正确的权限、访问权限和工具的团队成员来成功执行活动。如果执行活动时遇到问题，那么执行活动的团队成员有责任提供详细反馈，推进活动改进。
- 在活动构件的元数据中捕获所有权：在 AWS Systems Manager 之类的服务中通过文档和 AWS Lambda 函数自动执行的程序支持以标签形式捕获元数据信息。使用标签或资源组捕获资源所有权，详细说明所有权和联系信息。使用 AWS Organizations 创建标记策略，并确保捕获所有权和联系信息。

OPS02-BP03 确定对运营活动绩效负责的所有者

了解谁负责针对定义的工作负载执行特定活动，以及为什么负责。了解谁负责执行活动可让我们知晓谁来开展活动、验证结果并向活动所有者提供反馈。

建立此最佳实践的好处： 了解谁负责执行活动可让我们知晓需要采取行动时要通知谁以及谁将执行操作、验证结果并向活动所有者提供反馈。

未建立这种最佳实践的情况下暴露的风险等级： 高

实施指导

- 确定对运营活动绩效负责的所有者：了解环境中使用的流程和程序的责任分配
  - 确定流程和程序：确定为支持工作负载而开展的运营活动。将这些活动记录在易于发现的位置。
  - 确定各项活动的执行负责人：确定负责某项活动的团队。确保他们具有活动的详细信息，具备执行活动所需的技能以及正确的权限、访问权限和工具。他们必须了解活动执行条件（例如基于某个事件或计划）。显示这些信息，以便组织成员确定针对特定需求他们需要联系的人员（团队或个人）。

OPS02-BP04 团队成员知道自己的责任

了解您的角色具有哪些责任以及如何为业务成果做出贡献可帮助您确定任务的优先级以及自身角色的重要性。这使团队成员能够了解需求并做出适当响应。

建立此最佳实践的好处： 了解您的责任可帮助明确所做的决定、采取的行动以及需要将哪些活动交给适当的所有者。

未建立这种最佳实践的情况下暴露的风险等级： 高

实施指导

- 确保团队成员了解各自的角色和责任：确定团队成员的角色和责任，并确保他们了解对其角色的期望。显示这些信息，以便组织成员确定针对特定需求他们需要联系的人员（团队或个人）。

OPS02-BP05 制定用于确定责任和所有权的机制

在未确定个人或团队时，要为有权分配所有权或计划满足该需求的人定义升级路径。

建立此最佳实践的好处： 了解谁负责或拥有所有权可使您与合适的团队或团队成员联系，以提出请求或转换任务。确定谁有权分配责任或所有权或计划满足需求，可以降低不作为的风险并减少无法满足的需求。

未建立此最佳实践暴露的风险等级： 高

实施指导

- 制定用于确定责任和所有权的机制：为组织成员提供可访问机制，以发现和确定所有权和责任。这些机制将使他们能够根据特定的需求确定相关的联系人（团队或个人）。

OPS02-BP06 制定用于请求添加、更改和例外的机制

您可以向流程、程序和资源的所有者提出请求。对收益和风险进行评估之后，做出明智的决定，批准可行的和确认合适的请求。

建立此最佳实践的好处： 务必要建立相应机制，来请求添加、更改和例外，为团队的活动提供支持。如果没有这样的机制，当前状态将会限制创新。

未建立这种最佳实践的情况下暴露的风险等级： 中

实施指导

- 制定用于请求添加、更改和例外的机制：如果标准僵化，创新就会受到限制。为组织成员制定机制，向流程、程序和资源的所有者提出请求，以支持其业务需求。

OPS02-BP07 预先定义或协商团队间的职责

团队之间具有明确或协商好的协议，规定了团队之间的合作和相互支持方式（例如响应时间、服务级别目标或服务等级协议）。了解团队工作对业务成果的影响以及其他团队和组织的成果可以确定其任务的优先级，并帮助他们做出适当的响应。

当责任和所有权不确定或未知时，您将面临以下风险：没有及时处理必要的活动，以及在处理这些需求时可能出现工作冗余和潜在冲突。

建立此最佳实践的好处： 确定团队间的责任、相关目标和传达需求的方法，可以简化请求流程，并有助于确保提供必要的信息。这可以减少团队间因转换任务造成的延迟，并有助于为实现业务成果提供支持。

未建立此最佳实践暴露的风险等级： 低

实施指导

- 预先界定或协商团队间的职责：指定团队沟通方法以及相互支持所需的信息，有助于最大程度地减少因反复审核和澄清请求而造成的延误。如果就期望（例如响应时间或完成时间）达成了特定协议，团队将能够适当地制定有效的计划和资源。

OPS 3 组织文化如何为业务成果提供支持？

为您的团队成员提供支持，以便他们可以更有效地采取行动并为您的业务成果提供支持。

最佳实践

- OPS03-BP01 高管支持
- OPS03-BP02 赋能团队成员在结果有风险时采取行动
- OPS03-BP03 鼓励上报
- OPS03-BP04 沟通及时、清晰、可行
- OPS03-BP05 鼓励试验
- OPS03-BP06 支持和鼓励团队成员保持和增强他们的技能组合
- OPS03-BP07 为团队配置适当的资源
- OPS03-BP08 鼓励在团队内部和团队之间提出不同的观点

OPS03-BP01 高管支持

高层领导明确为组织设定期望并评估是否成功。高层领导是采用最佳实践和组织发展的发起人、倡导者和推动者

建立此最佳实践的好处： 积极参与的领导层、表达清晰的期望和共同的目标可确保团队成员了解组织对自己的期望。对成功进行评估可以确定阻碍成功的障碍，以便通过发起人、倡导者或他们的代表进行干预，从而消除这些障碍。

未建立这种最佳实践的情况下暴露的风险等级： 高

实施指导

- 高管支持：高级领导层明确为组织设定期望并评估是否成功。高层领导是采用最佳实践和组织发展的发起人、倡导者和推动者
  - 设定期望：为您的组织制定和发布目标，包括目标衡量方式。
  - 跟踪目标实现情况：定期衡量目标的逐步实现情况并分享结果，以便在结果有风险时可以采取适当的措施。
  - 为实现目标提供必要的资源：定期审核资源是否仍然合适，或者根据以下项目决定是否需要添加资源：新信息、目标变更、责任或您的业务环境。
  - 支持您的团队：与团队保持沟通，以便您了解他们的进展情况以及是否受到外部因素的影响。团队受外部因素影响时，需重新评估目标并适当地调整目标。确定阻碍团队进度的障碍。代表团队做出行动，帮助消除障碍，除去不必要的负担。
  - 推动最佳实践的采用：认可可量化收益的最佳实践并确定创建者和采用者。鼓励进一步采用，实现更大收益。
  - 推动团队发展：打造持续改进的文化。鼓励个人和组织的成长与发展。制定长期目标并为此奋斗，逐步实现成功。根据需求、业务目标和业务环境的变化调整此愿景。

OPS03-BP02 赋能团队成员在结果有风险时采取行动

工作负载所有者定义了指南和范围，赋能团队成员在结果有风险时做出响应。当事件超出定义的范围时，使用上报机制获取指示。

建立此最佳实践的好处： 在早期进行测试和验证更改，可以将解决问题的成本降至最低，并降低对客户的影响。经过测试之后再部署，可以最大限度地减少错误。

未建立此最佳实践暴露的风险等级： 高

实施指导

- 赋能团队成员在结果有风险时采取行动：为您的团队成员提供权限、工具和机会，实践有效响应所需的技能。
  - 为您的团队成员提供机会，实践响应所需的技能：提供替代的安全环境，以便在其中安全地对流程和程序进行测试和培训。进行实际演练，让团队成员在模拟的安全环境中获得响应现实意外事件的经验。
  - 定义并确认团队成员采取行动的权限：通过分配对他们所支持的工作负载和组件的权限和访问权限，来明确定义团队成员采取行动的权限。确认授权他们在结果存在风险时采取行动。

OPS03-BP03 鼓励上报

团队成员具有相应机制，如果他们认为结果存在风险，鼓励他们向决策者和利益相关者上报问题。应经常尽早上报，以便能够确定风险，并防止造成意外事件。

未建立这种最佳实践的情况下暴露的风险等级： 高

实施指导

- 鼓励经常尽早上报：从组织的角度认可，经常尽早上报是一种最佳实践。组织确认并接受，上报的内容最终可能证明并无依据，但最好要抓住机会预防意外事件的发生，而不要因为没有上报而错失机会。
  - 制定上报机制：我们设定明文程序来确定何时应上报以及上报方式。记录权限逐级提升（以采取行动或批准行动）的人员及其联系信息。上报应一直持续，直到团队成员认为他们已将风险移交给能够化解风险的人员，或者他们已联系到对运营该工作负载的风险和责任负责的人员。（即可以最终对工作负载做出决策的人员）。上报内容应包括风险的性质、工作负载的严重性、受影响的人、受到的影响以及紧迫性（即预计影响发生的时间）。
  - 保护上报的员工：制定政策保护团队成员，如果他们上报关于决策者或利益相关者未做出响应的问题，保护他们免遭报复。制定适当的机制，确定是否发生了这种情况并做出相应响应。

OPS03-BP04 沟通及时、清晰、可行

制定相应机制，用于将已知风险和计划内事件及时通知给团队成员。提供必要的相关信息、详细信息和时间（如果可能），为确定是否需要采取措施、需要采取什么措施以及及时采取措施提供支持。例如，提供软件漏洞通知可以加快修补过程；或者，提供计划内促销活动的通知可以实施变更冻结以避免发生服务中断的风险。

可以将计划内事件记录在变更日历或维护时间表中，以便团队成员可以确定哪些活动待处理。

在 AWS 上，可以使用 AWS Systems Manager 变更日历 来记录这些详细信息。它支持对日历状态进行程序检查，以确定日历在特定时间点对活动是打开还是关闭。运营活动可以根据特定的 已批准 时间窗进行规划，这些时间窗是为潜在的干扰性活动预留的。AWS Systems Manager 维护时段允许您根据实例和其他 支持资源 安排活动，从而自动执行活动并发现这些活动。

未建立这种最佳实践的情况下暴露的风险等级： 高

实施指导

- 沟通及时、清晰、可行：制定合理的机制，以清晰、可行的方式提供风险或计划内事件的通知，而且要引起足够的注意，以做出适当的响应。

- 在变更日历上记录计划内活动并发送通知：提供可访问的信息源，可在其中搜索计划内事件。通知来自同一系统的计划内事件。
- 跟踪可能影响工作负载的事件和活动：监控漏洞通知和补丁程序信息，以了解外部漏洞以及与工作负载组件相关的潜在风险。向团队成员发送通知，以便他们可以采取措施。

资源

相关文档：

- AWS Systems Manager 变更日历
- AWS Systems Manager 维护时段来自动执行修补托管系统的过程和安排修补活动

OPS03-BP05 鼓励试验

试验可加快学习速度，并使团队成员保持兴趣和参与热情。取得非预期结果也算试验成功，因为这种试验确定了无法实现成功的途径。团队成员不会因为取得非预期结果的成功试验而受到惩罚。创新必须进行试验，才能将创意转化为成果。

未建立此最佳实践暴露的风险等级： 中

实施指导

- 鼓励试验：鼓励试验，为学习和创新提供支持。
  - 试验各种技术：鼓励对现在或将来可能适用于实现业务成果的技术进行试验。这些知识可以为将来的创新提供有用信息。
  - 试验要目标明确：鼓励团队成员为实现特定目标而进行试验，或者对在不久的将来可能适用的技术进行试验。这些知识可以为创新提供有用信息。
  - 精心安排时间进行试验：将试验时间安排在团队成员没有日常工作的时候，以便他们专注于试验。
  - 提供资源以支持试验：为开展试验所需的资源提供资金（例如软件或云资源）。
  - 认可成功：认可试验产出的价值。应理解没有取得预期结果的试验也是成功的，这些试验帮助我们确定了无法取得成功的途径。团队成员不应因为试验未取得预期结果而受到处罚。

OPS03-BP06 支持和鼓励团队成员保持和增强他们的技能组合

团队必须增强自己的技能组合，以采用新技术；并随需求和职责的变化继续提供支持，以支持工作负载。新技术技能的增强通常能提升团队成员满意度并支持创新。支持您的团队成员获取和维护行业认

证，以验证和认可他们不断增强的技能。进行交叉培训，以促进知识转移并降低在您失去熟练掌握机构知识、经验丰富的团队成员时产生重大影响的风险。专门安排时间进行学习。

AWS 提供了许多资源，包括 AWS 入门资源中心， AWS Blog， AWS 在线技术讲座， AWS 活动和网络研讨会，以及 AWS Well-Architected 实验室，这些资源提供了指导、示例和详细演练，用以培训您的团队。

AWS 还在 Amazon Builders' Library 中分享了我们通过 AWS 运营 学到的最佳实践和模式 ；并通过 AWS Blog 和 AWS 官方播客分享了各种实用的教材。

您应该利用 AWS 提供的教育资源，例如 Well-Architected 实验室、 AWS Support （ AWS 知识中心， AWS 开发论坛和 AWS Support 中心）和 AWS 文档 来培训您的团队。请通过 AWS Support 中心联系 AWS Support，获取与 AWS 问题相关的帮助。

AWS 培训与认证 提供了一些免费培训，可以通过自定进度的数字课程，学习 AWS 的基础知识。您还可以注册讲师指导培训，进一步帮助培养您团队的 AWS 技能。

未建立此最佳实践暴露的风险等级： 中

实施指导

- 支持和鼓励团队成员保持和增强他们的技能组合：我们必须不断学习，这样才能采用新技术、支持创新，并随需求和责任的变化继续提供支持，以支持工作负载。
  - 提供教育资源：专门安排时间，提供培训材料、实验室资源，并支持参加会议和加入专业组织，以便有机会向讲师和同行学习。为初级团队成员提供与高级团队成员接触的机会，可以请高级团队成员作为他们的导师，或允许他们跟随高级团队成员学习并了解方法和技能。鼓励学习与工作没有直接关系的内容，拓展视野。
  - 团队教育和跨团队合作：为团队成员的继续教育需求做好规划。为团队成员提供（临时或永久）加入其他团队的机会，以分享技能和最佳实践，惠及整个组织
  - 支持获取和维护行业认证：支持团队成员获取和维护行业认证，以验证他们所学到的知识并认可他们的成就。

资源

相关文档：

- AWS 入门资源中心
- AWS Blog
- AWS Cloud 合规性

- AWS 开发论坛
- AWS 文档
- AWS 在线技术讲座
- AWS 活动和网络研讨会
- AWS 知识中心
- AWS Support
- AWS 培训与认证
- AWS Well-Architected 实验室，
- Amazon Builders' Library
- AWS 官方播客.

OPS03-BP07 为团队配置适当的资源

培养团队成员的能力，并提供工具和资源来支持工作负载需求。团队成员超负荷工作会增加人为错误导致事故发生的风险。投资于工具和资源（例如，对频繁执行的活动实现自动化）可以提高团队的效率，让他们为其他活动提供支持。

未建立这种最佳实践的情况下暴露的风险等级： 中

实施指导

- 为团队配置适当的资源：确保您了解团队取得的成功，以及推动团队成功或导致团队不成功的因素。采取行动为团队提供适当的资源。
  - 了解团队绩效：衡量团队运营成果的实现和资产的开发。跟踪输出和错误率随时间发生的变化。与团队沟通，了解会对他们工作产生影响的挑战（例如责任增加、技术变化、人员流失或支持的客户增加）。
  - 了解对团队绩效的影响：与团队保持沟通，以便您了解他们的进展情况以及是否受到外部因素的影响。团队受外部因素影响时，需重新评估目标并适当地调整目标。确定阻碍团队进度的障碍。代表团队做出行动，帮助消除障碍，除去不必要的负担。
  - 为团队提供必要资源，助力团队取得成功：定期审核资源是否仍然合适，或者是否需要添加新资源，并做出适当的调整，为团队提供支持。

OPS03-BP08 鼓励在团队内部和团队之间提出不同的观点

利用跨组织的多样性来寻求多种独特的见解。利用这种见解提高创新能力、对您的假设提出质疑，并降低确认偏差的风险。在团队内部提升包容性、多样性和可达性有助于获取有益的见解。

组织文化会直接影响团队成员的工作满意度和保留率。增强团队成员的参与度和能力，助力业务成功。

未建立这种最佳实践的情况下暴露的风险等级： 低

实施指导

- 寻求不同的观点和视角：鼓励所有人做出贡献。为弱势群体发声。在会议中轮换角色和职责。
  - 扩展角色和职责：让团队成员有机会尝试他们可能不会担任的角色。他们将从角色以及与其他团队成员的互动中获得经验和见解，而之前他们可能没有机会与他们互动。他们会将自己的经验和见解赋予新角色，以及就此与新团队成员沟通交流。随着见解的不断增多，可能会出现新的商机，或者可能会发现新的改进机会。让团队成员轮流体验他人日常执行的任务，了解执行这些任务的需求和影响。
  - 提供安全舒适的环境：制定相应的政策和控制措施，保护组织内团队成员的身心健康。团队成员应该能够彼此敞开心扉，而不是处在会受到报复的担惊受怕之中。当团队成员处于安全舒适的环境中时，才能有更高的参与热情、更高的工作成效。您的组织越多元化，您就越能更好地理解您所支持的人，包括客户。当您的团队成员感到舒服自在、能够畅所欲言并确信自己的意见会被听到时，他们会更愿意分享有价值的见解（例如营销机会、可访问性需求、尚待开发的细分市场、环境中未被发现的风险）。
  - 让团队成员充分参与：为员工提供必要的资源，让他们充分参与到所有与工作相关的活动中。每天都面临各种挑战的团队成员已不断开发应对这些挑战的技能。这些开发的技能独一无二，可以为您的组织带来巨大的效益。根据需要为团队成员提供住所将有助于他们增加对团队的贡献，从而提升您的效益。

准备

问题

- OPS 4 如何设计工作负载以便自己了解其状态？
- OPS 5 如何减少缺陷、简化修复和改进生产流程？
- OPS 6 您如何缓解部署风险？
- OPS 7 如何知道您已经准备好支持某种工作负载？

OPS 4 如何设计工作负载以便自己了解其状态？

将工作负载设计成能够提供所有组件（例如指标、日志和跟踪信息）的必要信息，以便您了解其内部状态。这让您能够在适当的时候提供有效的响应。

最佳实践

- OPS04-BP01 实施应用程序遥测
- OPS04-BP02 实施和配置工作负载遥测
- OPS04-BP03 实施用户活动遥测
- OPS04-BP04 实施依赖项遥测
- OPS04-BP05 实施事务跟踪

OPS04-BP01 实施应用程序遥测

应用程序遥测是实现工作负载可观测性的基础。您的应用程序应该发送遥测数据，提供对应用程序状态以及所实现业务成果的洞察。从故障排除到衡量新功能的影响，应用程序遥测可以为您构建、操作和演进工作负载的方法提供信息。

应用程序遥测数据包括指标和日志。指标是诊断信息，例如您的脉搏和体温。所有指标结合在一起，用于描述应用程序的状态。收集一段时间的指标，以便用于制定基准和检测异常。日志是应用程序发送的消息，说明其内部状态或所发生的事件。所记录事件的例子包括错误代码、事务标识符以及用户操作。

期望结果：

- 应用程序发送指标和日志，提供对其运行状况以及所取得业务成果的洞察。
- 工作负载中所有应用程序的指标和日志集中存储。

常见反模式：

- 您的应用程序无法发出遥测。出现问题时，只能通过客户获知。
- 客户反映您的应用程序没有响应。由于没有遥测，如果不亲自使用应用程序来了解当前的用户体验，就无法确认问题的存在，也无法确定问题的特征。

建立此最佳实践的好处：

- 您可以了解应用程序的运行状况、用户体验以及所取得的业务成果。
- 您可以更快地对应用程序运行状况中的更改做出反应。
- 您可以了解应用程序运行状况趋势。
- 您可以做出明智的决定来改进应用程序。
- 您可以更快地检测并解决应用程序问题。

未建立这种最佳实践的情况下暴露的风险等级： 高

实施指导

实施应用程序遥测由三个步骤组成：确定存储遥测数据的位置，确定描述应用程序状态的遥测数据，以及指示应用程序发送遥测数据。

例如，电子商务公司采用了基于微服务的架构。作为其架构设计流程的一部分，他们确定了可以帮助了解各个微服务状态的应用程序遥测。例如，用户购物车服务可以针对商品添加到购物车、放弃购物车以及将商品添加到购物车所用时间长度等事件，发送遥测数据。所有微服务将记录错误、警告和事务信息。遥测数据可以发送到 Amazon CloudWatch 进行存储和分析。

实施步骤

第一步是针对工作负载中的应用程序，确定用于遥测数据存储的集中位置。如果您还没有现有平台，Amazon CloudWatch 会提供遥测数据收集、控制面板、分析和事件生成功能。

若要确定您需要哪些遥测数据，可以从以下问题开始着手：

- 我的应用程序是否正常运行？
- 我的应用程序是否实现了业务成果？

  您的应用程序应该会发送日志和指标，综合起来即可解答这些问题。如果您无法利用现有的应用程序遥测解答这些问题，请与业务和工程设计利益相关方合作，创建可以做到这一点的遥测列表。在确定和开发新应用程序遥测的过程中，您可以请求 AWS 账户团队向您提供专家技术建议。

  在确定其他应用程序遥测之后，与工程设计利益相关方合作来检测应用程序。 适用于 OpenTelemetry 的 AWS Distro 提供收集应用程序遥测数据的 API、库和代理。 此示例展示了如何使用自定义指标检测 JavaScript 应用程序。

  客户如果希望了解 AWS 提供的可观测性服务，可以亲自参加 可观测性研讨会 ，或者请求 AWS 账户团队的支持来提供指导。此研讨会引导您了解 AWS 上的可观测性解决方案，并提供如何使用这些解决方案的动手实践示例。

  如需更深入地了解应用程序遥测，请阅读 Amazon Builders' Library 中的 检测分布式系统的运营可见性 文章。它说明了 Amazon 如何检测应用程序，并可供您用作开发自己的检测准则的指南。

实施计划的工作量级别： 中

资源

相关最佳实践：

the section called “OPS04-BP02 实施和配置工作负载遥测” – 应用程序遥测是工作负载遥测的组件。为了理解工作负载的整体运行状况，您需要了解组成工作负载的单独应用程序的运行状况。

the section called “OPS04-BP03 实施用户活动遥测” – 用户活动遥测通常是应用程序遥测的子集。添加商品到购物车、点击流或者已完成事务等用户活动可以提供对用户体验的洞察。

the section called “OPS04-BP04 实施依赖项遥测” – 依赖项检查与应用程序遥测相关，可以在应用程序中检测。如果您的应用程序依赖于外部依赖项，例如 DNS 或数据库，则应用程序可以发送有关可访问性、超时和其他事件的指标和日志。

the section called “OPS04-BP05 实施事务跟踪” – 跨工作负载跟踪事务需要各个应用程序发送有关如何处理共享事件的信息。单独应用程序处理这些事件的方式通过其应用程序遥测发送。

the section called “OPS08-BP02 定义工作负载指标” – 工作负载指标是工作负载运行状况的主要指标。主要应用程序指标是工作负载指标的一部分。

相关文档：

- AWS Builders Library – 检测分布式系统的运营可见性
- 适用于 OpenTelemetry 的 AWS Distro
- AWS Well-Architected 卓越运营白皮书 – 设计遥测
- 使用筛选条件根据日志事件创建指标
- 使用 Amazon CloudWatch 实施日志记录和监控
- 使用适用于 OpenTelemetry 的 AWS Distro 监控应用程序运行状况和性能
- 新增 – 如何使用 Amazon CloudWatch 代理更好地监控自定义应用程序指标
- AWS 上的可观测性
- 场景 – 发布指标到 CloudWatch
- 开始构建 – 如何高效地监控应用程序
- 将 CloudWatch 与 AWS SDK 结合使用

相关视频：

- AWS re:Invent 2021 – 开源方式的可观测性

- 使用 CloudWatch 代理从 Amazon EC2 实例收集指标和日志
- 如何为 AWS 工作负载轻松设置应用程序监控 – AWS 在线技术讲座
- 掌握无服务器应用程序的可观测性 – AWS 在线技术讲座
- AWS 上的开源可观测性 – AWS 虚拟研讨会

相关示例：

- AWS 日志记录和监控示例资源
- AWS 解决方案：Amazon CloudWatch 监控框架
- AWS 解决方案：集中式日志记录
- 可观测性研讨会

OPS04-BP02 实施和配置工作负载遥测

设计和配置工作负载，使其能够发出关于其内部状态和当前状态的信息，例如 API 调用量、HTTP 状态代码和扩展事件。使用这些信息帮助确定需要在什么时候响应。

使用 Amazon CloudWatch 等服务聚合工作负载组件中的日志和指标（例如， AWS CloudTrail 的 API 日志， AWS Lambda 指标， Amazon VPC 流日志和 其他服务）。

常见反模式：

- 您的客户抱怨性能不佳。您最近没有更改应用程序，因此您怀疑是工作负载组件的问题。由于没有遥测，您无法分析，难以确定是哪个或哪些组件导致了性能不佳。
- 无法访问您的应用程序。由于没有遥测，难以确定是否是网络问题。

建立此最佳实践的好处： 了解工作负载的内部情况可让您能够在必要时做出响应。

未建立此最佳实践暴露的风险等级： 高

实施指导

- 实施日志和指标遥测：构建工作负载，使其能够提供其内部状态和业务成果实现情况的信息。使用这些信息来确定需要在什么时候响应。
  - 使用 Amazon CloudWatch 获得对 VM 更好的可观测性 – AWS 在线技术讲座
  - Amazon CloudWatch 的工作原理
  - 什么是 Amazon CloudWatch？

- 使用 Amazon CloudWatch 指标
- 什么是 Amazon CloudWatch Logs？
  - 实施和配置工作负载遥测：设计和配置工作负载，使其能够发出关于其内部状态和当前状态的信息（例如 API 调用量、HTTP 状态代码和扩展事件）。
    - Amazon CloudWatch 指标和维度参考
    - AWS CloudTrail
    - 什么是 AWS CloudTrail？
    - VPC 流日志

资源

相关文档：

- AWS CloudTrail
- Amazon CloudWatch 文档
- Amazon CloudWatch 指标和维度参考
- Amazon CloudWatch 的工作原理
- 使用 Amazon CloudWatch 指标
- VPC 流日志
- 什么是 AWS CloudTrail？
- 什么是 Amazon CloudWatch Logs？
- 什么是 Amazon CloudWatch？

相关视频：

- AWS 上的应用程序性能管理
- 使用 Amazon CloudWatch 获得对 VM 更好的可观测性
- 使用 Amazon CloudWatch 获得对 VM 更好的可观测性 – AWS 在线技术讲座

OPS04-BP03 实施用户活动遥测

构建应用程序代码，使其能够发出关于用户活动的信息，例如点击流或者开始、放弃和完成的事务。使用这些信息来帮助了解应用程序的使用方式和使用量模式，并确定需要在什么时候响应。

常见反模式：

- 您的开发人员部署了无需用户遥测的新功能，并且利用率有所提高。您不能确定增加的利用率是由于新功能的使用，还是新代码导致的问题。
- 您的开发人员部署了无需用户遥测的新功能。如果不联系客户并询问他们，您就无法判断客户是否正在使用新功能。

建立此最佳实践的好处： 了解客户如何通过您的应用程序来确定使用模式、意外行为，以及在必要时让您做出响应。

未建立这种最佳实践的情况下暴露的风险等级： 中

实施指导

- 实施用户活动遥测：设计应用程序代码，使其能够发出关于用户活动的信息（例如点击流或者开始、放弃和完成的事务）。使用这些信息来帮助了解应用程序的使用方式和使用量模式，并确定需要在什么时候响应。

OPS04-BP04 实施依赖项遥测

设计和配置工作负载，使其能够提供关于其依赖的资源状态（例如可访问性或响应时间）的信息。外部依赖项的示例可以包括外部数据库、DNS 和网络连接。使用这些信息来确定需要在什么时候响应。

常见反模式：

- 如果不手动执行检查，了解 DNS 提供程序是否正常运行，就难以确定无法访问应用程序是否是 DNS 的问题。
- 您的购物车应用程序无法完成交易。如果不与信用卡处理提供商联系进行确认，就无法确定是否是他们的问题。

建立此最佳实践的好处： 了解依赖项的运行状况有助于您在必要时做出响应。

未建立此最佳实践暴露的风险等级： 中

实施指导

- 实施依赖项遥测：设计和配置工作负载，使其能够发出关于其状态及其依赖的系统状态的信息。例如：外部数据库、DNS、网络连接以及外部信用卡处理服务。

- Amazon CloudWatch 代理与 AWS Systems Manager 集成 – 适用于 Linux 和 Windows 的统一指标和日志收集
- 使用 CloudWatch 代理从 Amazon EC2 实例和本地服务器收集指标和日志

资源

相关文档：

- Amazon CloudWatch 代理与 AWS Systems Manager 集成 – 适用于 Linux 和 Windows 的统一指标和日志收集
- 使用 CloudWatch 代理从 Amazon EC2 实例和本地服务器收集指标和日志

相关示例：

- Well-Architected 实验室 – 依赖项监控

OPS04-BP05 实施事务跟踪

实施应用程序代码并配置工作负载组件，提供关于工作负载之间的事务流的信息。使用这些信息来确定需要在什么时候做出响应，并帮助您确定导致问题的因素。

在 AWS 中，您可以使用分布式跟踪服务（例如 AWS X-Ray）来收集和记录事务通过工作负载时的跟踪记录，生成地图以查看事务如何在工作负载和服务之间流动，深入了解组件之间的关系，并实时识别和分析问题。

常见反模式：

- 您跨多个账户实施了无服务器微服务架构。您的客户遇到间歇性性能问题。您无法确定是哪项功能还是哪个组件的问题，因为您缺少跟踪信息，无法明确指出应用程序哪里出现了性能问题以及导致问题的原因。
- 您尝试确定工作负载中的性能问题，以便在开发工作中解决它们。您无法查看应用程序组件以及与它们交互的服务之间的关系，难以确定问题出在哪里；这是因为您缺少跟踪信息，无法深入了解影响应用程序性能的具体服务和路径。

建立此最佳实践的好处： 了解跨工作负载的事务流可让您了解工作负载事务的预期行为及其在整个工作负载中的变化，使您能够在必要时做出响应。

未建立这种最佳实践的情况下暴露的风险等级： 低

实施指导

- 实施事务跟踪：设计应用程序和工作负载，使其发出有关系统组件间的事务流的信息，例如事务阶段、活动组件以及完成活动的时间。使用这些信息来确定正在进行的活动、已完成的活动以及已完成活动的结果。这可以帮助您确定需要在什么时候响应。例如，组件内的事务响应时间长于预期，这可能表明该组件存在问题。
  - AWS X-Ray
  - 什么是 AWS X-Ray？

资源

相关文档：

- AWS X-Ray
- 什么是 AWS X-Ray？

OPS 5 如何减少缺陷、简化修复和改进生产流程？

支持在生产时调整改进流程并支持重构、快速质量反馈和错误修复方法。这些方法可以加快有益更改进入生产环境的速度、减少产生的问题，并能够快速识别和修复通过部署活动引入的问题。

最佳实践

- OPS05-BP01 使用版本控制
- OPS05-BP02 测试并验证变更
- OPS05-BP03 使用配置管理系统
- OPS05-BP04 使用构建和部署管理系统
- OPS05-BP05 执行补丁管理
- OPS05-BP06 共享设计标准
- OPS05-BP07 实施提高代码质量的实践
- OPS05-BP08 使用多个环境
- OPS05-BP09 频繁进行可逆的小规模更改
- OPS05-BP10 完全自动化集成和部署

OPS05-BP01 使用版本控制

使用版本控制来跟踪更改和发布。

许多 AWS 服务都提供版本控制功能。使用修订或源代码控制系统（如 AWS CodeCommit ）管理代码和其他构件，如基础设施的版本控制的 AWS CloudFormation 模板。

常见反模式：

- 您一直在在工作站上开发和存储代码。工作站上发生了不可恢复的存储故障，您的代码丢失了。
- 用更改内容覆盖现有代码后，您重新启动应用程序，但其无法运行。您无法恢复为更改内容。
- 您对报告文件执行了写入锁定，而其他人需要对此文件进行编辑。他们与您联系要求您停止写入锁定，以便他们可以完成自己的任务。
- 您的研究团队一直在进行详细的分析，以便对未来的工作进行规划。有人不小心把购物单保存在最终报告上了。您无法还原更改，不得不重新创建报告。

建立此最佳实践的好处： 借助版本控制功能，您可以轻松地恢复到已知的良好状态、以前的版本，并降低资产丢失的风险。

未建立这种最佳实践的情况下暴露的风险等级： 高

实施指导

- 使用版本控制：在采用了版本控制的存储库中维护资产。这让您能够跟踪更改、部署新版本、检测对现有版本的更改，以及恢复到以前的版本（例如在发生故障时回滚到已知的良好状态）。将配置管理系统的版本控制功能集成到程序中。
  - AWS CodeCommit 简介
  - 什么是 AWS CodeCommit？

资源

相关文档：

- 什么是 AWS CodeCommit？

相关视频：

- AWS CodeCommit 简介

OPS05-BP02 测试并验证变更

测试并验证变更以便发现并减少错误。实现自动测试以便减少手动过程引起的错误，并减少测试工作量。

许多 AWS 服务都提供版本控制功能。使用修订或源代码控制系统（如 AWS CodeCommit ）管理代码和其他构件，如基础设施的版本控制的 AWS CloudFormation 模板。

常见反模式：

- 在将新代码部署到生产环境后，由于应用程序无法再运行，客户开始打电话投诉。
- 为了增强周边安全，您应用了新安全组。它运行以后产生了意想不到的后果，用户无法访问您的应用程序。
- 您修改了新函数调用的方法。另一个依赖于该方法的函数也无法运行。没有检测到问题，开始投产。由于一段时间内没有调用另一个函数，最终导致生产失败，但是没有找到原因。

建立此最佳实践的好处： 在早期进行测试和验证更改，可以将解决问题的成本降至最低，并降低对客户的影响。经过测试之后再部署，可以最大限度地减少错误。

未建立这种最佳实践的情况下暴露的风险等级： 高

实施指导

- 测试并验证变更：应该在所有生命周期阶段（例如开发、测试和生产阶段）测试更改并验证结果。使用测试结果来确认新功能，并减少部署失败的风险和影响。实现自动测试和验证，以便确保审核的一致性、减少手动过程引起的错误并减少工作量。
  - 什么是 AWS CodeBuild？
  - AWS CodeBuild 的本地构建支持

资源

相关文档：

- AWS 开发人员工具
- AWS CodeBuild 的本地构建支持
- 什么是 AWS CodeBuild？

OPS05-BP03 使用配置管理系统

使用配置管理系统来实现和跟踪配置更改。这些系统可以减少手动过程引起的错误，并减少部署更改的工作量。

静态配置管理在初始化资源时设置的值，这些值在资源的生命周期内预期保持一致。这样的例子包括为实例上的 Web 或应用程序服务器设置配置，或者定义 AWS 服务的配置（在 AWS Management Console 内或者通过 AWS CLI）。

动态配置管理在初始化时设置值，这些值在资源的生命周期内可能或预期会发生变化。例如，您可以设置一个功能切换，通过配置更改在代码中启用功能，或者在意外事件期间更改日志详细级别以捕获更多数据，然后在意外事件完成后更改回来，避免再不必要的日志记录及其相关费用。

如果您在实例、容器、无服务器函数或设备上运行的应用程序具有动态配置，您可以使用 AWS AppConfig 在您的环境中管理和部署它们。

在 AWS 上，您可以使用 AWS Config 跨账户和区域持续监控 AWS 资源 配置。这使您可以跟踪其配置历史记录，了解配置更改可能如何影响其他资源，并使用 AWS Config 规则 和 AWS Config 合规包根据预期或所需的配置审计它们。.

在 AWS 中，您可以使用像 AWS 开发人员工具 （例如，AWS CodeCommit、 AWS CodeBuild， AWS CodePipeline， AWS CodeDeploy和 AWS CodeStar）这样的服务来构建持续集成/持续部署（CI/CD）管道。

当计划的重要业务、运营活动或事件受到更改实施的影响时，建立更改日历并进行跟踪。围绕这些计划来调整活动以管理风险。 AWS Systems Manager 变更日历 提供了一种机制，可以记录更改开始或结束的时间块及更改原因，并与 其他 AWS 账户分享该信息。AWS Systems Manager Automation 脚本可以配置为符合更改日历状态。

AWS Systems Manager 维护时段 可用于安排在指定的时间执行 AWS SSM Run Command 或 Automation 脚本、AWS Lambda 调用或 AWS Step Functions 活动。在更改日历中标记这些活动，以便将其包含在您的评估中。

常见反模式：

- 您手动更新整个队列中的 Web 服务器配置，由于更新错误，许多服务器变得没有响应。
- 手动更新应用程序服务器队列需要花费很长时间。在变更过程中，如果配置不一致会导致意外行为发生。
- 有人更新了您的安全组，您的 Web 服务器无法访问了。如果不知道发生了哪些变更，您需要花费大量时间来调查问题，导致恢复时间延长。

建立此最佳实践的好处： 采用配置管理系统可以减少更改及对其进行跟踪的工作量，还可以降低手动程序导致错误的频率。

未建立这种最佳实践的情况下暴露的风险等级： 中

实施指导

- 使用配置管理系统：使用配置管理系统来跟踪并实施更改，以便减少手动过程引起的错误，并减少工作量。
  - 基础设施配置管理
  - AWS Config
  - 什么是 AWS Config？
  - AWS CloudFormation 简介
  - 什么是 AWS CloudFormation？
  - AWS OpsWorks
  - 什么是 AWS OpsWorks？
  - AWS Elastic Beanstalk 简介
  - 什么是 AWS Elastic Beanstalk？

资源

相关文档：

- AWS AppConfig
- AWS 开发人员工具
- AWS OpsWorks
- AWS Systems Manager 变更日历
- AWS Systems Manager 维护时段来自动执行修补托管系统的过程和安排修补活动
- 基础设施配置管理
- 什么是 AWS CloudFormation？
- 什么是 AWS Config？
- 什么是 AWS Elastic Beanstalk？
- 什么是 AWS OpsWorks？

相关视频：

- AWS CloudFormation 简介
- AWS Elastic Beanstalk 简介

OPS05-BP04 使用构建和部署管理系统

使用构建和部署管理系统。这些系统可以减少手动过程引起的错误，并减少部署更改的工作量。

在 AWS 中，您可以使用像 AWS 开发人员工具（例如，AWS CodeCommit、 AWS CodeBuild， AWS CodePipeline， AWS CodeDeploy和 AWS CodeStar）这样的服务来构建持续集成/持续部署（CI/CD）管道。

常见反模式：

- 在开发系统上编译代码后，您将可执行文件复制到生产系统上，但它无法启动。本地日志文件显示这是因为缺少依赖项。
- 您成功地在开发环境中构建了具有新功能的应用程序，并将代码送交质量检查（QA，Quality Assurance）。由于缺少静态资产，它没有通过质量检查。
- 星期五，经过大量的努力，您成功地在开发环境中手动构建了应用程序，包括新编码的功能。星期一，您无法重复这一成功构建应用程序的步骤。
- 您执行为新版本创建的测试。下周，您将设置测试环境，并执行所有现有的集成测试，然后执行性能测试。新代码产生了难以接受的性能影响，因此必须重新开发并测试。

建立此最佳实践的好处： 制定相应机制来管理活动的构建和部署。这样，您可以减少执行重复任务的工作量，让团队成员腾出时间专注于高价值的创造性任务，还可以减少手动程序导致的错误。

未建立这种最佳实践的情况下暴露的风险等级： 中

实施指导

- 使用构建和部署管理系统：使用构建和部署管理系统来跟踪并实施更改，以便减少手动过程引起的错误，并减少工作量。将集成和部署管道完全自动化，从代码签入到构建、测试、部署和验证都包含在内。这可以减少准备时间、提高更改频率，并减少工作量。
  - 什么是 AWS CodeBuild？
  - 面向软件开发的持续集成最佳实践
  - Slalom：AWS 上面向无服务器应用程序的 CI/CD

- AWS CodeDeploy 简介 – 使用 Amazon Web Services 自动完成软件部署
- 什么是 AWS CodeDeploy？

资源

相关文档：

- AWS 开发人员工具
- 什么是 AWS CodeBuild？
- 什么是 AWS CodeDeploy？

相关视频：

- 面向软件开发的持续集成最佳实践
- AWS CodeDeploy 简介 – 使用 Amazon Web Services 自动完成软件部署
- Slalom：AWS 上面向无服务器应用程序的 CI/CD

OPS05-BP05 执行补丁管理

执行补丁管理以便实现功能、解决问题并保持监管合规性。实现自动补丁管理以便减少手动过程引起的错误，并减少修补工作量。

补丁和漏洞管理是优势和风险管理活动的一部分。最好是具有不可变的基础设施和在已验证的已知良好状态下部署工作负载。如果该方法都不可行，那就只能进行修补。

更新系统映像、容器映像或 Lambda 自定义运行时和其他库 以消除漏洞，是补丁管理的一部分。您应使用以下工具来 管理适用于 Linux 或 Windows Server 映像的 Amazon 系统映像 (AMI) 的更新： EC2 Image Builder.您可以将 Amazon Elastic Container Registry 与现有管道配合使用以 管理 Amazon ECS 映像 和 管理 Amazon EKS 映像。AWS Lambda 包括 版本 管理功能。

在未事先在安全环境中测试的情况下，不应对生产系统执行修补操作。仅当补丁支持操作或业务结果时，才应该应用补丁。在 AWS 上，您可以使用 AWS Systems Manager 补丁管理器 和 AWS Systems Manager 维护时段来自动执行修补托管系统的过程和安排修补活动.

常见反模式：

- 您接到任务，需要在两个小时内应用所有新的安全补丁，但由于应用程序与补丁不兼容，导致了多次停机。

- 没有安装补丁的库会引发意外后果，这是因为未知方会利用其中的漏洞来访问您的工作负载。
- 您在未通知开发人员的情况下自动修补开发人员环境。您收到来自开发人员的多起投诉，称他们的环境不能按预期运行。
- 您尚未修补持久性实例上的现有商用软件。当您遇到软件问题并与供应商联系时，他们告知您已不再为该版本提供支持，您必须安装特定级别的补丁才能获得帮助。
- 您使用的加密软件最近发布了新补丁，对性能进行了重大改进。您未安装补丁的系统仍然存在性能问题，恰恰是因为没有安装补丁造成的。

建立此最佳实践的好处： 您可以通过建立补丁管理流程（包括修补标准和在整个环境中分发的方法）来实现收益并控制影响。这样一来，可以采用所需功能、解决问题并保持监管合规性。实施补丁管理系统和自动化，以减少部署补丁的工作量，并减少手动过程引起的错误。

未建立此最佳实践暴露的风险等级： 中

实施指导

- 补丁管理：修补系统以便纠正问题、获得所需的特性或功能、符合监管政策并满足供应商支持需求。在不可变系统中，使用适当的补丁集进行部署，以便实现所需结果。自动执行补丁管理机制以便缩短修补时间、减少手动过程引起的错误，并减少修补工作量。
  - AWS Systems Manager 补丁管理器

资源

相关文档：

- AWS 开发人员工具
- AWS Systems Manager 补丁管理器

相关视频：

- AWS 上面向无服务器应用程序的 CI/CD
- Ops 设计理念

相关示例：

- Well-Architected 实验室 – 清单和补丁管理

OPS05-BP06 共享设计标准

在不同团队间共享最佳实践，以便提高认识并最大程度地实现开发工作的效益。

在 AWS 上，您可以使用代码方法定义和管理应用程序、计算、基础设施和运营。这让您可以轻松地发布、分享和采用。

许多 AWS 服务和资源都可以设计为跨账户共享，从而使您能够跨团队分享所创建的资产和所学到的知识。例如，您可以与特定账户共享 CodeCommit 存储库、 Lambda 函数、 Amazon S3 存储桶和 AMI 。

发布新资源或更新时，请使用 Amazon SNS 提供 跨账户通知.订阅者可以使用 Lambda 获取新版本。

如果在组织中强制实施了共享标准，则必须存在相应的机制来以请求增加、更改标准和标准例外，以为团队的活动提供支持。如果没有这样的机制，标准将成为创新的约束。

常见反模式：

- 您创建了自己的用户身份验证机制，组织中的其他开发团队亦是如此。对于用户想要访问的系统的每一部分，他们都不得不使用一套单独的凭据。
- 您创建了自己的用户身份验证机制，组织中的其他开发团队亦是如此。您的组织必须满足一项新的合规性要求。现在，每个开发团队都必须投入资源来实施新的要求。
- 您创建了自己的屏幕布局，组织中的所有其他开发团队也各自创建了屏幕布局。用户抱怨界面不一致，难以导航。

建立此最佳实践的好处： 在标准满足多个应用程序或组织的要求的情况下，使用共享标准来支持最佳实践的采用并最大程度地实现开发工作的效益。

未建立此最佳实践暴露的风险等级： 中

实施指导

- 共享设计标准：在不同团队间共享现有的最佳实践、设计标准、检查清单、操作程序、指南和监管要求，以便降低复杂性并充分发挥开发工作的作用。确保建立针对设计标准的更改、补充和例外请求程序，以便支持持续改进和创新。确保团队了解已发布的内容，从而让他们能够利用内容，并减少返工和浪费的工作。
  - 授权访问 AWS 环境
  - 共享 AWS CodeCommit 存储库
  - AWS Lambda 函数的简单授权

- 将 AMI 与特定 AWS 账户共享
- 利用 AWS CloudFormation Designer URL 快速共享模板
- 将 AWS Lambda 与 Amazon SNS 配合使用

资源

相关文档：

- AWS Lambda 函数的简单授权
- 共享 AWS CodeCommit 存储库
- 将 AMI 与特定 AWS 账户共享
- 利用 AWS CloudFormation Designer URL 快速共享模板
- 将 AWS Lambda 与 Amazon SNS 配合使用

相关视频：

- 授权访问 AWS 环境

OPS05-BP07 实施提高代码质量的实践

实施能够提高代码质量并尽可能减少缺陷的最佳实践。一些示例包括测试驱动型开发、代码审查和标准采用。

在 AWS 上，您可以将 Amazon CodeGuru 等服务与管道集成，以 使用计划分析和机器学习来 识别潜在的代码和安全问题。CodeGuru 提供有关如何实施 AWS 最佳实践来解决这些问题的推荐。

常见反模式：

- 为了能更快地测试您的功能，您决定不集成标准输入过滤库。测试完成之后，您提交代码时没有合并入库。
- 对于正在处理的数据集，您经验不足，并不知道数据集中可能存在一系列边缘案例。这些边缘案例与您实施的代码不兼容。

建立此最佳实践的好处： 通过采用提高代码质量的实践，能够将引入生产中的问题降至最低。

未建立这种最佳实践的情况下暴露的风险等级： 中

实施指导

- 实施提高代码质量的实践：实施提高代码质量的实践，以便尽可能减少缺陷并降低部署代码的风险。例如测试驱动型开发、结对编程、代码审查和标准采用。
  - Amazon CodeGuru

资源

相关文档：

- Amazon CodeGuru

OPS05-BP08 使用多个环境

使用多个环境来试验、开发和测试您的工作负载。当环境接近于生产环境时，逐步加强控制，以确保工作负载在部署后能够按预期运行。

常见反模式：

- 您正在共享开发环境中执行开发，另一位开发人员将覆盖您的代码更改。
- 共享开发环境上严苛的安全控制令您无法试验新的服务和功能。
- 您在生产系统上执行负载测试，导致用户停机。
- 生产中发生了严重错误，导致数据丢失。在生产环境中，您尝试重新创建导致数据丢失的条件，以便能够确定它是如何发生的，并防止它再次发生。为了防止在测试期间再次丢失数据，您被迫采取措施，让用户无法使用应用程序。
- 您正在运行多租户服务，无法支持客户对专用环境的请求。
- 您不可能每次都测试，但在生产环境中会执行测试。
- 您认为单一环境的简单性比更改在环境中的影响范围更加重要。

建立此最佳实践的好处： 通过部署多个环境，可以让您为多个同时进行的开发、测试和生产环境提供支持，而不会在开发人员或用户社区间造成冲突。

未建立此最佳实践暴露的风险等级： 中

实施指导

- 使用多个环境：为开发人员提供控制机制最少的沙盒环境，以便支持试验。提供单独的开发环境以便支持并行工作，并提高开发的灵活性。在接近生产的环境中实施更严格的控制，让开发人员能够创

新。使用基础设施即代码和配置管理系统来部署与生产环境中的控制机制配置一致的环境，以便确保系统在部署后按照预期运行。关闭不使用的环境，以免空闲资源（例如晚上和周末的开发系统）产生费用。在负载测试时部署与生产等效的环境，以便实现有效结果。

- 什么是 AWS CloudFormation？
- 如何使用 AWS Lambda 按固定间隔停止和启动 Amazon EC2 实例？

资源

相关文档：

- 如何使用 AWS Lambda 按固定间隔停止和启动 Amazon EC2 实例？
- 什么是 AWS CloudFormation？

OPS05-BP09 频繁进行可逆的小规模更改

频繁进行可逆的小规模变更可以减少变更的范围和影响。这可以简化故障排除、支持更快的修复，并提供回滚更改的选项。

常见反模式：

- 您每季度都部署新版应用程序。
- 您经常更改数据库架构。
- 您执行手动就地更新，覆盖现有安装和配置。

建立此最佳实践的好处： 频繁部署小的更改可让您更快地发现开发工作带来的效益。更改很小时，更易于确定是否会带来意外后果。更改可逆时，由于简化了恢复，因此实施更改的风险更小。

未建立此最佳实践暴露的风险等级： 低

实施指导

- 频繁进行可逆的小规模更改：频繁进行可逆的小规模更改可以减小更改的范围和影响。这可以简化故障排除、支持更快的修复，并提供回滚更改的选项。这还可以加快企业实现价值的速度。

OPS05-BP10 完全自动化集成和部署

实现自动构建、部署和测试工作负载。这可以减少手动过程引起的错误，并减少部署更改的工作量。

使用 资源标签 和 AWS Resource Groups ，按照一致的 标记策略 应用元数据，以标识您的资源。标记您的资源，以便进行整理、成本核算、访问控制并有针对性地自动执行操作活动。

常见反模式：

- 星期五，您完成为分支功能编写新代码的工作。星期一，在运行代码质量测试脚本和各单元测试脚本后，您将代码签入计划发行的下一版本中。
- 您接到任务，需要为重要问题编写修复代码，该问题在生产中影响了大量客户。对修复代码进行测试后，您提交代码并通过电子邮件发送更改管理，请求批准以将其部署到生产环境中。

建立此最佳实践的好处： 通过自动构建和部署管理系统，可以减少由手动流程引发的错误，并减少部署更改的工作量，使您的团队成员能够专注于实现商业价值。

未建立此最佳实践暴露的风险等级： 低

实施指导

- 使用构建和部署管理系统：使用构建和部署管理系统来跟踪并实施更改，以便减少手动过程引起的错误，并减少工作量。将集成和部署管道完全自动化，从代码签入到构建、测试、部署和验证都包含在内。这可以减少准备时间、提高更改频率，并减少工作量。
  - 什么是 AWS CodeBuild？
  - 面向软件开发的持续集成最佳实践
  - Slalom：AWS 上面向无服务器应用程序的 CI/CD
  - AWS CodeDeploy 简介 – 使用 Amazon Web Services 自动完成软件部署
  - 什么是 AWS CodeDeploy？

资源

相关文档：

- 什么是 AWS CodeBuild？
- 什么是 AWS CodeDeploy？

相关视频：

- 面向软件开发的持续集成最佳实践
- AWS CodeDeploy 简介 – 使用 Amazon Web Services 自动完成软件部署

- Slalom：AWS 上面向无服务器应用程序的 CI/CD

OPS 6 您如何缓解部署风险？

采用提供快速质量反馈，并且若更改没有达到目标成效，则支持快速恢复的方法。使用这些实践可以减轻因部署更改而产生的问题的影响。

最佳实践

- OPS06-BP01 针对不成功的更改制定计划
- OPS05-BP02 测试并验证变更
- OPS06-BP03 使用部署管理系统
- OPS06-BP04 使用有限部署进行测试
- OPS06-BP05 使用并行环境进行部署
- OPS06-BP06 部署频繁、小规模、可逆的更改
- OPS06-BP07 完全自动化集成和部署
- OPS06-BP08 自动测试和回滚

OPS06-BP01 针对不成功的更改制定计划

制定计划，以便在变更没有达到目标成效时在生产环境中恢复到已知良好状态，或者进行修复。做好充分的准备，以备快速响应，最大限度缩短回滚时间。

常见反模式：

- 您执行部署以后应用程序变得不稳定，但是系统上似乎还有活动用户。您必须决定是回滚更改并影响活动用户，还是等到知道用户无论如何都可能受到影响后再回滚更改。
- 更改路由后，可以访问新环境，但是其中一个子网无法访问。您必须决定是回滚所有内容还是尝试修复无法访问的子网。在您做决定时，子网仍然无法访问。

建立此最佳实践的好处： 实施计划来缩短不成功更改的平均修复时间（MTTR，Mean Time To Recover），减少对最终用户的影响。

未建立这种最佳实践的情况下暴露的风险等级： 高

实施指导

- 针对不成功的更改制定计划：制定计划，以便在更改没有实现所需成果时在生产环境中恢复到已知良好状态（即回滚更改），或者进行修复（即前滚更改）。如果发现在失败后无法回滚的更改，请在提交更改之前做好准备。

OPS05-BP02 测试并验证变更

在所有生命周期阶段测试更改并验证结果，以便确认新功能并尽可能减少部署失败的风险和影响。

在 AWS 上，您可以创建临时并行环境，以降低试验和测试的风险、工作量及成本。使用 AWS CloudFormation 自动部署这些环境，以确保以一致的方式实施您的临时环境。

常见反模式：

- 您在应用程序中部署了一个很酷的新功能，它无法运行，而您却不知道。
- 您更新了证书。您不小心将证书安装到了错误的组件上。而您却不知道。

建立此最佳实践的好处： 在部署后对更改进行测试和验证，您可以及早发现问题，从而有机会减轻对客户的影响。

未建立这种最佳实践的情况下暴露的风险等级： 高

实施指导

- 测试并验证更改：在所有生命周期阶段（例如开发、测试和生产）测试更改并验证结果，以便确认新功能并尽可能减少部署失败的风险和影响。
  - AWS Cloud9
  - 什么是 AWS Cloud9？
  - 如何在发送代码之前在本地测试和调试 AWS CodeDeploy

资源

相关文档：

- AWS Cloud9
- AWS 开发人员工具
- 如何在发送代码之前在本地测试和调试 AWS CodeDeploy

- 什么是 AWS Cloud9？

OPS06-BP03 使用部署管理系统

使用部署管理系统来跟踪并实施更改。这可以减少手动过程引起的错误，并减少部署更改的工作量。

在 AWS 中，您可以使用像 AWS 开发人员工具（例如，AWS CodeCommit、 AWS CodeBuild， AWS CodePipeline， AWS CodeDeploy和 AWS CodeStar）这样的服务来构建持续集成/持续部署（CI/CD）管道。

常见反模式：

- 您手动将更新部署到整个队列中的应用程序服务器，由于更新错误，许多服务器变得没有响应。
- 手动部署到应用程序服务器队列需要花费很长时间。在更改过程中，如果版本不一致会导致意外行为发生。

建立此最佳实践的好处： 采用部署管理系统可以减少部署更改的工作量，还可以降低手动程序导致错误的频率。

未建立此最佳实践暴露的风险等级： 中

实施指导

- 使用部署管理系统：使用部署管理系统来跟踪并实施更改。这可以减少手动过程引起的错误，并减少部署更改的工作量。将集成和部署管道自动化，从代码签入到测试、部署和验证都包含在内。这可以减少准备时间、提高更改频率，并进一步减少工作量。
  - AWS CodeDeploy 简介 – 使用 Amazon Web Services 自动完成软件部署
  - 什么是 AWS CodeDeploy？
  - 什么是 AWS Elastic Beanstalk？
  - 什么是 Amazon API Gateway？

资源

相关文档：

- AWS CodeDeploy 用户指南
- AWS 开发人员工具
- 在 AWS CodeDeploy 中尝试示例蓝绿部署

- 什么是 AWS CodeDeploy？
- 什么是 AWS Elastic Beanstalk？
- 什么是 Amazon API Gateway？

相关视频：

- 使用 AWS 深入了解高级持续交付技术
- AWS CodeDeploy 简介 – 使用 Amazon Web Services 自动完成软件部署

OPS06-BP04 使用有限部署进行测试

与现有系统一起进行有限部署测试，以在全面部署前确认预期结果。例如使用 Canary 部署测试或一体化部署。

常见反模式：

- 您一次性将不成功的更改部署到所有生产环境中。而您却不知道。

建立此最佳实践的好处： 通过在完成有限部署后对更改进行测试和验证，您可以及早发现问题，从而有机会进一步减轻对客户的影响，将对客户的影响降至最低。

未建立这种最佳实践的情况下暴露的风险等级： 中

实施指导

- 使用有限部署进行测试：在全面部署之前使用有限的部署和现有系统进行测试，以确认实现所需成果。例如使用 Canary 部署测试或一体化部署。
  - AWS CodeDeploy 用户指南
  - 使用 AWS Elastic Beanstalk 进行蓝/绿部署
  - 设置 API Gateway 金丝雀发布部署
  - 在 AWS CodeDeploy 中尝试示例蓝绿部署
  - 在 AWS CodeDeploy 中使用部署配置

资源

相关文档：

- AWS CodeDeploy 用户指南
- 使用 AWS Elastic Beanstalk 进行蓝/绿部署
- 设置 API Gateway 金丝雀发布部署
- 在 AWS CodeDeploy 中尝试示例蓝绿部署
- 在 AWS CodeDeploy 中使用部署配置

OPS06-BP05 使用并行环境进行部署

在并行环境中实施变更，然后过渡到新环境。保留之前的环境，直到确认部署成功为止。这样可以支持回滚到以前的环境，从而尽可能缩短恢复时间。

常见反模式：

- 您通过修改现有系统来执行可变部署。发现更改不成功后，您被迫再次修改系统以还原旧版本，从而导致恢复时间延长。
- 在维护时段内，您停用旧环境，然后开始构建新环境。在这一过程进行了许多时间后，您发现部署中出现了无法恢复的问题。虽然非常疲惫，您还是不得不找回以前的部署过程，并开始重新构建旧环境。

建立此最佳实践的好处： 使用并行环境后，您可以预先部署新环境并在需要时过渡到新环境。如果新环境不成功，则可以转换回原始环境，完成快速恢复。

未建立这种最佳实践的情况下暴露的风险等级： 中

实施指导

- 使用并行环境进行部署：对并行环境实施更改，然后过渡或切换到新环境。保留之前的环境，直到确认部署成功为止。这样可以支持回滚到以前的环境，从而尽可能缩短恢复时间。例如在不可变基础设施中采用蓝/绿部署。
  - 在 AWS CodeDeploy 中使用部署配置
  - 使用 AWS Elastic Beanstalk 进行蓝/绿部署
  - 设置 API Gateway 金丝雀发布部署
  - 在 AWS CodeDeploy 中尝试示例蓝绿部署

资源

相关文档：

- AWS CodeDeploy 用户指南
- 使用 AWS Elastic Beanstalk 进行蓝/绿部署
- 设置 API Gateway 金丝雀发布部署
- 在 AWS CodeDeploy 中尝试示例蓝绿部署
- 在 AWS CodeDeploy 中使用部署配置

相关视频：

- 使用 AWS 深入了解高级持续交付技术

OPS06-BP06 部署频繁、小规模、可逆的更改

频繁进行可逆的小规模更改可以缩小变更的范围。这样可以简化故障排除工作、加快修复速度，并支持回滚更改。

常见反模式：

- 您每季度都部署新版应用程序。
- 您经常更改数据库架构。
- 您执行手动就地更新，覆盖现有安装和配置。

建立此最佳实践的好处： 频繁部署小的更改可让您更快地发现开发工作带来的效益。更改很小时，更易于确定是否会带来意外后果。更改可逆时，由于简化了恢复，因此实施更改的风险更小。

未建立此最佳实践暴露的风险等级： 低

实施指导

- 部署频繁、小规模、可逆的更改：频繁进行可逆的小规模更改可以缩小更改影响的范围。这样可以简化故障排除工作、加快修复速度，并支持回滚更改。

OPS06-BP07 完全自动化集成和部署

实现自动构建、部署和测试工作负载。这可以减少手动过程引起的错误，并减少部署更改的工作量。

使用 资源标签 和 AWS Resource Groups ，按照一致的 标记策略 应用元数据，以标识您的资源。标记您的资源，以便进行整理、成本核算、访问控制并有针对性地自动执行操作活动。

常见反模式：

- 星期五，您完成为分支功能编写新代码的工作。星期一，在运行代码质量测试脚本和各单元测试脚本后，您将代码签入计划发行的下一版本中。
- 您接到任务，需要为重要问题编写修复代码，该问题在生产中影响了大量客户。对修复代码进行测试后，您提交代码并通过电子邮件发送更改管理，请求批准以将其部署到生产环境中。

建立此最佳实践的好处： 通过自动构建和部署管理系统，可以减少由手动流程引发的错误，并减少部署更改的工作量，使您的团队成员能够专注于实现商业价值。

未建立这种最佳实践的情况下暴露的风险等级： 低

实施指导

- 使用构建和部署管理系统：使用构建和部署管理系统来跟踪并实施更改，以便减少手动过程引起的错误，并减少工作量。将集成和部署管道完全自动化，从代码签入到构建、测试、部署和验证都包含在内。这可以减少准备时间、提高更改频率，并减少工作量。
  - 什么是 AWS CodeBuild？
  - 面向软件开发的持续集成最佳实践
  - Slalom：AWS 上面向无服务器应用程序的 CI/CD
  - AWS CodeDeploy 简介 – 使用 Amazon Web Services 自动完成软件部署
  - 什么是 AWS CodeDeploy？
  - 使用 AWS 深入了解高级持续交付技术

资源

相关文档：

- 在 AWS CodeDeploy 中尝试示例蓝绿部署
- 什么是 AWS CodeBuild？
- 什么是 AWS CodeDeploy？

相关视频：

- 面向软件开发的持续集成最佳实践
- 使用 AWS 深入了解高级持续交付技术

- AWS CodeDeploy 简介 – 使用 Amazon Web Services 自动完成软件部署
- Slalom：AWS 上面向无服务器应用程序的 CI/CD

OPS06-BP08 自动测试和回滚

自动测试部署的环境以便确认目标效果。在没有达到预期结果时，自动回滚到之前的已知良好状态，尽可能地缩短恢复时间，并减少手动过程引起的错误。

常见反模式：

- 您为工作负载部署更改。您看到更改完成后，开始进行部署后测试。完成测试之后，您发现工作负载不可操作，而且客户断开了连接。然后，您开始回滚到之前的版本。经过较长时间检测，发现问题之后，通过手动重新部署会延长恢复时间。

建立此最佳实践的好处： 在部署之后对更改进行测试和验证，可以让您立即发现问题。自动回滚到以前的版本，可以将对客户的影响降至最低。

未建立这种最佳实践的情况下暴露的风险等级： 低

实施指导

- 自动测试和回滚：自动测试部署的环境以确认达成所需效果。在没有达到预期结果时，自动回滚到之前的已知良好状态，尽可能地缩短恢复时间，并减少手动过程引起的错误。例如，在部署之后执行详细的综合用户事务、验证结果，并在失败时回滚。
  - 使用 AWS CodeDeploy 重新部署和回滚部署

资源

相关文档：

- 使用 AWS CodeDeploy 重新部署和回滚部署

OPS 7 如何知道您已经准备好支持某种工作负载？

评估工作负载、流程及程序和工作人员的操作准备就绪情况，以便了解与工作负载相关的操作风险。

最佳实践

- OPS07-BP01 确保员工能力

- OPS07-BP02 确保以一致的方式对运维准备情况进行审查
- OPS07-BP03 使用运行手册执行程序
- OPS07-BP04 根据行动手册调查问题
- OPS07-BP05 做出明智的决策来部署系统和更改

OPS07-BP01 确保员工能力

通过一种机制来验证您是否有适当数量技术娴熟的员工来提供对运营需求的支持。根据需要进行员工培训并调整人员产能，以便保持有效的支持。

您需要拥有足够的团队成员来完成所有活动（包括待命的团队成员）。确保您的团队拥有必要的技能，以便能够成功完成关于您的工作负载、运营工具和 AWS 的培训。

AWS 提供了许多资源，包括 AWS 入门资源中心， AWS Blog， AWS 在线技术讲座， AWS 活动和网络研讨会，以及 AWS Well-Architected 实验室，这些资源提供了指导、示例和详细演练，用以培训您的团队。此外， AWS 培训与认证 提供了一些免费培训，可以通过自定进度的数字课程，学习 AWS 的基础知识。您还可以注册讲师指导培训，进一步帮助培养您团队的 AWS 技能。

常见反模式：

- 在以下情况下部署工作负载：团队成员不能熟练使用平台和服务。
- 在以下情况下部署工作负载：在预期的支持时间内没有团队成员可以提供支持。
- 在以下情况下部署工作负载：如果有团队成员正在休假或生病，则没有足够的团队成员来提供支持。
- 在以下情况下部署额外的工作负载：没有考虑团队成员为它和其他工作负载提供支持时的额外影响。

建立此最佳实践的好处： 拥有技能娴熟的团队成员能够为您的工作负载提供有效支持。

未建立这种最佳实践的情况下暴露的风险等级： 高

实施指导

- 员工能力：确认是否有足够的训练有素的人员来有效地支持工作负载。
  - 团队规模：确保您拥有足够的团队成员来执行运营活动，以及随时待命。
  - 团队技能：确保您的团队成员接受了足够的 AWS、工作负载及运营工具的培训，以履行其职责。
    - AWS 活动和网络研讨会
    - 欢迎参加 AWS 培训与认证

- 了解能力：在操作环境和工作负载发生变化时查看团队规模和技能，以确保有足够的能力保持卓越运营。进行适当调整，以确保团队规模和技能与团队所支持的工作负载的操作要求相匹配。

资源

相关文档：

- AWS Blog
- AWS 活动和网络研讨会
- AWS 入门资源中心
- AWS 在线技术讲座
- 欢迎参加 AWS 培训与认证

相关示例：

- Well-Architected 实验室

OPS07-BP02 确保以一致的方式对运维准备情况进行审查

使用运维准备情况审查（ORR，Operational Readiness Review），确保可以运营您的工作负载。ORR 是 Amazon 开发的一种机制，用于验证团队可以安全地运营其工作负载。ORR 是一个使用要求核对清单进行审查和检查的过程。ORR 是一种自助服务体验，供团队用于验证其工作负载。ORR 中包含的最佳实践源自我们多年构建软件的经验教训。

ORR 核对清单包括架构推荐、运维过程、事件管理和发布质量。我们的更正错误（CoE，Correction of Error）流程是这些项目的主要推动因素。您的事后分析应该可以推动自己的 ORR 演进。ORR 并不仅仅关系到遵循最佳实践，还关系到预防以前的事件再次发生。最后，ORR 中还可以包括安全性、监管和合规性要求。

在工作负载正式公开发布之前运行 ORR，然后在整个软件开发生命周期中运行 ORR。在发布之前运行 ORR 可以提升安全运营工作负载的能力。对工作负载定期重新运行 ORR 可以收集任何偏离最佳实践的情况。您可以准备用于新服务发布的 ORR 以及用于定期审查的 ORR。这可以帮助您遵循最新制定的最佳实践，并吸取从事后分析中学到的经验教训。随着您对云的使用日趋成熟，您可以将 ORR 要求作为默认设置整合到自己的架构中。

期望的结果： 您已准备好 ORR 核对清单，其中包括适合您组织的最佳实践。在工作负载发布之前运行 ORR。在整个工作负载生命周期中定期运行 ORR。

常见反模式：

- 您启动了工作负载，但不知道谁负责其运维工作。
- 在验证工作负载以便发布时，没有包括监管和安全性要求。
- 没有定期重新评估工作负载。
- 发布工作负载而没有准备好所需的规程。
- 您在多个工作负载中看到相同的根本原因反复导致出现故障。

建立此最佳实践的好处：

- 您的工作负载包括架构、流程和管理最佳实践。
- 学到的经验教训可合并到 ORR 流程中。
- 在工作负载发布时已准备好所需的规程。
- 在工作负载的整个软件生命周期中运行 ORR。

未建立这种最佳实践的情况下的风险等级： 高

实施指导

ORR 关系到两点：流程和核对清单。ORR 流程应该由您的组织采用并获得高管支持。至少，ORR 必须在工作负载正式公开发布之前已运行。在整个软件开发生命周期中运行 ORR 可确保软件始终遵循新的最佳实践或新要求。ORR 核对清单应包括配置项目、安全性和监管要求，以及组织的最佳实践。在一段时间后，您可以使用 AWS Config、 AWS Security Hub和 AWS Control Tower 防护机制等服务，将源自 ORR 的最佳实践整合到防护机制中，以实现自动化的最佳实践检测。

客户示例

在经历了多起生产事件之后，AnyCompany Retail 决定实施 ORR 流程。他们构建了核对清单，其中包括最佳实践、监管和合规性要求，以及从中断中学到的经验教训。在发布新工作负载之前，运行 ORR。每个工作负载会每年运行一次 ORR，其中包括一小组最佳实践，用于整合添加到 ORR 核对清单中的新最佳实践和要求。在一段时间后，AnyCompany Retail 使用 AWS Config 来检测一些最佳实践，以加快 ORR 流程。

实施步骤

如需详细了解 ORR，请阅读 运维准备情况审查（ORR）白皮书。其中详细介绍了 ORR 流程的历史，如何构建自己的 ORR 实践，以及如何制定自己的 ORR 核对清单。以下步骤是该文档的缩减版本。如需深入了解什么是 ORR 以及如何自行构建，建议您阅读该白皮书。

1. 让关键利益相关方聚在一起讨论，包括来自安全、运维和开发部门的代表。
2. 让每个利益相关方至少提一个要求。对于第一次迭代，请尝试将项目数限制为不超过三十个。
    - 附录 B：ORR 问题示例 源自运维准备情况审查（ORR）白皮书，包含您在开始着手时可借鉴的示例问题。
3. 在电子表格中收集您的要求。
    - 您可以使用 自定义剖析 （位于 AWS Well-Architected Tool 中）开发自己的 ORR，并跨账户以及在 AWS Organization 中分享它们。
4. 确定一个工作负载来运行 ORR。最好选择发布前的工作负载或者内部工作负载。
5. 运行 ORR 核对清单并记录任何发现结果。如果已经有防范措施，那么发现结果可能就不太重要。对于任何没有防范措施的发现结果，请将它们记录到项目的待办事项中，并在发布之前实施它们。
6. 在一段时间后，继续在 ORR 中添加最佳实践和要求。

具有 Enterprise Support 的 AWS Support 客户可以向其技术客户经理请求举行 运维准备情况审查研讨会 。该研讨会是一个交互式研讨会，采用 反推式工作方法 ，可帮助您制定自己的 ORR 核对清单。

实施计划的工作量级别： 高。在组织中采用 ORR 实践需要获得高管以及利益相关方的支持。使用整个组织中获得的反馈意见来构建和更新核对清单。

资源

相关最佳实践：

- OPS01-BP03 评估监管要求 – 监管要求非常适合包括在 ORR 核对清单中。
- OPS01-BP04 评估合规性要求 – 合规性要求有时候包括在 ORR 核对清单中。另一些时候它们可作为单独的流程。
- OPS03-BP07 为团队配置适当的资源 – 团队能力是很适合加入 ORR 要求的候选项。
- OPS06-BP01 针对不成功的更改制定计划 – 在发布工作负载之前，必须建立回滚或前滚计划。
- OPS07-BP01 确保员工能力 – 为了支持工作负载，您必须具备所需的人员。
- SEC01-BP03 识别并验证控制目标 – 安全控制目标会是非常合适的 ORR 要求。
- REL13-BP01 定义停机和数据丢失的恢复目标 – 灾难恢复计划是很好的 ORR 要求。
- COST02-BP01 根据组织的要求制定各种策略 – 成本管理策略非常适合包括在 ORR 核对清单中。

相关文档：

- AWS Control Tower – AWS Control Tower 中的防护机制

- AWS Well-Architected Tool – 自定义剖析
- Adrian Hornsby 提供的运维准备情况审查模板
- 运维准备情况审查（ORR）白皮书

相关视频：

- AWS Support 为您提供支持 | 构建高效的运维准备情况审查（ORR，Operational Readiness Review）

相关示例：

- 运维准备情况审查（ORR）剖析

相关服务：

- AWS Config
- AWS Control Tower
- AWS Security Hub
- AWS Well-Architected Tool

OPS07-BP03 使用运行手册执行程序

A 运行手册 是实现特定结果的书面流程。运行手册由某人为完成某件事而遵循的一系列步骤组成。早在航空发展的早期，运行手册便已用于运营。在云运营中，我们使用运行手册来降低风险并实现预期结果。简单而言，运行手册就是完成一项任务的核对清单。

运行手册是运营工作负载的重要组成部分。从新团队成员入职到部署一个主要版本，运行手册都是一个成文的流程，无论谁使用它们，都能获得一致的结果。运行手册应发布在一个中央位置，并随着流程的发展而更新，因为更新运行手册是变更管理流程的一个关键组成部分。它们还应包括关于错误处理、工具、权限、异常和问题发生时上报的指导。

随着贵组织日益成熟，开始自动编写运行手册。从简短且经常使用的运行手册开始。使用脚本语言来实现步骤自动化或使步骤更容易执行。当您自动化前几本运行手册后，您将花时间自动化更复杂的运行手册。随着时间的推移，大多数运行手册应以某种方式实现自动化。

期望结果： 您的团队有一系列执行工作负载任务的分步指南。运行手册包含期望结果、必要的工具和权限，以及关于错误处理的说明。它们存储在一个中央位置并经常更新。

常见反模式：

- 依靠记忆完成流程的每个步骤。
- 手动部署更改而不使用核对清单。
- 不同的团队成员执行相同的流程，但执行不同的步骤或取得不同的结果。
- 让运行手册与系统更改和自动化不同步。

建立此最佳实践的好处：

- 降低人工任务的错误率。
- 以一致的方式执行操作。
- 新的团队成员可以更早地开始执行任务。
- 可以自动编写运行手册以减少工作量。

未建立这种最佳实践的情况下暴露的风险等级： 中

实施指导

根据贵组织的成熟度级别，运行手册可以采用多种形式。它们至少应该包含一个分步文本文档。应明确指出期望结果。清楚地记录必要的特殊权限或工具。提供关于错误处理和出现问题时进行上报的详细指导。列出运行手册负责人，并将运行手册发布在一个中央位置。一旦运行手册编写完成，让您团队中的其他人运行它来进行验证。随着过程的发展，根据变更管理流程更新运行手册。

随着贵组织日益成熟，您的文本运行手册应实现自动化。使用诸如 AWS Systems Manager 自动化之类的服务，您可以将纯文本转换为可针对您的工作负载运行的自动化功能。这些自动化功能可以根据事件的发生而运行，从而减轻维持工作负载的运营负担。

客户示例

AnyCompany Retail 必须在软件部署期间执行数据库模式更新。云运营团队与数据库管理团队合作，构建了一个用于手动部署这些更改的运行手册。运行手册以核对清单的形式列出了流程中的每个步骤。其中有一节是关于出错时的错误处理。他们在内部 Wiki 上发布了该运行手册和其他运行手册。云运营团队计划在未来的冲刺阶段实现运行手册的自动化。

实施步骤

如果您没有现有的文档存储库，那么版本控制存储库是开始构建运行手册库的绝佳场所。您可以使用 Markdown 构建运行手册。我们提供了一个示例运行手册模板，您可以用它开始构建运行手册。

```
# Runbook Title ## Runbook Info | Runbook ID | Description | Tools Used
 | Special Permissions | Runbook Author | Last Updated | Escalation POC |
 |-------|-------|-------|-------|-------|-------|-------| | RUN001 | What is this
 runbook for? What is the desired outcome? | Tools | Permissions | Your Name |
 2022-09-21 | Escalation Name | ## Steps 1.Step one 2.Step two
```

1. 如果您当前尚没有文档存储库或 Wiki，请在版本控制系统中创建一个新的版本控制存储库。
2. 识别一个没有运行手册的流程。一个理想的流程是半定期执行的流程，步骤少，且故障影响小。
3. 在文档存储库中，使用模板创建新的草稿 Markdown 文档。填写 `Runbook Title` 以及 `Runbook Info ######`。
4. 从第一步开始，填写运行手册的 `Steps` 部分。
5. 将运行手册交给团队成员。让他们使用运行手册来验证这些步骤。如果有遗漏或需要澄清的地方，请更新运行手册。
6. 将运行手册发布到您的内部文档存储区。发布后，告诉您的团队和其他利益相关者。
7. 随着时间的推移，您将构建一个运行手册库。随着该库的增长，开始努力实现运行手册的自动化。

实施计划的工作量级别： 低。运行手册的最低标准是一个分步文本指南。实现运行手册自动化可能会增加实施工作量。

资源

相关最佳实践：

- OPS02-BP02 确定流程和程序所有者：运行手册应该有一个负责人负责维护。
- OPS07-BP04 根据行动手册调查问题：运行手册和行动手册彼此相似，但有一个关键区别：运行手册包含期望结果。在许多情况下，一旦行动手册确定了根本原因，就会触发运行手册。
- OPS10-BP01 使用流程来管理事件、意外事件和问题：运行手册是良好的事件、意外事件和问题管理实践的一部分。
- OPS10-BP02 针对每个提醒设置一个流程：应使用运行手册和行动手册来响应警报。随着时间的推移，应自动进行这些响应。
- OPS11-BP04 执行知识管理：维护运行手册是知识管理的一个关键部分。

相关文档：

- 利用自动化行动手册和运行手册实现卓越运营

- AWS Systems Manager：使用运行手册
- 适用于 AWS 大型迁移的迁移行动手册 - 任务 4：改进迁移运行手册
- 使用 AWS Systems Manager Automation 运行手册解决运营任务

相关视频：

- AWS re:Invent 2019：运行手册、事件报告和事件响应 DIY 指南（SEC318-R1）
- 如何在 AWS | Amazon Web Services 上实现 IT 运营自动化
- 将脚本集成到 AWS Systems Manager

相关示例：

- AWS Systems Manager：自动化演练
- AWS Systems Manager：从最新的快照运行手册中还原根卷
- 使用 Jupyter Notebook 和 CloudTrail Lake 构建 AWS 意外事件响应运行手册
- Gitlab - 运行手册
- Rubix - 用于在 Jupyter Notebook 中构建运行手册的 Python 库
- 使用 Document Builder 创建自定义运行手册
- Well-Architected 实验室：使用行动手册和运行手册自动完成操作

相关服务：

- AWS Systems Manager Automation

OPS07-BP04 根据行动手册调查问题

行动手册是用于调查事件的分步指南。发生事件时，行动手册用于开展调查，以及确定影响的范围和根本原因。行动手册可用于从失败的部署到安全事件的各种场景。在许多情况下，行动手册可确定根本原因，而运行手册可用来缓解其带来的风险。行动手册是贵组织事件响应计划的必要组成部分。

出色的行动手册有几个主要特点。它逐步指导用户完成事件发现过程。由外而内地思考，用户应执行哪些步骤来诊断事件？ 如果行动手册中需要特殊工具或提升的权限，请在行动手册中明确地定义。请制定沟通计划，以向利益相关者提供有关调查状态的最新信息，这是事件响应计划的一个重要组成部分。在无法确定根本原因的情况下，行动手册应制定上报计划。如果确定了根本原因，行动手册应指出介绍

如何解决根本原因的运行手册。应集中存储并定期维护行动手册。如果行动手册用于特定提醒，请向团队提供关于提醒中的行动手册的提示。

随着组织日趋成熟，可自动实施行动手册。从包含低风险事件的行动手册开始实施。使用脚本自动执行发现步骤。确保有配套的运行手册来缓解常见根本原因带来的风险。

期望的结果： 您的组织有针对常见事件的行动手册。行动手册集中存储在一个位置，可供团队成员使用。行动手册经常进行更新。对于任何已知的根本原因，将制定配套的运行手册。

常见反模式：

- 要调查事件，并没有标准方法。
- 团队成员依靠肌肉记忆或对机构的了解，对失败的部署进行排查。
- 新的团队成员将学习如何通过试错法来调查问题。
- 调查问题的最佳实践无法在不同团队之间共享。

建立此最佳实践的好处：

- 行动手册可帮助您减轻事件带来的影响。
- 不同的团队成员可使用同一行动手册，以一致的方式确定根本原因。
- 可以针对已知的根本原因制定运行手册，从而加快恢复速度。
- 团队成员根据行动手册能够更快地开始行动。
- 团队可以使用可重复的行动手册来扩展其流程。

未建立这种最佳实践的情况下暴露的风险等级： 中

实施指导

制定和使用行动手册的方式取决于组织的成熟度。如果您是初次使用云，请在中央文档存储库中以文本形式制定行动手册。随着组织日趋成熟，可以使用 Python 等脚本语言实现行动手册的半自动化。可以在 Jupyter notebook 中运行这些脚本来加快发现速度。先进的组织已针对可通过运行手册自动修正的常见问题，制定完全自动化的行动手册。

通过列出工作负载所发生的常见事件，开始制定行动手册。为风险较低且根本原因范围已缩小到几个问题的事件选择行动手册。在为较简单的场景制定行动手册后，可以着手处理风险较高的场景或根本原因尚不确定的场景。

随着贵组织日趋成熟，您的文本样式的行动手册应实现自动化。通过使用诸如 AWS Systems Manager Automations之类的服务，可以将纯文本转换为自动化代码。可以针对工作负载运行这些自动化代码，从而加快调查速度。可以激活这些自动化代码以响应事件，从而减少发现和解决事件所需的平均时间。

客户可以使用 AWS Systems Manager Incident Manager 来响应事件。此服务提供了单一界面对事件进行分类，在发现和缓解问题期间通知利益相关者，并在整个事件中进行协作。它使用 AWS Systems Manager Automations 加快检测和恢复的速度。

客户示例

生产事件影响了 AnyCompany Retail。随时待命的工程师根据行动手册调查了问题。随着他们逐步地解决问题，他们不断为行动手册中确定的关键利益相关者提供最新信息。工程师最终确定，根本原因是后端服务中出现竞态条件。根据运行手册，工程师重新启动了该服务，并使 AnyCompany Retail 重新联机。

实施步骤

如果您当前没有文档存储库，建议您为行动手册库创建版本控制存储库。您可以使用 Markdown 制定您的行动手册，该服务兼容大多数行动手册自动化系统。如果您从头开始制定行动手册，请使用以下行动手册示例模板。

```
# Playbook Title ## Playbook Info | Playbook ID | Description
 | Tools Used | Special Permissions | Playbook Author | Last
 Updated | Escalation POC | Stakeholders | Communication Plan |
 |-------|-------|-------|-------|-------|-------|-------|-------|-------| | RUN001
 | What is this playbook for? What incident is it used for? | Tools | Permissions |
 Your Name | 2022-09-21 | Escalation Name | Stakeholder Name | How will updates be
 communicated during the investigation? | ## Steps 1.Step one 2.Step two
```

1. 如果您当前没有文档存储库或 Wiki，请在版本控制系统中为行动手册创建一个新的版本控制存储库。
2. 确定需要调查的一个常见问题。它应该是根本原因范围限于几个问题且解决方案风险较低的场景。
3. 使用 Markdown 模板填写 `Playbook Name######## 部分以及 Playbook Info########`下的字段。
4. 填写问题排查步骤。尽可能清楚地知道要采取哪些行动，或者应调查哪些方面。
5. 将行动手册提供给团队成员，让他们仔细阅读并加以验证。如果发现有遗漏之处或某些内容不清楚，请更新行动手册。
6. 在文档存储库中发布您的行动手册，并告知您的团队和任何利益相关者。

7. 随着您添加更多的行动手册，这个行动手册库将会不断扩大。在您拥有多个行动手册后，可以开始使用 AWS Systems Manager Automations 等工具自动执行它们，从而使自动化操作和行动手册保持同步。

实施计划的工作量级别： 低。行动手册应该是集中存储在一个位置的文本文档。对于更加成熟的组织，将转为自动实施行动手册。

资源

相关最佳实践：

- OPS02-BP02 确定流程和程序所有者：行动手册应该有一个负责人来负责维护。
- OPS07-BP03 使用运行手册执行程序：运行手册和行动手册类似，但有一个关键区别：运行手册包含期望的结果。在许多情况下，一旦行动手册确定了根本原因，就会使用运行手册。
- OPS10-BP01 使用流程来管理事件、意外事件和问题：行动手册是良好的事件、意外事件和问题管理实践的一部分。
- OPS10-BP02 针对每个提醒设置一个流程：应使用运行手册和行动手册来响应警报。随着时间的推移，应自动进行这些响应。
- OPS11-BP04 执行知识管理：维护行动手册是知识管理的一个关键部分。

相关文档：

- 利用自动化行动手册和运行手册实现卓越运营
- AWS Systems Manager：使用运行手册
- 使用 AWS Systems Manager Automation 运行手册解决运营任务

相关视频：

- AWS re:Invent 2019：运行手册、事件报告和事件响应 DIY 指南（SEC318-R1）
- AWS Systems Manager Incident Manager – AWS 虚拟研讨会
- 将脚本集成到 AWS Systems Manager

相关示例：

- AWS 客户行动手册框架
- AWS Systems Manager：自动化演练

- 使用 Jupyter notebook 和 CloudTrail Lake 构建 AWS 事件响应运行手册
- Rubix – 用于在 Jupyter notebook 中构建运行手册的 Python 库
- 使用 Document Builder 创建自定义运行手册
- Well-Architected 实验室：根据行动手册和运行手册自动完成操作
- Well-Architected 实验室：使用 Jupyter 的事件响应行动手册

相关服务：

- AWS Systems Manager Automation
- AWS Systems Manager Incident Manager

OPS07-BP05 做出明智的决策来部署系统和更改

评估团队支持工作负载的能力以及工作负载的监管合规性。在决定是否将系统或更改投入生产环境时，将这些与部署的收益进行比较。了解收益和风险，以便做出明智的决策。

故障演练是一种演习，团队模拟发生故障的情况来制定防范策略。使用故障演练来预测故障，并根据需要创建程序。当您对用于评估工作负载的检查清单进行更改时，请计划要对不再符合条件的活动系统执行哪些操作。

常见反模式：

- 决定在以下情况下部署工作负载：不了解工作负载中存在安全风险。
- 决定在以下情况下部署工作负载：不了解它是否符合监管要求和标准。
- 决定在以下情况下部署工作负载：不了解您的团队是否可以为它提供支持。
- 决定在以下情况下部署工作负载：不了解组织会如何从中获益。

建立此最佳实践的好处： 拥有技能娴熟的团队成员能够为您的工作负载提供有效支持。

未建立这种最佳实践的情况下暴露的风险等级： 低

实施指导

- 做出部署工作负载和更改的明智决策：评估团队支持工作负载的能力以及工作负载的监管合规性。在决定是否将系统或更改投入生产环境时，将这些与部署的收益进行比较。了解收益和风险，并做出明智的决策。

运营

问题

- OPS 8 您如何了解工作负载的运行状况？
- OPS 9 您如何了解自己的运营状况？
- OPS 10 您如何应对工作负载事件和运营事件？

OPS 8 您如何了解工作负载的运行状况？

定义、记录和分析指标以便了解工作负载事件，从而采取适当的措施。

最佳实践

- OPS08-BP01 识别关键性能指标
- OPS08-BP02 定义工作负载指标
- OPS08-BP03 收集和分析工作负载指标
- OPS08-BP04 建立工作负载指标基准
- OPS08-BP05 了解工作负载的预期活动模式
- OPS08-BP06 在工作负载成果面临风险时发出提醒
- OPS08-BP07 在检测到工作负载异常时发出提醒
- OPS08-BP08 验证实现的成果以及 KPI 和指标的有效性

OPS08-BP01 识别关键性能指标

根据期望的业务成果（例如，订单率、客户保留率和利润与运营开支）和客户成果（例如，客户满意度）识别识别关键性能指标 (KPI)。评估 KPI 以便确定工作负载是否成功。

常见反模式：

- 业务领导会问您，工作负载在满足业务需求方面成效如何，但却没有确定成功的参考框架。
- 您无法确定您为组织运行的现有商用应用程序是否具有成本效益。

建立此最佳实践的好处： 通过识别识别关键性能指标，您可以将业务成果的实现情况作为对工作负载运行状况和是否成功的测试。

未建立这种最佳实践的情况下暴露的风险等级： 高

实施指导

- 识别关键性能指标：根据所需的业务成果和客户成果识别关键性能指标（KPI，Key Performance Indicator）。评估 KPI 以便确定工作负载是否成功。

OPS08-BP02 定义工作负载指标

定义工作负载指标来衡量 KPI（例如，放弃的购物车、下达的订单、成本、价格和分配的工作负载费用）的完成情况。定义工作负载指标以衡量工作负载的运行状况（例如，接口响应时间、错误率、提出的请求数、完成的请求数和利用率）。评估指标以便确定工作负载是否实现所需成果，并了解工作负载的运行状况。

您应将日志数据发送到像 CloudWatch Logs 这样的服务，并根据对必要日志内容的观察生成指标。

CloudWatch 具有一些专业功能，例如 适用于 .NET 和 SQL Server 的 Amazon CloudWatch Insights 和 Container Insights，这些功能可通过识别和设置专门支持的应用程序资源和技术堆栈的关键指标、日志和告警来为您提供帮助。

常见反模式：

- 您定义了标准指标，这些指标没有关联任何 KPI，也并非针对任何工作负载量身定制。
- 您的指标计算中存在会产生无效结果的错误。
- 您没有为工作负载定义任何指标。
- 您只衡量可用性。

建立此最佳实践的好处： 通过定义和评估工作负载指标，您可以确定工作负载的运行状况并衡量业务成果的实现情况。

未建立此最佳实践暴露的风险等级： 高

实施指导

- 定义工作负载指标：定义工作负载指标来衡量 KPI 的实现情况。定义工作负载指标来衡量工作负载及其各个组件的运行状况。评估指标以便确定工作负载是否实现所需成果，并了解工作负载的运行状况。
  - 发布自定义指标
  - 搜索和筛选日志数据
  - Amazon CloudWatch 指标和维度参考

资源

相关文档：

- Amazon CloudWatch 指标和维度参考
- 发布自定义指标
- 搜索和筛选日志数据

OPS08-BP03 收集和分析工作负载指标

定期主动检查各种指标，以便发现趋势并确定哪里需要做出适当响应。

您应汇总应用程序、工作负载组件、服务以及对服务（如 CloudWatch Logs）的 API 调用的日志数据。通过对必要的日志内容进行观察生成指标，从而深入了解运营活动的表现。

在 AWS 上，您可以使用 Amazon DevOps Guru的机器学习功能分析工作负载指标并识别运营问题。AWS DevOps Guru 会提供运营问题通知，并给出 有针对性的主动 建议，以解决问题并保持应用程序正常运行。

在 AWS 责任共担模式中，部分监控会通过以下控制面板提供给您： AWS Health Dashboard。此控制面板会在 AWS 遇到可能会影响您的事件时提供提醒和修正指导。拥有商业支持和企业支持订阅的客户还可以获取 AWS Health API，从而实现与事件管理系统的集成。

在 AWS 上，您可以 将您的日志数据导出到 Amazon S3 或者 直接将日志发送 到 Amazon S3 以便长期存储。使用 AWS Glue，您可以在 Amazon S3 中发现并准备您的日志数据以供分析，并将相关元数据存储在以下位置： AWSAWS Glue Data Catalog。 Amazon Athena通过与 AWS Glue 的原生集成，可用于分析您的日志数据，并使用标准 SQL 进行查询。使用像 Amazon QuickSight 这样的商业智能工具，您可以直观显示、浏览和分析您的数据。

另一种 解决方案 是使用 Amazon OpenSearch Service 和 OpenSearch 控制面板 来收集、分析和显示跨多个账户和 AWS 区域的 AWS 日志。

常见反模式：

- 网络设计团队询问您当前的网络带宽利用率。您提供当前指标，网络利用率为 35%。他们降低了电路容量，这样可以节省成本，但是导致了大量的连接问题，这是因为您的时间点测量没有反映出利用率的趋势。
- 您的路由器出现故障。它一直以越来越高的频率记录非关键内存错误，直到出现故障。您没有发现这种趋势，因此没有在路由器引起服务中断之前更换故障内存。

建立此最佳实践的好处： 通过收集和分析工作负载指标，您可以了解工作负载的运行状况，并可以洞悉可能影响工作负载或业务成果完成情况的趋势。

未建立这种最佳实践的情况下暴露的风险等级： 高

实施指导

- 收集和分析工作负载指标：定期主动检查各种指标，以便发现趋势并确定哪里需要做出适当响应。
  - 使用 Amazon CloudWatch 指标
  - Amazon CloudWatch 指标和维度参考
  - 使用 CloudWatch 代理从 Amazon EC2 实例和本地服务器收集指标和日志

资源

相关文档：

- Amazon Athena
- Amazon CloudWatch 指标和维度参考
- Amazon DevOps Guru
- AWS Glue
- AWSAWS Glue Data Catalog
- Amazon OpenSearch Service
- AWS Health Dashboard
- Amazon QuickSight
- 使用 CloudWatch 代理从 Amazon EC2 实例和本地服务器收集指标和日志
- 使用 Amazon CloudWatch 指标

OPS08-BP04 建立工作负载指标基准

建立指标基准以便提供预期值，作为比较和识别性能不足和性能过剩组件的依据。确定改进、调查和干预的阈值。

常见反模式：

- 服务器以 95% 的 CPU 利用率运行，您被问及这种情况是好是坏。由于没有为该服务器的 CPU 利用率设定基准，因此您也不知道是好是坏。

建立此最佳实践的好处： 通过定义基准指标值，您可以评估当前指标值和指标趋势，从而确定是否需要采取措施。

未建立这种最佳实践的情况下暴露的风险等级： 中

实施指导

- 建立工作负载指标基准：建立工作负载指标基准，以便提供期望值作为比较依据。
  - 创建 Amazon CloudWatch 警报

资源

相关文档：

- 创建 Amazon CloudWatch 警报

OPS08-BP05 了解工作负载的预期活动模式

通过建立工作负载活动的模式来识别异常行为，以便您可以在需要时做出适当的响应。

CloudWatch 通过 CloudWatch 异常检测 功能来应用统计和机器学习算法，以生成代表正常指标行为的预期值范围。

Amazon DevOps Guru 可通过事件关联、日志分析并应用机器学习来分析工作负载遥测数据，用于确定异常行为。在检测到意外行为时，它会提供 相关的指标和事件， 并给出应对该行为的推荐方案。

常见反模式：

- 您正在查看网络利用率日志，发现网络利用率在上午 11:30 至下午 1:30 之间上升，然后在下午 4:30 至 6:00 之间再次上升。您不知道这是否正常。
- 您的 Web 服务器在每天凌晨 3:00 重新启动。您不知道这是否是预期行为。

建立此最佳实践的好处： 通过学习行为模式，您可以识别意外行为并在必要时采取措施。

未建立这种最佳实践的情况下暴露的风险等级： 中

实施指导

- 了解工作负载的预期活动模式：建立工作负载活动模式以便确定行为何时不符合预期值，从而根据需要做出适当响应。

资源

相关文档：

- Amazon DevOps Guru
- CloudWatch 异常检测

OPS08-BP06 在工作负载成果面临风险时发出提醒

在工作负载成果面临风险时发出提醒，从而在必要时做出适当响应。

理想情况下，您之前已经确定能够作为发出提醒依据的指标阈值，或可以用于触发自动响应的事件。

在 AWS 上，您可以使用 Amazon CloudWatch Synthetics 创建金丝雀脚本，通过执行与客户相同的操作，监控您的端点和 API。通过生成的遥测数据以及 发掘的洞察， 您可以在客户受到损害之前确定问题。

您也可以使用 CloudWatch Logs Insights 和专门构建的查询语言以交互方式搜索和分析您的日志数据。CloudWatch Logs Insights 自动 发现 AWS 服务日志中的字段以及 JSON 格式的自定义日志事件。它会随您的日志量和查询复杂性而扩展，并在数秒内为您提供答案，从而帮助您搜索引发事件的因素。

常见反模式：

- 您的网络断开连接。没有人发现这一情况。没有人尝试确定原因或采取措施来恢复网络连接。
- 安装补丁后，您的持久性实例开始无法访问，这会对用户造成影响。您的用户创建了支持案例。没有人收到通知。没有人采取措施。

建立此最佳实践的好处： 如果可以发现业务成果处于危险之中并提醒需要采取措施，您就有机会预防意外事件的发生或者减轻意外事件的影响。

未建立此最佳实践暴露的风险等级： 中

实施指导

- 在工作负载成果面临风险时发出提醒：在工作负载成果面临风险时发出提醒，以便您能够根据需要做出适当响应。
  - 什么是 Amazon CloudWatch Events？
  - 创建 Amazon CloudWatch 警报
  - 使用 Amazon SNS 通知调用 Lambda 函数

资源

相关文档：

- Amazon CloudWatch Synthetics
- CloudWatch Logs Insights
- 创建 Amazon CloudWatch 警报
- 使用 Amazon SNS 通知调用 Lambda 函数
- 什么是 Amazon CloudWatch Events？

OPS08-BP07 在检测到工作负载异常时发出提醒

在检测到工作负载异常时发出提醒，从而在必要时做出适当响应。

您对一段时间内工作负载指标的分析可能会建立行为模式，您可以对这些模式进行充分量化，以定义事件或发出警报作为响应。

经过训练后， CloudWatch 异常检测 功能可用于 对 检测到的异常发出警报，或将期望值叠加到指标数据 图表 上，以进行持续的比较。

常见反模式：

- 您零售网站的销量突然急剧增加。没有人发现这一情况。没有人尝试找出导致这种激增的原因。没有人采取措施来让客户在额外负载下仍然保持优质体验。
- 应用补丁后，您的持久性服务器频繁重启，这会对用户造成影响。通常情况下，服务器最多重启三次。没有人发现这一情况。没有人尝试确定发生这种情况的原因。

建立此最佳实践的好处： 通过了解工作负载行为的模式，您可以发现意外行为，并在必要时采取措施。

未建立这种最佳实践的情况下暴露的风险等级： 低

实施指导

- 在检测到工作负载异常时发出提醒：在检测到工作负载异常时发出提醒，以便您能够根据需要做出适当响应。
  - 什么是 Amazon CloudWatch Events？
  - 创建 Amazon CloudWatch 警报

- 使用 Amazon SNS 通知调用 Lambda 函数

资源

相关文档：

- 创建 Amazon CloudWatch 警报
- CloudWatch 异常检测
- 使用 Amazon SNS 通知调用 Lambda 函数
- 什么是 Amazon CloudWatch Events？

OPS08-BP08 验证实现的成果以及 KPI 和指标的有效性

在业务层面查看工作负载的运行情况，以便确定自己是否满足需求，并确定需要改进哪些方面才能实现业务目标。验证 KPI 和指标的有效性并在需要时进行修改。

AWS 还通过 AWS 服务 API 和 SDK（例如，Grafana、Kibana 和 Logstash）支持第三方日志分析系统和商业智能工具。

常见反模式：

- 从未将页面响应时间视作影响客户满意度的因素。您从未对页面响应时间设定指标或阈值。您的客户投诉称响应缓慢。
- 您尚未达到最低响应时间目标。为了缩短响应时间，您已经纵向扩展了应用程序服务器。现在，响应时间缩短，远远超出了目标；而且还有大量已付费的未使用容量。

建立此最佳实践的好处： 通过审核和修订 KPI 及指标，您可以了解工作负载如何支持业务成果的实现，并可以确定需要对哪些方面进行改进以实现业务目标。

未建立这种最佳实践的情况下暴露的风险等级： 低

实施指导

- 验证实现的成果以及 KPI 和指标的有效性：在业务层面查看工作负载运营情况，以便帮助您确定自己是否满足需求，并确定需要改进哪些方面才能实现业务目标。验证 KPI 和指标的有效性并在需要时进行修改。
  - 使用 Amazon CloudWatch 控制面板
  - 什么是日志分析？

资源

相关文档：

- 使用 Amazon CloudWatch 控制面板
- 什么是日志分析？

OPS 9 您如何了解自己的运营状况？

定义、记录和分析运营指标以便了解运营事件，从而采取适当的措施。

最佳实践

- OPS09-BP01 识别关键性能指标
- OPS09-BP02 定义运营指标
- OPS09-BP03 收集和分析运营指标
- OPS09-BP04 建立运营指标基准
- OPS09-BP05 了解运营的预期活动模式
- OPS09-BP06 在运营成果面临风险时发出提醒
- OPS09-BP07 在检测到运营异常时发出提醒
- OPS09-BP08 验证实现的成果以及 KPI 和指标的有效性

OPS09-BP01 识别关键性能指标

根据期望的业务成果（如交付新功能）和客户成果（如客户支持案例）识别关键性能指标（KPI，Key Performance Indicator）。评估 KPI 以便确定运营是否成功。

常见反模式：

- 业务领导会问您，运营在完成业务目标方面成效如何，但却没有确定成功的参考框架。
- 您无法确定维护时段是否会对业务成果产生影响。

建立此最佳实践的好处： 通过识别识别关键性能指标，您可以将业务成果的实现情况作为对运营运行状况和是否成功的测试。

未建立这种最佳实践的情况下暴露的风险等级： 高

实施指导

- 识别关键性能指标：根据所需的业务成果和客户成果识别关键性能指标（KPI，Key Performance Indicator）。评估 KPI 以便确定运营是否成功。

OPS09-BP02 定义运营指标

定义运营指标以衡量 KPI 的实现情况（例如，成功的部署和失败的部署）。定义运营指标以衡量运营活动的运行状况（例如，事件的平均检测时间 (MTTD) 和事件的平均恢复时间 (MTTR)）。评估指标以便确定运营是否已实现期望的成果，并了解运营活动的运行状况。

常见反模式：

- 根据团队认为的合理情况来确定运营指标。
- 您的指标计算中存在会产生不正确结果的错误。
- 您没有为运营活动定义任何指标。

建立此最佳实践的好处： 通过定义和评估运营指标，您可以确定运营活动的运行状况并衡量业务成果实现情况。

未建立这种最佳实践的情况下暴露的风险等级： 高

实施指导

- 定义运营指标：定义运营指标来衡量 KPI 的实现情况。定义运营指标来衡量运营状况及其活动的运行状况。评估指标以便确定运营是否实现所需成果，并了解运营状况。
  - 发布自定义指标
  - 搜索和筛选日志数据
  - Amazon CloudWatch 指标和维度参考

资源

相关文档：

- AWS Answers：集中式日志记录
- Amazon CloudWatch 指标和维度参考
- 使用 Amazon CloudWatch Events 检测管道状态的更改并做出反应

- 发布自定义指标
- 搜索和筛选日志数据

相关视频：

- 制定监控计划

OPS09-BP03 收集和分析运营指标

定期主动审核各种指标，以便发现趋势并确定哪里需要做出适当响应。

您应该将来自操作活动执行和操作 API 调用的日志数据聚合到像 CloudWatch Logs 这样的服务中。根据对必要日志内容的观察生成指标，从而深入了解运营活动的性能。

在 AWS 上，您可以 将您的日志数据导出到 Amazon S3 或者 直接将日志发送 到 Amazon S3 以便长期存储。使用 AWS Glue，您可以在 Amazon S3 中发现并准备您的日志数据以供分析，并将相关元数据存储在以下位置： AWSAWS Glue Data Catalog. Amazon Athena通过与 AWS Glue 的原生集成，可用于分析您的日志数据，并使用标准 SQL 进行查询。使用像 Amazon QuickSight 这样的商业智能工具，您可以直观显示、浏览和分析您的数据。

常见反模式：

- 一个识别关键性能指标是始终如一地交付新功能。您没有衡量部署频率的方法。
- 您记录部署、回滚部署、安装补丁和回滚补丁，以跟踪您的运营活动，但是没有人审核指标。
- 您有一个恢复时间目标，要在十五分钟内将丢失的数据库恢复，这是在部署系统且还没有用户时定义的。现在，您有成千上万的用户，并且已经运营了两年。最近一次恢复花费了两个多小时。没有对此进行记录，也没有人知道。

建立此最佳实践的好处： 通过收集和分析运营指标，您可以了解运营活动的运行状况，并可以洞察可能影响运营或业务成果完成情况的趋势。

未建立这种最佳实践的情况下暴露的风险等级： 高

实施指导

- 收集和分析运营指标：定期主动检查各种指标，以便发现趋势并确定哪里需要做出适当响应。
  - 使用 Amazon CloudWatch 指标
  - Amazon CloudWatch 指标和维度参考

- 使用 CloudWatch 代理从 Amazon EC2 实例和本地服务器收集指标和日志

资源

相关文档：

- Amazon Athena
- Amazon CloudWatch 指标和维度参考
- Amazon QuickSight
- AWS Glue
- AWSAWS Glue Data Catalog
- 使用 CloudWatch 代理从 Amazon EC2 实例和本地服务器收集指标和日志
- 使用 Amazon CloudWatch 指标

OPS09-BP04 建立运营指标基准

建立指标基准以便提供预期值，作为比较和识别运营活动执行不足和运营活动执行过度的依据。

常见反模式：

- 您被问及预期的部署时长。您尚未测量部署所需的时间，也无法确定预期时间。
- 您被问及从应用程序服务器问题中恢复所需的时间。您不知道从首次联系客户到恢复完成的时长。您不知道从首次通过监控发现问题到恢复完成的时长。
- 您被问及周末需要多少支持人员。您不知道周末通常有多少支持案例，无法估算。
- 您有一个恢复时间目标，要在十五分钟内将丢失的数据库恢复，这是在部署系统且还没有用户时定义的。现在，您有成千上万的用户，并且已经运营了两年。您不知道数据库的还原时间是如何变化的。

建立此最佳实践的好处： 通过定义基准指标值，您可以评估当前指标值和指标趋势，从而确定是否需要采取措施。

未建立此最佳实践暴露的风险等级： 中

实施指导

- 了解运营的预期活动模式：建立运营活动模式以便确定行为何时不符合预期值，从而根据需要做出适当响应。

OPS09-BP05 了解运营的预期活动模式

建立运营活动的模式来识别异常行为，以便您在必要时做出适当响应。

常见反模式：

- 最近，您的部署失败率大幅增加。您单独处理每个故障。您没有发现，故障是由对部署管理系统不熟悉的新员工所执行的部署引发的。

建立此最佳实践的好处： 通过学习行为模式，您可以识别意外行为并在必要时采取措施。

未建立此最佳实践暴露的风险等级： 中

实施指导

- 了解运营的预期活动模式：建立运营活动模式以便确定行为何时不符合预期值，从而根据需要做出适当响应。

OPS09-BP06 在运营成果面临风险时发出提醒

任何时候，只要运营成果存在风险，就必须引发警报并采取操作。运营成果是为生产工作负载提供支持的任意活动。其范围极广，从开发应用程序新版本到从中断中恢复，无所不包。需要像重视业务成果一样重视运营成果。

软件团队应确定关键运营指标和活动，并为其设定警报。警报必须及时并且内容可付诸行动。引发警报时，必须附带对相应运行手册或行动手册的引用。没有相应操作的警报会导致用户疲于应对警报。

期望的结果： 运营活动存在风险时，发送警报来督促采取行动。警报应包含引发警报的背景信息，并指向行动手册（提供调查方法）或运行手册（提供防范方法）。在可能时，运行手册应自动运行并发送通知。

常见反模式：

- 您在调查一起事件并建立了支持案例。支持案例指明违反了服务等级协议（SLA，Service Level Agreement），但没有引发警报。
- 原本计划在午夜进行生产环境部署，但由于最后时刻进行代码更改而延迟。没有引发警报，部署挂起。
- 出现生产中断，但没有发送警报。
- 您的部署时间始终落后于预计时间。没有采取任何调查操作。

建立此最佳实践的好处：

- 在运营成果存在风险时引发警报有助于防患于未然，提升支持工作负载的能力。
- 由于实现了积极的运营成果，业务成果得到改善。
- 对运营问题的检测和修复能力得到改进。
- 整体的运营健康状况得以提升。

未建立这种最佳实践的情况下暴露的风险等级： 中

实施指导

您必须先定义运营成果，然后才能在运营成果上设置警报。这个过程首先要定义哪些运营活动对您的组织来说最重要。是需要在两个小时内部署到生产环境，还是在设定的时间内响应支持案例？ 您的组织必须定义关键运营活动以及衡量方式，这样才能对其进行监控、改进和设定警报。您需要一个集中位置来存储和分析工作负载及运营遥测数据。应该能够使用同一套机制，在运营成果存在风险时引发警报。

客户示例

在 AnyCompany Retail 的例行部署期间触发了 CloudWatch 警报。已经超过了部署的准备时间。Amazon EventBridge 在 AWS Systems Manager OpsCenter 中创建了 OpsItem。云运营团队使用行动手册调查问题，确定架构更改用时超过了预期时间。他们向待命开发人员发出警报并继续监控部署。部署完成后，云运营团队解决了 OpsItem。该团队将在事后检查期间分析事件。

实施步骤

1. 如果您尚未确定运营 KPI、指标和活动，请针对这一问题实施前述最佳实践（OPS09-BP01 到 OPS09-BP05）。
   - AWS Support 客户如果具有 企业支持 ，就可以向其技术客户经理请求举行 运营 KPI 研讨会 。这一协作式研讨会免费提供，可以帮助您根据业务目标定义运营 KPI 和指标。请联系您的技术客户经理了解详情。
2. 在您建立运营活动、KPI 和指标之后，可以在监控平台上配置警报。警报应该有关联的操作，例如行动手册或运行手册。应该避免没有操作的警报。
3. 在经过一段时间之后，您应该评估运营指标、KPI 以及活动来确定改进领域。作为对警报的响应，在运行手册和行动手册中收集操作员的反馈，确定改进领域。
4. 警报应包括用于将它们标记为误报的机制。此机制应该引发对指标阈值的审查。

实施计划的工作量级别： 中。在实施此最佳实践之前，必须落实多个最佳实践。在确定运营活动并建立运营 KPI 之后，应该建立警报。

资源

相关最佳实践：

- OPS02-BP03 确定对运营活动绩效负责的所有者：每个运营活动和成果都应该确定负责人。此人在成果存在风险时应收到警报。
- OPS03-BP02 赋能团队成员在结果有风险时采取行动：在引发警报时，您的团队应该有人采取行动来修复问题。
- OPS09-BP01 识别关键性能指标：在运营成果上发出警报的第一步是确定运营 KPI。
- OPS09-BP02 定义运营指标：在开始生成警报之前建立此最佳实践。
- OPS09-BP03 收集和分析运营指标：建立警报需要集中收集运营指标。
- OPS09-BP04 建立运营指标基准：运营指标基准提供了调节警报和避免用户疲于应对警报的能力。
- OPS09-BP05 了解运营的预期活动模式：您可以通过了解运营事件的活动模式来提高警报的准确性。
- OPS09-BP08 验证实现的成果以及 KPI 和指标的有效性：评估所取得的运营成果以确保 KPI 和指标有效。
- OPS10-BP02 针对每个提醒设置一个流程：每个警报应该具有关联的运行手册或行动手册，并向接收警报的人员提供背景信息。
- OPS11-BP02 在意外事件发生后执行分析：在警报之后开展事后分析，确定改进领域。

相关文档：

- AWS 部署管道参考架构：应用程序管道架构
- GitLab：敏捷性/DevOps 指标入门

相关视频：

- 使用 AWS Systems Manager OpsCenter 聚合和解决运营问题
- 将 AWS Systems Manager OpsCenter 与 Amazon CloudWatch 警报集成
- 使用 Amazon EventBridge 将数据来源与 AWS Systems Manager OpsCenter 集成

相关示例：

- 使用 Amazon EC2 Systems Manager Automation 和 AWS Health 为 Amazon EC2 通知和其他情况自动执行修正操作
- AWS 管理和监管工具研讨会 – Operations 2022
- 在 AWS 上使用 DevOps 监控控制面板提取、分析和可视化指标

相关服务：

- Amazon EventBridge
- AWS Support 主动服务 – 运营 KPI 研讨会
- AWS Systems Manager OpsCenter
- CloudWatch 事件

OPS09-BP07 在检测到运营异常时发出提醒

在检测到运营异常时发出提醒，从而在必要时做出适当响应。

您对一段时间内运营指标的分析可能会建立行为模式，您可以对这些模式进行充分量化，以定义事件或发出警报作为响应。

经过训练后， CloudWatch 异常检测 功能可用于 对 检测到的异常发出警报，或将期望值叠加到指标数据 图表 上，以进行持续的比较。

Amazon DevOps Guru 可通过事件关联、日志分析并应用机器学习来分析工作负载遥测数据，用于确定异常行为。所获得的 见解 与相关数据和推荐一起呈现。

常见反模式：

- 您在对实例队列应用补丁。您在测试环境中成功地对补丁进行了测试。在您队列中很大比例的实例中，补丁应用都以失败告终。您没有执行任何操作。
- 您注意到，有的部署是从星期五结束时开始的。您的组织将预定义的维护时段安排在星期二和星期四。您没有执行任何操作。

建立此最佳实践的好处： 通过了解运营行为的模式，您可以识别意外行为并在必要时采取措施。

未建立此最佳实践暴露的风险等级： 低

实施指导

- 在检测到运营异常时发出提醒：在检测到运营异常时发出提醒，从而根据需要做出适当响应。

- 什么是 Amazon CloudWatch Events？
- 创建 Amazon CloudWatch 警报
- 使用 Amazon SNS 通知调用 Lambda 函数

资源

相关文档：

- Amazon DevOps Guru
- CloudWatch 异常检测
- 创建 Amazon CloudWatch 警报
- 使用 Amazon CloudWatch Events 检测管道状态的更改并做出反应
- 使用 Amazon SNS 通知调用 Lambda 函数
- 什么是 Amazon CloudWatch Events？

OPS09-BP08 验证实现的成果以及 KPI 和指标的有效性

在业务层面查看运营活动，以便帮助您确定自己是否满足需求，并确定需要改进哪些方面才能实现业务目标。验证 KPI 和指标的有效性并在需要时进行修改。

AWS 还通过 AWS 服务 API 和 SDK（例如，Grafana、Kibana 和 Logstash）支持第三方日志分析系统和商业智能工具。

常见反模式：

- 随着开发团队数量的增加，部署的频率也随之增加。您定义的预期部署频率是每周一次。而您现在已定期每日部署。如果您的部署系统出现问题，无法进行部署，那么几天之内都不会被发现。
- 之前，您的业务仅在星期一至星期五的核心业务时间提供支持。您针对事件建立了下一工作日响应时间目标。您最近开始提供 24x7 全天候支持，响应时间目标为 2 小时。您的夜班员工不堪重负，客户也不满意。没有迹象表明事件响应时间有问题，因为您在针对下一工作日目标进行报告。

建立此最佳实践的好处： 通过审核和修订 KPI 及指标，您可以了解工作负载如何支持业务成果的实现，并可以确定需要对哪些方面进行改进以实现业务目标。

未建立此最佳实践暴露的风险等级： 低

实施指导

- 验证实现的成果以及 KPI 和指标的有效性：在业务层面查看运营活动，以便帮助您确定自己是否满足需求，并确定需要改进哪些方面才能实现业务目标。验证 KPI 和指标的有效性并在需要时进行修改。
  - 使用 Amazon CloudWatch 控制面板
  - 什么是日志分析？

资源

相关文档：

- 使用 Amazon CloudWatch 控制面板
- 什么是日志分析？

OPS 10 您如何应对工作负载事件和运营事件？

制定和验证用于响应事件的程序，以便尽可能减少其对工作负载的干扰。

最佳实践

- OPS10-BP01 使用流程来管理事件、意外事件和问题
- OPS10-BP02 针对每个提醒设置一个流程
- OPS10-BP03 根据业务影响确定运营事件的优先顺序
- OPS10-BP04 定义上报路径
- OPS10-BP05 启用推送通知
- OPS10-BP06 通过控制面板展现状况信息
- OPS10-BP07 自动响应事件

OPS10-BP01 使用流程来管理事件、意外事件和问题

贵组织拥有处理事件、意外事件和问题的流程。事件 是在工作负载中发生但可能不需要干预的事情。意外事件 是需要干预的事件。 问题 是需要干预或无法解决的反复发生的事件。您需要一些流程来减轻这些事件对业务的影响，并确保做出适当的响应。

当您的工作负载发生意外事件和问题时，您需要一些流程来处理它们。您将如何与利益相关者沟通事件的状态？ 谁负责监督领导应对工作？ 您用什么工具来减轻事件的影响？ 这些是您建立可靠的响应流程所需回答的一些问题的例子。

这些流程必须记录在一个中央位置，并可供参与您工作负载的任何人使用。如果您没有中央 Wiki 或文档存储区，可以使用版本控制存储库。随着流程的发展，您将不断更新这些计划。

接下来将需要对问题进行自动化。这些事情占用了您的时间，限制了您的创新能力。首先构建一个可重复的流程来缓解问题。随着时间的推移，将重点放在自动化缓解或修复根本问题上。这样就可以腾出时间来改进您的工作负载。

期望结果： 贵组织拥有处理事件、意外事件和问题的流程。这些流程被记录下来并存储在一个中央位置。它们随着流程的更改而更新。

常见反模式：

- 周末发生了一起意外事件，值班工程师不知道该怎么办。
- 一位客户向您发送一封电子邮件，说应用程序关闭了。您重新启动服务器以修复该问题。这种情况经常发生。
- 有一起意外事件，多个团队独立工作，试图解决该问题。
- 部署发生在您的工作负载中，而不会被记录下来。

建立此最佳实践的好处：

- 您有一条关于工作负载中事件的审计跟踪。
- 从意外事件中恢复的时间缩短了。
- 团队成员能够一致地解决意外事件和问题。
- 调查意外事件时，大家更加团结一致。

未建立这种最佳实践的情况下暴露的风险等级： 高

实施指导

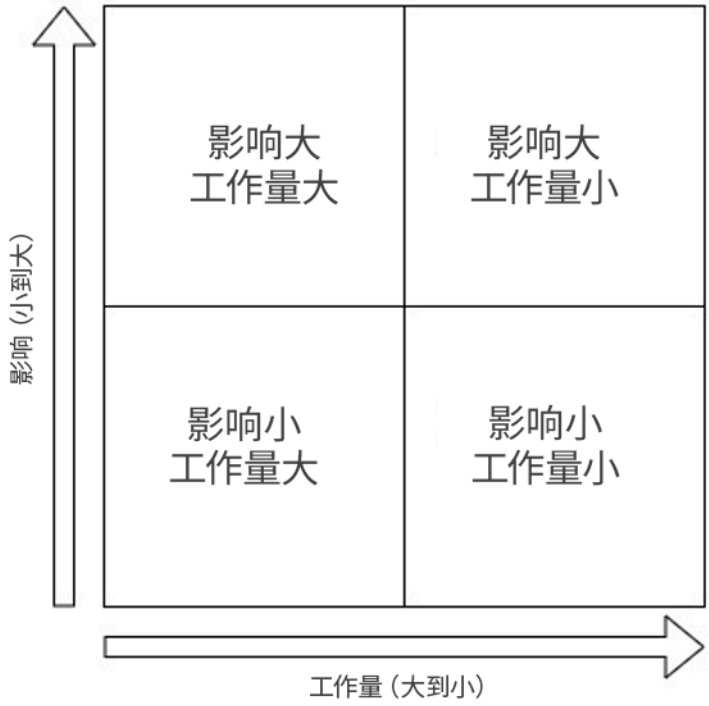
实施这种最佳实践意味着您正在跟踪工作负载事件。您建立了处理意外事件和问题的流程。这些流程被记录下来、共享并经常更新。发现问题，确定优先级，并加以解决。

客户示例

AnyCompany Retail 的内部 Wiki 中有一部分专门用于事件、意外事件和问题管理的流程。所有事件均发送至 Amazon EventBridge。问题在 AWS Systems Manager OpsCenter 中被识别为 OpsItems，并按优先级进行修复，减少了无差别的劳动。当流程发生变化时，它们会在内部 Wiki 中进行更新。他们使用 AWS Systems Manager Incident Manager 来管理意外事件并协调缓解工作。

实施步骤

1. 事件
    - 跟踪工作负载中发生的事件，即使不需要人工干预。
    - 与工作负载利益相关者合作，制定一份应跟踪的事件清单。一些示例包括已完成的部署或成功的修补。
    - 您可以使用 Amazon EventBridge 或 Amazon Simple Notification Service 之类的服务生成自定义事件以进行跟踪。
2. 意外事件
    - 首先要确定意外事件的沟通计划。必须告知哪些利益相关者？ 您将如何让他们了解情况？ 谁负责监督协调工作？ 我们建议建立一个内部聊天渠道进行沟通和协调。
    - 为支持您工作负载的团队定义上报路径，特别是在团队没有随时待命的轮换情况下。根据您的支持级别，您还可以向 AWS Support 提交工单。
    - 创建一个调查该意外事件的行动手册。这应该包括沟通计划和详细的调查步骤。在您的调查中包括检查 AWS Health Dashboard 。
    - 记录意外事件响应计划。沟通意外事件管理计划，以便内部和外部客户了解参与规则以及对他们的期望。就使用方法对您的团队成员进行培训。
    - 客户可以使用 Incident Manager 来建立和管理他们的意外事件响应计划。
    - 企业支持客户可以向他们的技术客户经理请求参加 意外事件管理研讨会 。这场有指导意义的研讨会可测试您现有的意外事件响应计划，并帮助您找出需要改进之处。
3. 问题
    - 必须在您的 ITSM 系统中识别和跟踪问题。
    - 确定所有已知问题，并根据修复工作量和对工作负载的影响来确定它们的优先级。



- 先解决影响大、工作量小的问题。一旦这些问题得到解决，就继续处理那些属于“影响小且工作量小”象限的问题。
- 随着您的工作负载增长和扩展，您可以使用 Systems Manager OpsCenter 来识别这些问题，为它们附上运行手册，并跟踪它们。

实施计划的工作量级别： 中。您需要一个流程和工具来实施这种最佳实践。记录您的流程，让与工作负载相关的任何人都可以使用它们。经常更新它们。您建立了一个管理问题、缓解问题或解决问题的流程。

资源

相关最佳实践：

- OPS07-BP03 使用运行手册执行程序：已知问题需要一个相关的运行手册，以使缓解工作保持一致。
- OPS07-BP04 根据行动手册调查问题：必须使用行动手册对意外事件进行调查。
- OPS11-BP02 在意外事件发生后执行分析：从意外事件中恢复之后，务必要进行事后分析。

相关文档：

- Atlassian - DevOps 时代的意外事件管理

- AWS 安全意外事件响应指南
- DevOps 和 SRE 时代的意外事件管理
- PagerDuty - 什么是意外事件管理？

相关视频：

- AWS re:Invent 2020：分布式组织中的意外事件管理
- AWS re:Invent 2021 - 使用事件驱动型架构构建下一代应用程序
- AWS 支持您 | 探讨事件管理桌面练习
- AWS Systems Manager Incident Manager - AWS 虚拟研讨会
- AWS 后续举措主讲 Incident Manager | AWS 事件

相关示例：

- AWS 管理和监管工具研讨会 - OpsCenter
- AWS 主动式服务 – 意外事件管理研讨会
- 使用 Amazon EventBridge 构建事件驱动型应用程序
- 在 AWS 上构建事件驱动型架构

相关服务：

- Amazon EventBridge
- Amazon SNS
- AWS Health Dashboard
- AWS Systems Manager Incident Manager
- AWS Systems Manager OpsCenter

OPS10-BP02 针对每个提醒设置一个流程

针对引发提醒的任何事件制定明确的响应措施（运维手册或管理手册），并明确指定负责人。这样可以确保您及时有效地响应运营事件，并防止可以针对其采取措施的事件被不重要的通知所掩盖。

常见反模式：

- 监控系统会向您显示已批准的连接流和其他消息。由于消息量过大，导致您错过了需要干预的周期性错误消息。
- 您收到提醒，指示网站停机。发生这种情况时，没有明确的流程。您被迫采用临时方法来诊断和解决问题。边处理边开发流程会延长恢复时间。

建立此最佳实践的好处： 仅在需要采取措施时发出提醒可以防止低价值提醒遮掩高价值提醒。制定一个可随时采取措施的提醒流程，可以对环境中的事件做出一致而迅速的响应。

未建立此最佳实践暴露的风险等级： 高

实施指导

- 提醒响应流程：对于引发提醒的任何事件，都要制定明确的响应措施（运行手册或管理手册），并明确指定负责其成功完成的负责人（例如个人、团队或角色）。响应的执行可能是自动的，也可能由其他团队完成，但是负责人应负责确保响应流程获得预期的成果。设置这些流程可以确保您及时有效地响应运营事件，并防止可以针对其采取措施的事件被不重要的通知所掩盖。例如，可以实施自动扩展来扩展 Web 前端，但是运营团队应负责确保自动扩展规则和限制符合工作负载需求。

资源

相关文档：

- Amazon CloudWatch 功能
- 什么是 Amazon CloudWatch Events？

相关视频：

- 制定监控计划

OPS10-BP03 根据业务影响确定运营事件的优先顺序

确保在多个事件需要干预时，优先处理对业务最为重要的事件。人身伤亡、经济损失、名誉或信任损害都是一种影响。

常见反模式：

- 您收到一个支持请求，需为用户添加打印机配置。在处理该问题时，您收到一个支持请求，称您的零售网站停机。为您的用户完成打印机配置后，您着手处理网站问题。

- 您收到通知，指示您的零售网站和薪资系统都发生停机。您不知道应该优先处理哪个问题。

建立此最佳实践的好处： 优先响应对业务影响最大的意外事件，可以帮助您管理这种影响。

未建立此最佳实践暴露的风险等级： 中

实施指导

- 根据业务影响确定运营事件的优先顺序：确保在多个事件需要干预时，优先处理对业务最为重要的事件。影响可能包括人身伤亡、经济损失、违规、名誉或信任损害。

OPS10-BP04 定义上报路径

在运维手册和管理手册中定义上报路径，包括触发上报的事件和上报程序。明确指定每项措施的负责人，以便确保有效而及时地响应运营事件。

在采取措施之前，确定何时需要人为决定。与决策者合作，提前做出决策，这样 MTTR 便不会因为等待响应而延长。

常见反模式：

- 您的零售网站停机。您不了解用于网站恢复的运行手册。您开始打电话求助同事。
- 您收到一个关于应用程序无法访问的支持案例。您没有系统管理权限。您不知道谁具有权限。您尝试与创建案例的系统负责人联系，但没有得到响应。您无法联系到系统负责人，而您的同事对此也不太熟悉。

建立此最佳实践的好处： 通过定义上报、上报触发器和上报程序，您可以适当的影响速率系统地向意外事件添加资源。

未建立这种最佳实践的情况下暴露的风险等级： 中

实施指导

- 定义上报路径：在运维手册和管理手册中定义上报路径，包括触发升级的事件和升级程序。例如，当运维手册无法解决问题或者预定义的时间已经过去时，将问题从支持工程师升级给高级支持工程师。当管理手册无法确定修复路径或者预定义的时间已经过去时，将问题从高级工程师升级给开发团队也是一种正确的升级路径。明确指定每项措施的负责人，以便确保有效而及时地响应运营事件。升级可以涉及第三方。例如某个网络连接提供商或软件供应商。升级可以涉及负责受影响的系统并且获得授权的决策者。

OPS10-BP05 启用推送通知

在用户使用的服务受到影响以及这些服务的运行状况再次恢复正常时，直接与用户联系（例如通过电子邮件或 SMS），确保用户采取适当的措施。

常见反模式：

- 您的应用程序遭到了分布式拒绝服务意外事件，并且已经几天没有响应。没有错误消息。您尚未发送通知电子邮件。您尚未发送文本通知。您尚未在社交媒体上共享信息。您的客户很沮丧，正在寻找其他可以为他们提供支持的供应商。
- 星期一，您的应用程序在安装补丁后出现问题，停机数小时。星期二，您的应用程序在部署代码后出现问题，而且在数小时内都处于不可靠状态。星期三，为了规避与出现故障的补丁相关的安全漏洞，您的应用程序部署了代码，但之后却出现问题，在部署代码后几个小时内，应用程序都不可用。星期四，您的客户很沮丧，开始寻找其他可以为他们提供支持的供应商。
- 本周末，您的应用程序会因维护而停机。您没有告知客户。您的部分客户已预先安排了需使用您的应用程序的活动。发现您的应用程序不可用后，他们感到非常沮丧。

建立此最佳实践的好处： 通过定义通知、通知触发器和通知程序，您可以通知客户并在工作负载问题影响客户时做出响应。

未建立这种最佳实践的情况下暴露的风险等级： 中

实施指导

- 启用推送通知：在用户使用的服务受到影响以及这些服务的运行状况恢复正常时，直接与用户联系（例如通过电子邮件或 SMS），以便用户得以采取相应的措施。
  - Amazon SES 功能
  - 什么是 Amazon SES？
  - 设置 Amazon SNS 通知

资源

相关文档：

- Amazon SES 功能
- 设置 Amazon SNS 通知
- 什么是 Amazon SES？

OPS10-BP06 通过控制面板展现状况信息

提供为目标受众（例如内部技术团队、领导和客户）专门设计的控制面板，以传达业务当前的运营状况并提供值得关注的指标。

您可以使用 Amazon CloudWatch 控制面板， 在 CloudWatch 控制台中可自定义的主页上创建控制面板。借助像 Amazon QuickSight 这样的商业智能服务，您可以创建和发布工作负载和运营状况（例如，订单达成率、连接的用户和交易时间）的交互式控制面板。您可以创建控制面板，用来显示指标的系统级和业务级视图。

常见反模式：

- 您根据请求运行有关管理应用程序当前使用情况的报告。
- 在意外事件发生期间，相关系统负责人每二十分钟与您联系一次，想要知道问题是否已解决。

建立此最佳实践的好处： 创建控制面板后，您可以让客户自助访问信息，这样可让他们获知相关信息，并确定是否需要采取措施。

未建立此最佳实践暴露的风险等级： 中

实施指导

- 通过控制面板展现状况信息：提供为目标受众（例如内部技术团队、领导和客户）专门设计的控制面板，以传达业务当前的运营状况并提供值得关注的指标。提供用于获取状态信息的自助选项，可以减少负责处理状态请求的运营团队的中断。示例包括 Amazon CloudWatch 控制面板和 AWS Health Dashboard。
  - CloudWatch 控制面板创建并使用自定义指标视图

资源

相关文档：

- Amazon QuickSight
- CloudWatch 控制面板创建并使用自定义指标视图

OPS10-BP07 自动响应事件

自动响应事件以便减少由手动流程引起的错误，并确保响应及时并且一致。

有多种方法可以在 AWS 上自动执行运行手册和行动手册操作。要响应 AWS 资源中的状态更改事件或您自己的自定义事件，您应创建 CloudWatch Events 规则 以通过 CloudWatch 目标（例如，Lambda 函数、Amazon Simple Notification Service（Amazon SNS）主题、Amazon ECS 任务和 AWS Systems Manager Automation）触发响应。

要响应超过资源阈值的指标（例如，等待时间），您应创建 CloudWatch 警报 以使用 Amazon EC2 操作或 Auto Scaling 操作执行一个或多个操作，或者向 Amazon SNS 主题发送通知。如果您需要执行自定义操作以响应警报，请通过 Amazon SNS 通知调用 Lambda。使用 Amazon SNS 发布事件通知和升级消息，以便让人们了解情况。

AWS 还通过 AWS 服务 API 和 SDK 支持第三方系统。AWS 合作伙伴和第三方提供了许多用于监控、通知和响应的监控工具。其中一些工具包括 New Relic、Splunk、Loggly、SumoLogic 和 Datadog。

您应该保留关键的手动程序，以备在自动程序出故障时使用。

常见反模式：

- 开发人员检查其代码。发生此事件后，本可开始构建然后执行测试，但您没执行任何操作。
- 在停止运行前，您的应用程序记录了一个特定的错误。重新启动应用程序的流程易于理解，可编写成脚本。您可以使用日志事件来调用脚本并重新启动应用程序。否则的话，如果错误发生在星期天凌晨 3 点，您作为负责修复系统的随叫随到的资源，将不得不起床去处理。

建立此最佳实践的好处： 通过自动响应事件，您可以缩短响应时间并减少人工活动中发生的错误。

未建立此最佳实践暴露的风险等级： 低

实施指导

- 自动响应事件：自动响应事件以便减少由手动流程引起的错误，并确保响应及时并且一致。
  - 什么是 Amazon CloudWatch Events？
  - 创建在发生事件时触发的 CloudWatch Events 规则
  - 创建在 AWS API 调用上使用 AWS CloudTrail 触发的 CloudWatch Events 规则
  - 来自支持的服务的 CloudWatch Events 事件示例

资源

相关文档：

- Amazon CloudWatch 功能

- 来自支持的服务的 CloudWatch Events 事件示例
- 创建在 AWS API 调用上使用 AWS CloudTrail 触发的 CloudWatch Events 规则
- 创建在发生事件时触发的 CloudWatch Events 规则
- 什么是 Amazon CloudWatch Events？

相关视频：

- 制定监控计划

相关示例：

演进

问题

- OPS 11 如何改进运营？

OPS 11 如何改进运营？

分配专用的时间和资源用于持续增量改进，以便提高运营的有效性和效率。

最佳实践

- OPS11-BP01 设置持续改进流程
- OPS11-BP02 在意外事件发生后执行分析
- OPS11-BP03 实施反馈环路
- OPS11-BP04 执行知识管理
- OPS11-BP05 确定推动改进的因素
- OPS11-BP06 验证分析结果
- OPS11-BP07 审核运营指标
- OPS11-BP08 记录和分享经验教训
- OPS11-BP09 分配时间进行改进

OPS11-BP01 设置持续改进流程

定期评估各种改进机会并确定其优先顺序，以便集中精力处理可以实现最大收益的工作。

常见反模式：

- 您记录了创建开发或测试环境所需的程序。您可以使用 CloudFormation 自动执行该流程，也可以从控制台手动执行。
- 您的测试显示，绝大部分的 CPU 资源都用于应用程序内一小部分效率低下的功能。您可以将重点放在改进它们并降低成本上，但是您需要创建新的可用性特性。

建立此最佳实践的好处： 持续改进机制支持定期评估各种改进机会、确定其优先顺序，以及集中精力处理可以实现最大收益的工作。

未建立此最佳实践暴露的风险等级： 高

实施指导

- 定义持续改进流程：定期评估各种改进机会并确定其优先顺序，以便将精力集中在可以实现最大收益的工作上。实施更改以便改进，并评估成果以便确定是否成功。如果成果不符合目标并且仍然需要改进，则寻求其他行动方案。运营流程中应该分配专用的时间和资源，以便实现持续增量改进。

OPS11-BP02 在意外事件发生后执行分析

审核影响客户的事件，确定导致这些事件的因素和预防措施。利用这些信息来制定缓解措施，以限制或防止再次发生同类事件。制定程序以迅速有效地做出响应。根据目标受众，适当传达事件成因和纠正措施。

常见反模式：

- 您管理应用程序服务器。大约每 23 小时 55 分钟，所有活动会话都会终止。您已尝试找出应用程序服务器上出现的问题。您怀疑可能是网络问题，但由于网络团队工作繁忙无法为您提供支持，因此无法与他们合作。由于缺乏可遵循的预定义流程，因此难以获取支持并收集必要的信息来确定发生了什么情况。
- 您的工作负载中出现了数据丢失的情况。这是第一次发生，原因不明。您认为它不重要，因为可以重新创建数据。数据丢失对客户的影响开始变得愈发频繁。还原丢失的数据时，这也会增加您的操作负担。

建立此最佳实践的好处： 设置预定义的流程，以确定导致意外事件发生的要素、条件、操作和事件，从而帮助您找到改进机会。

未建立这种最佳实践的情况下暴露的风险等级： 高

实施指导

- 通过流程来确定事件成因：审查所有影响客户的意外事件。设置流程来确定和记录导致意外事件的因素，以便制定缓解措施来限制或防止事件再次发生，并且您还可以据此制定及时有效的应对措施。在适当的情况下向目标受众说明根本原因。

OPS11-BP03 实施反馈环路

反馈环路提供了可操作的见解，进而推动决策的制定。将反馈环路融入过程和工作负载中。这可帮助您确定问题和需要改进的领域。它们还可以验证在改进方面所做的投入。这些反馈环路为持续改进工作负载奠定了基础。

反馈环路分为两大类： 即时反馈 和 回顾性分析。通过审查运营活动的绩效和成果来收集即时反馈。此反馈来自团队成员、客户或活动的自动化输出。通过 A/B 测试和发布新功能等方式接收即时反馈，这对于快速失效机制至关重要。

定期执行回顾性分析，可以获得在运营成果审核和指标审核过程中产生的反馈。这些回顾在冲刺结束时进行、有节奏地进行或者在重大发布或事件之后进行。这种类型的反馈环路验证了在运营或工作负载方面的投入。它有助于衡量成功并验证您的策略。

期望的结果： 您可以使用即时反馈和回顾性分析来加快改进。有一种机制可用于捕获用户和团队成员的反馈。回顾性分析用于确定可推动改进的趋势。

常见反模式：

- 您推出了一项新功能，但无法接收客户对此新功能的反馈。
- 在投资进行运营改进后，您无需回顾来验证它们。
- 您可以收集客户反馈，但不用定期进行审查。
- 反馈环路会产生建议的操作项，但它们不包括在软件开发过程中。
- 对于所提出的改进事项，客户不会收到关于它们的反馈意见。

建立此最佳实践的好处：

- 您可以反过来从客户出发，以便推动新的功能。
- 您的组织文化能够更快地对变更做出回应。
- 趋势用于确定改进机会。
- 回顾将验证对工作负载和运营所做的投入。

未建立这种最佳实践的情况下暴露的风险等级： 高

实施指导

实施此最佳实践意味着同时使用即时反馈和回顾性分析。这些反馈环路将推动改进。有许多适用于即时反馈的机制，包括调查、客户投票或反馈表。您的组织还使用回顾来确定改进机会并验证计划。

客户示例

AnyCompany Retail 创建了一个 Web 表单，客户可使用此表单提供反馈或报告问题。在每周 Scrum 期间，软件开发团队将评估用户反馈。反馈定期用于引导相应平台的发展。他们在每个冲刺结束时进行回顾，确定需要改进的项目。

实施步骤

1. 即时反馈
   - 您需要一种机制来接收由客户和团队成员提供的反馈，也可以将您的运营活动配置为交付自动反馈。
   - 您的组织需要一个流程，来审查此反馈、确定要改进的方面并安排改进。
   - 必须将反馈纳入您的软件开发过程中。
   - 在实施改进时，请对反馈提交者进行跟进。
     - 您可以使用 AWS Systems Manager OpsCenter 将这些改进创建为 OpsItem 并进行跟踪。
2. 回顾性分析
   - 在开发周期结束时、按设定的节奏或在主要发布后进行回顾。
   - 召开回顾性会议，让工作负载中涉及的利益相关者参加。
   - 在白板或电子表格上创建三个列：“停止”、“开始”和“继续”。
     - 停止 针对的是您希望团队停止执行的任何工作。
     - 开始 针对的是要开始付诸行动的想法。
     - 继续 针对的是要继续执行的项目。
   - 在会议室里四处走动，从利益相关者那里收集反馈。
   - 确定反馈的优先级。将操作和利益相关者分配给任何“开始”或“继续”项目。
   - 将操作纳入软件开发过程中，并在实施改进时将状态更新传达给利益相关者。

实施计划的工作量级别： 中。要实施此最佳实践，您需要一种方法来获取并分析即时反馈。此外，您需要建立一个回顾性分析过程。

资源

相关最佳实践：

- OPS01-BP01 评估外部客户需求：反馈环路是一种用于收集外部客户需求的机制。
- OPS01-BP02 评估内部客户需求：内部利益相关者可以使用反馈环路来传达需求和要求。
- OPS11-BP02 在意外事件发生后执行分析：事件后分析是发生事件后进行回顾性分析的重要形式。
- OPS11-BP07 审核运营指标：运营指标审查可确定趋势和需要改进的方面。

相关文档：

- 构建 CCOE 时应避免的 7 个陷阱
- Atlassian 团队行动手册 – 回顾
- 电子邮件定义：反馈环路
- 建立基于 AWS Well-Architected Framework 审查的反馈环路
- IBM Garage 方法 – 保持回顾
- Investopedia – PDCS 循环
- Tim Cochran 所著的《最大限度地提高开发人员效率》
- 运营准备情况审查（ORR）白皮书 – 迭代
- TIL CSI – 持续服务改进
- 当丰田转向电子商务：Amazon 的精益方法

相关视频：

- 构建有效的客户反馈环路

相关示例：

- Astuto – 客户反馈开源工具
- AWS 解决方案 – AWS 上的 QnABot
- Fider – 客户反馈整理平台

相关服务：

- AWS Systems Manager OpsCenter

OPS11-BP04 执行知识管理

执行机制，以方便您的团队成员及时发现和访问他们正在寻找的信息，并确定信息是最新且完整的。制定适当的机制，以确定所需的内容、需要更新的内容以及应存档的内容（以便不再引用它们）。

常见反模式：

- 一位沮丧的客户创建了一个针对新产品功能请求的支持案例，希望以此解决遇到的问题。它被添加到优先改进列表中。

未建立此最佳实践暴露的风险等级： 高

实施指导

- 知识管理：确保采取了机制，以方便您的团队成员及时发现和访问他们正在寻找的信息，并确定信息是最新且完整的。维护机制，以确定所需的内容、需要更新的内容以及应存档的内容（以便不再引用它们）。

OPS11-BP05 确定推动改进的因素

确定推动改进的因素，以便评估各种机会并确定其优先顺序。

在 AWS 上，您可以聚合所有运营活动、工作负载和基础设施的日志，以创建详细的活动历史记录。然后，您可以根据推动因素，使用 AWS 工具分析您在一段时间内的运营状况和工作负载运行状况（例如，确定趋势、将事件和活动与结果相关联，并在环境之间/跨系统进行比较和对比），以发现改进机会。

您应该使用 CloudTrail 跟踪 API 活动（通过 AWS Management Console、CLI、开发工具包和 API），以了解您账户中发生的情况。您可以使用 CloudTrail 和 CloudWatch 跟踪您的 AWS 开发人员工具部署活动。这将向您的 CloudWatch Logs 日志数据添加部署的详细活动历史记录及其结果。

将您的日志数据导出到 Amazon S3 以便长期存储。使用 AWS Glue，您可以在 Amazon S3 中发现并准备您的日志数据以供分析。使用 Amazon Athena，借助其与 AWS Glue 的原生集成来分析您的日志数据。使用像 Amazon QuickSight 这样的商业智能工具，您可以直观显示、浏览和分析您的数据

常见反模式：

- 您有一个脚本，可以正常运行但不完美。您投入时间重新编写。现在，它完美无缺。

- 您的初创公司正试图从风险投资人那里获得另一笔资金。他们希望您证明符合 PCI DSS。您希望让他们满意，因此您记录合规性，但却错过了客户的交付日期，导致客户流失。您没有做错，但是现在您怀疑这样做是否合适。

建立此最佳实践的好处： 确定希望用于改进的标准后，您可以最大程度地减小基于事件的动机或情感投入所带来的影响。

未建立这种最佳实践的情况下暴露的风险等级： 中

实施指导

- 了解推动改进的因素：您只应该在能够实现所需成果的情况下更改某个系统。
  - 需要的功能：在评估改进机会时评估需要的特性和功能。
    - AWS 的新增功能
  - 无法接受的问题：在评估改进机会时评估无法接受的问题、错误和漏洞。
    - AWS 最新安全公告
    - AWS Trusted Advisor
  - 合规性要求：在分析改进机会时，评估保持监管和政策合规性或获取第三方支持所需的更新和更改。
    - AWS 合规性
    - AWS 合规性计划
    - AWS 合规性最新新闻

资源

相关文档：

- Amazon Athena
- Amazon QuickSight
- AWS 合规性
- AWS 合规性最新新闻
- AWS 合规性计划
- AWS Glue
- AWS 最新安全公告
- AWS Trusted Advisor

- 将您的日志数据导出到 Amazon S3
- AWS 的新增功能

OPS11-BP06 验证分析结果

与跨职能团队和业务负责人共同查看分析结果和响应措施。通过这些工作来建立共识、发现其他影响并确定行动方案。适当调整响应措施。

常见反模式：

- 您看到某个系统上的 CPU 利用率达到 95%，因此优先考虑寻找一种方法来减少系统上的负载。您认为最佳行动方案是进行扩展。系统是一个转码器，您可对它进行扩展，让它始终以 95% 的 CPU 利用率运行。如果您先与系统负责人联系，他们会向您做出解释。您浪费了时间。
- 系统负责人坚持认为他们的系统是关键任务型系统。系统未置于高安全性环境中。为了提高安全性，您需要对关键任务型系统实施额外的检测和预防控制措施。您通知系统负责人工作已完成，并向他收取额外资源的费用。在收到此通知后的沟通过程中，系统负责人了解到对于关键任务型系统有一个正式定义，而他的系统并不满足。

建立此最佳实践的好处： 通过与业务负责人和主题专家一起验证分析结果，您可以建立共识并更有效地指导改进。

未建立这种最佳实践的情况下暴露的风险等级： 中

实施指导

- 验证分析结果：与业务负责人和主题专家沟通，以确保对您收集的数据的价值达成共识和一致。确定其他问题、潜在影响并制定行动方案。

OPS11-BP07 审核运营指标

定期与来自不同业务领域的跨团队参与者对运营指标进行回顾性分析。通过这些分析来确定改进机会和可能的行动方案，并分享经验教训。

寻找在所有环境（例如，开发、测试和生产环境）中改进的机会。

常见反模式：

- 维护时段导致一次重要的零售促销中断。如果存在其他影响业务的事件，可以延迟标准维护时段，而业务部门对此并不知晓。

- 由于使用了组织中常用的错误库，导致了长时间的停机。自此之后，您已经迁移到可靠的库。您组织中的其他团队尚未意识到风险的存在。如果您定期开会并审核此意外事件，他们应该注意到这种风险。
- 转码器的性能一直在不断下降，这对媒体团队产生了影响。但这还不算多严重。真正糟糕的是，除非情况严重到足以引发意外事件，否则您将难以发现。如果您与媒体团队一起审核运营指标，就有机会发现指标的变化，同时认识到他们的经验并利用这些经验将问题解决。
- 您没有审核对客户 SLA 的满足程度。您目前正趋向于无法满足客户 SLA。如果无法满足客户 SLA，将会受到经济处罚。如果您定期开会审核这些 SLA 的指标，您将有机会发现并解决这一问题。

建立此最佳实践的好处： 您可以通过会议定期审核运营指标、事件和意外事件，在团队之间保持共识、分享经验教训，以及确定改进的优先级和目标。

未建立此最佳实践暴露的风险等级： 中

实施指导

- 审核运营指标：定期与来自不同业务领域的跨团队参与者对运营指标进行回顾性分析。与包括业务、开发和运营团队在内的利益相关方共同分析通过即时反馈和回顾性分析得到的发现，并分享经验教训。根据他们的见解来确定改进机会和可能的行动方案。
  - Amazon CloudWatch
  - 使用 Amazon CloudWatch 指标
  - 发布自定义指标
  - Amazon CloudWatch 指标和维度参考

资源

相关文档：

- Amazon CloudWatch
- Amazon CloudWatch 指标和维度参考
- 发布自定义指标
- 使用 Amazon CloudWatch 指标

OPS11-BP08 记录和分享经验教训

记录和分享在运营活动中获得的经验教训，以便在内部和不同团队中利用。

您应该分享团队学到的经验教训，以增加整个组织的效益。您需要分享信息和资源，以防止出现可避免的错误并简化开发工作。这让您可以专注于交付所需的功能。

使用 AWS Identity and Access Management (IAM) 定义权限，以允许对您要在账户内和账户之间共享的资源进行受控访问。然后，您应该使用版本受控的 AWS CodeCommit 存储库来分享应用程序库、脚本程序、程序文档和其他系统文档。您可以共享对 AMI 的访问权限并授权跨账户使用 Lambda 函数，以此来分享您的计算标准。您还应将您的基础设施标准共享为 AWS CloudFormation 模板。

通过 AWS API 和 SDK，您可以集成外部和第三方工具和存储库（例如，GitHub、BitBucket 和 SourceForge）。在分享您学到的和开发的内容时，请注意设定权限以确保共享存储库的完整性。

常见反模式：

- 由于使用了组织中常用的错误库，导致了长时间的停机。自此之后，您已经迁移到可靠的库。您组织中的其他团队尚未意识到风险的存在。如果您记录并分享对于此库的经验，他们就可能会注意到风险。
- 您已经确定了内部共享微服务中导致会话中断的边缘案例。为了避免这一边缘案例的出现，您更新了对服务的调用。您组织中的其他团队尚未意识到风险的存在。如果您记录并分享对于此库的经验，他们就可能会注意到风险。
- 您已找到一种方法，可以显著降低其中一个微服务的 CPU 利用率要求。您不知道其他团队是否可以利用这种技术。如果您记录并分享对于此库的经验，他们将有机会加以利用。

建立此最佳实践的好处： 分享经验教训可以为改进提供支持，并最大程度地从经验中获益。

未建立这种最佳实践的情况下暴露的风险等级： 低

实施指导

- 记录和分享经验教训：设置程序来记录在运营活动执行和回顾性分析过程中获得的经验教训，供其他团队利用。
  - 分享经验教训：设置程序在不同团队中分享经验教训和相关构件。例如，通过可以访问的 Wiki 共享更新后的程序、指南、管理机制和最佳实践。通过公共存储库共享脚本、代码和库。
    - 授权访问 AWS 环境
    - 共享 AWS CodeCommit 存储库
    - AWS Lambda 函数的简单授权
    - 将 AMI 与特定 AWS 账户共享
    - 利用 AWS CloudFormation Designer URL 快速共享模板

- 将 AWS Lambda 与 Amazon SNS 配合使用

资源

相关文档：

- AWS Lambda 函数的简单授权
- 共享 AWS CodeCommit 存储库
- 将 AMI 与特定 AWS 账户共享
- 利用 AWS CloudFormation Designer URL 快速共享模板
- 将 AWS Lambda 与 Amazon SNS 配合使用

相关视频：

- 授权访问 AWS 环境

OPS11-BP09 分配时间进行改进

流程中专用的时间和资源可以实现持续增量改进。

在 AWS 上，您可以创建临时的环境副本，从而降低试验和测试的风险、工作量及成本。这些重复的环境可用于测试分析、试验、开发和测试计划改进时所得出的结论。

常见反模式：

- 您的应用程序服务器中存在一个已知性能问题。它被添加到每个计划内功能实施之后的待办事项中。如果计划功能的添加速率保持不变，那么性能问题将永远无法解决。
- 为了支持持续改进，您批准管理员和开发人员利用他们所有的额外时间来选择和实施改进。没有完成任何改进。

建立此最佳实践的好处： 通过在流程中投入专用时间和资源，您可以实现持续增量改进。

未建立这种最佳实践的情况下暴露的风险等级： 低

实施指导

- 分配时间进行改进：在流程中抽出专门的时间和资源，用于实现持续增量改进。实施更改以便改进，并评估结果以确定是否成功。如果结果不符合目标，并且仍然需要改进，则寻求其他行动方案。

安全性

主题

- 安全基础知识
- 身份与权限管控
- 检测
- 基础设施保护
- 数据保护
- 事件响应

安全基础知识

问题

- SEC 1 如何安全地操作您的工作负载？

SEC 1 如何安全地操作您的工作负载？

为了安全地操作您的工作负载，您必须对安全性的各个方面应用总体最佳实践。采用您在组织和工作负载层面的卓越运营中定义的要求和流程，并将它们应用到各个方面。及时了解最新的 AWS、行业建议以及威胁情报信息可帮助您改进您的威胁模型和控制目标。实现安全流程、测试和验证的自动化可扩展您的安全运营。

最佳实践

- SEC01-BP01 使用账户分隔工作负载
- SEC01-BP02 保护 AWS 账户
- SEC01-BP03 识别并验证控制目标
- SEC01-BP04 及时了解最新的安全威胁
- SEC01-BP05 及时了解最新的安全建议
- SEC01-BP06 在管道中自动测试和验证安全控制措施
- SEC01-BP07 使用威胁模型识别风险并确定其优先级
- SEC01-BP08 定期评估和实施新的安全服务和功能

SEC01-BP01 使用账户分隔工作负载

从安全性和基础设施入手，随着工作负载的增长，使您的组织能够设置通用防护。这种方法在工作负载之间提供了边界和控制。强烈建议执行账户级分离，以使生产环境与开发和测试环境分离，或者在需要处理外部合规性要求（例如 PCI-DSS 或 HIPAA）所定义的各级敏感数据的工作负载与无需处理这些数据的工作负载之间提供强大的逻辑边界。

未建立这种最佳实践的情况下暴露的风险等级： 高

实施指导

- 使用 AWS Organizations：使用 AWS Organizations 集中实现针对多个 AWS 账户的基于策略的管理。
  - 开始使用 AWS Organizations
  - 如何使用服务控制策略来设置您在 AWS Organization 中的跨账户权限防护机制
- 考虑使用 AWS Control Tower：AWS Control Tower 基于最佳实践，提供了一种简单的方法来设置和管理新的、安全的多账户 AWS 环境。
  - AWS Control Tower

资源

相关文档：

- IAM 最佳实践
- 安全公告
- AWS 安全性审计指导原则

相关视频：

- 使用 AWS Organizations 管理多账户 AWS 环境
- 架构完善的安全性最佳实践
- 使用 AWS Control Tower 监管多账户 AWS 环境

SEC01-BP02 保护 AWS 账户

您的 AWS 账户可以通过很多方法进行保护，包括保护 根用户且不使用它，并及时更新联系人信息。随着您的工作负载增长和扩展，您可以使用 AWS Organizations 集中管理和控制您在 AWS 中的账户。AWS Organizations 可以帮助您管理账户、设置控制以及跨账户配置服务。

未建立此最佳实践暴露的风险等级： 高

实施指导

- 使用 AWS Organizations：使用 AWS Organizations 集中实现针对多个 AWS 账户的基于策略的管理。
  - 开始使用 AWS Organizations
  - 如何使用服务控制策略来设置您在 AWS Organization 中的跨账户权限防护机制
- 限制 AWS 根用户的使用：只使用根用户执行明确需要根用户的任务。
  - 需要 AWS 账户根用户凭证的 AWS 任务
- 为根用户启用多重身份验证（MFA）：如果没有使用 AWS Organizations 为您管理根用户，请在 AWS 账户根用户上启用 MFA。
  - 根用户
- 定期更改根用户密码：更改根用户密码可降低使用已保存的密码的风险。如果您未使用 AWS Organizations 且有其他人具有物理访问权限，那么这一点尤为重要。
  - 更改 AWS 账户根用户密码
- 使用 AWS 账户根用户时启用通知：自动接收通知可降低风险。
  - 如何在 AWS 账户的根访问密钥被使用时接收通知
- 限制对新添加的区域的访问：对于新的 AWS 区域，诸如用户和角色之类的 IAM 资源将仅传播到您启用的区域。
  - 设置权限，为即将推出的 AWS 区域启用账户
- 考虑使用 AWS CloudFormation StackSets：CloudFormation StackSets 可用于通过已批准的模板将资源（包括 IAM 策略、角色和组）部署到不同的 AWS 账户和区域中。
  - 使用 CloudFormation StackSets

资源

相关文档：

- AWS Control Tower

- AWS 安全性审计指导原则
- IAM 最佳实践
- 安全公告

相关视频：

- 利用自动化和监管，支持大规模采用 AWS
- 架构完善的安全性最佳实践

相关示例：

- 实验室：AWS 账户和根用户

SEC01-BP03 识别并验证控制目标

根据您的合规性要求以及从威胁模型中发现的风险，获得并验证您需要应用于工作负载的控制目标和控制措施。持续验证控制目标和控制措施可帮助您衡量风险缓解措施的有效性。

未建立这种最佳实践的情况下暴露的风险等级： 高

实施指导

- 确定合规性要求：了解您的工作负载必须符合的组织、法律和合规性要求。
- 确定 AWS 合规性资源：确定 AWS 帮助您实现合规性的资源。
  - https://aws.amazon.com/compliance/
  - https://aws.amazon.com/artifact/

资源

相关文档：

- AWS 安全性审计指导原则
- 安全公告

相关视频：

- AWS Security Hub：管理安全警报和自动执行合规性检查

- 架构完善的安全性最佳实践

SEC01-BP04 及时了解最新的安全威胁

通过及时了解最新的安全威胁，帮助您定义并实施适当的控制措施，识别攻击媒介。使用 AWS Managed Services 可以更轻松地接收 AWS 账户中意外或异常行为的通知。在您的安全信息流程中，使用 AWS 合作伙伴工具或第三方威胁信息源进行调查。此 通用漏洞披露（CVE，Common Vulnerabilities and Exposures）列表 包含公开披露的网络安全漏洞，可供您用于掌握最新信息。

未建立此最佳实践暴露的风险等级： 高

实施指导

- 订阅威胁情报来源：定期查看来自多个来源、与您在工作负载中所用技术相关的威胁情报信息。
  - 通用漏洞披露列表
- 考虑使用 AWS Shield Advanced 服务：如果您的工作负载可通过互联网访问，则该服务可让您近乎实时地了解情报来源。

资源

相关文档：

- AWS 安全性审计指导原则
- AWS Shield
- 安全公告

相关视频：

- 架构完善的安全性最佳实践

SEC01-BP05 及时了解最新的安全建议

及时了解最新的 AWS 和行业安全建议，以改善您的工作负载安全状况。AWS 安全公告 包含有关安全性和隐私通知的重要信息。

未建立此最佳实践暴露的风险等级： 高

实施指导

- 关注 AWS 更新：订阅或定期查看新建议、提示与诀窍。
  - AWS Well-Architected 实验室
  - AWS 安全性博客
  - AWS 服务文档
- 订阅行业新闻：定期查看来自多个来源、与您在工作负载中所用技术相关的新闻动态。
  - 示例：通用漏洞披露列表

资源

相关文档：

- 安全公告

相关视频：

- 架构完善的安全性最佳实践

SEC01-BP06 在管道中自动测试和验证安全控制措施

为安全机制建立可靠的基准和模板，并将其作为构建、管道和流程的一部分进行测试和验证。利用工具和自动化功能，持续测试并验证所有的安全控制措施。例如，对机器镜像和基础设施即代码模板等项目进行扫描，以发现安全漏洞、异常以及与每个阶段的既定基准的偏差。AWS CloudFormation Guard 可帮助您验证 CloudFormation 模板是否安全，为您节省时间并减少配置错误风险。

减少引入到生产环境中的安全性错误配置的数量至关重要 — 在构建过程中，可以执行的质量控制和可以减少的缺陷越多越好。设计持续集成和持续部署 (CI/CD) 管道，以便尽可能测试安全问题。CI/CD 管道提供了在构建和交付的每个阶段增强安全性的机会。还必须确保 CI/CD 安全工具始终是最新版本，以减轻不断变化的威胁。

跟踪对工作负载配置进行的更改，帮助您进行合规性审计、更改管理以及可能适用于您的调查。您可以使用 AWS Config 记录和评估您的 AWS 和第三方资源。这使您可以依据规则及合规包（合规包是带有补救操作的规则集合），连续审计和评估您的整体合规情况。

更改跟踪应包括计划更改，计划更改可能是组织更改控制流程（有时也称作 MACD，即移动、添加、更改、删除（Move, Add, Change, Delete））、临时更改或意外更改（如意外事件）的一部分。更改可

能出现在基础设施中，但也可能涉及其他类别，如代码存储库中的更改、机器镜像和应用程序清单更改、流程和策略更改或文档更改。

未建立此最佳实践暴露的风险等级： 中

实施指导

- 自动管理配置：使用配置管理服务或工具自动实施安全配置并对其进行验证。
  - AWS Systems Manager
  - AWS CloudFormation
  - 在 AWS 上设置 CI/CD 管道

资源

相关文档：

- 如何使用服务控制策略来设置您在 AWS Organization 中的跨账户权限防护机制

相关视频：

- 使用 AWS Organizations 管理多账户 AWS 环境
- 架构完善的安全性最佳实践

SEC01-BP07 使用威胁模型识别风险并确定其优先级

使用威胁模型识别并维护一个最新的潜在威胁登记表。确定您的威胁优先级并调整您的安全控制措施，以进行防范、检测和响应。在不断变化的安全环境中，重新审视和维护此登记表。

威胁建模提供了系统化的方法，用于在设计流程的早期阶段协助查找和解决安全问题。这个时机越早越好，因为相比生命周期的后期，前期补救成本更低。

威胁建模流程的常规核心步骤包括：

1. 确定资产、参与者、入口点、组件、使用案例和信任级别，并在设计图中包括这些内容。
2. 确定威胁列表。
3. 对于每个威胁，确定防范措施，这可以包括实施安全控制措施。
4. 创建并检查风险矩阵，以确定是否采取了足够的措施来防范威胁。

在工作负载（或工作负载功能）级别进行威胁建模最为有效，这可以确保获取所有上下文信息用于评估。随着安全形势的变化，请重新检查并维护此矩阵。

未建立此最佳实践暴露的风险等级： 低

实施指导

- 创建威胁模型：威胁模型可以帮助您识别和解决潜在的安全威胁。
  - NIST：以数据为中心的系统威胁建模指南

资源

相关文档：

- AWS 安全性审计指导原则
- 安全公告

相关视频：

- 架构完善的安全性最佳实践

SEC01-BP08 定期评估和实施新的安全服务和功能

评估并实施 AWS 和 AWS 合作伙伴提供的安全服务和功能，以改善您的工作负载安全状况。AWS 安全博客重点介绍新的 AWS 服务和功能、实施指导和常规安全指南。AWS 的最新内容 是一个很好的工具，可帮助您随时了解所有新的 AWS 功能、服务和公告。

未建立这种最佳实践的情况下暴露的风险等级： 低

实施指导

- 规划定期审核：创建审核活动日历，包括遵守合规性要求、评估新的 AWS 安全功能和服务，以及及时了解行业最新动态。
- 发现 AWS 服务和功能：发现适用于您使用的服务的安全功能，并在新功能发布时查看这些功能。
  - AWS 安全性博客
  - AWS 安全公告
  - AWS 服务文档

- 定义 AWS 服务上线流程：定义用于上线新 AWS 服务的流程。包括您如何评估新 AWS 服务的功能，以及针对工作负载的合规性要求。
- 测试新的服务和功能：当有新的服务和功能发布时，在与生产环境非常相似的非生产环境中对其进行测试。
- 实施其他防御机制：实施自动化机制来保护您的工作负载，并探索可用选项。
  - 按照 AWS Config 规则 修正不合规的 AWS 资源

资源

相关视频：

- 架构完善的安全性最佳实践

身份与权限管控

问题

- SEC 2 如何管理人员和机器的身份验证？
- SEC 3 如何管理人员和机器的权限？

SEC 2 如何管理人员和机器的身份验证？

在访问和运行安全的 AWS 工作负载时，您需要管理两种类型的身份。了解管理和授予访问权限所需的身份类型，这有助于确保正确的身份能够在正确的条件下访问正确的资源。

人员身份：您的管理员、开发人员、操作员和最终用户需要确定身份才能访问您的 AWS 环境和应用程序。这些是您的组织成员或您与之协作的外部用户，以及通过 Web 浏览器、客户端应用程序或交互式命令行工具与您的 AWS 资源交互的用户。

机器身份：您的服务应用程序、操作工具和工作负载需要一个身份来向 AWS 服务发出请求，例如，读取数据。这些身份包括在 AWS 环境中运行的机器，例如 Amazon EC2 实例或 AWS Lambda 函数。您还可以管理需要访问权限的外部各方的机器身份。此外，您可能还有需要访问您 AWS 环境的 AWS 之外的机器。

最佳实践

- SEC02-BP01 使用强大的登录机制
- SEC02-BP02 使用临时凭证

- SEC02-BP03 安全存储和使用密钥
- SEC02-BP04 依赖集中式身份提供者
- SEC02-BP05 定期审计和轮换凭证
- SEC02-BP06 利用用户组和属性

SEC02-BP01 使用强大的登录机制

强制执行最小密码长度策略，并指导您的用户避免使用常见或重复使用过的密码。使用软件或硬件机制实施 Multi-Factor Authentication (MFA)，以提供一层额外的保护。例如，当使用 IAM Identity Center 作为身份源时，请为 MFA 配置“背景认知”或“始终开启”设置，并允许用户注册自己的 MFA 设备以加快采用速度。当使用外部身份提供程序 (IdP) 时，请为 MFA 配置您的 IdP。

未建立这种最佳实践的情况下暴露的风险等级： 高

实施指导

- 创建 Identity and Access Management（IAM）策略来实施 MFA 登录：创建客户管理的一项 IAM 策略，禁止其他所有 IAM 操作（除了允许用户在“我的安全凭证”页面上代入角色、更改自己的凭证、以及管理其 MFA 设备）。
- 在身份提供者中启用 MFA：在您使用的身份提供者中启用 MFA 或者启用单点登录服务，例如 AWS IAM Identity Center。
- 配置强密码策略：配置强 密码策略 （在 IAM 和联合身份系统中）来防护暴力攻击。

- 定期轮换凭证：确保工作负载管理员定期更改其密码和访问密钥（如果使用）。

资源

相关文档：

- 开始使用 AWS Secrets Manager
- IAM 最佳实践
- 身份提供程序和联合
- AWS 账户根用户
- 开始使用 AWS Secrets Manager
- 临时安全凭证
- 安全合作伙伴解决方案：访问和访问控制

- 临时安全凭证
- AWS 账户根用户

相关视频：

- 有关大规模管理、检索和轮换密钥的最佳实践
- 使用 IAM Identity Center 大规模管理用户权限
- 在每个层面掌握身份

SEC02-BP02 使用临时凭证

需要身份以动态获取 临时凭证。对于员工身份，使用 AWS IAM Identity Center 或与 AWS Identity and Access Management（IAM）角色联合访问 AWS 账户。对于机器身份，例如 Amazon Elastic Compute Cloud（Amazon EC2）实例或 AWS Lambda 函数，要求使用 IAM 角色，而不是拥有长期访问密钥的 IAM 用户。

对于使用 AWS Management Console 的人员身份，要求用户获取临时凭证并联合到 AWS 中。您可以使用 AWS IAM Identity Center 用户门户来完成此操作。对于需要访问 CLI 的用户，请确保他们使用 AWS CLI v2，它支持与 IAM Identity Center 直接集成。用户可以创建链接到 IAM Identity Center 账户和角色的 CLI 配置文件。CLI 会自动从 IAM Identity Center 检索 AWS 凭证，并代表您刷新这些凭证。这样就无需从 IAM Identity Center 控制台复制并粘贴临时 AWS 凭证。对于开发工具包，用户应依靠 AWS Security Token Service（AWS STS）来代入角色，以接收临时凭证。在某些情况下，使用临时凭证可能并不现实。您应了解存储访问密钥的风险、经常轮换这些密钥，并尽可能要求使用多重身份验证（MFA）作为一项条件。使用最后访问的信息来确定何时轮换或删除访问密钥。

当您需要授权使用方访问您的 AWS 资源时，请使用 Amazon Cognito 身份池，并为他们分配一组临时的有限权限凭证，以使他们能够访问您的 AWS 资源。通过您创建的 IAM 角色 控制每个用户的权限。您可以定义规则，以根据用户的 ID 令牌中的声明，为每个用户选择角色。您可以为通过身份验证的用户定义一个默认角色。对于未通过身份验证的访客用户，您还可以定义一个拥有有限权限的单独 IAM 角色。

对于机器身份，您应依靠 IAM 角色授予对 AWS 的访问权限。对于 Amazon Elastic Compute Cloud（Amazon EC2）实例，您可以使用 适用于 Amazon EC2 的角色。您可以将 IAM 角色附加到您的 Amazon EC2 实例，以使您在 Amazon EC2 上运行的应用程序能够使用 AWS 创建的临时安全凭证，并通过实例元数据服务（IMDS，Instance Metadata Service）自动进行轮换。此 最新版本 的 IMDS 可防御暴露临时凭证的漏洞，应该予以实施。要使用密钥或密码访问 Amazon EC2 实例，AWS Systems Manager 是一种更安全的方法，它允许您使用预安装的代理来访问和管理实例，而无需使用

存储的密钥。此外，您也可以使用其他 AWS 服务（例如 AWS Lambda）来配置 IAM 服务角色，以授权此服务利用临时凭证执行 AWS 操作。在无法使用临时凭证的情况下，请使用编程工具，例如 AWS Secrets Manager，来自动完成凭证轮换和管理。

定期审计和轮换凭证：（最好通过自动化工具）定期验证，以确保实施正确的控制措施。对于人员身份，您应要求用户定期更改他们的密码并弃用访问密钥，以支持临时凭证。在从 IAM 用户转向集中身份时，您可以 生成凭证报告 以审计 IAM 用户。我们还建议您在身份提供者中实施 MFA 设置。您可以设置 AWS Config 规则 来监控这些设置。对于机器身份，您应依靠使用 IAM 角色的临时凭证。当无法执行此操作时，需要经常审计和轮换访问密钥。

安全存储和使用密钥： 对于并非与 IAM 相关且无法利用临时凭证的凭证，如数据库登录，请使用一种专门用于处理密钥管理的服务，比如 Secrets Manager。借助 Secrets Manager，您可以使用 支持的服务轻松管理、轮换和安全地存储加密密钥。为访问密钥而执行的调用将记录到 AWS CloudTrail 中以用于审计，IAM 权限可以为它们授予最低访问权限。

未建立这种最佳实践的情况下暴露的风险等级： 高

实施指导

- 实施最低权限策略：向 IAM 组和角色分配具有最低权限的访问策略，以反映所定义的用户角色或职能。
  - 授予最小特权
- 删除不必要的权限：通过删除不必要的权限来实施最低权限。
  - 通过查看用户活动缩小策略范围
  - 查看角色访问权限

- 考虑使用权限边界：权限边界是一项高级功能，它使用托管策略设置基于身份的策略可向 IAM 实体授予的最高权限。实体的权限边界仅允许实体执行其基于身份的策略及其权限边界都允许的操作。
  - 实验室：IAM 权限边界委派角色创建
- 考虑为权限分配资源标签：您可以使用标签来控制对支持标记的 AWS 资源的访问。您还可以对 IAM 用户和角色进行标记，以控制他们可以访问的内容。
  - 实验室：基于 IAM 标签的 EC2 访问控制
  - 基于属性的访问控制 (ABAC)

资源

相关文档：

- 开始使用 AWS Secrets Manager
- IAM 最佳实践
- 身份提供程序和联合
- 安全合作伙伴解决方案：访问和访问控制
- 临时安全凭证
- AWS 账户根用户

相关视频：

- 有关大规模管理、检索和轮换密钥的最佳实践
- 使用 AWS IAM Identity Center 大规模管理用户权限
- 在每个层面掌握身份

SEC02-BP03 安全存储和使用密钥

对于需要密钥（例如用于第三方应用程序的密码）的员工和机器身份，请根据最新的行业标准，在专业服务中存储并自动轮换它们。对于并非与 IAM 相关且无法利用临时凭证的凭证，如数据库登录，请使用一种专门用于处理密钥管理的服务，比如 AWS Secrets Manager。借助 Secrets Manager，您可以使用支持的服务轻松管理、轮换和安全存储加密密钥。为访问密钥而执行的调用将记录到 AWS CloudTrail 中以用于审计，IAM 权限可以为它们授予最低访问权限。

未建立此最佳实践暴露的风险等级： 高

实施指导

- 使用 AWS Secrets Manager： AWS Secrets Manager 是一项 AWS 服务，让您能够轻松地管理密钥。密钥可以是数据库凭证、密码、第三方 API 密钥甚至任意文本。

资源

相关文档：

- 开始使用 AWS Secrets Manager
- 身份提供程序和联合

相关视频：

- 有关大规模管理、检索和轮换密钥的最佳实践

SEC02-BP04 依赖集中式身份提供者

对于员工身份，依赖身份提供商，使您能够在集中位置管理身份。这样，您就可以更轻松地管理跨多个应用程序和服务的访问权限，因为您在从单一位置创建、管理和撤销访问权限。例如，如果有人离开了您的组织，您可以从一个位置撤销此人对所有应用程序和服务（包括 AWS）的访问权限。这样就降低了对多个凭证的需求，并提供了与现有的人力资源 (HR) 流程集成的机会。

要与单独的 AWS 账户联合，您可以将用于 AWS 的集中身份与基于 SAML 2.0 并支持 AWS Identity and Access Management 的提供程序结合使用。无论是由您在 AWS 中托管的提供程序、AWS 外部的提供程序还是由 AWS Partner 提供的提供程序，您都可以使用，只要这些提供程序与 SAML 2.0 协议兼容。您可以使用您的 AWS 账户与您选择的提供程序之间的联合，为用户或应用程序授予访问权限，以使他们能通过使用 SAML 断言获得临时安全凭证，以调用 AWS API 操作。基于 Web 的单点登录同样受支持，因此允许用户从您的登录网站登录到 AWS Management Console。

要与您的 AWS Organizations 中的多个账户联合，您可以在 AWS IAM Identity Center (IAM Identity Center)中配置您的身份源，并指定您的用户和组的存储位置。配置之后，您的身份提供程序将是您的事实来源，并可以使用跨域身份管理系统 (SCIM) v2.0 协议来 同步 信息。随后，您可以查找用户或组，并授予他们 IAM Identity Center 访问权限，以使他们能够访问 AWS 账户和/或云应用程序。

IAM Identity Center 与 AWS Organizations 集成，这样，您就可以配置您的身份提供程序，然后为您的组织中管理的 现有账户和新账户授予访问权限 。IAM Identity Center 为您提供了一个默认存储库，您可以使用它来管理您的用户和组。如果您选择使用 IAM Identity Center 存储库，请创建您的用户和组，并为他们分配对您的 AWS 账户和应用程序的访问权限级别，同时铭记最低权限最佳实践。您也可以选择使用 SAML 2.0 连接到您的外部身份提供程序 ，或 连接到您的 Microsoft AD 目录 （使用 AWS Directory Service）。配置之后，您可以通过您的中央身份提供者进行身份验证，以登录到 AWS Management Console或 AWS 移动应用程序。

要管理您的工作负载的最终用户或消费者，例如移动应用程序，您可以使用 Amazon Cognito。它为您的 Web 和移动应用程序提供了身份验证、授权和用户管理功能。您的用户可以直接使用用户名和密码登录，也可以通过第三方（例如 Amazon、Apple、Facebook 或 Google）登录。

未建立此最佳实践暴露的风险等级： 高

实施指导

- 集中管理访问：创建 Identity and Access Management（IAM）身份提供者实体，以在您的 AWS 账户与身份提供者（IdP）之间建立信任关系。IAM 支持与 OpenID Connect（OIDC）或 SAML 2.0（Security Assertion Markup Language 2.0，安全断言标记语言 2.0）兼容的 IdP。

  - 身份提供程序和联合
- 集中应用程序访问：考虑使用 Amazon Cognito 实现应用程序集中式访问。借助 Amazon Cognito，您可以快速轻松地将用户注册、登录和访问控制添加到 Web 和移动应用程序中。 Amazon Cognito 可扩展至数百万用户，并支持使用社交身份提供者（如 Facebook、Google 和 Amazon）登录，以及通过企业身份提供者使用 SAML 2.0 登录。

- 删除旧的 IAM 用户和组：在您开始使用身份提供者（IdP）后，请删除不再需要的 IAM 用户和组。
  - 查找未使用的凭证
  - 删除 IAM 组

资源

相关文档：

- IAM 最佳实践
- 安全合作伙伴解决方案：访问和访问控制
- 临时安全凭证
- AWS 账户根用户

相关视频：

- 有关大规模管理、检索和轮换密钥的最佳实践
- 使用 AWS IAM Identity Center 大规模管理用户权限
- 在每个层面掌握身份

SEC02-BP05 定期审计和轮换凭证

当您无法依赖临时凭证并需要长期凭证时，请审计凭证，以确保实施了定义的控制措施（例如多重身份验证（MFA））、凭证定期轮换且具有适当的访问级别。（最好通过自动化工具）定期验证，以确保实施正确的控制措施。对于人员身份，您应要求用户定期更改他们的密码并弃用访问密钥，以支持临时凭证。在从 AWS Identity and Access Management（IAM）用户转向集中身份时，您可以 生成凭证报告 以审计 IAM 用户。我们还建议您在身份提供者中实施 MFA 设置。您可以设置 AWS Config 规则 来监控这些设置。对于机器身份，您应依靠使用 IAM 角色的临时凭证。当无法执行此操作时，需要经常审计和轮换访问密钥。

未建立这种最佳实践的情况下暴露的风险等级： 中

实施指导

- 定期审计凭证：使用凭证报告以及 Identity and Access Management（IAM）Access Analyzer 审计 IAM 凭证和权限。
  - IAM Access Analyzer
  - 获取凭证报告
  - 实验室：自动化 IAM 用户清理
- 使用访问级别审核 IAM 权限：为了提高您的 AWS 账户的安全性，请定期审核和监控每个 IAM 策略。请确保您的策略授予仅执行必要操作所需的最低权限。
  - 使用访问级别审核 IAM 权限
- 考虑自动创建和更新 IAM 资源：AWS CloudFormation 可用于自动部署 IAM 资源（包括角色和策略），以减少人为错误，因为可以验证模板和控制版本。
  - 实验室：自动部署 IAM 组和角色

资源

相关文档：

- 开始使用 AWS Secrets Manager
- IAM 最佳实践
- 身份提供程序和联合
- 安全合作伙伴解决方案：访问和访问控制
- 临时安全凭证

相关视频：

- 有关大规模管理、检索和轮换密钥的最佳实践
- 使用 AWS IAM Identity Center 大规模管理用户权限
- 在每个层面掌握身份

SEC02-BP06 利用用户组和属性

随着您管理的用户数量不断增加，您需要确定如何组织这些用户，以便能够实现规模管理。将具有常见安全要求的用户置于由您的身份提供程序定义的组中，并建立机制以确保用于访问控制的用户属性（例如部门或位置）正确无误且已更新。使用这些组和属性（而不是单个用户）来控制访问权限。这样，您就可以通过使用 权限集一次性更改用户的组成员身份或属性来集中管理访问，而不是在需要更改用户的访问权限时更新多个单独策略。您可以使用 AWS IAM Identity Center（IAM Identity Center）来管理用户组和属性。IAM Identity Center 支持最常用的属性，无论是在创建用户时手动输入的属性还是使用同步引擎自动预置的属性，例如跨域身份管理系统（SCIM，Cross-Domain Identity Management）规范中定义的那些属性。

将具有常见安全要求的用户置于由您的身份提供程序定义的组中，并建立机制以确保用于访问控制的用户属性（例如部门或位置）正确无误且已更新。使用这些组和属性（而不是单个用户）来控制访问。这使您可以通过一次性更改用户的组成员身份或属性来集中管理访问，而不是在用户的访问需要更改时更新多个单独策略。

未建立这种最佳实践的情况下暴露的风险等级： 低

实施指导

- 如果您在使用 AWS IAM Identity Center（IAM Identity Center）配置组：IAM Identity Center 使您能够配置用户组，并为组分配所需的权限级别。
  - AWS Single Sign-On – 管理身份
- 了解基于属性的访问控制（ABAC，Attribute-Based Access Control）：ABAC 是一种基于属性定义权限的授权策略。
  - 什么是适用于 AWS 的 ABAC？
  - 实验室：基于 IAM 标签的 EC2 访问控制

资源

相关文档：

- 开始使用 AWS Secrets Manager
- IAM 最佳实践
- 身份提供程序和联合
- AWS 账户根用户

相关视频：

- 有关大规模管理、检索和轮换密钥的最佳实践
- 使用 AWS IAM Identity Center 大规模管理用户权限
- 在每个层面掌握身份

相关示例：

- 实验室：基于 IAM 标签的 EC2 访问控制

SEC 3 如何管理人员和机器的权限？

管理权限以控制对需要访问 AWS 和您的工作负载的人员和机器身份的访问。权限用于控制哪些人可以在什么条件下访问哪些内容。

最佳实践

- SEC03-BP01 定义访问要求
- SEC03-BP02 授予最低访问权限
- SEC03-BP03 建立紧急访问流程
- SEC03-BP04 持续减少权限
- SEC03-BP05 为您的组织定义权限防护机制
- SEC03-BP06 基于生命周期管理访问权限
- SEC03-BP07 分析公共和跨账户访问
- SEC03-BP08 安全地共享资源

SEC03-BP01 定义访问要求

管理员、最终用户或其他组件都需要访问您工作负载的每个组件或资源。明确定义哪些人员或事物应当有权访问每个组件，选择用于进行身份验证和授权的适当身份类型和方法。

常见反模式：

- 在应用程序中进行硬编码或存储密码。
- 向每个用户授予自定义权限。
- 使用长期有效的凭证。

未建立这种最佳实践的情况下暴露的风险等级： 高

实施指导

管理员、最终用户或其他组件都需要访问您工作负载的每个组件或资源。明确定义哪些人员或事物应当有权访问每个组件，选择用于进行身份验证和授权的适当身份类型和方法。

应提供对组织内 AWS 账户 的常规访问，方法是使用 联合身份访问 或集中式身份提供者。您还应将身份管理集中处理，确保对于 AWS 将访问集成到员工访问生命周期中已建立了既定做法。例如，当员工转岗到具有不同访问级别的职位时，该员工的小组成员资格也应进行更改以反映新的访问要求。

在定义非人类身份的访问要求时，请确定哪些应用程序和组件需要访问权限以及如何向其授予权限。建议使用通过最低权限访问模型构建的 IAM 角色。AWS 托管策略 提供了预定义的 IAM 策略，这些策略涵盖了大多数常见使用案例。

AWS 服务（例如 AWS Secrets Manager 和 AWS Systems Manager Parameter Store）可以帮助在无法使用 IAM 角色的情况下，安全地将密码与应用程序或工作负载分离。在 Secrets Manager 中，您可以为凭证建立自动轮换。您可以通过 Systems Manager 使用您在创建参数时指定的唯一名称，来引用脚本、命令、SSM 文档、配置和自动化工作流中的参数。

您可以使用 AWS Identity and Access Management Roles Anywhere 获取 IAM 中的临时安全凭证，这种凭证适用于在 AWS 外部运行的工作负载。您的工作负载可以使用 IAM 策略 和 IAM 角色 ，也就是您为访问 AWS 资源在 AWS 应用程序中所用的策略和角色。

如果可能，请优先选择短期临时凭证而不是长期静态凭证。在一些场景中，需要具有编程访问权限和长期凭证的 IAM 用户，此时请使用 访问密钥上次使用的信息 来轮换和删除访问密钥。

资源

相关文档：

- 基于属性的访问控制（ABAC）
- AWS IAM Identity Center
- IAM Roles Anywhere
- 适用于 IAM Identity Center 的 AWS 托管策略
- AWS IAM 策略条件
- IAM 使用案例
- 删除不必要的凭证
- 策略的使用

- 如何根据 AWS 账户、OU 或组织来控制对 AWS 资源的访问
- 使用 AWS Secrets Manager 中的增强搜索来轻松标识、安排和管理密钥

相关视频：

- 在 60 分钟以内成为 IAM 策略高手
- 职责分离、最低权限、委托和 CI/CD
- 简化身份和访问管理以实施创新

SEC03-BP02 授予最低访问权限

通过允许在特定条件下访问特定 AWS 资源上的特定操作，仅授予身份所需的访问权限。依靠组和身份属性来大规模动态设置权限，而不是为单个用户定义权限。例如，您可以允许一组开发人员访问，以便仅管理其项目的资源。这样，当开发人员被从组中移除时，开发人员的访问权限将在使用该组进行访问控制的任何位置被撤消，而无需对访问策略进行任何更改。

常见反模式：

- 默认为向用户授予管理员权限。
- 使用根账户进行日常活动。

未建立这种最佳实践的情况下暴露的风险等级： 高

实施指导

建立 最小权限 原则可确保身份只能执行完成特定任务所需的最小功能集，同时实现可用性和效率的平衡。按照此原则进行操作可以限制意外访问，并有助于确保您能够审计哪些用户有权访问哪些资源。在 AWS 中，默认情况下，身份不具有任何权限（根用户除外）。根用户的凭证应受到严格控制，并且应该仅用于少数 特定任务。

您可以使用策略来明确授予附加到 IAM 或资源实体的权限，例如联合身份或计算机所使用的 IAM 角色或者某些资源（例如 S3 存储桶）。当您创建并附加策略时，您可以指定服务操作、资源以及为使 AWS 允许访问而必须满足的条件。AWS 支持多种条件以帮助您缩小访问权限范围。例如，使用 `PrincipalOrgID` 条件键时，将会验证 AWS Organizations 的标识符，以便能够授权在您的 AWS 组织内访问。

您还可以控制 AWS 服务代表您发出的请求，例如要求 AWS CloudFormation 创建一个 AWS Lambda 函数，方法是使用 `CalledVia` 条件键。您应该对不同的策略类型进行分层，以有效地限制账户内的总

体权限。例如，您可以允许应用程序团队创建他们自己的 IAM 策略，但使用 权限边界 来限制他们可以授予的最高权限。

有几种 AWS 功能有助于您扩展权限管理并遵守最低权限原则。基于属性的访问控制 允许您根据资源标签 来限制权限，从而根据应用于资源的标签和调用的 IAM 主体做出授权决定。这使您能够将标记和权限策略结合使用，以实现精细的资源访问，而无需许多自定义策略。

另一种加速创建最低权限策略的方法是，在活动运行后基于 CloudTrail 权限生成策略。IAM Access Analyzer 会自动基于活动生成 IAM 策略。您也可以在组织或个人账户级别，使用 IAM Access Advisor 来跟踪上次获取的关于某个具体策略的信息。

确立一种节奏，按此节奏查看这些详细信息并删除不需要的权限。您应在 AWS 组织内建立权限防护机制，以控制任何成员账户中的最高权限。诸如 AWS Control Tower 这样的服务具有规范性的托管式预防控制机制， 并允许您定义自己的控制机制。

资源

相关文档：

- IAM 实体的权限边界
- 用于编写最低权限 IAM 策略的方法
- 通过基于访问活动生成的 IAM 策略，IAM Access Analyzer 可让您更轻松实施最低权限
- 使用上次获取的信息来细化权限
- IAM 策略类型及其使用时间
- 使用 IAM 策略模拟器测试 IAM 策略
- AWS Control Tower 中的防护机制
- 零信任架构：AWS 视角
- 如何使用 CloudFormation StackSets 实施最低权限原则

相关视频：

- 新一代权限管理
- 零信任：AWS 视角
- 如何使用权限边界限制 IAM 用户和角色以防止权限升级？

相关示例：

- 实验室：创建 IAM 权限边界委派角色

SEC03-BP03 建立紧急访问流程

万一发生自动化流程或管道问题，此流程允许紧急访问您的工作负载。这将帮助您依赖最低权限访问，但确保用户可以在需要时获得相应的访问级别。例如，为管理员建立用来验证和批准其请求的流程，如用于提供访问权限的紧急 AWS 跨账户角色，或者管理员在验证和批准紧急请求时所遵循的特定流程。

常见反模式：

- 未建立紧急流程，无法从现有身份配置中断状态中恢复。
- 授予长期提升权限以进行问题排查或恢复。

未建立这种最佳实践的情况下暴露的风险等级： 中

实施指导

建立紧急访问会采用多种形式，您应为此做好准备。首先是主要身份提供者失败。在此情况下，您应依赖具有所需权限的另一种访问方法进行恢复。此方法可以是后备身份提供者或 IAM 用户。第二种方法应受到 严格的控制和监控， 并在使用时发送通知。紧急访问身份应来自专用于此目的的账户，并且其权限只相当于专为恢复而设计的角色。

有些紧急访问需要临时提升管理访问权限，您还应为此做好准备。一个常见的场景是，将更改权限限制为用于部署更改的自动化流程。如果此流程出现问题，用户可能需要申请提升的权限，才能还原功能。在此情况下，请建立一个流程，使用户能够申请提升的访问权限，并使管理员能够验证和批准请求。还应在流程中提供实施计划，详细说明有关预置访问权限和设置break-glass紧急角色的最佳实践指南 SEC10-BP05 预置访问权限。

资源

相关文档：

- 在 AWS 上监控和通知
- 管理临时提升的访问权限

相关视频：

- 在 60 分钟以内成为 IAM 策略高手

SEC03-BP04 持续减少权限

当团队和工作负载确定他们需要哪些访问权限时，删除他们不再使用的权限，并建立审核流程以实现最低权限。持续监控和减少未使用的身份和权限。

当团队和项目刚刚起步时，您有时会选择授予宽泛的访问权限（在开发或测试环境中），以激励创新和敏捷性。我们建议您持续评估访问权限，特别是在生产环境中，将访问权限限制为仅提供所需的权限，实现最低权限。AWS 提供了访问权限分析功能，以帮助您识别未使用的访问权限。为了帮助您识别未使用的用户、角色、权限和凭证，AWS 会分析访问活动，并提供关于访问密钥和角色的上次使用情况的信息。您可以使用 上次访问时间戳 来 识别未使用的用户和角色并将它们移除。此外，您还可以查看关于服务和操作的上次访问情况的信息，并 收紧特定用户和角色的权限。例如，您可以使用关于上次访问情况的信息，确定您的应用程序角色需要执行的特定 Amazon Simple Storage Service（Amazon S3）操作，并只允许访问这些操作。AWS Management Console中提供了这些功能，您也可以对这些功能进行编程，以便将它们整合到您的基础设施工作流程和自动化工具中。

未建立这种最佳实践的情况下暴露的风险等级： 中

实施指导

- 配置 AWS Identity and Access Management（IAM）Access Analyzer：AWS IAM Access Analyzer 帮助您识别组织和账户中与外部实体共享的资源，例如 Amazon Simple Storage Service（Amazon S3）存储桶或 IAM 角色。
  - AWS IAM Access Analyzer

资源

相关文档：

- 基于属性的访问控制 (ABAC)
- 授予最小特权
- 删除不必要的凭证
- 策略的使用

相关视频：

- 在最多 60 分钟的时间内成为 IAM 策略高手
- 职责分离、最低权限、委托和 CI/CD

SEC03-BP05 为您的组织定义权限防护机制

建立通用控件以限制对组织中所有身份的访问。例如，您可以限制对特定 AWS 区域 的访问，或防止操作员删除通用资源，例如用于您的核心安全团队的 IAM 角色。

常见反模式：

- 在组织管理员账户中运行工作负载。
- 在同一账户中运行生产工作负载和非生产工作负载。

未建立这种最佳实践的情况下暴露的风险等级： 中

实施指导

当您在 AWS 中的工作负载增多并管理这些额外的工作负载时，您应使用账户分离这些工作负载，并使用 AWS Organizations 管理这些账户。我们建议您建立常用权限防护机制，以限制您所在组织中的所有身份的访问权限。例如，您可以限制对特定 AWS 区域 的访问，或防止您的团队删除常见资源，例如您的核心安全团队使用的 IAM 角色。

您可以首先实施示例服务控制策略，例如禁止用户禁用密钥服务。SCP 使用 IAM 策略语言，并允许您建立所有 IAM 主体（用户和角色）都要遵循的控制机制。您可以限制对特定服务操作和资源的访问，并根据特定的条件限制访问，以满足您所在组织的访问控制需求。如有必要，您可以为您的防护机制定义异常情况。例如，您可以为账户中除特定管理员角色以外的所有 IAM 实体限制服务操作。

我们建议您避免在管理账户中运行工作负载。应该使用管理账户来管理和部署将影响成员账户的安全防护机制。一些 AWS 服务支持使用委派管理员账户。在可能的情况下，您应使用此委派账户，而不是使用管理账户。您应严格限制对组织管理员账户的访问。

通过使用多账户策略，您可以更灵活地将防护机制应用于工作负载。AWS Security Reference Architecture 提供了有关如何设计账户结构的规范性指南。AWS Control Tower 等 AWS 服务提供了一些功能，可集中管理整个组织内的预防性和检测性控制机制。为组织中的每个账户或 OU 定义明确的用途，并根据该用途限制控制机制。

资源

相关文档：

- AWS Organizations
- 服务控制策略（SCP，Service Control Policy）
- 在多账户环境中充分利用服务控制策略

- AWS Security Reference Architecture（AWS SRA）

相关视频：

- 使用服务控制策略实施预防性防护机制
- 使用 AWS Control Tower 实施大规模管理
- AWS Identity and Access Management 深入探讨

SEC03-BP06 基于生命周期管理访问权限

将访问控制措施与操作员和应用程序生命周期以及您的集中联合身份提供者集成。例如，在用户离开组织或角色发生变化时删除用户的访问权限。

当您使用不同的账户管理工作负载时，您有时需要在这些账户之间共享资源。我们建议您使用 AWS Resource Access Manager (AWS RAM) 来共享资源。使用此服务，您可以轻松、安全地在您的 AWS Organizations 和组织部门内共享 AWS 资源。使用 AWS RAM，当账户移进和移出与之共享资源的组织或组织部门时，会自动授予或撤销对共享资源的访问权限。这样有助于您确保只与您的目标账户共享资源。

未建立此最佳实践暴露的风险等级： 低

实施指导

用户访问生命周期：针对加入的人员、工作职能变更和离开的人员实施用户访问生命周期策略，以确保只有在职用户具有访问权限。

资源

相关文档：

- 基于属性的访问控制 (ABAC)
- 授予最小特权
- IAM Access Analyzer
- 删除不必要的凭证
- 策略的使用

相关视频：

- 在最多 60 分钟的时间内成为 IAM 策略高手
- 职责分离、最低权限、委托和 CI/CD

SEC03-BP07 分析公共和跨账户访问

持续监控重点关注公共访问和跨账户访问的调查结果。将公共访问和跨账户访问限制为仅限需要此类访问的资源。

常见反模式：

- 在管理跨账户访问和对资源的公开访问时，没有遵循流程。

未建立这种最佳实践的情况下暴露的风险等级： 低

实施指导

在 AWS 中，您可以授权访问另一个账户中的资源。您可以使用附加到资源的策略（例如，Amazon Simple Storage Service（Amazon S3）存储桶策略）授予直接跨账户访问权限，或者通过允许某个身份代入另一个账户中的 IAM 角色来授予此类访问权限。使用资源策略时，请验证将访问权限授予了您组织中的身份，并且您确定是要公开这些资源。建立一个流程来审批所有需要可公开访问的资源。

IAM Access Analyzer 使用 可证明的安全性 来标识从账户的外部访问某个资源时的所有访问路径。它持续审核资源策略，并报告公开访问和跨账户访问的结果，以使您能够轻松分析可能非常宽泛的访问权限。请考虑配置 IAM Access Analyzer 与 AWS Organizations 来验证您可以监控所有账户。使用 IAM Access Analyzer，您还可以 预览 Access Analyzer 调查结果，然后再部署资源权限。这样，您便可以验证策略更改仅按照意图，授权对您资源的公共和跨账户访问。在设计多账户访问权限时，您可以使用信任策略来控制在何种情况下允许代入某个角色。例如，您可以限制特定的源 IP 范围才能代入角色。

您也可以使用 AWS Config 报告和修复资源 中任何意外设置为公开访问的配置（通过 AWS Config 策略检查）。诸如 AWS Control Tower 和 AWS Security Hub 等服务简化了跨 AWS Organizations 的检查和防护机制的部署，可以识别并修复公开暴露的资源。例如，AWS Control Tower 具有托管防护机制，可以检测是否有任何 Amazon EBS 快照可由所有 AWS 账户恢复。

资源

相关文档：

- 使用 AWS Identity and Access Management Access Analyzer
- AWS Control Tower 中的防护机制

- AWS 基础安全最佳实践标准
- AWS Config 托管规则
- AWS Trusted Advisor 检查参考

相关视频：

- 保护多账户环境的最佳实践
- 深入探究 IAM Access Analyzer

SEC03-BP08 安全地共享资源

管理对跨账户或您的 AWS Organizations 内的共享资源的使用。监控共享资源并查看共享资源访问。

常见反模式：

- 在向第三方授予跨账户访问权限时使用默认 IAM 信任策略。

未建立这种最佳实践的情况下暴露的风险等级： 低

实施指导

使用多个 AWS 账户管理工作负载时，您可能需要在账户之间共享资源。这种共享通常是某个 AWS Organizations 内的跨账户共享。多种 AWS 服务，例如 AWS Security Hub、Amazon GuardDuty和 AWS Backup 均具备与 Organizations 集成的跨账户功能。您可以使用 AWS Resource Access Manager 分享其他共用资源，例如 VPC 子网或中转网关连接、AWS Network Firewall或 Amazon SageMaker Runtime 管道。如果您希望确保账户仅在自己的 Organizations 内共享资源，我们建议使用 服务控制策略（SCP，Service Control Policy） 来防止向外部主体授予访问权限。

在共享资源时，您应采取相关措施来防止意外的访问。我们建议您将基于身份的控制与网络控制结合起来， 为组织创建数据边界。这些控制措施应施加严格的限制，确定哪些资源可以共享，并防止共享或暴露不应被外泄的资源。例如，作为数据边界的一部分，您可以使用 VPC 端点策略和 `aws:PrincipalOrgId` 条件，确保访问您 Amazon S3 存储桶的身份属于您的组织。

在一些情况下，您可能需要允许 Organizations 之外的资源或者向第三方授予对您账户的访问权限。例如，合作伙伴提供的监控解决方案可能会需要访问您账户内部的资源。在这些情况下，您应该创建 IAM 跨账户角色，并仅向该角色提供第三方所需的权限。您还应该使用 外部 ID 条件制定信任策略。使用外部 ID 时，您应该为每个第三方生成唯一的 ID。该唯一 ID 不应由第三方提供，也不应由其控制。如果第三方不再需要访问您的环境，您应删除该角色。在任何情况下，您都应该避免向第三方提供长期

IAM 凭证。保持对其他原生支持分享功能的 AWS 服务的关注。例如，AWS Well-Architected Tool 允许 将工作负载 与其他 AWS 账户分享。

在使用 Amazon S3 等服务时，建议 禁用 Amazon S3 存储桶的 ACL 并使用 IAM 策略来定义访问控制。要限制对 Amazon S3 的 源自 Amazon CloudFront的访问，请从来源访问身份（OAI）迁移到来源访问控制（OAC），后者支持使用 AWS KMS的服务器端加密等附加功能。

资源

相关文档：

- 存储桶拥有者向并非其拥有的对象授予跨账户权限
- 如何将信任策略与 IAM 结合使用
- 在 AWS 上构建数据边界
- 如何在向第三方授予对 AWS 资源的访问权限时使用外部 ID

相关视频：

- 使用 AWS Resource Access Manager 实现精细访问
- 使用 VPC 端点保护您的数据边界
- 在 AWS 上建立数据边界

检测

问题

- SEC 4 您如何检测和调查安全事件？

SEC 4 您如何检测和调查安全事件？

通过日志和指标来记录和分析事件，以便了解信息。针对安全事件和潜在的威胁采取措施，以便保护您的工作负载。

最佳实践

- SEC04-BP01 配置服务和应用程序日志记录
- SEC04-BP02 集中分析日志、结果和指标
- SEC04-BP03 自动响应事件

- SEC04-BP04 实施可操作的安全事件

SEC04-BP01 配置服务和应用程序日志记录

在整个工作负载中配置日志记录，包括应用程序日志、资源日志和 AWS 服务日志。例如，确保已为组织内的所有账户启用 AWS CloudTrail、Amazon CloudWatch Logs、Amazon GuardDuty 和 AWS Security Hub。

基本做法是在账户级别建立一套检测机制。这套基本机制的目的是记录和检测对您账户中的所有资源执行的多种操作。它们允许您构建全面的检测能力和一些用于添加功能的选项，包括自动修复和合作伙伴集成。

在 AWS 中，可以实施这套基本机制的服务包括：

- AWS CloudTrail 可提供 AWS 账户活动的事件历史记录，包括通过 AWS Management Console、AWS 开发工具包、命令行工具和其他 AWS 服务执行的操作。
- AWS Config 监控和记录您的 AWS 资源配置，并允许您对照所需的配置自动执行评估和修复。
- Amazon GuardDuty 是一种威胁检测服务，可持续监控恶意活动和未经授权的行为，从而保护您的 AWS 账户和工作负载。
- AWS Security Hub 集中聚合、组织和优先处理来自多个 AWS 服务和可选第三方产品的安全警报或调查结果，以使您全面了解安全警报和合规性状态。

在账户级别构建基础时，很多核心 AWS 服务（例如 Amazon Virtual Private Cloud Console（Amazon VPC））提供了服务级别的日志记录功能。Amazon VPC 流日志 可让您捕获有关传入和传出网络接口的 IP 流量的信息，这些信息可提供对于连接历史记录的宝贵见解，并根据异常行为触发自动操作。

对于并非起源于 AWS 服务的 Amazon Elastic Compute Cloud（Amazon EC2）实例和基于应用程序的日志记录，可以使用以下工具来存储和分析日志： Amazon CloudWatch Logs。云 代理 将从正在运行的操作系统和应用程序收集日志，并自动存储这些日志。当这些日志在 CloudWatch Logs 中可用之后，您即可 实时处理它们，或者使用 CloudWatch Logs Insights进行深入分析。

与收集和聚合日志同样重要的是，要能够从复杂的架构生成的大量日志和事件数据中提取有意义的见解。请参阅 《可靠性支柱》白皮书 的 “监控”部分 以获取详细信息。日志自身可能包含敏感数据 – 当应用程序数据以错误的方式进入 CloudWatch Logs 代理捕获的日志文件中，或者为日志聚合功能配置了跨区域日志记录并且在跨境传输某些类型的信息时需要注意一些法律事项时。

一种方法是使用提供日志时在事件上触发的 AWS Lambda 函数，以筛选和编辑日志数据，然后将其转发到中央日志记录位置，例如 Amazon Simple Storage Service（Amazon S3）存储桶。未编辑的日

志可以保留在本地存储桶中，直到合理的时间结束（由法律和您的法律团队决定），届时 Amazon S3 生命周期规则会自动将它们删除。可以在 Amazon S3 中使用 Amazon S3 对象锁定功能进一步保护日志，您可在其中使用“一次写入多次读取”(WORM) 模式来存储对象。

未建立这种最佳实践的情况下暴露的风险等级： 高

实施指导

- 启用 AWS 服务的日志记录：启用 AWS 服务的日志记录以满足您的要求。日志记录功能包括：Amazon VPC 流日志、Elastic Load Balancing（ELB）日志、Amazon S3 存储桶日志、CloudFront 访问日志、Amazon Route 53 查询日志和 Amazon Relational Database Service（Amazon RDS）日志。
  - AWS Answers：原生的 AWS 安全日志记录功能
- 评估并记录特定于操作系统和应用程序的日志，以便检测可疑行为。
  - 开始使用 CloudWatch Logs
  - 开发人员工具和日志分析
- 对日志采取适当的控制：日志中可能包含敏感信息，只有获得授权的用户可以访问。考虑限制对 Amazon S3 存储桶和 CloudWatch Logs 日志组的访问权限。
  - Amazon CloudWatch 的身份验证与访问控制
  - Amazon S3 中的身份与访问管理
- 配置 Amazon GuardDuty：AWS 账户 是一种威胁检测服务，可持续监控恶意活动和未经授权的行为，从而保护您的 GuardDuty 和工作负载。使用实验启用 GuardDuty 并配置通过电子邮件发送的自动化警报。
- 在 CloudTrail 中配置自定义跟踪：配置跟踪可将日志保存比默认时长更长的时间，以便日后进行分析。
- 支持 AWS Config：AWS Config 可以提供 AWS 账户 中的 AWS 资源配置详细信息。其中包括资源彼此之间的关系以及它们以前的配置，让您可以了解配置和关系随时间的变化。
- 支持 AWS Security Hub通过 Security Hub，您可以全面了解自己在 AWS 中的安全状态，帮助您检查是否符合安全行业标准和最佳实践。Security Hub 会收集 AWS 账户、服务和支持的第三方合作伙伴产品的数据，帮助您分析您的安全趋势，并确定最高优先级的安全问题。

资源

相关文档：

- Amazon CloudWatch

- Amazon EventBridge
- 开始使用：Amazon CloudWatch Logs
- 安全合作伙伴解决方案：日志记录和监控

相关视频：

- 集中监控资源配置和合规性
- 修正 Amazon GuardDuty 和 AWS Security Hub 调查结果
- 云中的威胁管理：Amazon GuardDuty 和 AWS Security Hub

相关示例：

- 实验室：自动部署检测性控制

SEC04-BP02 集中分析日志、结果和指标

安全运营团队依靠收集日志和使用搜索工具来发现需要关注的潜在事件，这些事件可能代表未经授权的活动或无意的更改。但是，仅仅分析收集的数据和手动处理信息不足以应对从复杂架构流出的大量信息。单凭分析和报告无法及时分配合适的资源来处理事件。

建立成熟的安全运维团队的最佳实践是，将安全事件和调查结果的流程深度集成到通知和工作流系统中，例如票证系统、错误或问题系统或者其他安全信息和事件管理（SIEM，Security Information and Event Management）系统。这样，工作流可以摆脱电子邮件和静态报告，让您能够路由、上报和管理事件或调查结果。许多组织也在逐步将安全警报集成到他们的聊天或协作以及开发人员工作效率平台中。对于正在踏上自动化之旅的组织，在规划首要自动化任务时，一个由 API 驱动的低延迟票证系统能够提供极高的灵活性。

这种最佳实践不仅适用于从描述用户活动或网络事件的日志消息生成的安全事件，还适用于在基础设施本身检测到的更改生成的安全事件。当面对一些更改，而且这些更改的不受欢迎程度足够微妙，以致于目前无法使用 AWS Identity and Access Management（IAM）和 AWS Organizations 配置的组合来防止这些更改发生时，为了保持和验证安全架构，必须能够检测更改、确定更改是否适当，然后将这些信息路由到正确的修复工作流程。

Amazon GuardDuty 和 AWS Security Hub 为日志记录提供了聚合、重复数据删除和分析机制，您也可以通过其他 AWS 服务提供这些机制。GuardDuty 可从 AWS CloudTrail 管理和数据事件、VPC DNS 日志以及 VPC 流日志等来源提取、聚合和分析信息。Security Hub 能够提取、聚合和分析来自 GuardDuty、AWS Config、Amazon Inspector、Amazon Macie、AWS Firewall Manager 以及 AWS

Marketplace 中提供的大量第三方安全产品的输出，如果您相应构建了自己的代码，还将包括这些代码。GuardDuty 和 Security Hub 都有一个管理员-成员模型，此模型可以跨多个账户聚合调查结果和见解，拥有本地 SIEM 的客户通常将 Security Hub 用作 AWS 端日志和警报预处理器和聚合器，随后即可通过基于 AWS Lambda 的处理器和转发服务器提取 Amazon EventBridge。

未建立这种最佳实践的情况下暴露的风险等级： 高

实施指导

- 评估日志处理功能：评估用于处理日志的选项
  - 使用 Amazon OpenSearch Service 来记录和监控（几乎）所有内容
  - 寻找专门提供日志记录和监控解决方案的合作伙伴
- 作为分析 CloudTrail 日志的开始，请测试 Amazon Athena。
  - 配置 Athena 分析 CloudTrail 日志
- 在 AWS 中实施集中式日志记录：请参阅以下 AWS 示例解决方案来集中处理多个来源的日志记录。
  - 集中日志记录解决方案
- 通过合作伙伴集中处理日志记录：APN 合作伙伴拥有可以帮助您集中分析日志的解决方案。
  - 日志记录和监控

资源

相关文档：

- AWS Answers：集中式日志记录
- AWS Security Hub
- Amazon CloudWatch
- Amazon EventBridge
- 开始使用：Amazon CloudWatch Logs
- 安全合作伙伴解决方案：日志记录和监控

相关视频：

- 集中监控资源配置和合规性
- 修正 Amazon GuardDuty 和 AWS Security Hub 调查结果

- 云中的威胁管理：Amazon GuardDuty 和 AWS Security Hub

SEC04-BP03 自动响应事件

使用自动化流程调查和修复事件可减少人工处理工作量和人为错误，从而扩展调查功能。定期审核将帮助您优化自动化工具，并实现持续迭代。

在 AWS 中，可以使用 Amazon EventBridge，调查感兴趣的事件以及自动化工作流程可能发生的意外变化的相关信息。此服务提供可扩展的规则引擎，可代理原生 AWS 事件格式（例如 AWS CloudTrail 事件）以及您可以从应用程序中生成的自定义事件。Amazon GuardDuty 还允许您将这些事件路由到构建意外事件响应系统（AWS Step Functions）的工作流程系统中，或者路由到中央安全账户或存储桶中以执行进一步分析。

检测更改并将此信息路由到正确的工作流的操作也可以使用 AWS Config 规则 和 合规包完成。AWS Config 会检测对范围内服务的更改（虽然延迟会比 EventBridge 更高），并生成可使用 AWS Config 规则 进行解析的事件，以便进行回滚、强制实施合规性策略以及将信息转发到相关系统（如变更管理平台和运营票证系统）。除了编写您自己的 Lambda 函数以响应 AWS Config 事件，您还可以充分利用 AWS Config 规则 开发工具包以及 一组开源 AWS Config 规则。合规包是 AWS Config 规则 和修复操作的集合，您可将其作为以 YAML 模板格式创作的单个实体进行部署。一个 示例合规包模板， 面向 Well-Architected 安全性支柱提供。

未建立这种最佳实践的情况下暴露的风险等级： 中

实施指导

- 使用 GuardDuty 实施自动化警报：GuardDuty 是一种威胁检测服务，可持续监控恶意活动和未经授权的行为，从而保护您的 AWS 账户和工作负载。启用 GuardDuty 并配置自动化警报。
- 自动执行调查流程：制定自动化流程来调查事件并向管理员报告信息，以便节省时间。
  - 实验室：Amazon GuardDuty 动手实践

资源

相关文档：

- AWS Answers：集中式日志记录
- AWS Security Hub
- Amazon CloudWatch
- Amazon EventBridge

- 开始使用：Amazon CloudWatch Logs
- 安全合作伙伴解决方案：日志记录和监控
- 设置 Amazon GuardDuty

相关视频：

- 集中监控资源配置和合规性
- 修正 Amazon GuardDuty 和 AWS Security Hub 调查结果
- 云中的威胁管理：Amazon GuardDuty 和 AWS Security Hub

相关示例：

- 实验室：自动部署检测性控制

SEC04-BP04 实施可操作的安全事件

创建发送给团队并将由团队处理的警报。确保警报包含团队采取措施所需的相关信息。对于您的每个检测性机制，您还应调查一个以 运行手册 或者 行动手册形式存在的流程。例如，当您启用 Amazon GuardDuty时，它会生成不同的 调查结果。您的每个调查结果类型都应具有一个运行手册条目，例如，如果发现了 特洛伊木马程序， 您的运行手册的简单说明可以指示某个人员进行调查和修复。

未建立这种最佳实践的情况下暴露的风险等级： 低

实施指导

- 发现可用于 AWS 服务的指标：发现可通过 Amazon CloudWatch 用于您正在使用的服务的指标。
  - AWS 服务文档
  - 使用 Amazon CloudWatch 指标
- 配置 Amazon CloudWatch 告警。
  - 使用 Amazon CloudWatch 告警

资源

相关文档：

- Amazon CloudWatch

- Amazon EventBridge
- 安全合作伙伴解决方案：日志记录和监控

相关视频：

- 集中监控资源配置和合规性
- 修正 Amazon GuardDuty 和 AWS Security Hub 调查结果
- 云中的威胁管理：Amazon GuardDuty 和 AWS Security Hub

基础设施保护

问题

- SEC 5 如何保护您的网络资源？
- SEC 6 如何保护计算资源？

SEC 5 如何保护您的网络资源？

任何以某种形式连接至网络的工作负载（互联网或私有网络）都需要多层防御，以帮助防御基于外部和内部网络的威胁。

最佳实践

- SEC05-BP01 创建网络层
- SEC05-BP02 控制所有层的流量
- SEC05-BP03 自动执行网络防护
- SEC05-BP04 实施检查和保护

SEC05-BP01 创建网络层

将具有相同可访问性需求的组件分组为若干层。例如，应将虚拟私有云（VPC）中无需进行互联网访问的数据库集群，放在无法向/从互联网路由的子网中。在不使用 VPC 操作的无服务器工作负载中，使用微服务进行类似的分层和隔离可实现相同目的。

具有相同可访问性要求的组件（例如 Amazon Elastic Compute Cloud（Amazon EC2）实例、Amazon Relational Database Service（Amazon RDS）数据库集群和 AWS Lambda 函数）可细分为由子网构成的层。例如，应将 VPC 中无需进行互联网访问的 Amazon RDS 数据库集群放在无法向/从互联

网路由的子网中。此分层控制方法可减轻单层错误配置的影响，这种错误可能允许意外访问。对于 Lambda，您可以在 VPC 中运行您的函数，以充分利用基于 VPC 的控制。

对于可能包含数千个 VPC、AWS 账户和本地网络的网络连接，您应使用 AWS Transit Gateway。它充当一个枢纽，以控制如何在类似于辐条的所有互联网络之间路由流量。Amazon Virtual Private Cloud 与 AWS Transit Gateway 之间的流量保留在 AWS 私有网络中，可减少外部威胁媒介，例如分布式拒绝服务（DDoS，Distributed Denial of Service）攻击和常见漏洞（SQL 注入、跨站点脚本、跨站点请求伪造或滥用损坏的身份验证代码等等）。AWS Transit Gateway 区域间对等连接也会对区域间流量加密，而且不会出现任何单点故障或带宽瓶颈。

未建立这种最佳实践的情况下暴露的风险等级： 高

实施指导

- 在 VPC 中创建子网：为每层创建子网（在包含多个可用区的组中）并关联路由表以控制路由。
  - VPC 和子网
  - 路由表

资源

相关文档：

- AWS Firewall Manager
- Amazon Inspector
- Amazon VPC 安全性
- 开始使用 AWS WAF

相关视频：

- 用于各种 VPC 的 AWS Transit Gateway 参考架构
- 使用 Amazon CloudFront、AWS WAF 和 AWS Shield 提供应用程序加速和保护

相关示例：

- 实验室：自动部署 VPC

SEC05-BP02 控制所有层的流量

当构建您的网络拓扑时，您应检查每个组件的连接要求。例如，某个组件是否需要互联网可访问性（入站和出站）、连接到 VPC 的能力、边缘服务和外部数据中心。

使用 VPC，您可以使用您设置的私有 IPv4 地址范围或者 AWS 选择的 IPv6 地址范围来定义跨 AWS 区域的网络拓扑。对于入站和出站流量，您应采用深度防御方法应用多种控制，包括使用安全组（状态检测防火墙）、网络 ACL、子网和路由表。在 VPC 中，您可以在可用区中创建子网。每个子网都可以拥有一个关联的路由表，此表定义了用于管理流量在子网内所采用路径的路由规则。您可以将要连接到互联网或 NAT 网关的路由连接到 VPC 或使其经过另一个 VPC，以定义互联网可路由子网。

当在 VPC 内启动某个实例、Amazon Relational Database Service（Amazon RDS）数据库或其他服务时，它的每个网络接口都有自己的安全组。此防火墙位于操作系统层之外，可用于定义允许入站和出站流量的规则。您还可以定义安全组之间的关系。例如，通过参考对相关的实例应用的安全组，数据库层安全组中的实例仅接受来自应用程序层内实例的流量。除非您在使用非 TCP 协议，否则不必在以下情况下允许互联网直接访问 Amazon Elastic Compute Cloud（Amazon EC2）实例（甚至使用安全组禁止使用的端口）：没有负载均衡器或 CloudFront。这样有助于防止通过操作系统或应用程序问题进行意外访问。您还可以为子网附加网络 ACL，它将用作无状态防火墙。您应配置网络 ACL 以缩小各层之间允许的流量范围，但请注意，您需要定义入站和出站规则。

一些 AWS 服务要求组件访问互联网进行 API 调用，其目标是 AWS API 端点 所在的位置。另外一些 AWS 服务使用 VPC 端点 ，这些端点位于您的 Amazon VPC 中。很多 AWS 服务（包括 Amazon S3 和 Amazon DynamoDB）都支持 VPC 端点，并且已在 AWS PrivateLink中广泛使用此技术。我们建议您使用此方法来访问 AWS 服务、第三方服务以及安全地托管在其他 VPC 中您自己的服务。AWS PrivateLink 上的所有网络流量保持在 AWS 骨干网中，永远不会通过互联网。连接只能由服务的使用方启动，不能由服务的提供方启动。为外部服务访问使用 AWS PrivateLink 让您可以创建没有互联网访问的气隙 VPC，帮助您保护 VPC 免受外部威胁因素的影响。第三方服务可以使用 AWS PrivateLink 允许其客户通过私有 IP 地址，从其 VPC 连接到服务。对于需要出站连接到互联网的 VPC 资产，可以让它们通过 AWS 托管的 NAT 网关、仅出站的互联网网关或者您创建并管理的 Web 代理进行仅出站（单向）连接。

未建立这种最佳实践的情况下暴露的风险等级： 高

实施指导

- 控制 VPC 中的网络流量：实施 VPC 最佳实践来控制流量。
  - Amazon VPC 安全性
  - VPC 端点
  - Amazon VPC 安全组

  - 网络 ACL
- 控制边缘站点的流量：实施边缘服务（例如 Amazon CloudFront），以提供一层额外的保护和其他功能。
  - Amazon CloudFront 使用案例
  - AWS Global Accelerator
  - AWS Web Application Firewall（AWS WAF）
  - Amazon Route 53
  - Amazon VPC 传入路由
- 控制私有网络流量：实施保护工作负载专有流量的服务。
  - Amazon VPC 对等连接
  - Amazon VPC 端点服务（AWS PrivateLink）
  - Amazon VPC Transit Gateway
  - AWS Direct Connect
  - AWS Site-to-Site VPN
  - AWS Client VPN
  - Amazon S3 接入点

资源

相关文档：

- AWS Firewall Manager
- Amazon Inspector
- 开始使用 AWS WAF

相关视频：

- 用于各种 VPC 的 AWS Transit Gateway 参考架构
- 使用 Amazon CloudFront、AWS WAF 和 AWS Shield 提供应用程序加速和保护

相关示例：

- 实验室：自动部署 VPC

SEC05-BP03 自动执行网络防护

自动运行保护机制，以提供基于威胁情报和异常检测的自我防御网络。例如可应对最新的威胁并减轻它们的影响的那些入侵检测和预防工具。您可以通过实施 Web 应用程序防火墙来实现自动化的网络保护，例如使用 AWS WAF Security Automations 解决方案（https://github.com/awslabs/aws-waf-security-automations）来自动拦截来自已知威胁媒介相关 IP 地址的请求。

未建立这种最佳实践的情况下暴露的风险等级： 中

实施指导

- 自动执行基于 Web 流量的保护：AWS 提供了使用 AWS CloudFormation 自动部署一组 AWS WAF 规则的解决方案，旨在筛选常见的基于 Web 的攻击。用户可以从预配置的保护功能中进行选择，这些功能定义 AWS WAF Web 访问控制列表（Web ACL）中包含的规则。
  - AWS WAF 安全自动化
- 考虑使用 AWS Partner 解决方案：AWS 合作伙伴提供数百种业界领先的产品，这些产品与您的本地环境中的现有控制措施等效、相同或与之集成。这些产品对现有 AWS 服务起到补充作用，使您能够在云和本地部署环境中部署全面的安全架构，进而实现更无缝的体验。
  - 基础设施安全性

资源

相关文档：

- AWS Firewall Manager
- Amazon Inspector
- Amazon VPC 安全性
- 开始使用 AWS WAF

相关视频：

- 用于各种 VPC 的 AWS Transit Gateway 参考架构
- 使用 Amazon CloudFront、AWS WAF 和 AWS Shield 提供应用程序加速和保护

相关示例：

- 实验室：自动部署 VPC

SEC05-BP04 实施检查和保护

检查和筛选每层的流量。您可以使用 VPC Network Access Analyzer检测 VPC 配置中可能存在的意外访问。您可以指定网络访问需求，然后确定不能满足这些要求的潜在网络路径。对于通过基于 HTTP 的协议处理的组件，Web 应用程序防火墙可帮助防止常见的攻击。 AWS WAF 是一个 Web 应用程序防火墙，可监控和拦截与转发到 Amazon API Gateway API、Amazon CloudFront 或 Application Load Balancer 的可配置规则匹配的 HTTP(s) 请求。要开始使用 AWS WAF，您可以将 AWS 托管式规则 与您自己的规则结合使用，也可以使用现有的 合作伙伴集成。

要管理 AWS WAF、AWS Shield Advanced 保护以及跨 AWS Organizations 的 Amazon VPC 安全组，您可以使用 AWS Firewall Manager。它允许您跨账户和应用程序集中配置和管理防火墙规则，从而更轻松地扩展常见规则的实施。通过使用 AWS Shield Advanced或 能够自动拦截向 您的 Web 应用程序发送非必要请求的解决方案，它还使您能够快速响应攻击。Firewall Manager 也可以与 AWS Network Firewall结合使用。AWS Network Firewall 是一种托管服务，使用规则引擎为您提供对有状态和无状态网络流量的精细控制。它支持 与 Suricata 兼容的 开源入侵防御系统（IPS，Intrusion Prevention System）规范，以便使用规则来保护您的工作负载。

未建立这种最佳实践的情况下暴露的风险等级： 低

实施指导

- 配置 Amazon GuardDuty：GuardDuty 是一种威胁检测服务，可持续监控恶意活动和未经授权的行为，从而保护您的 AWS 账户 和工作负载。启用 GuardDuty 并配置自动化警报。
  - Amazon GuardDuty
  - 实验室：自动部署检测性控制
- 配置虚拟私有云（VPC）流日志：VPC 流日志功能使您能够进一步捕获有关传入和传出 VPC 中网络接口的 IP 流量信息。流日志数据可以发布到 Amazon CloudWatch Logs 和 Amazon Simple Storage Service（Amazon S3）。创建流日志后，您可以在选定目标中检索和查看其数据。
- 考虑使用 VPC 流量径向：流量镜像是一项 Amazon VPC 功能，您可以用它从 Amazon Elastic Compute Cloud（Amazon EC2）实例的弹性网络接口复制网络流量，然后将其发送到带外安全和监控设备，以进行内容检查、威胁监控和故障排除。
  - VPC 流量镜像

资源

相关文档：

- AWS Firewall Manager

- Amazon Inspector
- Amazon VPC 安全性
- 开始使用 AWS WAF

相关视频：

- 用于各种 VPC 的 AWS Transit Gateway 参考架构
- 使用 Amazon CloudFront、AWS WAF 和 AWS Shield 提供应用程序加速和保护

相关示例：

- 实验室：自动部署 VPC

SEC 6 如何保护计算资源？

工作负载内的计算资源需要采用多层防御，才有助于免受内部和外部威胁。计算资源包括 EC2 实例、容器、AWS Lambda 函数、数据库服务、IoT 设备等。

最佳实践

- SEC06-BP01 执行漏洞管理
- SEC06-BP02 缩小攻击面
- SEC06-BP03 实施托管服务
- SEC06-BP04 自动保护计算
- SEC06-BP05 帮助人员远程执行操作
- SEC06-BP06 验证软件完整性

SEC06-BP01 执行漏洞管理

频繁扫描和修补您的代码、依赖项和基础设施中的漏洞，以帮助防御新的威胁。

从计算基础设施的配置开始，您可以使用 AWS CloudFormation 自动创建和更新资源。通过 CloudFormation，您可以使用 AWS 示例或者自行编写，创建以 YAML 或 JSON 格式编写的模板。这样您便可以创建默认安全的基础设施模板，通过 CloudFormation Guard进行验证，从而节省时间并减少配置错误的风险。您可以使用持续交付来构建基础设施并部署应用程序，例如，使用 AWS CodePipeline来自动进行构建、测试和发布。

您负责自己 AWS 资源的补丁管理，包括 Amazon Elastic Compute Cloud（Amazon EC2）实例、亚马逊云机器镜像（AMI）以及众多其他计算资源。对于 Amazon EC2 实例，AWS Systems Manager 使用安全相关的更新和其他类型的更新来自动执行修补托管实例的流程。您可以使用 Patch Manager 为操作系统和应用程序应用修补程序。（在 Windows Server 上，应用程序支持仅限于 Microsoft 应用程序的更新。）您可以使用 Patch Manager 在 Windows 实例上安装服务包，以及在 Linux 实例上执行次要版本升级。您可以按操作系统类型修补 Amazon EC2 实例集或本地服务器和虚拟机队列。这包括 Windows Server、Amazon Linux、Amazon Linux 2、CentOS、Debian Server、Oracle Linux、Red Hat Enterprise Linux（RHEL）、SUSE Linux Enterprise Server（SLES）和 Ubuntu Server 的受支持版本。您可以扫描实例以单独查看缺失补丁的报告，也可以扫描并自动安装所有缺失的补丁。

未建立此最佳实践暴露的风险等级： 高

实施指导

- 配置 Amazon Inspector：Amazon Inspector 测试 Amazon Elastic Compute Cloud（Amazon EC2）实例的网络可访问性，以及在这些实例上运行应用程序的安全状态。Amazon Inspector 评估应用程序的风险、漏洞以及相较于最佳实践的偏差。
  - 什么是 Amazon Inspector？
- 扫描源代码：扫描库和依赖项，以确定是否有漏洞。
  - Amazon CodeGuru
  - OWASP：源代码分析工具

资源

相关文档：

- AWS Systems Manager
- 使用 Amazon EC2 Systems Manager 替换堡垒主机
- AWS Lambda 安全性概述

相关视频：

- 在 Amazon EKS 上运行高安全性工作负载
- 保护无服务器和容器服务
- 有关 Amazon EC2 实例元数据服务的安全最佳实践

相关示例：

- 实验室：自动部署 Web 应用程序防火墙

SEC06-BP02 缩小攻击面

通过强化操作系统，并尽量减少所使用的组件、库和外部可用的服务，缩小暴露在意外访问下的危险。首先减少未使用的组件，无论它们是操作系统程序包、应用程序（适用于基于 Amazon Elastic Compute Cloud（Amazon EC2）的工作负载）还是您代码中的外部软件模块（适用于所有工作负载）。您可以找到许多面向常见的操作系统和服务器软件的强化和安全配置指南。例如，您可以从 互联网安全中心 开始并进行迭代。

在 Amazon EC2 中，您可以创建自己的亚马逊云机器镜像（AMI）并进行修补和强化，以帮助您满足企业的特定安全要求。您应用到 AMI 上的补丁和其他安全控制措施在其创建时生效，它们并非动态的，除非您在启动之后进行了修改，例如，使用 AWS Systems Manager 进行修改。

您可以使用 EC2 Image Builder 简化构建安全 AMI 的过程。EC2 Image Builder 可大幅减少创建和维护黄金镜像所需的工作，无需编写和维护自动化过程。在有软件更新可用时，Image Builder 自动生成新的镜像，无需用户手动迭代镜像工作版本。通过 EC2 Image Builder，您可以使用 AWS 提供的测试和自己的测试，在将镜像部署到生产环境中之前轻松地验证镜像的功能和安全性。您还可以应用 AWS 提供的安全设置来进一步保护自己的镜像，满足内部安全标准。例如，您可以使用 AWS 提供的模板，生成符合安全技术实施指南（STIG，Security Technical Implementation Guide）标准的镜像。

使用第三方静态代码分析工具，您可以确定常见的安全问题，例如未检查的函数输入边界，以及适用的通用漏披露（CVE，Common Vulnerabilities and Exposures）。您可以对所支持的语言使用 Amazon CodeGuru 。您还可以使用第三方依赖关系检查工具，确定代码链接的库是否是最新版本、它们是否不含 CVE，并确保您拥有符合您软件政策要求的许可条件。

使用 Amazon Inspector，您可以针对 CVE，对您的实例执行配置评估、根据安全基准执行评估以及实现缺陷通知自动化。Amazon Inspector 在生产实例或构建管道中运行，它会在发现结果时通知开发人员和工程师。您可以通过编程方式访问调查结果，并将您的团队引导至待办事项和错误跟踪系统。 EC2 Image Builder 可通过自动化修补、AWS 提供的安全策略实施和其他自定义来维护服务器映像 (AMI)。当使用容器时，在您的构建管道中对您的映像存储库定期实施 ECR 映像扫描 ，以便在您的容器中查找 CVE。

尽管 Amazon Inspector 和其他工具能够有效地确定配置和存在的任何 CVE，但也需要使用其他方法在应用程序级别测试您的工作负载。 模糊 是一种众所周知的查错方法，可自动将格式不正确的数据注入到您应用程序的输入字段和其他区域来查错。

未建立此最佳实践暴露的风险等级： 高

实施指导

- 强化操作系统：配置操作系统以符合最佳实践。
  - 保护 Amazon Linux
  - 保护 Microsoft Windows Server
- 强化容器化资源：配置容器化资源以符合安全最佳实践。
- 实施 AWS Lambda 最佳实践。
  - AWS Lambda 最佳实践

资源

相关文档：

- AWS Systems Manager
- 使用 Amazon EC2 Systems Manager 替换堡垒主机
- AWS Lambda 安全性概述

相关视频：

- 在 Amazon EKS 上运行高安全性工作负载
- 保护无服务器和容器服务
- 有关 Amazon EC2 实例元数据服务的安全最佳实践

相关示例：

- 实验室：自动部署 Web 应用程序防火墙

SEC06-BP03 实施托管服务

实施用于托管资源的服务，例如 Amazon Relational Database Service（Amazon RDS）、AWS Lambda 和 Amazon Elastic Container Service（Amazon ECS），以便在责任共担模式中减少安全维护任务。例如，Amazon RDS 可帮助您设置、操作和扩展关系数据库，并自动执行管理任务，例如硬件预置、数据库设置、修补和备份。这意味着您将有更多的空闲时间，因此可以专注于通过 AWS Well-Architected Framework 中所述的其他方法来保护您的应用程序。使用 Lambda，无需使用预置或托管服务器即可运行代码，因此您只需在代码级别专注于连接、调用和安全性，而不是基础设施或操作系统级别。

未建立此最佳实践暴露的风险等级： 中

实施指导

- 探索可用的服务：探索、测试和实施管理资源的服务，例如 Amazon RDS、AWS Lambda 和 Amazon ECS。

资源

相关文档：

- AWS 网站
- AWS Systems Manager
- 使用 Amazon EC2 Systems Manager 替换堡垒主机
- AWS Lambda 安全性概述

相关视频：

- 在 Amazon EKS 上运行高安全性工作负载
- 保护无服务器和容器服务
- 有关 Amazon EC2 实例元数据服务的安全最佳实践

相关示例：

- 实验室：AWS Certificate Manager 请求公有证书

SEC06-BP04 自动保护计算

自动执行计算保护机制，包括管理漏洞、缩小攻击面和管理资源。此自动化将帮助您投入时间以保护工作负载的其他方面，并降低人为犯错的风险。

未建立这种最佳实践的情况下暴露的风险等级： 中

实施指导

- 自动管理配置：使用配置管理服务或工具自动实施安全配置并对其进行验证。
  - AWS Systems Manager

    - AWS CloudFormation
    - 实验室：自动部署 VPC
    - 实验室：自动部署 EC2 Web 应用程序

- 自动修补 Amazon Elastic Compute Cloud（Amazon EC2）实例：AWS Systems Manager Patch Manager 使用安全相关的更新和其他类型的更新来自动执行修补托管实例的流程。您可以使用 Patch Manager 为操作系统和应用程序应用修补程序。
    - AWS Systems Manager 补丁管理器
    - 使用 AWS Systems Manager Automation 集中完成多账户和多区域的修补

- 实施入侵检测和预防：实施入侵检测和预防工具，以监控并停止实例上的恶意活动。
- 考虑使用 AWS Partner 解决方案：AWS 合作伙伴提供数百种业界领先的产品，这些产品与您的本地环境中的现有控制措施等效、相同或与之集成。这些产品对现有 AWS 服务起到补充作用，使您能够在云和本地部署环境中部署全面的安全架构，进而实现更无缝的体验。
    - 基础设施安全性

资源

相关文档：

- AWS CloudFormation
- AWS Systems Manager
- AWS Systems Manager 补丁管理器
- 使用 AWS Systems Manager Automation 集中完成多账户和多区域的修补
- 基础设施安全性
- 使用 Amazon EC2 Systems Manager 替换堡垒主机
- AWS Lambda 安全性概述

相关视频：

- 在 Amazon EKS 上运行高安全性工作负载
- 保护无服务器和容器服务
- 有关 Amazon EC2 实例元数据服务的安全最佳实践

相关示例：

- 实验室：自动部署 Web 应用程序防火墙
- 实验室：自动部署 EC2 Web 应用程序

SEC06-BP05 帮助人员远程执行操作

移除交互式访问功能可降低人为错误的风险以及手动配置或管理的可能性。例如，通过更改管理工作流，使用基础设施即代码部署 Amazon Elastic Compute Cloud（Amazon EC2）实例，然后使用 AWS Systems Manager 等工具管理 Amazon EC2 实例，而不是允许直接访问或通过堡垒主机进行访问。AWS Systems Manager 可以使用 自动化 工作流、文档 （行动手册）和 Run Command等功能自动执行多种维护和部署任务。AWS CloudFormation 堆栈从管道进行构建，并能够自动执行您的基础设施部署和管理任务，而无需直接使用 AWS Management Console或 API。

未建立此最佳实践暴露的风险等级： 低

实施指导

- 替换控制台访问：用 AWS Systems Manager Run Command 替换实例的控制台访问（SSH 或 RDP），以自动管理任务。

- AWS Systems Manager Run Command

资源

相关文档：

- AWS Systems Manager
- AWS Systems Manager Run Command
- 使用 Amazon EC2 Systems Manager 替换堡垒主机
- AWS Lambda 安全性概述

相关视频：

- 在 Amazon EKS 上运行高安全性工作负载
- 保护无服务器和容器服务
- 有关 Amazon EC2 实例元数据服务的安全最佳实践

相关示例：

- 实验室：自动部署 Web 应用程序防火墙

SEC06-BP06 验证软件完整性

实施一些机制（例如代码签名），以确保工作负载中使用的软件、代码和库来自可信的来源且未被篡改。例如，您应验证二进制文件和脚本的代码签名证书以确认作者，并确保证书自作者创建以来未被篡改。AWS Signer 通过集中管理代码签名生命周期，包括签名证书以及公有和私有密钥，帮助确保代码的可信度和完整性。您可以了解如何对 AWS Lambda使用代码签名的高级模式和最佳实践。此外，通过将您下载的软件的校验和与提供商提供的校验和进行对比，可帮助确保它未被篡改。

未建立这种最佳实践的情况下暴露的风险等级： 低

实施指导

- 调查机制：代码签名是一种可用来验证软件完整性的机制。
  - NIST：代码签名的安全注意事项

资源

相关文档：

- AWS Signer
- 新增 – 代码签名，用于 AWS Lambda 的可信度和完整性控制措施

数据保护

问题

- SEC 7 如何对数据进行分类？
- SEC 8 如何保护静态数据？
- SEC 9 如何保护传输中的数据？

SEC 7 如何对数据进行分类？

分类提供了一种基于关键性和敏感度对数据进行分类的方法，以帮助您确定适当的保护和保留控制措施。

最佳实践

- SEC07-BP01 识别工作负载内的数据
- SEC07-BP02 定义数据保护控制措施
- SEC07-BP03 自动识别和分类
- SEC07-BP04 定义数据生命周期管理

SEC07-BP01 识别工作负载内的数据

您需要了解您的工作负载正在处理的数据的类型和分类、相关的业务流程、数据所有者、适用的法律和合规性要求、数据的存储位置以及因此需要实施的控制措施。这可能包括用于指明数据是可公开访问、仅供内部使用（例如客户的个人可识别信息 (PII)）还是受到更加严格的访问限制（例如知识产权、法律特权数据或敏感数据等等）的分类。通过谨慎管理适当的数据分级系统以及每个工作负载的保护要求级别，您可以匹配适用于数据的控制和访问或保护级别。例如，公开内容可供任何人访问，而重要内容则以受保护的方式进行加密和存储，需要授权访问密钥才能解密。

未建立这种最佳实践的情况下暴露的风险等级： 高

实施指导

- 考虑使用 Amazon Macie 发现数据：Macie 可识别敏感数据，例如个人身份信息（PII，Personally Identifiable Information）或知识产权。
  - Amazon Macie

资源

相关文档：

- Amazon Macie
- 数据分类白皮书
- 开始使用 Amazon Macie

相关视频：

- 新 Amazon Macie 简介

SEC07-BP02 定义数据保护控制措施

根据数据分类级别保护数据。例如，使用相关建议保护分类为公共的数据，同时使用其他控制措施保护敏感数据。

通过使用资源标签、根据敏感度（可能还包括限制性条款、飞地或感兴趣的社区）划分 AWS 账户、IAM 策略、AWS Organizations SCP、AWS Key Management Service（AWS KMS）和 AWS CloudHSM，您可以定义并实施您的数据分类和加密保护策略。例如，如果您的项目具有包含极关键数据的 S3 存储桶或者处理机密数据的 Amazon Elastic Compute Cloud（Amazon EC2）实例，则可以使用 `Project=ABC` 标签对其进行标记。只有您的直属团队知道项目代码的含义，它提供了一种使用基于属性的访问控制措施的方法。您可以通过关键策略和授权定义对 AWS KMS 加密密钥的访问级别，以确保只有适当的服务可以通过安全机制访问敏感内容。如果您正在根据标签做出授权决定，您应确保在 AWS Organizations 中使用标签策略适当定义对于标签的权限。

未建立此最佳实践暴露的风险等级： 高

实施指导

- 定义您的数据识别和分类架构：对数据执行标识和分类，用于评估您要存储的数据的潜在影响和类型，并确定谁可以访问数据。
  - AWS 文档
- 发现可用的 AWS 控制措施：对于您正在使用或计划使用的 AWS 服务，发现安全控制措施。许多服务在其文档中都会提供一个安全部分。
  - AWS 文档
- 确定 AWS 合规性资源：确定 AWS 为您提供帮助的资源。
  - https://aws.amazon.com/compliance/

资源

相关文档：

- AWS 文档
- 数据分类白皮书
- 开始使用 Amazon Macie
- 缺少文本

相关视频：

- 新 Amazon Macie 简介

SEC07-BP03 自动识别和分类

自动识别和分类数据可帮助您实施正确的控制措施。在这方面实现自动化而不是允许人员直接访问，可以降低人为犯错和漏洞的风险。您应使用 Amazon Macie等工具执行评估，这些工具使用机器学习来自动发现、分类和保护 Amazon Macie 中的敏感数据。AWS 可以识别个人身份信息（PII，Personally Identifiable Information）或知识产权之类的敏感数据，并为您提供控制面板和警报，让您了解此类数据的访问或移动情况。

未建立此最佳实践暴露的风险等级： 中

实施指导

- 使用 Amazon Simple Storage Service（Amazon S3）清单：Amazon S3 清单是可以用来审核和报告对象的复制和加密状态的工具之一。
  - Amazon S3 清单
- 考虑使用 Amazon Macie：Amazon Macie 使用机器学习来自动发现存储在 Amazon S3 中的数据，并对其进行分类。
  - Amazon Macie

资源

相关文档：

- Amazon Macie
- Amazon S3 清单
- 数据分类白皮书
- 开始使用 Amazon Macie

相关视频：

- 新 Amazon Macie 简介

SEC07-BP04 定义数据生命周期管理

您定义的生命周期策略应基于敏感度级别以及法律和组织要求。应考虑您的数据保留期限、数据销毁流程、数据访问管理、数据转换和数据共享等方面。当选择数据分类方法时，请平衡可用性与访问权限。您还应考虑多种访问级别及其细微差别，以便针对每个级别实施安全且有效的方法。始终采用深度防御方法并减少人工访问数据次数以及数据转换、删除或复制机制。例如，要求用户对应用程序执行严格身份验证，并为应用程序而不是用户授予执行远程操作的必要访问权限。此外，确保用户来自可信网络路径并要求其获取解密密钥。使用控制面板和自动报告等工具为用户提供数据信息，而不是让他们直接访问数据。

未建立这种最佳实践的情况下暴露的风险等级： 低

实施指导

- 识别数据类型：确定您正在工作负载中存储或处理的数据类型。这些数据可以是文本、图像、二进制数据库等。

资源

相关文档：

- 数据分类白皮书
- 开始使用 Amazon Macie

相关视频：

- 新 Amazon Macie 简介

SEC 8 如何保护静态数据？

通过实施多个控制措施来保护静态数据，以降低未经授权的访问或处理不当带来的风险。

最佳实践

- SEC08-BP01 实施安全密钥管理
- SEC08-BP02 强制实施静态加密
- SEC08-BP03 自动执行静态数据保护
- SEC08-BP04 强制实施访问控制

- SEC08-BP05 使用机制限制对数据的访问

SEC08-BP01 实施安全密钥管理

通过定义加密方法，包括密钥存储、轮换和访问控制，有助于您防止内容被未经授权的用户访问或不必要地暴露给经过授权的用户。AWS Key Management Service（AWS KMS）可以帮助您管理加密密钥，并可 与许多 AWS 服务集成。该服务可以为 AWS KMS 密钥提供持久、安全和冗余的存储。您可以定义密钥别名以及密钥级策略。这些策略可以帮助您定义关键管理员以及关键用户。此外，AWS CloudHSM（HSM）是一个基于云的硬件安全模块，使您可以在 AWS Cloud 上轻松生成和使用自己的加密密钥。它使用经 FIPS 140-2 第 3 级验证的 HSM，帮助您满足企业、合同和监管合规性要求，以确保数据安全。

未建立这种最佳实践的情况下暴露的风险等级： 高

实施指导

- 实施 AWS KMS：使用 AWS KMS 可以轻松地创建和管理密钥，并控制在各种 AWS 服务和应用程序中使用加密。AWS KMS 是一项安全且具有弹性的服务，使用经过 FIPS 140-2 验证的硬件安全模块来保护您的密钥。
  - 开始使用：AWS Key Management Service（AWS KMS）
- 考虑使用 AWS 加密开发工具包：当您的应用程序需要加密客户端数据时，使用包含 AWS KMS 集成的 AWS 加密开发工具包。
  - AWS 加密开发工具包

资源

相关文档：

- AWS Key Management Service
- AWS 加密服务和工具
- 开始使用：AWS Key Management Service（AWS KMS）
- 利用加密保护 Amazon S3 数据

相关视频：

- AWS 中的加密原理
- 在 AWS 上保护您的数据块存储

SEC08-BP02 强制实施静态加密

您应确保只以加密的方式存储数据。AWS Key Management Service（AWS KMS）与大量 AWS 服务无缝集成，使您能够更轻松地加密所有静态数据。例如，在 Amazon Simple Storage Service（Amazon S3）中，您可以对存储桶设置 默认加密 ，以自动加密所有的新对象。此外，Amazon Elastic Compute Cloud (Amazon EC2) 和 Amazon S3 支持通过设置默认加密来强制加密。您可以使用 AWS Config 规则 自动检查您已使用了加密，例如针对 Amazon Elastic Block Store（Amazon EBS）卷、 Amazon Relational Database Service（Amazon RDS）实例和 Amazon S3 存储桶。

未建立这种最佳实践的情况下暴露的风险等级： 高

实施指导

- 对 Amazon Simple Storage Service（Amazon S3）强制实施静态加密：实施 Amazon S3 存储桶默认加密。
  - 如何为 S3 存储桶启用默认加密？
- 使用 AWS Secrets Manager：Secrets Manager 是一项 AWS 服务，让您能够轻松地管理密钥。密钥可以是数据库凭证、密码、第三方 API 密钥甚至任意文本。
  - AWS Secrets Manager
- 为新的 EBS 卷配置默认加密：指定所有新创建的 EBS 卷要以加密形式创建，并选择使用 AWS 提供的默认密钥或您创建的密钥。
  - EBS 卷的默认加密
- 配置加密亚马逊云机器镜像（AMI）：通过复制启用加密功能的现有 AMI，可自动加密根卷和快照。
  - 有加密快照的 AMI
- 配置 Amazon Relational Database Service（Amazon RDS）加密：通过启用加密选项，配置对您的 Amazon RDS 数据库集群和静态快照的加密。
  - 加密 Amazon RDS 资源
- 在其他 AWS 服务中配置加密：对于您使用的 AWS 服务，请确定加密功能。
  - AWS 文档

资源

相关文档：

- 有加密快照的 AMI

- AWS 加密工具
- AWS 文档
- AWS 加密开发工具包
- AWS KMS 加密详情白皮书
- AWS Key Management Service
- AWS Secrets Manager
- AWS 加密服务和工具
- Amazon EBS 加密
- EBS 卷的默认加密
- 加密 Amazon RDS 资源
- 如何为 S3 存储桶启用默认加密？
- 利用加密保护 Amazon S3 数据

相关视频：

- AWS 中的加密原理
- 在 AWS 上保护您的数据块存储

SEC08-BP03 自动执行静态数据保护

利用自动化工具持续验证和实施静态数据控制措施，例如，确保只存在经过加密的存储资源。您可以自动确认所有 EBS 卷都已经过加密，方法是 使用 AWS Config 规则。AWS Security Hub 还可以按照安全标准执行自动化检查，以验证多种不同的控制措施。此外，您的 AWS Config 规则可以自动 修复不合规的资源。

未建立此最佳实践暴露的风险等级： 中

实施指导

静态数据 代表您在工作负载期间的任意时间段内保留在非易失性存储器中的任何数据。其中包括数据块存储、对象存储、数据库、存档、IoT 设备和用来保留数据的任何其他存储介质。在实施了加密和适当的访问控制时，保护静态数据可以降低未经授权访问的风险。

强制实施静态加密：您应确保只以加密的方式存储数据。AWS KMS 与很多 AWS 服务无缝集成，使您能够更轻松地加密所有静态数据。例如，在 Amazon Simple Storage Service（Amazon S3）中，您

可以对存储桶设置 默认加密 ，以自动加密所有的新对象。此外，Amazon EC2 和 Amazon S3 支持通过设置默认加密来强制加密。您可以使用 AWS 托管 Config 规则 自动检查您已使用了加密，例如针对 EBS 卷、Amazon Relational Database Service（Amazon RDS）实例和 Amazon S3 存储桶。

资源

相关文档：

- AWS 加密工具
- AWS 加密开发工具包

相关视频：

- AWS 中的加密原理
- 在 AWS 上保护您的数据块存储

SEC08-BP04 强制实施访问控制

强制实施包含最低权限和机制的访问控制，包括备份、隔离和版本控制，以帮助保护您的静态数据。防止操作员授予对您的数据的公共访问权限。

各种控制措施，包括访问权限（使用最低权限）、备份（请参阅 可靠性白皮书）、分离和版本控制，都可以帮助保护您的静态数据。您应使用本白皮书中前面介绍的检测性机制（包括 CloudTrail）和服务级别日志（例如 Amazon Simple Storage Service（Amazon S3）访问日志），审计对您的数据进行的访问。您应清点可公开访问的数据，并计划如何随着时间的推移减少可用的数据量。Amazon S3 Glacier 文件库锁定和 Amazon S3 对象锁定这两项功能可提供强制访问控制 – 利用合规性选项锁定文件库策略之后，在锁定过期之前，即使根用户也无法对其进行更改。此机制符合 SEC、CFTC 和 FINRA 的图书和记录管理要求。有关更多详细信息，请参阅 本白皮书。

未建立这种最佳实践的情况下暴露的风险等级： 低

实施指导

- 强制实施访问控制：强制实施最低权限访问控制，包括对加密密钥的访问。
  - 管理对 Amazon S3 资源的访问权限简介
- 根据不同分类级别隔离数据：针对 AWS Organizations 管理的数据分类级别使用不同的 AWS 账户。
  - AWS Organizations

- 查看 AWS KMS 策略：查看 AWS KMS 策略中授予的访问级别。
  - 管理对 AWS KMS 资源的访问权限概览
- 查看 Amazon S3 存储桶和对象权限：定期查看 Amazon S3 存储桶策略中授予的访问级别。最佳做法是配置不可公开读取或写入的存储桶。考虑使用 AWS Config 检测可公开访问的存储桶，并使用 Amazon CloudFront 提供 Amazon S3 中的内容。
  - AWS Config 规则
  - Amazon S3 + Amazon CloudFront：云中的天作之合
- 启用 Amazon S3 版本控制和对象锁定。
  - 使用版本控制
  - 使用 Amazon S3 对象锁定来锁定对象
- 使用 Amazon S3 清单：Amazon S3 清单是可以用来审核和报告对象的复制和加密状态的工具之一。
  - Amazon S3 清单
- 查看 Amazon EBS 和 AMI 共享权限：共享权限允许将镜像和卷共享到您的工作负载外部的 AWS 账户。
  - 共享 Amazon EBS 快照
  - 共享 AMI

资源

相关文档：

- AWS KMS 加密详情白皮书

相关视频：

- 在 AWS 上保护您的数据块存储

SEC08-BP05 使用机制限制对数据的访问

禁止所有用户直接访问正常运行环境中的敏感数据和系统。例如，利用变更管理工作流程，借助工具管理 Amazon Elastic Compute Cloud（Amazon EC2）实例，而不是允许直接访问或通过堡垒主机进行访问。这可以使用 AWS Systems Manager Automation来实现，此功能将使用 包含您的任务执行步骤的 自动化文档。这些文档可以存储在源代码控制中、在运行之前接受对等审核，并接受全面测试以便最大程度降低风险（与 shell 访问相比）。企业用户可以使用一个仪表板而不是通过直接访问数据存

储库来执行查询。当未使用 CI/CD 管道时，确定需要利用哪些控制措施和流程来充分提供通常禁用的 Break Glass 访问机制。

未建立这种最佳实践的情况下暴露的风险等级： 低

实施指导

- 实施可限制对数据的访问的机制：这些机制包括使用控制面板（例如 Amazon QuickSight），以向用户显示数据，而不是直接查询。
  - Amazon QuickSight
- 自动管理配置：远程执行操作，使用配置管理服务或工具自动实施安全配置并对其进行验证。避免使用堡垒主机或直接访问 EC2 实例。
  - AWS Systems Manager
  - AWS CloudFormation
  - AWS 上适用于 AWS CloudFormation 模板的 CI/CD 管道

资源

相关文档：

- AWS KMS 加密详情白皮书

相关视频：

- AWS 中的加密原理
- 在 AWS 上保护您的数据块存储

SEC 9 如何保护传输中的数据？

通过实施多个控制措施来保护传输中的数据，以降低未经授权的访问或数据丢失所带来的风险。

最佳实践

- SEC09-BP01 实施安全密钥和证书管理
- SEC09-BP02 执行传输中加密
- SEC09-BP03 自动检测意外数据访问
- SEC09-BP04 对网络通信进行身份验证

SEC09-BP01 实施安全密钥和证书管理

安全地存储加密密钥和证书，并按照适当的时间间隔和使用严格的访问控制措施来轮换这些密钥和证书。实现这一目的的最佳方法是使用托管服务，例如 AWS Certificate Manager（ACM）。它能够让您轻松预置、管理和部署公有和私有传输层安全性 (TLS) 证书，以便与 AWS 服务和您的内部互联资源配合使用。TLS 证书用于保障网络通信的安全性、确立网站在互联网上的身份和资源在私有网络上的身份。ACM 与 Elastic Load Balancer（ELB）、AWS 分配以及 API Gateway 上的 API 等 AWS 资源集成，还负责处理自动证书续订事宜。如果您使用 ACM 来部署私有根 CA，则它可以提供要在 Amazon Elastic Compute Cloud（Amazon EC2）实例、容器等对象中使用的证书和私有密钥。

未建立这种最佳实践的情况下暴露的风险等级： 高

实施指导

- 实施安全密钥和证书管理：实施您定义的安全密钥和证书管理解决方案。
  - AWS Certificate Manager
  - 如何在 AWS 中托管和管理整个私有证书基础设施
- 实施安全协议：使用传输层安全性协议（TLS，Transport Layer Security）或 IPsec 等支持身份验证和机密性的协议，以便减少数据篡改或丢失的风险。查看 AWS 文档，了解与您正在使用的服务相关的协议和安全性信息。

资源

相关文档：

- AWS 文档

SEC09-BP02 执行传输中加密

根据相应的标准和建议，实施您定义的加密要求，以帮助您满足组织、法律和合规性要求。AWS 服务提供使用 TLS 的 HTTPS 端点进行通信，从而可以在与 AWS API 通信时提供传输中加密。可以使用安全组在 VPC 中审计和拦截不安全的协议，例如 HTTP。HTTP 请求也可以 自动重定向到 Amazon CloudFront 中的 HTTPS 或 应用程序负载均衡器。您可以完全控制计算资源，以便在整个服务中实施加密。您也可以利用 VPN 连接从外部网络连接到您的 VPC 中，以便于对流量进行加密。如果您有特殊要求，可以使用 AWS Marketplace 中提供的第三方解决方案。

未建立这种最佳实践的情况下暴露的风险等级： 高

实施指导

- 实施传输中加密：您定义的加密要求应基于最新的标准和最佳实践，且仅允许使用安全协议。例如，仅配置一个安全组，以允许通过 HTTPS 协议访问应用程序负载均衡器或 Amazon Elastic Compute Cloud（Amazon EC2）实例。
- 在边缘服务中配置安全协议：使用 Amazon CloudFront 和要求的密码来配置 HTTPS。
  - 将 HTTPS 与 CloudFront 搭配使用
- 将 VPN 用于外部连接：考虑使用 IPsec 虚拟专用网络（VPN）来保护点对点或网络对网络连接，以实现数据隐私性和完整性。
  - VPN 连接
- 在负载均衡器中配置安全协议：使用 HTTPS 侦听器来保护指向负载均衡器的连接。
  - 适用于应用程序负载均衡器的 HTTPS 侦听器
- 为实例配置安全协议：考虑在实例上配置 HTTPS 加密。
  - 教程：将 Amazon Linux 2 上的 Apache Web 服务器配置为使用 SSL/TLS
- 在 Amazon Relational Database Service（Amazon RDS）中配置安全协议：使用安全套接字层（SSL，Secure Socket Layer）或传输层安全性协议（TLS，Transport Layer Security）来加密与数据库实例的连接。
  - 使用 SSL 加密与数据库实例的连接
- 在 Amazon Redshift 中配置安全协议：将集群配置为要求安全套接字层（SSL，Secure Socket Layer）或传输层安全性协议（TLS，Transport Layer Security）连接。
  - 针对连接配置安全选项
- 在其他 AWS 服务中配置安全协议：对于您使用的 AWS 服务，请确定传输中加密功能。

资源

相关文档：

- AWS 文档

SEC09-BP03 自动检测意外数据访问

使用 Amazon GuardDuty 等工具自动检测可疑活动或尝试将数据移动到定义的边界之外。例如，GuardDuty 可以通过以下方法，检测异常的 Amazon Simple Storage Service（Amazon S3）读取活动： Exfiltration:S3/AnomalousBehavior finding。除了 GuardDuty 以外，还可以将Amazon VPC 流日志（用于捕获网络流量信息）与 Amazon EventBridge 配合使用，以触发对已成功和被拒绝的异常连

接的检测。Amazon S3 Access Analyzer 可以帮助评估您的 Amazon S3 存储桶中的哪些数据可供哪些人访问。

未建立此最佳实践暴露的风险等级： 中

实施指导

- 自动检测意外数据访问：使用工具或检测机制自动检测试图将数据移出定义边界的行为；例如，检测正在将数据复制到无法识别的主机的数据库系统。
  - VPC 流日志
- 考虑 Amazon Macie：Amazon Macie 是一项完全托管式数据安全和数据隐私服务，该服务使用机器学习和模式匹配发现和保护 AWS 中的敏感数据。
  - Amazon Macie

资源

相关文档：

- VPC 流日志
- Amazon Macie

SEC09-BP04 对网络通信进行身份验证

使用传输层安全性 (TLS) 或 IPsec 等支持身份验证的协议来验证通信的身份。

使用支持身份验证的网络协议，可以在双方之间建立信任。此功能将增强协议中使用的加密方法，以降低通信被篡改或拦截的风险。实施身份验证时常用的协议包括（很多 AWS 服务中使用的）传输层安全性（TLS）和（ AWS Virtual Private Network（AWS VPN）中使用的）IPsec。

未建立这种最佳实践的情况下暴露的风险等级： 低

实施指导

- 实施安全协议：使用 TLS 或 IPsec 等支持身份验证和机密性的安全协议，以便减少数据篡改或丢失的风险。查看 AWS 文档， 了解与您正在使用的服务相关的协议和安全性。

资源

相关文档：

- AWS 文档

事件响应

问题

- SEC 10 如何预测、响应事件以及从事件中恢复？

SEC 10 如何预测、响应事件以及从事件中恢复？

准备工作对于及时有效地调查、响应安全事件以及从安全事件中恢复至关重要，可以尽可能减少对组织的破坏。

最佳实践

- SEC10-BP01 确定关键人员和外部资源
- SEC10-BP02 制定事件管理计划
- SEC10-BP03 准备取证能力
- SEC10-BP04 自动控制功能
- SEC10-BP05 预置访问权限
- SEC10-BP06 预先部署工具
- SEC10-BP07 执行实际试用

SEC10-BP01 确定关键人员和外部资源

确定能够帮助您的组织响应事件的内部和外部人员、资源、以及法律义务。

当您与其他团队（例如法律顾问、领导、业务利益相关者、AWS Support 服务等）一起在云中定义您的事件响应方法时，您必须确定关键人员、利益相关者和相关联系人。为了降低依赖性并缩短响应时间，请确保为您的团队、专家安全团队和响应者开展培训，以使他们了解您使用的服务并有机会练习动手实践。

我们鼓励您寻找外部 AWS 安全合作伙伴，他们应当能够为您带来外部专业知识和不同的视角，以增强您的响应能力。您的可靠安全合作伙伴可以帮助您发现您可能并不熟悉的潜在风险或威胁。

未建立此最佳实践暴露的风险等级： 高

实施指导

- 确定组织中的关键人员：制定联系人列表，列出组织内需要参与意外事件响应和恢复的人员。
- 确定外部合作伙伴：根据需要联系能够帮助您响应意外事件并从中恢复的外部合作伙伴。

资源

相关文档：

- AWS 意外事件响应指南

相关视频：

- 准备和响应 AWS 环境中的安全意外事件

相关示例：

SEC10-BP02 制定事件管理计划

制定计划，以帮助您响应事件、在事件期间进行沟通以及从事件中恢复。例如，您可以根据您的工作负载和组织中最可能遇到的情况开始制定事件响应计划。在计划中说明如何在内部和外部进行沟通和上报。

未建立这种最佳实践的情况下暴露的风险等级： 高

实施指导

对于响应、缓解安全事件的潜在影响并从中恢复来说，事件管理计划是至关重要的。事件管理计划是一个结构化的过程，用于及时地确定、补救和响应安全事件。

云的许多操作角色和要求都与本地环境中的相同。在创建事件管理计划时，应考虑最符合业务成果和合规性要求的响应和恢复策略，这一点非常重要。例如，如果您在 AWS 中运行在美国符合 FedRAMP 标准的工作负载，则遵守《NIST SP 800-61 计算机安全处理指南》会很有帮助。同样，在使用欧洲 PII（Personally Identifiable Information，个人身份信息）数据运行工作负载时，请考虑如何根据 欧盟通用数据保护条例（GDPR，General Data Protection Regulation）。

在为 AWS 中运行的工作负载构建事件管理计划时，请首先使用 AWS 责任共担模式，以便构建针对事件响应的深度防御方法。在此模式中，AWS 负责管理云本身的安全，云内部的安全则由您负责。这意味着您将保留控制权，并对选择实施的安全控制机制负责。此 AWS 安全事件响应指南 详细介绍了构建以云为中心的事件管理计划的关键概念和基本指南。

必须不断地迭代有效的事件管理计划，使其与您的云运营目标保持一致。在创建和改进事件管理计划时，请考虑使用下面详述的实施计划。

- 针对事件响应进行的培训和训练： 当与定义的基线存在偏差（例如，部署错误或配置错误）时，您可能需要做出响应并展开调查。要成功做到这一点，您必须了解可用于 AWS 环境中的安全事件响应的控制机制和功能，以及准备、培训和训练参与事件响应的云团队时要考虑的流程。
  - 行动手册 和 运行手册 是有效机制，用于在训练如何应对事件时建立一致性。首先，在事件响应期间构建经常运行的过程的初始列表，并在学习或使用新过程时继续迭代。
  - 通过计划的 实际试用，将行动手册和运行手册社交化。在实际试用期间，在受控环境中模拟事件响应，帮助您的团队回顾如何进行响应，以及验证参与事件响应的团队是否熟悉工作流。审查模拟事件的结果以明确改进措施，并确定有关进一步的训练或其他工具的需求。
  - 在安全方面，人人有责。通过让通常运行工作负载的所有人员参与进来，建立事件管理流程的集体性知识。这项工作应该涵盖业务的所有方面，即运营、测试、开发、安全性、业务运营和业务部门的负责人均应参与其中。
- 将事件管理计划归档： 将一些工具和过程归档，而这些工具和过程用于记录、处理、传达进度，以及提供有关活动事件的通知。事件管理计划旨在确保尽快恢复正常操作、将业务影响降至最低并始终通知所有相关方。事件示例包括（但不限于）网络连接丢失或降级、进程或 API 无响应、计划的任务未执行（例如，修补失败）、应用程序数据或服务不可用、因安全事件导致计划外服务中断、凭证泄露或错误配置。
  - 确定负责解决事件的主要所有者，例如工作负载所有者。清楚地指明负责处理事件的人员以及沟通方式。当有多方（例如外部供应商）参与事件解决过程时，请考虑构建 责任（RACI）矩阵，详细说明参与事件解决过程的各个团队或人员的角色和职责。

    RACI 矩阵详述了以下各方的职责：
    - R： 责任（Responsible） 方，负责完成任务。
    - A： 负责（Accountable） 方或利益相关者，负责最终裁定是否已成功完成特定任务。
    - C： 咨询（Consulted） 方，将向其征求意见，通常是主题专家。
    - I： 告知（Informed） 方，将告知其进度，通常仅在任务完成或有可交付结果时才告知。
- 对事件进行分类： 通过根据严重性和影响分数来定义事件并为其分类，可以使用结构化方法来分类和解决事件。以下建议说明了一个用于量化事件的 影响-解决紧急矩阵 。例如，影响小且紧急程度低的事件被视为严重性较低的事件。
  - 高（H）： 您的业务将受到重大影响。与 AWS 资源相关的应用程序的关键功能不可用。它们将被预留给对生产系统影响最大的事件。由于补救具有时效性，事件的影响会迅速扩大。
  - 中（M）： 与 AWS 资源相关的商业服务或应用程序会受到一定程度的影响，并且处于降级状态。有助于实现服务等级目标（SLO，Service Level Objective）的应用程序在服务等级协议

（SLA，Service Level Agreement）限制内受到影响。系统可以在性能降低的情况下运行，而不会对财务和声誉产生很大影响。

  - 低（L）：与 AWS 资源相关的商业服务或应用程序的非关键功能受到影响。系统可以在性能降低的情况下运行，对财务和声誉产生的影响极小。

- 使安全控制机制标准化：使安全控制机制标准化的目标是实现操作结果的一致性、可追溯性和可重复性。加快实施对事件响应至关重要的关键活动的标准化，例如：
  - 身份和访问权限管理：建立用于控制对数据的访问以及管理人类和机器身份的权限的机制。将您自己的身份和访问权限管理扩展到云，并使用具有单点登录和基于角色的权限的联合安全性来优化访问权限管理。有关标准化访问权限管理的最佳实践建议和改进计划，请参阅《安全性支柱》白皮书的“身份和访问权限管理”部分。
  - 漏洞管理：建立机制来识别 AWS 环境中的漏洞，攻击者可能会利用这些漏洞来破坏和滥用您的系统。将预防性和检测性控制机制用作安全机制，以响应安全事件并缓解安全事件可能带来的影响。将威胁建模等流程标准化，使之成为基础设施构建和应用程序交付生命周期的一部分。
  - 配置管理：对于在 AWS Cloud 中部署资源，定义标准配置，并使其过程实现自动化。通过实施基础设施和资源预置的标准化，有助于降低因错误部署或意外的人为错误配置而带来的错误配置风险。请参阅《卓越运营支柱》白皮书的“设计原则”部分，以获取实施此控制机制的指南和改进计划。
  - 用于审计控制机制的日志记录和监控：实施一些机制来监控资源的故障、性能下降和安全问题。通过使这些控制机制标准化，还对系统中发生的活动进行审计跟踪，有助于及时对问题进行分类和补救。SEC04（“您如何检测和调查安全事件？”）下的最佳实践提供了有关实施此控制机制的指南。
- 使用自动化：利用自动化，可以及时地大规模解决事件。AWS 提供了多种服务，以在事件响应策略的上下文中实施自动化。请专注于在自动化和人工干预之间找到适当的平衡。您在行动手册和运行手册中构建事件响应时，系统会自动执行可重复的步骤。使用 AWS Systems Manager Incident Manager 等 AWS 服务更快地解决 IT 事件。使用开发人员工具提供版本控制，并自动实施 Amazon Machine Images (AMI) 和基础设施即代码（IaC）部署，而无需人工干预。在适用的情况下，使用 Amazon GuardDuty、Amazon Inspector、AWS Security Hub、AWS Config 和 Amazon Macie 等托管服务，来自动实施检测和合规性评估。使用 Amazon DevOps Guru 等机器学习服务来优化检测功能，在异常操作模式问题出现之前检测出它们。
- 进行根本原因分析并汲取经验教训：实施一些机制，以在事件后响应审查过程中记录经验教训。当事件的根本原因揭示出更大的缺陷、设计缺陷、配置错误或再次发生的可能性时，它就会被归类为问题。在这种情况下，请分析并解决问题，最大程度地减小对正常操作的中断。

资源

相关文档：

- AWS 安全事件响应指南
- NIST：计算机安全事件处理指南

相关视频：

- 在 AWS 中自动化事件响应和取证
- 运行手册、事件报告和事件响应 DIY 指南
- 准备和响应 AWS 环境中的安全事件

相关示例：

- 实验室：使用 Jupyter 的事件响应行动手册 – AWS IAM
- 实验：使用 AWS 控制台和 CLI 的事件响应

SEC10-BP03 准备取证能力

对于意外事件响应者来说，了解取证调查在什么情况下、如何适合您的响应计划非常重要。您的组织应定义要收集的证据以及收集过程中使用的工具。确定并准备适当的取证调查能力，包括外部专家、工具和自动化。您应预先做出的一个关键决定是，是否从实时系统中收集数据。如果系统断电或重新启动，某些数据（例如，易失性内存或活动网络连接的内容）将会丢失。

您的响应团队可以结合使用 AWS Systems Manager、Amazon EventBridge 和 AWS Lambda 等工具，在操作系统和 VPC 流量镜像中自动运行取证工具以捕获网络数据包，从而收集非持久化证据。使用定制取证工作站以及可供响应者访问的工具，在专用安全账户中进行其他活动，例如日志分析或分析磁盘镜像。

定期将相关日志发送到数据存储，实现高持久性和完整性。响应者应有权访问这些日志。AWS 提供了多种工具来简化日志调查，例如 Amazon Athena、Amazon OpenSearch Service（OpenSearch Service）和 Amazon CloudWatch Logs Insights。此外，使用 Amazon Simple Storage Service（Amazon S3）Object Lock 安全地保留证据。该服务遵循 WORM（一次写入多次读取，write-once-read-many）模型，防止对象在定义的时段内被删除或覆盖。由于取证调查技术需要专家培训，因此您可能需要聘请外部专家。

未建立这种最佳实践的情况下暴露的风险等级： 中

实施指导

- 确定取证能力：分析组织的取证调查能力、可用工具和外部专家。
- 自动化事件响应和取证

资源

相关文档：

- 如何在 AWS 中自动实施取证磁盘收集

SEC10-BP04 自动控制功能

自动控制意外事件并从意外事件中恢复，以缩短响应时间和减少对组织的影响。

当您从行动手册中创建并练习流程和工具之后，您可以将此逻辑解构到基于代码的解决方案中，很多响应者可以将此逻辑用作工具来自动进行响应，因此消除了响应者的分歧或猜测。这样可以加快响应的生命周期。下一个目标是允许此代码被警报或事件自身而不是被人类响应者调用以实现完全自动化，从而创建由事件驱动的响应。这些过程还应自动将相关数据添加到您的安全系统中。例如，涉及来自不需要的 IP 地址的流量的意外事件可以自动填充 AWS WAF 阻止列表或 Network Firewall 规则组，从而防止进一步的活动。

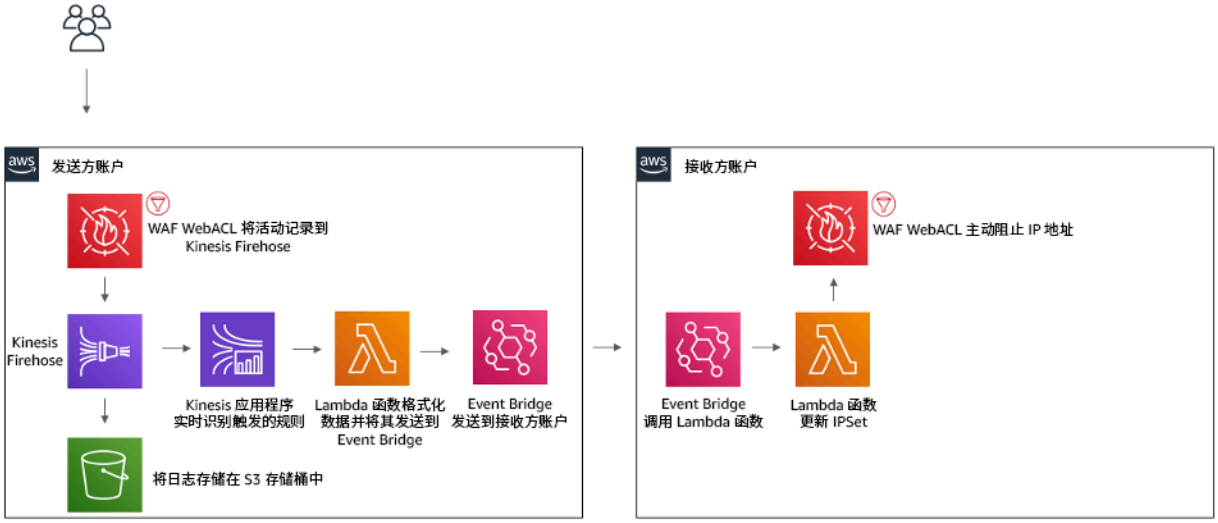


图 3：AWS WAF 自动阻止已知的恶意 IP 地址。

使用由事件驱动的响应系统，检测性机制会触发一个响应机制，以自动修复事件。您可以使用由事件驱动的响应能力，以缩短检测机制与响应机制之间的价值实现时间。要创建这个由事件驱动的架构，您可以使用 AWS Lambda，这是一项无服务器计算服务，可运行您的代码以响应事件并为您自

动管理底层计算资源。例如，假设您有一个 AWS 账户并为其启用了 AWS CloudTrail 服务。如果已禁用 AWS CloudTrail（通过 `cloudtrail:StopLogging` API 调用），则您可以使用 Amazon EventBridge 监控特定的 `cloudtrail:StopLogging` 事件，并通过调用 AWS Lambda 函数来调用 `cloudtrail:StartLogging` 以重新启动日志记录功能。

未建立这种最佳实践的情况下暴露的风险等级： 中

实施指导

自动控制功能。

资源

相关文档：

- AWS 意外事件响应指南

相关视频：

- 准备和响应 AWS 环境中的安全意外事件

SEC10-BP05 预置访问权限

确保事件响应者将正确的访问权限预置到 AWS 中，以缩短调查到恢复所需的时间。

常见反模式：

- 使用根账户进行事件响应。
- 变更现有用户账户。
- 在提供实时权限提升时直接操作 IAM 权限。

未建立这种最佳实践的情况下暴露的风险等级： 中

实施指导

AWS 建议尽可能减少或消除对长期有效凭证的依赖，转而使用临时凭证和 实时 权限提升机制。长期有效的凭证容易带来安全风险，并且会增加运营开销。对于大多数管理任务以及事件响应任务，建议您对管理访问实施 身份联合验证 以及 临时上报。在此模型中，用户请求提升到更高级别的权限（例如事件响应角色），如果用户符合提升条件，则会向审批者发送请求。如果请求获得批准，用户将收到一组临时的 AWS 凭证 ，可用于完成用户任务。在这些凭证过期后，用户必须提交新的提升请求。

在大多数事件响应场景中，建议使用临时权限提升。执行此操作的正确方法是使用 AWS Security Token Service 和 会话策略 来限定访问范围。

在一些场景中，联合身份不可用，例如：

- 与被盗用的身份提供者（IdP）相关的中断。
- 导致联合访问管理系统损坏的错误配置或人为错误。
- 恶意活动，例如分布式拒绝服务（DDoS，Distributed Denial of Service）事件或导致系统不可用的活动。

在上述情况下，应配置紧急 Break Glass 访问，以允许调查事件并及时给予补救。我们建议您使用 具有适当权限的 IAM 用户，来执行任务和访问 AWS 资源。请仅将根凭证用于 需要根用户访问权限的任务。要确认事件响应者对 AWS 和其他相关系统是否具有正确的访问权限级别，建议预置专用的用户账户。用户账户需要特许的访问权限，并且必须受到严格的控制和监视。在构建账户时，必须使用执行必要任务所需的最少权限，并且访问级别应基于作为事件管理计划的一部分创建的行动手册。

最好使用专门构建的专用用户和角色。通过添加 IAM 策略来临时提升用户或角色的访问权限，既会导致无法清楚地了解用户在事件期间拥有哪些访问权限，又会带来无法撤销提升的权限的风险。

请务必删除尽可能多的依赖项，以确保能在尽可能多的故障场景中获得访问权限。为了支持此操作，可创建一个行动手册，验证是否在专用的安全账户中创建事件响应用户作为 AWS Identity and Access Management 用户，而不是通过任何现有的联合身份验证或单点登录（SSO）解决方案管理他们。每个响应者都必须有自己的指定账户。账户配置必须实施 强密码策略 和多重身份验证（MFA）。如果事件响应行动手册仅需要对 AWS Management Console的访问权限，则用户不应配置访问密钥，并且应明确禁止用户创建访问密钥。可以使用 IAM 策略或服务控制策略（SCP，Service Control Policy）进行此配置，如 AWS 安全最佳实践（适用于 AWS Organizations SCP）中所述。用户仅能够在其他账户中代入事件响应角色，而不应具有其他任何权限。

在事件处理期间，可能需要向其他内部或外部人员授予访问权限，以支持调查、补救或恢复活动。在这种情况下，可以使用前面提到的行动手册机制，并且必须创建一个流程，确保在事件结束后立即撤消其他任何访问权限。

要确保能正确地监控和审计对事件响应角色的使用，至关重要的一点是，为此目的创建的 IAM 用户账户不会在人员之间共享，并且不会使用 AWS 账户 根用户，除非 特定任务要求这样做。如果需要根用户（例如，对特定账户的 IAM 访问权限不可用），请使用单独的流程和可用的行动手册来验证根用户密码和 MFA 令牌的可用性。

要为事件响应角色配置 IAM 策略，请考虑使用 IAM Access Analyzer 来生成基于 AWS CloudTrail 日志的策略。为此，请在非生产账户中向事件响应角色授予管理员访问权限，并运行行动手册。完成后，会

创建一个策略，仅允许已执行的操作。之后，可以跨所有账户将此策略应用于所有事件响应角色。您可能希望为每个行动手册创建一个单独的 IAM 策略，以便更轻松地进行管理和审计。示例行动手册可能包括针对勒索软件、数据泄露、丢失生产访问权限和其他场景的响应计划。

使用事件响应用户账户可在 其他 AWS 账户 中代入专用的事件响应 IAM 角色。必须将这些角色配置为仅可由安全账户中的用户代入，并且信任关系必须要求调用主体已使用 MFA 进行身份验证。角色必须使用严格界定的 IAM 策略来控制访问。确保这些角色的所有 `AssumeRole` 请求都记录在 CloudTrail 中并发出提醒，并确保记录使用这些角色执行的任何操作。

强烈建议清楚地命名 IAM 用户账户和 IAM 角色，以便在 CloudTrail 日志中轻松找到他们。例如，将 IAM 账户命名为 *<USER_ID>*-BREAK-GLASS# 并将 IAM 角色命名为 BREAK-GLASS-ROLE。

CloudTrail 用于记录 AWS 账户中的 API 活动，并且应该用于 配置关于使用事件响应角色的提醒。。请参阅博文，了解有关配置使用根密钥时的提醒。可以修改说明以配置 Amazon CloudWatch 指标筛选条件，从而筛选 `AssumeRole` 事件（与事件响应 IAM 角色相关）：

```
{ $.eventName = "AssumeRole" && $.requestParameters.roleArn =
 "<INCIDENT_RESPONSE_ROLE_ARN>" && $.userIdentity.invokedBy NOT EXISTS && $.eventType !
= "AwsServiceEvent" }
```

由于事件响应角色可能具有高级别的访问权限，因此，请务必将这些提醒转至广泛的群体，并及时采取适当的行动。

在事件处理期间，响应者可能需要访问不受 IAM 直接保护的系统。它们可能包括 Amazon Elastic Compute Cloud 实例、Amazon Relational Database Service 数据库或软件即服务（SaaS）平台。强烈建议不要使用 SSH 或 RDP 等本机协议，而是使用AWS Systems Manager Session Manager 对 Amazon EC2 实例进行所有管理访问。可以使用安全且经过审计的 IAM 控制此访问。此外，还可以使用 AWS Systems Manager Run Command 文档自动实施行动手册的部分内容，这可以减少用户出错的机会并缩短恢复时间。对于访问数据库和第三方工具，我们建议将访问凭证存储在 AWS Secrets Manager 中，并向事件响应者角色授予访问权限。

最后，事件响应 IAM 用户账户的管理应该添加到您的 合并人员、移动人员和离开人员流程中，并定期进行检查和测试，以确认只允许预期访问。

资源

相关文档：

- 管理对 AWS 环境的临时提升的访问权限
- AWS 安全事件响应指南

- AWS Elastic Disaster Recovery
- AWS Systems Manager Incident Manager
- 为 IAM 用户设置账户密码策略
- 在 AWS 中使用多重身份验证（MFA）
- 使用 MFA 配置跨账户存取
- 使用 IAM Access Analyzer 生成 IAM 策略
- 多账户环境中的 AWS Organizations 服务控制策略的最佳实践
- 如何在使用 AWS 账户的根访问密钥时接收通知
- 使用 IAM 托管策略创建精细会话权限

相关视频：

- 在 AWS 中自动化事件响应和取证
- 运行手册、事件报告和事件响应 DIY 指南
- 准备和响应 AWS 环境中的安全事件

相关示例：

- 实验室：AWS 账户设置和根用户
- 实验：使用 AWS 控制台和 CLI 的事件响应

SEC10-BP06 预先部署工具

确保安全人员将适当的工具预先部署到 AWS 中，以缩短调查到恢复的时间。

要自动化安全工程和运营功能，您可以使用 AWS 提供的一整套 API 和工具。您可以完全自动执行身份管理、网络安全、数据保护和监控功能，并使用您已采用的常见软件开发方法交付这些功能。当构建安全自动化时，您的系统可以监控、审核和启动响应，您不必安排人员监控您的安全位置并对事件做出人为响应。跨 AWS 服务，自动向意外事件响应者提供可搜索的相关日志数据的有效方法是启用 Amazon Detective.

如果您的事件响应团队继续以同样的方式响应警报，警报可能会让他们应接不暇。随着时间的推移，团队对警报的敏感性可能会下降，并可能在处理正常情况时犯错或者错过异常警报。利用一些功能自动处理重复和正常的警报，并将敏感、特殊的事件交由人员来处理，这样有助于避免疲于应对警报。集成

异常检测系统（例如 Amazon GuardDuty、AWS CloudTrail Insights 和 Amazon CloudWatch Anomaly Detection）可以减轻常见的基于阈值的警报负担。

您可以通过编程方式自动执行此流程中的步骤，从而改进手动流程。为事件定义修复模式之后，您可以将此模式分解为可执行的逻辑，并编写代码以执行此逻辑。随后，响应者即可执行此代码以修复问题。随着时间的推移，您可以自动化越来越多的步骤，并最终自动处理各类常见事件。

对于在您的 Amazon Elastic Compute Cloud（Amazon EC2）实例的操作系统内运行的工具，您应使用 AWS Systems Manager Run Command 执行评估，它可以使用您安装在 Amazon EC2 实例操作系统中的代理，安全地远程管理实例。它需要使用 Systems Manager 代理（SSM 代理），很多亚马逊云机器镜像（AMI，Amazon Machine Image）中都默认安装了此代理。但请注意，一旦某个实例受损，此实例上运行的工具或代理所做出的任何响应都应被视为不可信赖的响应。

未建立此最佳实践暴露的风险等级： 低

实施指导

- 预先部署工具：确保安全人员在 AWS 中预先部署了适当的工具，以便对意外事件做出适当响应。
  - 实验：使用 AWS Management Console和 CLI 响应意外事件
  - 使用 Jupyter 的意外事件响应行动手册 – AWS IAM
  - AWS 安全自动化
- 实施资源标记：用信息标记资源（例如，正在调查的资源的代码），以便在意外事件期间确定资源。
  - AWS 标记策略

资源

相关文档：

- AWS 意外事件响应指南

相关视频：

- 运行手册、事件报告和事件响应 DIY 指南

SEC10-BP07 执行实际试用

实际试用（也称为模拟或练习）是一些内部事件，可提供结构化机会，使您能够在逼真的场景中练习您的事件管理计划和流程。这些事件应让响应者练习在真实场景中使用的相同工具和技术，甚至应该模仿

真实环境。实际试用主要涉及做好准备，并以迭代方式提高您的响应能力。有些原因能够让您发现开展实际试用活动的价值，这些原因包括：

- 验证准备情况
- 建立信心 – 从模拟中学习以及开展员工培训
- 履行合规或合同义务
- 生成资格鉴定构件
- 敏捷 – 增量改进
- 速度更快并且不断改进的工具
- 优化沟通和上报
- 适应罕见和意外的情况

由于这些原因，通过参与模拟活动而学到的东西能够让组织有效地应对压力事件。开展既逼真又有益的模拟活动可能是一项非常困难的练习。尽管对可处理常见事件的流程或自动化进行测试能够实现一些优势，但只有参与创造性的 安全意外事件响应模拟（SIRS，Security Incident Response Simulation） 活动以测试您应对意外情况的能力并持续改进时，这些测试才能体现价值。

创建为您的环境、团队和工具定制的自定义模拟。找出一个问题，并围绕该问题设计您的模拟。这样的问题可以包括凭证泄露、服务器与不必要的系统通信或者导致未经授权暴露的错误配置。确定由熟悉组织的工程师创建场景，并确定另一个团队来参与其中。场景应是逼真的且具有挑战性，这样才有价值。它应包含掌握日志记录、通知、上报和执行运行手册或自动化的机会。在模拟过程中，响应者应练习他们的技术和组织技能，领导者应培养他们的意外事件管理技能。在模拟结束时，庆祝团队取得的成效，并寻找迭代、重复和扩展到深度模拟的方法。

AWS 已创建意外事件响应运行手册模板， 您不仅可以使用该模板准备您的响应工作，还可以将该模板用作模拟的基础。在规划时，可以将模拟分为五个阶段。

证据收集： 在这个阶段，团队将通过各种方式获得提醒，例如内部票证系统、来自监控工具的提醒、匿名提示甚至是公共新闻。之后，团队开始审查基础设施和应用程序日志来确定问题来源。此步骤还将涉及内部上报和意外事件领导力。在确定后，团队将继续控制意外事件

控制意外事件： 团队确定发生了意外事件并确立了问题来源。现在，团队应采取行动来控制意外事件，例如，通过禁用已泄露的凭证、隔离计算资源或撤销角色的权限。

解决意外事件： 现在，团队已控制意外事件，他们将努力减少应用程序或基础设施配置中任何易受攻击的漏洞。这可能包括轮换用于工作负载的所有凭证、修改访问控制列表（ACL，Access Control List）或更改网络配置。

未建立这种最佳实践的情况下暴露的风险等级： 中

实施指导

- 运行 实际试用：运行模拟 意外事件 响应 事件（实际试用） 来处理涉及关键人员和管理的各种威胁。
- 记录经验教训：从运行 实际试用 中获得的经验教训应成为反馈循环的一部分以改进流程。

资源

相关文档：

- AWS 意外事件响应指南
- AWS 弹性灾难恢复

相关视频：

- 运行手册、事件报告和事件响应 DIY 指南

可靠性

主题

- 基础
- 工作负载架构
- 变更管理
- 故障管理

基础

问题

- REL 1 如何管理服务配额和限制？
- REL 2 如何规划网络拓扑？

REL 1 如何管理服务配额和限制？

基于云的工作负载架构存在服务配额（也被称作服务限制）。存在这些配额是为了防止意外预置超过您所需的资源，并对 API 操作的请求速率进行限制，以保护服务不会遭到滥用。还存在资源限制，例如，将比特推入光缆的速率，或物理磁盘上的存储量。

最佳实践

- REL01-BP01 了解服务限额和限制
- REL01-BP02 跨多个账户和区域管理服务限额
- REL01-BP03 通过架构适应固定服务限额和限制
- REL01-BP04 监控和管理限额
- REL01-BP05 自动管理限额
- REL01-BP06 确保在当前限额与最大使用量之间存在足够的差距，以便应对失效转移

REL01-BP01 了解服务限额和限制

您要知道您的工作负载架构的默认配额和配额提高请求。您还要了解哪些资源限制（如磁盘或网络）可能会对您产生影响。

Service Quotas 是一项 AWS 服务，可帮助您在一个地方管理 100 多项 AWS 服务的限额。除了查找限额值，您还可以在 Service Quotas 控制台或通过 AWS 开发工具包请求增加限额并跟踪。AWS Trusted Advisor 提供服务限额检查，显示您的服务使用情况，以及某些服务在某些方面的限额。不同服务的默认服务限额也可在对应服务的 AWS 文档中找到，例如，请参阅 Amazon VPC 配额.通过配置使用计划，可在 API Gateway 内设置限流 API 的速率限制。可通过配置对应的服务进行设置的其他限制包括预置 IOPS、已分配的 RDS 存储，以及 EBS 卷分配等。Amazon Elastic Compute Cloud (Amazon EC2) 有自己的服务限制控制面板，可帮助您管理您的实例、Amazon Elastic Block Store (Amazon EBS) 和弹性 IP 地址限制。如果在某用例中，服务配额会对您的应用程序的性能造成影响，而且您无法根据自身需求对其进行调整，请联系 AWS Support 了解是否有解决的办法。

常见反模式：

- 部署工作负载，但未考虑所使用的 AWS 服务上的服务限额。
- 设计工作负载，但未不调查和考虑 AWS 服务的设计限制。
- 部署大量使用的工作负载来替换已知的现有工作负载，但没有配置必要的限额或预先联系 AWS Support。
- 通过计划事件来将流量引导至您的工作负载，但没有配置必要的限额或预先联系 AWS Support。

建立此最佳实践的好处： 了解服务配额，API 限流限制和设计限制，使您能够在设计、实施和运营工作负载时考虑到这些限制因素。

未建立此最佳实践暴露的风险等级： 高

实施指导

- 在发布的文档和 Service Quotas 中查看 AWS 服务限额
  - AWS Service Quotas（以前称为限制）
- 通过查看部署代码确定工作负载所需的所有服务。
- 使用 AWS Config 查找 AWS 账户中使用的所有 AWS 资源。
  - AWS Config 支持的 AWS 资源类型和资源关系
- 您也可以使用 AWS CloudFormation 确定使用的 AWS 资源。查看 AWS Management Console中创建的资源或通过 list-stack-resources CLI 命令创建的资源。您还可以查看配置为要在模板自身部署的资源。
  - 查看 AWS Management Console上的 AWS CloudFormation 堆栈数据和资源
  - 适用于 CloudFormation 的 AWS CLI：list-stack-resources
- 确定适用的服务配额。通过 Trusted Advisor 和 Service Quotas 使用能够以编程方式访问的信息。

资源

相关文档：

- AWS Marketplace：可以帮助跟踪限制的 CMDB 产品
- AWS Service Quotas（以前称为服务限制）
- AWS Trusted Advisor 最佳实践检查（见“服务限制”部分）
- AWS Answers 上的 AWS Limit Monitor
- Amazon EC2 服务限制
- 什么是 Service Quotas？

相关视频：

- AWS Live re:Inforce 2019 – Service Quotas

REL01-BP02 跨多个账户和区域管理服务限额

如果您目前使用多个 AWS 账户或 AWS 区域，请确保在运行生产工作负载的所有环境中都请求适当的限额。

每个账户的服务配额都可被跟踪。除非另有说明，否则每个限额都针对的是特定的 AWS 区域。除生产环境以外，还要管理所有适用的非生产环境中的配额，以避免妨碍测试与开发。

常见反模式：

- 允许一个隔离区内的资源利用率增加，但没有相关机制保持其他隔离区中的容量。
- 手动单独设置隔离区中的所有限额。
- 无法确定区域级别隔离的部署是否具有足够的大小，在某个部署丢失时容纳来自其他区域的流量增长。

建立此最佳实践的好处： 在隔离区不可用时，确保您能够处理当前负载，这有助于减少在失效转移期间发生的错误数，并且不会导致您的客户遭遇服务拒绝访问的情况。

未建立这种最佳实践的情况下暴露的风险等级： 高

实施指导

- 根据您的服务要求、延迟、法规和灾难恢复（DR，Disaster Recovery）要求选择相关账户和区域。
- 确定跨所有相关账户、区域和可用区的服务限额。限制的范围具体到账户和区域。
- 什么是 Service Quotas？

资源

相关文档：

- AWS Marketplace：可以帮助跟踪限制的 CMDB 产品
- AWS Service Quotas（以前称为服务限制）
- AWS Trusted Advisor 最佳实践检查（见“服务限制”部分）
- AWS Answers 上的 AWS Limit Monitor
- Amazon EC2 服务限制
- 什么是 Service Quotas？

相关视频：

- AWS Live re:Inforce 2019 – Service Quotas

REL01-BP03 通过架构适应固定服务限额和限制

请注意不可更改的服务配额和物理资源，并且在设计架构时要防止这些因素影响可靠性。

其中的示例包括网络带宽、AWS Lambda 有效负载大小、限制 API Gateway 的突发速率，以及并发用户连接至 Amazon Redshift 集群。

常见反模式：

- 执行基准测试时间过短，利用突发限制，但随后希望服务在该容量下持续执行。
- 选择每个用户或客户使用一项服务资源的设计时，没有意识到设计存在限制，这些限制将导致扩展时设计失败。

建立此最佳实践的好处： 跟踪您工作负载其他部分中的 AWS 服务的固定限额和限制，例如连接限制、IP 地址限制和第三方服务限制，以便能够检测到何时接近限额，并使您能够在超出限额之前解决限额问题。

未建立这种最佳实践的情况下暴露的风险等级： 中

实施指导

- 了解固定服务限额：了解固定服务限额和限制，并围绕这些因素设计架构。
  - AWS Service Quotas

资源

相关文档：

- AWS Marketplace：可以帮助跟踪限制的 CMDB 产品
- AWS Service Quotas（以前称为服务限制）
- AWS Trusted Advisor 最佳实践检查（见“服务限制”部分）
- AWS Answers 上的 AWS Limit Monitor
- Amazon EC2 服务限制

- 什么是 Service Quotas？

相关视频：

- AWS Live re:Inforce 2019 – Service Quotas

REL01-BP04 监控和管理限额

评估您的可能使用情况，并适当提高您的限额，支持使用量按计划增长。

对于支持的服务，您可以通过配置 CloudWatch 警报来监控使用情况，并在接近配额时发出提醒，从而管理您的配额。这些警报可以从 Service Quotas 或 Trusted Advisor 触发。您还可以使用 CloudWatch Logs 上的指标筛选条件来搜索与提取日志中的模式，确定使用量是否快达到配额阈值。

常见反模式：

- 配置警报，以在快达到 Service Quotas 时发出提醒，但没有关于如何响应提醒的流程。
- 只为 Service Quotas 支持的服务配置警报，不监控其他服务。

建立此最佳实践的好处： 自动跟踪 AWS 服务限额，并根据这些限额监控您的使用情况，使您可以了解何时达到限额。您也可以使用这些监控数据来评估何时可以降低限额，从而节省成本。

未建立这种最佳实践的情况下暴露的风险等级： 中

实施指导

- 监控和管理限额：评估您在 AWS 上的可能使用情况，适当提高您的区域服务限额，并支持使用量按计划增长。
  - 获取当前资源使用量（例如存储桶、实例）。使用 Amazon EC2 DescribeInstances API 等服务 API 操作来收集当前资源使用情况信息。
  - 捕获当前限额：使用 AWS Service Quotas、AWS Trusted Advisor 和 AWS 文档。
    - AWS Service Quotas 是一项 AWS 服务，使用该服务可帮助您在一个地方管理 100 多项 AWS 服务的限额。
    - 根据 Trusted Advisor 服务限制来确定您当前的服务限制。
    - 使用服务 API 操作来确定当前服务限额（如果支持）。
    - 记录已发出的限额提高请求及其状态。限额提高请求获得批准后，请确保更新您的记录以反映限额更改。

资源

相关文档：

- AWS Marketplace：可以帮助跟踪限制的 CMDB 产品
- AWS Service Quotas（以前称为服务限制）
- 服务限制的 AWS Trusted Advisor 最佳实践检查
- AWS Answers 上的 AWS Limit Monitor
- Amazon EC2 服务限制
- 什么是 Service Quotas？
- 使用 Amazon CloudWatch 警报监控 Service Quotas

相关视频：

- AWS Live re:Inforce 2019 – Service Quotas

REL01-BP05 自动管理限额

实施工具以便在接近阈值时向您发送提醒。您可以自动发出限额提高请求：通过使用 AWS Service Quotas API，您可以自动发出限额提高请求。

如果将您的配置管理数据库 (CMDB) 或票证系统与 Service Quotas 集成，您可以自动跟踪配额提高请求和当前配额。除了 AWS 开发工具包之外，Service Quotas 还使用 AWS Command Line Interface（AWS CLI）提供自动化。

常见反模式：

- 以电子表格的形式跟踪配额和使用情况。
- 每天、每周或每月运行使用情况报告，然后将使用情况与配额进行比较。

建立此最佳实践的好处： 自动跟踪 AWS 服务限额，并根据这些限额监控您的使用情况，从而让您了解何时接近限额。您可以设置自动流程，帮助您在需要时提出配额提高请求。当使用情况趋向于相反的方向时，您可能需要考虑降低一些限额，以实现降低风险（如果凭据被盗）和节省成本的效果。

未建立此最佳实践暴露的风险等级： 中

实施指导

- 设置自动监控：通过开发工具包实施各种工具，以便在接近阈值时向您发出提醒。
  - 利用 Service Quotas，通过自动限额监控解决方案（例如 AWS Limit Monitor 或从 AWS Marketplace 获得的产品）来增强服务。
    - 什么是 Service Quotas？
    - AWS 上的限额监控 – AWS 解决方案
  - 使用 Amazon SNS 和 AWS Service Quotas API 来根据限额阈值来设置触发响应。
  - 测试自动化。
    - 配置限制阈值。
    - 与来自 AWS Config、部署管道、Amazon EventBridge 或第三方的更改事件集成。
    - 人工设置低限额阈值来测试响应。
    - 设置触发器以根据通知采取适当措施，并在必要时联系 AWS Support。
    - 人工触发更改事件。
    - 运行实际测试以测试限额提高更改流程。

资源

相关文档：

- APN 合作伙伴：可帮助进行配置管理的合作伙伴
- AWS Marketplace：可以帮助跟踪限制的 CMDB 产品
- AWS Service Quotas（以前称为服务限制）
- AWS Trusted Advisor 最佳实践检查（见“服务限制”部分）
- AWS 上的限额监控 – AWS 解决方案
- Amazon EC2 服务限制
- 什么是 Service Quotas？

相关视频：

- AWS Live re:Inforce 2019 – Service Quotas

REL01-BP06 确保在当前限额与最大使用量之间存在足够的差距，以便应对失效转移

当资源出现故障时，它可能仍会被计入限额，直到被成功终止。在出现故障的资源被终止之前，请确保您的配额涵盖所有出现故障的资源与其替换资源的叠加。在计算此差距时，应将可用区故障考虑在内。

常见反模式：

- 根据当前需求设置服务限额，而不考虑失效转移场景。

建立此最佳实践的好处： 当事件可能影响可用性时，云可让您实施策略来减小这些事件造成的影响或从这些事件中恢复。此类策略通常包括创建额外资源来替换出现故障的资源。您的限额策略必须适应这些额外资源。

未建立此最佳实践暴露的风险等级： 中

实施指导

- 确保您的服务限额与最高使用量之间有足够的差距，以便应对失效转移。
  - 根据您的部署模式、可用性要求和使用量增长情况确定服务限额。
  - 根据需要请求增加限额。预计完成限额提高请求所需的时间。
    - 确定可靠性要求（也称为“X 个 9”）。
    - 构建故障场景（例如组件、可用区或区域缺失）。
    - 确定部署方法（例如金丝雀部署、蓝/绿部署、红/黑部署或滚动部署）。
    - 在当前限制中包含适当的缓冲区（例如 15%）。
    - 预计使用量增长（例如监控使用量趋势）。

资源

相关文档：

- AWS Marketplace：可以帮助跟踪限制的 CMDB 产品
- AWS Service Quotas（以前称为服务限制）
- AWS Trusted Advisor 最佳实践检查（见“服务限制”部分）
- Amazon EC2 服务限制
- 什么是 Service Quotas？

相关视频：

- AWS Live re:Inforce 2019 – Service Quotas

REL 2 如何规划网络拓扑？

工作负载通常存在于多个环境中。其中包括多个云环境（可公开访问云和私有云），可能还包括现有数据中心基础设施。相关计划必须涵盖网络注意事项，如系统内部和系统间连接、公有 IP 地址管理、私有 IP 地址管理，以及域名解析。

最佳实践

- REL02-BP01 为工作负载公有端点使用高度可用的网络连接
- REL02-BP02 为云环境和本地部署环境之间的私有网络预置冗余连接
- REL02-BP03 确保 IP 子网分配考虑扩展和可用性
- REL02-BP04 轴辐式拓扑优先于多对多网格
- REL02-BP05 在互相连接的所有私有地址空间中强制实施非重叠的私有 IP 地址范围

REL02-BP01 为工作负载公有端点使用高度可用的网络连接

这些端点及其路由必须高度可用。为此，需使用高度可用的 DNS、内容分发网络 (CDN)、API Gateway、负载均衡或反向代理。

Amazon Route 53、AWS Global Accelerator、Amazon CloudFront、Amazon API Gateway 和 Elastic Load Balancing（ELB）都提供了高度可用的公共端点。您还可以选择评估 AWS Marketplace 软件设备是否适用于负载均衡和代理。

您的工作负载所提供的服务的用户，无论其是最终用户或其他服务，都要在这些服务终端节点上发起请求。您可以使用多个 AWS 资源以提供高度可用的端点。

Elastic Load Balancing 提供跨可用区的负载均衡，执行第 4 层（TCP）或第 7 层（http/https）路由，与 AWS WAF 集成，并与 AWS Auto Scaling 集成以帮助构建自我修复基础设施，吸收流量增长，并同时在流量减少时释放资源。

Amazon Route 53 是一项可扩展且高度可用的域名系统（DNS，Domain Name System）服务，可将用户请求连接至在 AWS 中运行的基础设施，如 Amazon EC2 实例、Elastic Load Balancing 负载均衡器或 Amazon S3 存储桶，此外还可用于将用户路由至 AWS 以外的基础设施。

AWS Global Accelerator 是一项网络层服务，您可以用它将流量引导至 AWS 全球网络中的最佳端点。

分布式拒绝服务（DDoS，Distributed Denial of Service）攻击会引发使您的用户的合法流量无法进入并降低可用性的风险。AWS Shield 提供自动防护，无需额外成本即可避免您的工作负载上的 AWS 服

务端点遭受此类攻击。您可以使用 APN 合作伙伴和 AWS Marketplace 提供的虚拟设备来增强这些功能，以满足您的需求。

常见反模式：

- 在实例或容器中使用公有互联网地址并通过 DNS 管理与它们的连接。
- 使用互联网协议地址而非域名查找服务。
- 为较大地理区域提供内容（网页、静态资产、媒体文件），而不使用内容分发网络。

建立此最佳实践的好处： 通过在工作负载中实施高度可用的服务，您将清楚自己的工作负载可供用户使用。

未建立此最佳实践暴露的风险等级： 高

实施指导

确保为工作负载用户提供高度可用的连接：Amazon Route 53、AWS Global Accelerator、Amazon CloudFront、Amazon API Gateway 和 Elastic Load Balancing（ELB）都提供高度可用的面向公众的端点。您还可以选择评估 AWS Marketplace 软件设备是否适用于负载均衡和代理。

- 确保您与用户之间具有高度可用的连接。
- 确保使用高度可用的 DNS 来管理应用程序端点域名。
  - 如果您的用户通过互联网访问应用程序，请使用服务 API 操作以确保正确使用互联网网关。另外，请确认托管应用程序端点的子网的路由表条目正确无误。
    - DescribeInternetGateways
    - DescribeRouteTables
- 确保在应用程序前使用高度可用的反向代理或负载均衡器。
  - 如果用户通过本地部署环境访问您的应用程序，请确保 AWS 与本地部署环境之间的连接高度可用。
  - 使用 Route 53 管理您的域名。
    - 什么是 Amazon Route 53？
  - 使用符合您要求的第三方 DNS 提供商。
  - 使用 Elastic Load Balancing。
    - 什么是 Elastic Load Balancing？
  - 使用符合您要求的 AWS Marketplace 设备。

资源

相关文档：

- AWS 合作伙伴：可帮助您规划联网的合作伙伴
- AWS Direct Connect 弹性建议
- 适用于网络基础设施的 AWS Marketplace
- Amazon Virtual Private Cloud 连接选项白皮书
- 多数据中心 HA 网络连接
- 使用 Direct Connect 弹性工具包开始操作
- VPC 终端节点和 VPC 终端节点服务 (AWS PrivateLink)
- 什么是 AWS Global Accelerator？
- 什么是 Amazon VPC？
- 什么是 Transit Gateway？
- 什么是 Amazon CloudFront？
- 什么是 Amazon Route 53？
- 什么是 Elastic Load Balancing？
- 使用 Direct Connect 网关

相关视频：

- AWS re:Invent 2018：Amazon VPC 的高级 VPC 设计和新功能（NET303）
- AWS re:Invent 2019：适用于众多 VPC 的 AWS Transit Gateway 参考架构（NET406-R1）

REL02-BP02 为云环境和本地部署环境之间的私有网络预置冗余连接

在单独部署的私有网络之间使用多个 AWS Direct Connect 连接或 VPN 隧道。使用多个 Direct Connect 位置实现高可用性。如果使用多个 AWS 区域，请确保其中至少有两个区域存在冗余。您可能想要评估终止 VPN 的 AWS Marketplace 设备。如果您使用 AWS Marketplace 设备，请在不同的可用区中部署冗余实例以实现高可用性。

AWS Direct Connect 是一项云服务，可以在本地环境和 AWS 之间轻松建立专用网络连接。使用 Direct Connect Gateway，您的本地数据中心可以连接至跨多个 AWS 区域的多个 AWS VPC。

此类冗余可解决会对连接弹性造成影响的潜在故障：

- 您将如何灵活应对拓扑中的故障？
- 如果您配置错了某些内容并删除了连接，会发生什么情况？
- 您是否能应对服务流量或使用量的意外增加？
- 您是否能够吸收未遂的分布式拒绝服务 (DDoS) 攻击？

若通过 VPN 将您的 VPC 连接至本地数据中心，您应该考虑在选择运行该设备所需的供应商和实例大小时所需要的弹性和带宽要求。如果您使用的 VPN 设备在其实施中没有弹性，则您应该通过第二个设备获取冗余连接。对于所有这些场景，您需要定义可接受的恢复时间并进行测试，以确保您可以满足这些要求。

如果选择通过 Direct Connect 连接将您的 VPC 连接至数据中心，而且您要求连接具有高可用性，请从每个数据中心获得冗余 Direct Connect 连接。冗余连接应使用来自与第一个不同位置的其他 Direct Connect 连接。如果您有多个数据中心，则确保连接在不同的位置终止。使用 Direct Connect 弹性工具包 以帮助您完成设置。

如果您选择使用 AWS VPN 并通过互联网失效转移到 VPN，一定要了解，它支持每个 VPN 隧道高达 1.25-Gbps 的吞吐量，但在多个 AWS 托管 VPN 隧道终止于同一 VGW 的情况下，不支持将等价多路径（ECMP，Equal Cost Multi Path）用于出站流量。我们不建议您使用 AWS 托管 VPN 作为 Direct Connect 连接的备份，除非您可以接受失效转移期间的速度低于 1 Gbps。

您还可以使用 VPC 端点将您的 VPC 私下连接至受支持的 AWS 服务和 VPC 端点服务，它们得到 AWS PrivateLink 的支持而无需通过公有互联网传输。终端节点为虚拟设备。它们是水平扩展、冗余，而且高度可用的 VPC 组件。它们支持在您的 VPC 和服务实例之间进行通信，而不会对您的网络流量施加可用性风险或带宽限制。

常见反模式：

- 在本地网络和 AWS 之间只有一个连接供应方。
- 使用 AWS Direct Connect 连接的连接功能，但只有一个连接。
- 您的 VPN 连接只有一条路径。

建立此最佳实践的好处： 通过在云环境和公司或本地部署环境之间实施冗余连接，您可以确保两个环境之间的依赖服务能够可靠通信。

未建立这种最佳实践的情况下暴露的风险等级： 高

实施指导

- 确保您在 AWS 和本地部署环境之间具有高度可用的连接。在单独部署的私有网络之间使用多个 AWS Direct Connect 连接或 VPN 隧道。使用多个 Direct Connect 位置实现高可用性。如果使用多个 AWS 区域，请确保其中至少有两个区域存在冗余。您可能想要评估终止 VPN 的 AWS Marketplace 设备。如果您使用 AWS Marketplace 设备，请在不同的可用区中部署冗余实例以实现高可用性。
  - 确保您拥有面向本地部署环境的冗余连接。您可能需要面向多个 AWS 区域的冗余连接，以满足可用性需求。
    - AWS Direct Connect 弹性建议
    - 使用冗余 Site-to-Site VPN 连接以提供故障转移
      - 使用服务 API 操作确定正确使用了 Direct Connect 线路。
        - DescribeConnections
        - DescribeConnectionsOnInterconnect
        - DescribeDirectConnectGatewayAssociations
        - DescribeDirectConnectGatewayAttachments
        - DescribeDirectConnectGateways
        - DescribeHostedConnections
        - DescribeInterconnects
      - 如果您仅有一个 Direct Connect 连接或没有此连接，请设置连接虚拟私有网关的冗余 VPN 隧道。
        - 什么是 AWS Site-to-Site VPN？
  - 捕获您的当前连接（例如，Direct Connect、虚拟私有网关、AWS Marketplace 设备）。
    - 使用服务 API 操作查询 Direct Connect 连接的配置。
      - DescribeConnections
      - DescribeConnectionsOnInterconnect
      - DescribeDirectConnectGatewayAssociations
      - DescribeDirectConnectGatewayAttachments
      - DescribeDirectConnectGateways
      - DescribeHostedConnections
      - DescribeInterconnects
    - 使用服务 API 操作收集路由表使用的虚拟私有网关。

    - DescribeVpnGateways
    - DescribeRouteTables
  - 使用服务 API 操作收集路由表使用的 AWS Marketplace 应用程序。
    - DescribeRouteTables

资源

相关文档：

- AWS 合作伙伴：可帮助您规划联网的合作伙伴
- AWS Direct Connect 弹性建议
- 适用于网络基础设施的 AWS Marketplace
- Amazon Virtual Private Cloud 连接选项白皮书
- 多数据中心 HA 网络连接
- 使用冗余 Site-to-Site VPN 连接以提供故障转移
- 使用 Direct Connect 弹性工具包开始操作
- VPC 终端节点和 VPC 终端节点服务 (AWS PrivateLink)
- 什么是 Amazon VPC？
- 什么是 Transit Gateway？
- 什么是 AWS Site-to-Site VPN？
- 使用 Direct Connect 网关

相关视频：

- AWS re:Invent 2018：Amazon VPC 的高级 VPC 设计和新功能（NET303）
- AWS re:Invent 2019：适用于众多 VPC 的 AWS Transit Gateway 参考架构（NET406-R1）

REL02-BP03 确保 IP 子网分配考虑扩展和可用性

Amazon VPC IP 地址范围必须足够大，以满足工作负载的要求，包括考虑未来的扩展以及跨可用区为子网分配 IP 地址。这包括负载均衡器、EC2 实例和基于容器的应用程序。

当您规划网络拓扑时，第一步是定义 IP 地址空间本身。应（按照 RFC 1918 准则）为每个 VPC 分配私有 IP 地址范围。作为此流程的一部分，要满足以下要求：

- 在每个区域中为多个 VPC 留出 IP 地址空间。
- 在 VPC 内，为跨多个可用区的多个子网留出空间。
- 始终在 VPC 内保留未使用的 CIDR 块空间以用于未来扩展。
- 确保有 IP 地址空间以满足您可能使用的任何 EC2 实例临时性队列的需求，如适用于机器学习的 Spot 队列、Amazon EMR 集群或 Amazon Redshift 集群。
- 注意，每个子网 CIDR 块中的前四个 IP 地址和最后一个 IP 地址将被预留而无法供您使用。
- 您应计划部署大型 VPC CIDR 块。注意，最初被分配到您的 VPC 的 VPC CIDR 块无法被更改或删除，但您可以向 VPC 添加额外的非重叠的 CIDR 块。虽然无法更改 IPv4 CIDR，但可以对 IPv6 CIDR 进行更改。请记住，部署最大的 VPC (/16) 会产生超过 65000 个 IP 地址。单单在基础 10.x.x.x IP 地址空间内，您可以预置 255 个这样的 VPC。因此，您可以趋向于过大数量而不是过小数量，这样更容易管理 VPC。

常见反模式：

- 创建小型 VPC。
- 创建小型子网，然后在业务增长过程中向配置添加子网。
- 错误估计 Elastic Load Balancer 可以使用的 IP 地址数量。
- 在相同子网中部署多个高流量负载均衡器。

建立此最佳实践的好处： 这可确保您能适应工作负载增长要求，并在扩展过程中继续提供可用性。

未建立这种最佳实践的情况下暴露的风险等级： 中

实施指导

- 规划您的网络以适应增长、符合监管合规性以及实现与其他服务的集成。如果没有合理的规划，则增长可能会被低估、监管合规性可能会发生变化并且收购或私有网络连接可能难以实施。
  - 根据您的服务要求、延迟、法规和灾难恢复（DR，Disaster Recovery）要求选择相关 AWS 账户和区域。
  - 确定您的区域 VPC 部署需求。
  - 确定 VPC 的大小。
    - 确定您是否要部署多 VPC 连接。
      - 什么是 Transit Gateway？
      - 单区域多 VPC 连接

- 确定您是否需要隔离网络以满足法规要求。
- 使 VPC 尽可能大。最初为 VPC 分配的 VPC CIDR 块无法更改或删除，但您可以向 VPC 添加额外的非重叠 CIDR 块。但是，这样可能会分割您的地址范围。
- 使 VPC 尽可能大。最初为 VPC 分配的 VPC CIDR 块无法更改或删除，但您可以向 VPC 添加额外的非重叠 CIDR 块。但是，这样可能会分割您的地址范围。

资源

相关文档：

- AWS 合作伙伴：可帮助您规划联网的合作伙伴
- 适用于网络基础设施的 AWS Marketplace
- Amazon Virtual Private Cloud 连接选项白皮书
- 多数据中心 HA 网络连接
- 单区域多 VPC 连接
- 什么是 Amazon VPC？

相关视频：

- AWS re:Invent 2018：Amazon VPC 的高级 VPC 设计和新功能（NET303）
- AWS re:Invent 2019：适用于众多 VPC 的 AWS Transit Gateway 参考架构（NET406-R1）

REL02-BP04 轴辐式拓扑优先于多对多网格

如果通过 VPC 对等连接、AWS Direct Connect 或 VPN 连接的网络地址空间超过两个（例如，VPC 和本地网络），则使用与 AWS Transit Gateway 所提供的模型类似的轴辐式模型。

如果只有两个此类网络，您可以简单地使其相互连接，但随着网络数量的增加，这种网格连接的复杂性将变得无法接受。AWS Transit Gateway 提供易于维护的轴辐式模型，允许在您的多个网络中对流量进行路由。

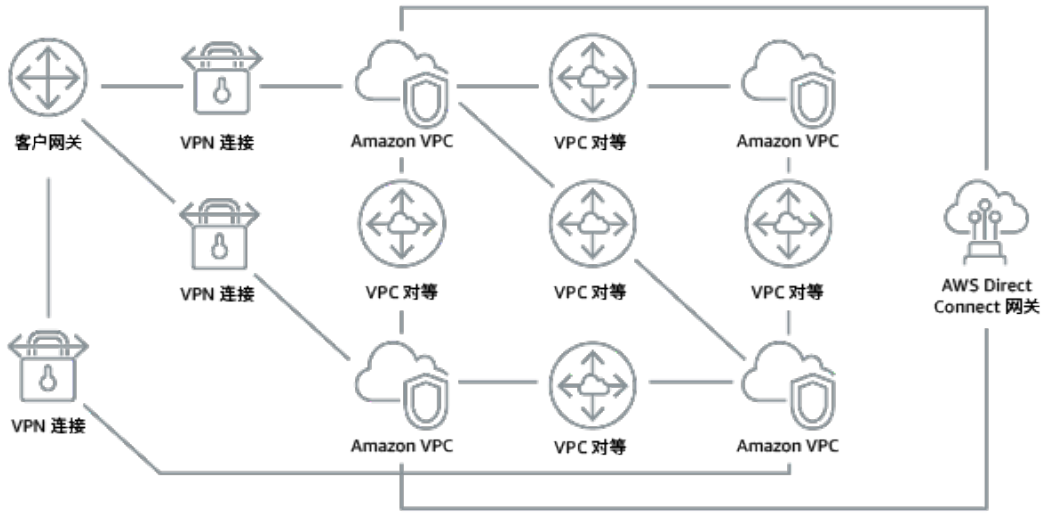


图 1：没有 AWS Transit Gateway：您需要将各个 Amazon VPC 对等连接并使用 VPN 连接来连接到各个现场位置，其复杂程度会随着它的扩展而递增。

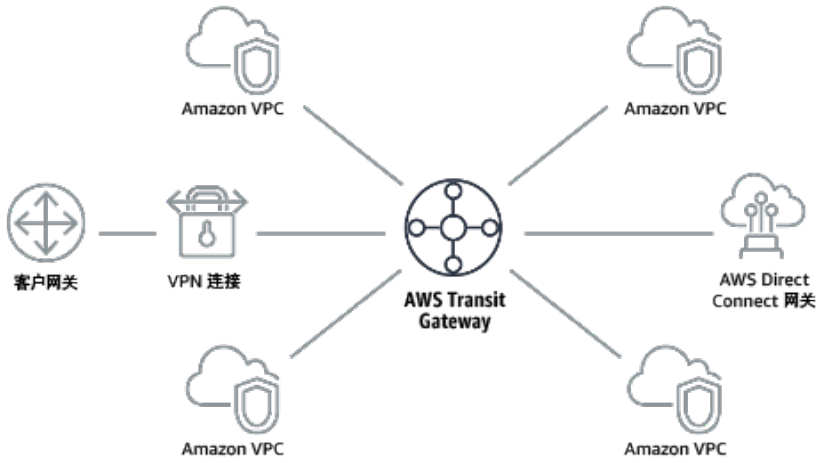


图 2：使用 AWS Transit Gateway：您只需将每个 Amazon VPC 或 VPN 连接至 AWS Transit Gateway，而且它会在每个 VPC 或 VPN 之间往返路由流量。

常见反模式：

- 使用 VPC 对等连接来连接两个以上的 VPC。
- 为每个 VPC 建立多个 BGP 会话，从而建立跨多个 AWS 区域的虚拟私有云（VPC，Virtual Private Cloud）连接。

建立此最佳实践的好处： 随着网络数量增加，这种复杂的网格连接将变得无法维持。AWS Transit Gateway 提供易于维护的轴辐式模型，允许在您的多个网络中路由流量。

未建立这种最佳实践的情况下暴露的风险等级： 中

实施指导

- 轴辐式拓扑优先于多对多网格。如果通过 VPC 对等连接、AWS Direct Connect 或 VPN 连接的网络地址空间超过两个（VPC，本地网络），则使用与 AWS Transit Gateway 所提供的模型类似的轴辐式模型。
  - 如果只有两个此类网络，您只需将其相互连接即可，但随着网络数量的增长，这种复杂的网格连接将变得无法维持。AWS Transit Gateway 提供易于维护的轴辐式模型，允许在您的多个网络中路由流量。
    - 什么是 Transit Gateway？

资源

相关文档：

- AWS 合作伙伴：可帮助您规划联网的合作伙伴
- 适用于网络基础设施的 AWS Marketplace
- 多数据中心 HA 网络连接
- VPC 终端节点和 VPC 终端节点服务 (AWS PrivateLink)
- 什么是 Amazon VPC？
- 什么是 Transit Gateway？

相关视频：

- AWS re:Invent 2018：Amazon VPC 的高级 VPC 设计和新功能（NET303）
- AWS re:Invent 2019：适用于众多 VPC 的 AWS Transit Gateway 参考架构（NET406-R1）

REL02-BP05 在互相连接的所有私有地址空间中强制实施非重叠的私有 IP 地址范围

多个 VPC 通过对等连接或 VPN 连接时，各个 VPC 的 IP 地址范围不得重叠。与之类似，您必须避免 VPC 和本地部署环境或与其他您使用的云提供商之间出现 IP 地址冲突。您还必须能够在需要时分配私有 IP 地址范围。

IP 地址管理 (IPAM) 系统可以帮助解决这个问题。AWS Marketplace 提供多个 IPAM。

常见反模式：

- 在 VPC 中使用与本地部署或企业网络相同的 IP 范围。
- 不必追踪用于部署工作负载的 VPC 的 IP 范围。

建立此最佳实践的好处： 主动规划网络可确保您不会遇到互连网络中多次出现相同 IP 地址的情况。这可防止使用不同应用程序的工作负载部分出现路由问题。

未建立此最佳实践暴露的风险等级： 中

实施指导

- 监控和管理您的 CIDR 使用。评估您在 AWS 上的可能使用量、将 CIDR 范围添加到现有 VPC 并创建 VPC 以便使用量实现计划增长。
  - 捕获当前的 CIDR 使用量（例如，VPC、子网）
    - 使用服务 API 操作收集当前的 CIDR 使用量数据。
  - 捕获您当前的子网使用量。
    - 使用服务 API 操作在每个区域中按 VPC 收集子网。
      - DescribeSubnets
    - 记录当前使用量。
    - 确定您是否创建了任何重叠 IP 范围。
    - 计算备用容量。
    - 记录重叠的 IP 范围。您可以迁移到新地址范围，或使用 AWS Marketplace 的网络和端口转换（NAT）设备（如果需要连接重叠范围）。

资源

相关文档：

- AWS 合作伙伴：可帮助您规划联网的合作伙伴
- 适用于网络基础设施的 AWS Marketplace
- Amazon Virtual Private Cloud 连接选项白皮书
- 多数据中心 HA 网络连接
- 什么是 Amazon VPC？

- 什么是 IPAM？

相关视频：

- AWS re:Invent 2018：Amazon VPC 的高级 VPC 设计和新功能（NET303）
- AWS re:Invent 2019：适用于众多 VPC 的 AWS Transit Gateway 参考架构（NET406-R1）

工作负载架构

问题

- REL 3 如何设计工作负载服务架构？
- REL 4 您如何在分布式系统中设计交互以预防发生故障？
- REL 5 您如何在分布式系统中进行交互设计，从而缓解或经受住故障影响？

REL 3 如何设计工作负载服务架构？

使用面向服务的架构 (SOA) 或微服务架构构建高度可扩展的可靠工作负载。面向服务的架构 (SOA) 可通过服务接口使软件组件可重复使用。微服务架构则进一步让组件变得更小、更简单。

最佳实践

- REL03-BP01 选择如何划分工作负载
- REL03-BP02 构建专注于特定业务领域和功能的服务
- REL03-BP03 根据 API 提供服务合同

REL03-BP01 选择如何划分工作负载

在确定应用程序的弹性要求时，工作负载划分很重要。应尽可能避免使用整体式架构。相反，应仔细考虑哪些应用程序组件可以分解为多项微服务。根据您的应用程序要求，最终应尽可能采用服务导向型架构（SOA）与微服务组合的形式。能够实现无状态的工作负载更容易部署为微服务。

期望结果： 工作负载应该可支持、可扩展，并尽可能地松散耦合。

在选择如何划分工作负载时，要权衡其优点和复杂性。适用于即将首次发布的新产品的功能有别于从一开始就构建用于扩展的工作负载的需求。重构一个现有的整体架构时，您需要考虑应用程序对无状态分解的支持程度。通过将服务分解为较小的部分，可以让职责明确的小型团队来开发和管理它们。然而，较小的服务会带来复杂性，包括可能会增加延迟，调试变得更复杂，而且加重运营负担。

常见反模式：

- 如示例所示，微服务 Death Star 是这样一种情况：原子组件变得高度相互依赖，牵一发而动全身，使组件像一块整体一样死板而又脆弱。

建立此实践的好处：

- 更多特定分段可以提高敏捷性、组织灵活性和可扩展性。
- 减小了服务中断的影响。
- 应用程序组件可能有不同的可用性要求，可通过更加原子化的分段来满足这些要求。
- 支持工作负载的团队职责分明。

未建立这种最佳实践的情况下暴露的风险等级： 高

实施指导

请根据工作负载的分段方式选择架构类型。选择 SOA 或微服务架构（极少数情况下是整体架构）。即便在刚开始选择整体架构，您必须确保它是模块化的，而且由于您的产品随着采用的用户增加而扩展，它最终也可转变成为 SOA 或微服务架构。SOA 和微服务各自提供较小的区段，它们是现代可扩展和可靠架构的首选，但您需要认真权衡利弊，尤其在部署微服务架构时。

一项主要的权衡是您现在使用的是分布式计算架构，可能更难实现用户延迟要求，还增加了调试和跟踪用户交互的复杂性。您可以使用 AWS X-Ray 来帮助解决此问题。需要考虑的另一个影响是，您管理的应用程序数量增加时，运营复杂性也会随之增加，这要求部署多个相互独立组件。

整体架构、服务导向型架构和微服务架构

实施步骤

- 确定构建或重构应用程序所需的适当架构。SOA 和微服务分别提供较小分段，这是现代可扩展的可靠架构的首选。要在实现较小分段的同时避免一些微服务复杂性，SOA 是很好的折中方案。有关更多详细信息，请参阅 微服务利弊权衡.
- 如果您的工作负载适合，并且您的组织可以支持，则应使用微服务架构来实现最佳敏捷性和可靠性。有关更多详细信息，请参阅 在 AWS 上实施微服务。
- 考虑遵循 Strangler Fig 模式， 将整体架构重构为较小的组件。这包括逐步用新的应用程序和服务替换特定的应用程序组件。AWS Migration Hub Refactor Spaces 作为增量重构的起点。有关更多详细信息，请参阅 使用绞杀者模式无缝迁移本地传统工作负载。
- 实施微服务可能需要采用一种服务发现机制，让这些分布式服务能够相互通信。AWS App Mesh 可用于服务导向型架构，以实现可靠的发现和服务访问。 AWS Cloud Map 也可用于动态的基于 DNS 的服务发现。
- 如果您要从整体架构迁移到 SOA，Amazon MQ 可作为服务总线来帮助弥合这一差距（在云中重新设计传统应用程序时）。
- 对于具有单个共享数据库的现有整体架构，请选择将数据重组为较小分段的方式。可以按业务部门、访问模式或数据结构来划分。在重构过程的这一阶段，应选择是使用关系型还是非关系型（NoSQL）数据库来继续操作。有关更多详细信息，请参阅 从 SQL 到 NoSQL.

实施计划的工作量级别： 高

资源

相关最佳实践：

- REL03-BP02 构建专注于特定业务领域和功能的服务

相关文档：

- Amazon API Gateway：使用 OpenAPI 配置 REST API
- 什么是服务导向型架构？
- 边界上下文（领域驱动设计的中心模式）
- 在 AWS 上实施微服务
- 微服务利弊权衡
- 微服务 – 一个全新架构术语的定义

- AWS 上的微服务
- 什么是 AWS App Mesh？

相关示例：

- 迭代应用程序现代化研讨会

相关视频：

- 利用 AWS 上的微服务实现卓越

REL03-BP02 构建专注于特定业务领域和功能的服务

面向服务的架构（SOA，Service-Oriented Architecture）采用由业务需求定义的划分明确的功能来构建服务。微服务则使用领域模型和有界上下文对此进行进一步限制，使每项服务都只用于一种用途。专注于特定功能让您可以区分不同服务的可靠性要求，并且更有针对性地锁定投资目标。简洁的业务问题和与每项服务关联的小型团队也简化了组织扩展。

在设计微服务架构时，借助于领域驱动设计 (DDD) 对使用实体的业务问题进行建模十分有帮助。以 Amazon.com 网站为例，实体可能包括包装、配送、时间表、价格、折扣和货币。然后，该模型会使用 边界上下文进一步细分为更小的模型，具有相似功能和属性的实体在边界上下文中被分到一组。因此，在 Amazon.com 例子中，包装、配送和时间表是装运上下文的一部分，而价格、折扣和货币是定价上下文的一部分。通过将模型细分为不同的上下文，即可得到如何确定微服务边界的模板。

未建立这种最佳实践的情况下暴露的风险等级： 高

实施指导

- 根据业务领域及各自的功能设计工作负载。专注于特定功能让您可以区分不同服务的可靠性要求，并且更有针对性地锁定投资目标。简洁的业务问题和与每项服务关联的小型团队也简化了组织扩展。
  - 执行领域分析，为您的工作负载制定领域驱动设计 (DDD) 方案。然后，您可以选择一个架构类型，以满足您的工作负载需求。

    - 如何将整体式架构分解为多项微服务
    - 在被遗留系统包围时通过 DDD 开始着手
    - Eric Evans“领域驱动设计：解决软件核心的复杂性”
    - 在 AWS 上实施微服务
- 将服务分解成尽可能小的组件。借助微服务架构，您可以将工作负载分解成功能最小的组件，以便支持组织的可扩展性和敏捷性。
  - 根据工作负载及其设计目标、限制和任何其他使用注意事项来定义 API。
    - 定义 API。
      - API 定义应允许增加参数。
    - 定义设计可用性。
      - 您的 API 可以具有针对不同功能的多个设计目标。
    - 设置限制
      - 通过测试来确定工作负载的功能限制。

资源

相关文档：

- Amazon API Gateway：使用 OpenAPI 配置 REST API
- 边界上下文（领域驱动设计的中心模式）
- Eric Evans“领域驱动设计：解决软件核心的复杂性”
- 在被遗留系统包围时通过 DDD 开始着手
- 如何将整体式架构分解为多项微服务
- 在 AWS 上实施微服务
- 微服务利弊权衡
- 微服务 – 一个全新架构术语的定义
- AWS 上的微服务

REL03-BP03 根据 API 提供服务合同

服务合同是团队之间关于服务集成的成文协议，它包括机器可读的 API 定义、速率限制和性能预期。版本控制策略让客户能够继续使用现有的 API，并在更新的 API 准备就绪时将他们的应用程序迁移到

此类 API。只要遵守合同，即可随时进行部署。服务提供商团队可以使用自己选择的技术堆栈来满足 API 合同要求。同样，服务使用者可以使用自己的技术。

微服务将面向服务的架构（SOA，Service-Oriented Architecture）的概念提升到创建具有最小功能集的服务。每项服务都会发布一个 API，以及使用相应服务的设计目标、限制和其他注意事项。这会通过调用 应用程序 建立合同。这可以实现三个主要优势：

- 服务具有一个需要解决的简明的业务问题，以及出现该业务问题的小型团队。这有助于更好地扩展组织。
- 只要满足 API 和其他合同要求，团队就可以随时进行部署。
- 只要满足 API 和其他合同要求，团队就可以使用他们想用的任何技术堆栈。

Amazon API Gateway 是一种完全托管式服务，可以帮助开发人员轻松创建、发布、维护、监控和保护任意规模的 API。它负责处理多达数十万个并发 API 调用的接受和处理过程中涉及的所有任务，包括流量管理、授权和访问控制、监控以及 API 版本管理。采用 OpenAPI 规范 (OAS)，亦即之前的 Swagger 规范，您可以定义 API 合同并将其导入到 API Gateway。然后，您便可以通过 API Gateway 对 API 进行版本控制与部署。

未建立这种最佳实践的情况下暴露的风险等级： 低

实施指导

- 按照不同 API 提供服务合同：服务合同是团队之间关于服务集成的记录在案的协议，它包括机器可读的 API 定义、速率限制和性能预期等。
  - Amazon API Gateway：使用 OpenAPI 配置 REST API
    - 版本控制策略让客户能够继续使用现有的 API，并在更新的 API 准备就绪时将他们的应用程序迁移到此类 API。
    - Amazon API Gateway 是一种完全托管式服务，可以帮助开发人员轻松创建任意规模的 API。采用 OpenAPI 规范（OAS，OpenAPI Specification），亦即之前的 Swagger 规范，您可以定义 API 合同并将其导入到 API Gateway。然后，您便可以通过 API Gateway 对 API 进行版本控制与部署。

资源

相关文档：

- Amazon API Gateway：使用 OpenAPI 配置 REST API

- 边界上下文（领域驱动设计的中心模式）
- 在 AWS 上实施微服务
- 微服务利弊权衡
- 微服务 – 一个全新架构术语的定义
- AWS 上的微服务

REL 4 您如何在分布式系统中设计交互以预防发生故障？

分布式系统依赖于通信网络实现组件（例如服务器或服务）的互联。尽管这些网络中存在数据丢失或延迟，但是您的工作负载必须可靠运行。分布式系统组件的运行方式不得对其他组件或工作负载产生负面影响。这些最佳实践能够预防故障，并改善平均故障间隔时间（MTBF）。

最佳实践

- REL04-BP01 确定需要哪种类型的分布式系统
- REL04-BP02 实施松耦合的依赖关系
- REL04-BP03 持续工作
- REL04-BP04 使所有响应幂等

REL04-BP01 确定需要哪种类型的分布式系统

硬性实时分布式系统需要同步并快速地做出响应，而软性实时系统有更宽松的以分钟计的时间窗口，或更多响应。离线系统会对响应进行批处理或异步处理。硬性实时分布式系统具有最严格的可靠性要求。

硬性实时分布式系统 要面对分布式系统中的 最艰巨的挑战，又被称作请求/回复服务。因为收到请求的时间不可预测，而响应必须迅速（例如，客户正急切地等待响应）。此类示例包括，前端 Web 服务器、指令管道、信用卡交易，以及每个 AWS API 和语音通话。

未建立这种最佳实践的情况下暴露的风险等级： 高

实施指导

- 确定需要哪种类型的分布式系统。分布式系统要应对的挑战包括延迟、扩展、了解联网 API、对数据进行集结与解集，以及 Paxos 等算法的复杂性。随着系统规模扩大并呈现出更明显的分布态势，我们现在需要在日常生活中面对过去只存在于理论中的边缘情况。
  - Amazon Builders' Library：分布式系统相关挑战

- 硬性实时分布式系统需要快速的同步响应。
- 软性实时系统则有更宽松的以分钟计的时间窗口，或更好响应。
- 离线系统会对响应进行批处理或异步处理。
- 硬性实时分布式系统具有最严格的可靠性要求。

资源

相关文档：

- Amazon EC2：确保幂等性
- Amazon Builders' Library：分布式系统相关挑战
- Amazon Builders' Library：可靠性、持续工作和安然无忧
- 什么是 Amazon EventBridge？
- 什么是 Amazon Simple Queue Service？

相关视频：

- 2019 年 AWS 纽约峰会：介绍事件驱动型架构和 Amazon EventBridge（MAD205）
- AWS re:Invent 2018：闭环系统和开放思维：如何掌控不同规模的系统（ARC337）（包括松耦合、持续工作和静态稳定性）
- AWS re:Invent 2019：迁移到事件驱动型架构（SVS308）

REL04-BP02 实施松耦合的依赖关系

队列系统、流系统、工作流和负载均衡器等依赖关系是松耦合的。松耦合有助于隔离某个组件的行为与依赖于它的其他组件的行为，从而提升弹性和敏捷性。

如果对一个组件的更改会强迫其他依赖于它的组件也发生更改，则它们之间的关系为 紧密 耦合。 松散 耦合会打破这种依赖关系，使存在依赖关系的组件只需了解经过版本控制而且已发布的接口。在依赖项之间实施松散耦合将隔离一个组件中的故障，防止对其他组件造成影响。

松散耦合让您可以为组件增加额外的代码或功能，同时在最大程度上降低依赖于它的组件的风险。而且，随着您可以横向扩展，或甚至更改依赖项的底层实施，可扩展性也得到改善。

要通过松散耦合进一步提升弹性，在可能的情况下采用异步组件交互。若确定对请求进行注册已足够，则此模型适用于无需立即响应的任何交互。它包含一个生成事件的组件和另外一个使用事件的组件。两

个组件不会通过直接点对点交互，但通常经由中间持久存储层集成，例如，SQS 队列或诸如 Amazon Kinesis 或 AWS Step Functions 流数据平台。

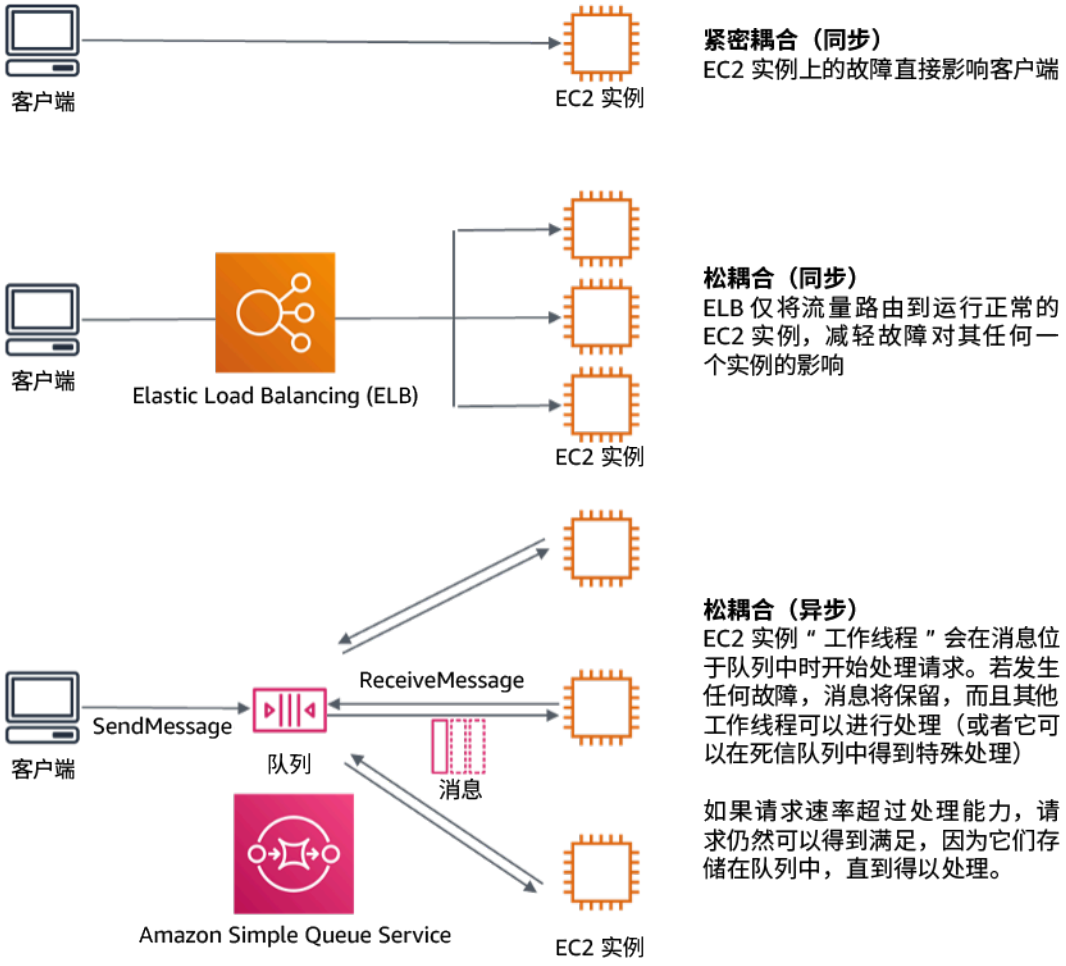


图 4：队列系统和负载均衡器等依赖关系是松散耦合的。

Amazon SQS 队列和 Elastic Load Balancer 只是为松散耦合增加中间层的两种方式。您还可以使用 Amazon EventBridge 在 AWS Cloud 中构建事件驱动型架构，而前者可从其依赖的服务（事件使用器）中提取客户端（事件产生器）。当您需要进行高吞吐量、基于推送的多对多消息收发时，Amazon Simple Notification Service（Amazon SNS）是可供选择的高效解决方案。通过 Amazon SNS 主题，您的发布者系统可以呈扇形将消息分发到大量订阅者终端节点以便进行并行处理。

虽然队列具有多项优点，但在大多数硬性实时系统中，早于阈值时间（通常为秒）的请求应被视为过时（客户端已放弃而且不再等待响应）而不被处理。因此，较新（而且可能依然有效）的请求会被处理。

常见反模式：

- 将单一实例作为工作负载的一部分部署。

- 直接在工作负载层之间调用 API，不具备故障转移或异步处理请求的功能。

建立此最佳实践的好处： 松耦合有助于隔离某个组件的行为与依赖于它的其他组件的行为，从而提升弹性和敏捷性。组件中的故障相互隔离。

未建立此最佳实践暴露的风险等级： 高

实施指导

- 实施松耦合的依赖关系。队列系统、流系统、工作流和负载均衡器等依赖关系是松耦合的。松耦合有助于隔离某个组件的行为与依赖于它的其他组件的行为，从而提升弹性和敏捷性。
  - AWS re:Invent 2019：迁移到事件驱动型架构（SVS308）
  - 什么是 Amazon EventBridge？
  - 什么是 Amazon Simple Queue Service？
    - 您可借助 Amazon EventBridge 构建松耦合的分布式事件驱动型架构。
      - 2019 年 AWS 纽约峰会：介绍事件驱动型架构和 Amazon EventBridge（MAD205）
    - 如果对一个组件的更改会强迫其他依赖于它的组件也发生更改，则它们之间的关系为紧耦合。松耦合会打破这种依赖关系，使存在依赖关系的组件只需了解经过版本控制而且已发布的接口。
    - 让组件在可能的情况下进行异步交互。此模型适用于无需立即响应，只需确认请求已注册就足够的任何交互。
      - AWS re:Invent 2019：使用 Amazon SQS 和 Lambda 的可扩展无服务器事件驱动型应用程序（API304）

资源

相关文档：

- AWS re:Invent 2019：迁移到事件驱动型架构（SVS308）
- Amazon EC2：确保幂等性
- Amazon Builders' Library：分布式系统相关挑战
- Amazon Builders' Library：可靠性、持续工作和安然无忧
- 什么是 Amazon EventBridge？
- 什么是 Amazon Simple Queue Service？

相关视频：

- 2019 年 AWS 纽约峰会：介绍事件驱动型架构和 Amazon EventBridge（MAD205）
- AWS re:Invent 2018：闭环系统和开放思维：如何掌控不同规模的系统（ARC337）（包括松耦合、持续工作和静态稳定性）
- AWS re:Invent 2019：迁移到事件驱动型架构（SVS308）
- AWS re:Invent 2019：使用 Amazon SQS 和 Lambda 的可扩展无服务器事件驱动型应用程序（API304）

REL04-BP03 持续工作

系统会在负载中存在剧烈快速更改时失败。例如，如果您的工作负载执行的一项运行状况检查监控着数千个服务器的运行状况，每次都应发送相同大小的有效负载（当前状态的完整快照）。无论是否有服务器或有多少服务器发生故障，运行状况检查系统都会持续工作，而不会有剧烈、快速的变动。

例如，如果运行状况检查系统正在监控 100000 个服务器，它的标称负载低于通常而言较低的服务器故障率。但如果发生重大事件使一半的服务器运行不正常，则运行状况检查系统会因为尝试更新通知系统以及向其客户端传送状态而变得不堪重负。因此，运行状况检查系统每次应发送当前状态的完整快照。100000 个服务器的运行状况，若每个以一个比特代表，则仅需要 12.5-KB 有效负载。无论是没有服务器发生故障还是所有服务器都发生故障，运行状况检查系统都会持续工作，而大幅度骤变也不会威胁到系统的稳定性。这实际上是 Amazon Route 53 处理端点（例如 IP 地址）的运行状况检查来确定最终用户如何路由到这些端点的方式。

未建立这种最佳实践的情况下暴露的风险等级： 低

实施指导

- 持续工作，使系统不会在负载出现骤变时失败。
- 实施松耦合的依赖关系。队列系统、流系统、工作流和负载均衡器等依赖关系是松耦合的。松耦合有助于隔离某个组件的行为与依赖于它的其他组件的行为，从而提升弹性和敏捷性。
  - Amazon Builders' Library：可靠性、持续工作和安然无忧
  - AWS re:Invent 2018：闭环系统和开放思维：如何掌控不同规模的系统（ARC337）（包括持续工作）
    - 例如，如果运行状况检查系统正在监控 10 万台服务器工程设计工作负载，不论成功或失败的次数，有效负载大小均能保持稳定。

资源

相关文档：

- Amazon EC2：确保幂等性
- Amazon Builders' Library：分布式系统相关挑战
- Amazon Builders' Library：可靠性、持续工作和安然无忧

相关视频：

- 2019 年 AWS 纽约峰会：介绍事件驱动型架构和 Amazon EventBridge（MAD205）
- AWS re:Invent 2018：闭环系统和开放思维：如何掌控不同规模的系统（ARC337）（包括持续工作）
- AWS re:Invent 2018：闭环系统和开放思维：如何掌控不同规模的系统（ARC337）（包括松耦合、持续工作和静态稳定性）
- AWS re:Invent 2019：迁移到事件驱动型架构（SVS308）

REL04-BP04 使所有响应幂等

幂等服务承诺每个请求只完成一次，因此发起多个相同请求与进行单个请求的效果相同。幂等服务使客户端可以轻松进行重试，而不必担心错误地处理多次。要执行此操作，客户端可以发出具有幂等性令牌的 API 请求，每当重复请求时都会使用同一令牌。幂等服务 API 使用令牌返回响应，该响应与首次完成请求时返回的响应相同。

在分布式系统中，至多（客户端仅发起一个请求）或至少（持续发起请求直到客户端收到成功确认）执行某项操作一次并不难。难就难在要保证某项操作具有幕等性，亦即它被 恰好 执行一次，从而使发起多个相同的请求与发起一个请求的效果相同。在 API 中使用幕等性令牌，服务可以一次或多次收到变异请求，而不会创建重复记录或产生副作用。

未建立这种最佳实践的情况下暴露的风险等级： 中

实施指导

- 使所有响应幂等。幂等服务承诺每个请求只完成一次，因此发起多个相同请求与进行单个请求的效果相同。
  - 客户端可以发出具有幂等性令牌的 API 请求，每当重复请求时都会使用同一令牌。幂等服务 API 使用令牌返回响应，该响应与首次完成请求时返回的响应相同。
    - Amazon EC2：确保幂等性

资源

相关文档：

- Amazon EC2：确保幂等性
- Amazon Builders' Library：分布式系统相关挑战
- Amazon Builders' Library：可靠性、持续工作和安然无忧

相关视频：

- 2019 年 AWS 纽约峰会：介绍事件驱动型架构和 Amazon EventBridge（MAD205）
- AWS re:Invent 2018：闭环系统和开放思维：如何掌控不同规模的系统（ARC337）（包括松耦合、持续工作和静态稳定性）
- AWS re:Invent 2019：迁移到事件驱动型架构（SVS308）

REL 5 您如何在分布式系统中进行交互设计，从而缓解或经受住故障影响？

分布式系统依赖于通信网络以便使组件互相连接（如服务器或服务等）。尽管这些网络中存在数据丢失或延迟，但是您的工作负载必须可靠运行。分布式系统组件的运行方式不得对其他组件或工作负载产生负面影响。这些最佳实践使工作负载能够承受压力或故障，从中更快地恢复，并且降低此类伤害的影响。其结果是缩短平均恢复时间（MTTR）。

最佳实践

- REL05-BP01 实施轻松降级以将适用的硬依赖关系转换为软依赖关系
- REL05-BP02 限制请求
- REL05-BP03 控制与限制重试调用
- REL05-BP04 快速失效机制和限制队列
- REL05-BP05 设置客户端超时
- REL05-BP06 尽可能使服务为无状态
- REL05-BP07 实施紧急杠杆

REL05-BP01 实施轻松降级以将适用的硬依赖关系转换为软依赖关系

某个组件的依赖关系运行不正常时，该组件仍可在性能降低的条件下运行。例如，当依赖关系调用失败时，进行故障转移，使用预先确定的静态响应。

假设被服务 A 调用的服务 B 反过来调用服务 C。

图 5：若服务 B 调用服务 C 失败。服务 B 向服务 A 返回降级响应。

当服务 B 调用服务 C 时，它会收到错误或超时消息。而服务 B，因为缺少来自服务 C（及其所包含数据）的响应，则会返回它能够做出的响应。它可以是最后缓存的正确值，或服务 B 可以使用预先确定的静态响应取代它收到的来自服务 C 的响应。然后，向调用方（即服务 A）返回降级响应。若无此静态响应，服务 C 的故障将级联传递至服务 B 和服务 A，因此而丧失可用性。

按照硬依赖关系可用性等式中的倍乘系数（见 使用硬依赖关系计算可用性），C 的可用性的任何降低将严重影响 B 的有效可用性。通过返回静态响应，服务 B 能够缓解 C 中的故障的影响，而且，虽然被降级，可使服务 C 看起来似乎 100% 可用（假设它在错误的情况下可靠地返回静态响应）。注意，静态响应是返回错误的简单替代，而不是使用其他方式对响应进行重新计算的尝试。此类采用完全不同的机制试图达到相同结果的尝试被称作回退行为，是一种要被避免的反模式。

优雅降级的另一个例子是 断路器模式.当故障为临时性时，应采用重试策略。若情况并非如此，而且操作很有可能失败，则断路器模式会防止客户端执行可能失败的请求。系统照常处理请求时，断路器会处于关闭状态，让请求正常通过。当远程系统开始返回错误或出现高延迟，断路器就会打开，依赖项被忽略，或者结果会被更轻松获取但较不全面的响应（可能只是响应缓存）所取代。系统将定期尝试调用依赖项，以确定它是否已恢复。出现这种情况时，断路器将处于关闭状态。

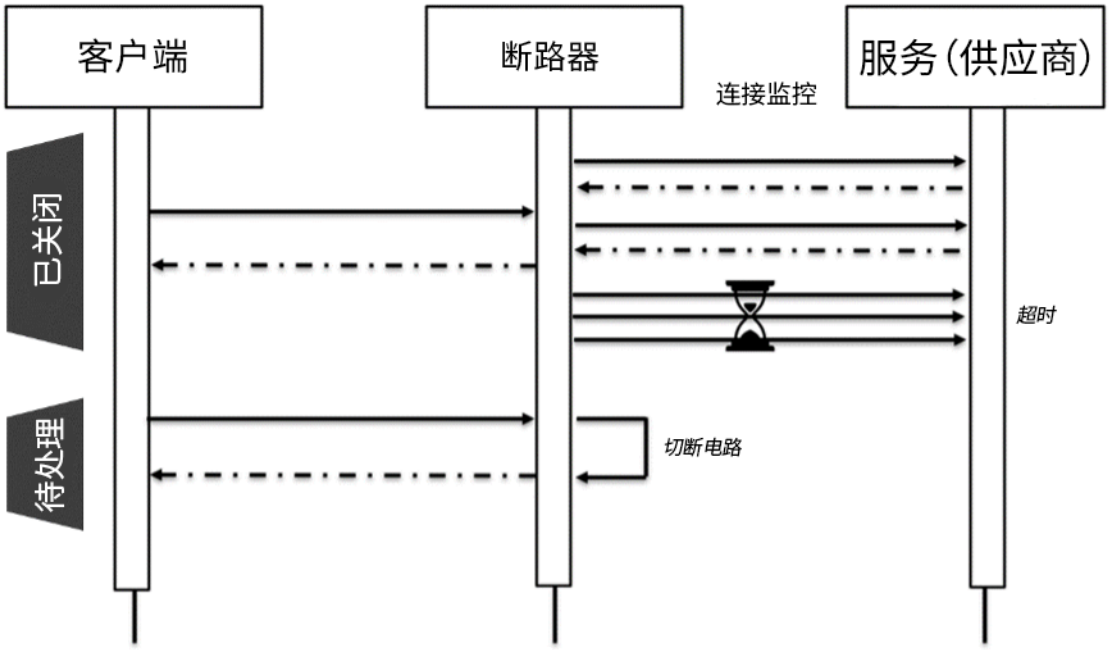


图 6：断路器显示关闭或开启状态。

除了图表中显示的关闭和开启状态，在开启状态内的可配置时间段以后，断路器可能会变为半开启状态。在此状态中，它会以远低于正常水平的速率定期尝试调用服务。此探测器用于检查服务的运行状况。在半开启状态中多次成功以后，断路器会转为关闭，并恢复正常请求。

未建立这种最佳实践的情况下暴露的风险等级： 高

实施指导

- 实施轻松降级以将适用的硬依赖关系转换为软依赖关系。某个组件的依赖关系运行不正常时，该组件仍可在性能降低的条件下运行。例如，当依赖关系调用失败时，进行故障转移，使用预先确定的静态响应。
  - 通过返回静态响应，您的工作负载会缓解其依赖项中发生的故障。
    - Well-Architected 实验室：第 300 级：实施运行状况检查和管理依赖项以提高可靠性
  - 在重试操作可能失败时检测到该情况，并防止您的客户端使用断路器模式进行失败调用。
    - CircuitBreaker

资源

相关文档：

- Amazon API Gateway：对 API 请求限流以提高吞吐量
- CircuitBreaker（对《发布它！》一书中的“断路器”部分进行的总结）
- AWS 中的错误重试和指数回退
- Michael Nygard《发布它！ 设计和部署生产就绪的软件》（Release It! Design and Deploy Production-Ready Software）
- Amazon Builders' Library：避免在分布式系统中回退
- Amazon Builders' Library：避免无法克服的队列积压
- Amazon Builders' Library：缓存挑战和策略
- Amazon Builders' Library：为超时、重试和回退引入抖动

相关视频：

- 重试、回退和抖动：AWS re:Invent 2019：介绍 Amazon Builders’ Library（DOP328）

相关示例：

- Well-Architected 实验室：第 300 级：实施运行状况检查和管理依赖项以提高可靠性

REL05-BP02 限制请求

限制请求是对需求意外增加做出响应的缓解模式。部分请求会得到执行，但超出定义限制的请求会被拒绝，并返回说明它们已被限制的消息。客户端预期将会回退，并且放弃请求或以较低速率进行重试。

您的服务应设计为可以应对每个节点或单元格所能处理的已知请求容量。此容量可以通过负载测试确定。然后，您需要跟踪请求到达速率，如果临时到达速率超过此限制，则相应的响应是发出信号表明请求已被限制。这允许用户进行重试，或许会向可能具有可用容量的另一个节点或单元格发出请求。Amazon API Gateway 提供一些限制请求的方法。Amazon SQS 和 Amazon Kinesis 可对请求进行缓冲，平滑请求速率并降低对可异步处理的请求进行限制的需求。

未建立此最佳实践暴露的风险等级： 高

实施指导

- 限制请求。这是对按需求意外增加做出响应的缓解模式。部分请求会得到执行，但超出定义限制的请求会被拒绝，并返回说明它们已被限制的消息。客户端预期将会回退，并且放弃请求或以较低速率进行重试。
  - 使用 Amazon API Gateway

- 对 API 请求限流以提高吞吐量

资源

相关文档：

- Amazon API Gateway：对 API 请求限流以提高吞吐量
- AWS 中的错误重试和指数回退
- Amazon Builders' Library：避免在分布式系统中回退
- Amazon Builders' Library：避免无法克服的队列积压
- Amazon Builders' Library：为超时、重试和回退引入抖动
- 对 API 请求限流以提高吞吐量

相关视频：

- 重试、回退和抖动：AWS re:Invent 2019：介绍 Amazon Builders' Library（DOP328）

REL05-BP03 控制与限制重试调用

在逐渐延长的间隔以后使用指数回退进行重试。引入抖动使此类重试间隔随机化，并限制重试的最大次数。

分布式软件系统中的常见组件包括服务器、负载均衡器、数据库和 DNS 服务器。在操作中，受故障影响，任何此类组件都可能开始生成错误。处理错误的默认方式为，在客户端实施重试。此方法可提高应用程序的可靠性和可用性。不过，如果规模较大，而且客户端在错误发生时立即重试失败的操作，网络中的新请求和重试请求可能会快速导致饱和，并争用网络带宽。这可能导致 重试风暴， 从而降低服务的可用性。此模式可能会继续，直到发生全系统故障。

为避免出现此情况，应使用回退算法，如常用的 指数回退 。指数回退算法会逐渐降低执行重试的速率，从而避免网络阻塞。

很多开发工具包和软件库（包括 AWS 的开发工具包和软件库）都实施此类算法的一种版本。但是， 别心存侥幸地认为已采用回退算法，始终要进行测试和验证。

仅依靠回退还不够，因为在分布式系统中，所有客户端都可能同步回退，形成重试调用集群。Marc Brooker 在他的博文 指数回退和抖动中解释了如何修改指数回退中的 wait() 函数以防止形成重试调用集群。他给出的解决办法 是 在 wait() 函数中增加抖动。要避免过长时间的重试，实施应为回退设置一个最大值限制。

最后，务必要配置 最大重试次数 或已用时间，在此以后，重试将失败。AWS 开发工具包将默认实施此操作，而且也可以对它进行配置。针对堆栈中较低的服务，最大重试限制为 0 或 1 可以限制风险，而且在将重试委派给堆栈较高的服务时依然有效。

未建立这种最佳实践的情况下暴露的风险等级： 高

实施指导

- 控制与限制重试调用。在逐渐延长的间隔以后使用指数回退进行重试。引入抖动使此类重试间隔随机化，并限制重试的最大次数。
  - AWS 中的错误重试和指数回退
    - 默认情况下，Amazon SDK 实施重试和指数回退。在调用自己的依赖服务时，您需要在依赖关系层中执行类似的逻辑。根据您的使用案例确定超时以及何时停止重试。

资源

相关文档：

- Amazon API Gateway：对 API 请求限流以提高吞吐量
- AWS 中的错误重试和指数回退
- Amazon Builders' Library：避免在分布式系统中回退
- Amazon Builders' Library：避免无法克服的队列积压
- Amazon Builders' Library：缓存挑战和策略
- Amazon Builders' Library：为超时、重试和回退引入抖动

相关视频：

- 重试、回退和抖动：AWS re:Invent 2019：介绍 Amazon Builders' Library（DOP328）

REL05-BP04 快速失效机制和限制队列

如果工作负载无法成功响应请求，则快速试错。这样可释放与请求关联的资源，并允许该服务在资源不足的情况下恢复。如果工作负载能够成功响应，但请求速率过高，则使用队列来对请求进行缓冲。不过，不要允许使用长队列，它可能导致处理已被客户端放弃的过时请求。

此最佳实践适用于请求的服务器端，或接收方。

请注意，可在系统的多个级别创建队列，它们可能严重妨碍快速恢复的能力，因为需要先处理较旧的过时请求（虽然不再需要响应），然后才会处理较新的请求。另外还要注意队列所在的位置。它们通常会隐藏在工作流或记录到数据库的工作中。

未建立此最佳实践暴露的风险等级： 高

实施指导

- 快速失效机制和限制队列。如果工作负载无法成功响应请求，则快速试错。这样可释放与请求关联的资源，并允许该服务在资源不足的情况下恢复。如果工作负载能够成功响应，但请求速率过高，则使用队列来对请求进行缓冲。不过，不要允许使用长队列，它可能导致处理已被客户端放弃的过时请求。
  - 在服务面临压力时执行快速失效机制。
    - 快速试错
  - 限制队列：在基于队列的系统中，如果在停止处理后消息仍不断涌入，则消息债务可能造成大量积压，从而增加处理时间。工作完成太晚，以至于结果无法发挥作用，从根本上导致了队列原本要避免的可用性打击问题。
    - Amazon Builders' Library：避免无法克服的队列积压

资源

相关文档：

- AWS 中的错误重试和指数回退
- 快速试错
- Amazon Builders' Library：避免在分布式系统中回退
- Amazon Builders' Library：避免无法克服的队列积压
- Amazon Builders' Library：缓存挑战和策略
- Amazon Builders' Library：为超时、重试和回退引入抖动

相关视频：

- 重试、回退和抖动：AWS re:Invent 2019：介绍 Amazon Builders’ Library（DOP328）

REL05-BP05 设置客户端超时

适当设置超时，对它们进行系统性验证，而且不要依靠默认值，因为默认值通常设置得过高。

此最佳实践适用于请求的客户端，或发送方。

为任何远程调用和大体上任何跨流程调用设置连接超时和请求超时。许多框架具有内置超时功能，但仍需谨慎，因为许多默认值为无限值或过高。过高的值会降低超时的实用性，因为客户端等待超时发生时，系统会继续消耗资源。过低的值可能因为要重试过多请求而导致后端流量增加以及延迟变长。在有些情况下，由于要对全部请求进行重试，从而可能导致完全中断。

要了解关于 Amazon 如何利用超时、重试和抖动回退的更多信息，请参阅 构建者库：超时、重试和抖动回退.

未建立这种最佳实践的情况下暴露的风险等级： 高

实施指导

- 为任何远程调用和大体上任何跨流程调用设置连接超时和请求超时。许多框架具有内置超时功能，但仍需谨慎，因为许多默认值为无限值或过高。过高的值会降低超时的实用性，因为客户端等待超时发生时，系统会继续消耗资源。过低的值可能因为要重试过多请求而导致后端流量增加以及延迟变长。在有些情况下，由于要对全部请求进行重试，从而可能导致完全中断。
  - AWS 开发工具包：重试次数和超时

资源

相关文档：

- AWS 开发工具包：重试次数和超时
- Amazon API Gateway：对 API 请求限流以提高吞吐量
- AWS 中的错误重试和指数回退
- Amazon Builders' Library：为超时、重试和回退引入抖动

相关视频：

- 重试、回退和抖动：AWS re:Invent 2019：介绍 Amazon Builders’ Library（DOP328）

REL05-BP06 尽可能使服务为无状态

服务应该不需要状态，或者在不同的客户端请求之间卸载状态，磁盘上和内存中本地存储的数据不存在依赖关系。这使任意替换服务器成为可能，而且不会对可用性产生影响。Amazon ElastiCache 或 Amazon DynamoDB 是卸载状态的理想目标位置。

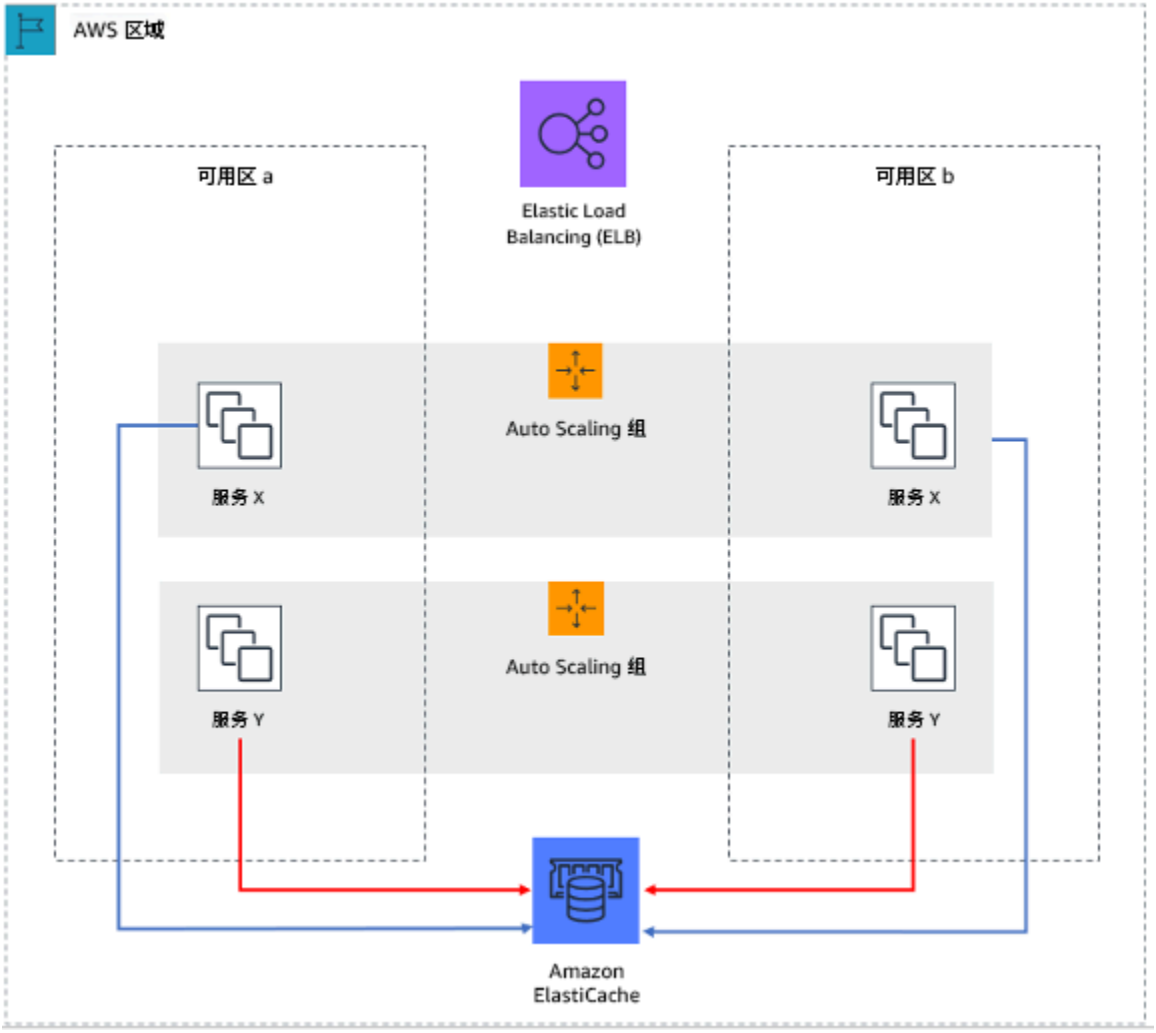


图 7：在此无状态 Web 应用程序中，会话状态会被卸载到 Amazon ElastiCache。

当用户或服务与应用程序进行交互，它们通常会执行一系列交互并构成一次会话。对于用户来说，会话是他们在使用应用程序时持续存在于请求之间的特殊数据。无状态应用程序是无需掌握之前交互而且不会存储会话信息的应用程序。

若采用无状态设计，则您可以使用无服务器计算服务，如 AWS Lambda 或 AWS Fargate。

除了服务器替换，无状态应用程序的另一项优点是，由于任何可用的计算资源（如 EC2 实例和 AWS Lambda 函数）都可以处理任何请求，因此它们可以进行横向扩展。

未建立此最佳实践暴露的风险等级： 中

实施指导

- 让应用程序无状态。无状态应用程序支持水平扩展，并且可以承受单个节点故障。

- 删除可能存储在请求参数中的状态。
- 在检查是否需要状态之后，将任何状态追踪移动到具有弹性的多区域缓存或数据存储（如 Amazon ElastiCache、Amazon RDS、Amazon DynamoDB 或第三方分布式数据解决方案）。存储无法移动到弹性数据存储的状态。
  - 某些数据（例如 cookie）可能在标头或查询参数中传递。
  - 进行重构，从而删除可能在请求中快速传递的状态。
  - 提交请求时实际上并不需要某些数据，这些数据可以按需检索。
  - 删除可以异步检索的数据。
  - 确定满足所需状态要求的数据存储。
  - 考虑针对非关系型数据使用 NoSQL 数据库。

资源

相关文档：

- Amazon Builders' Library：避免在分布式系统中回退
- Amazon Builders' Library：避免无法克服的队列积压
- Amazon Builders' Library：缓存挑战和策略

REL05-BP07 实施紧急杠杆

紧急杠杆是可帮助您的工作负载减轻可用性影响的快速流程。

未建立这种最佳实践的情况下暴露的风险等级： 中

实施指导

- 实施紧急杠杆。这些是可帮助您的工作负载减轻可用性影响的快速流程。即使未找到根本原因，它们也可以运行。理想的紧急杠杆可通过提供完全确定的激活与停用标准，将解析器的认知负担降低到零。杠杆通常需要手动操作，但也可实现自动化
  - 杠杆示例包括，
    - 阻止所有机器人流量
    - 为静态页面而非动态页面提供服务
    - 减少对依赖项的调用频率
    - 限制来自依赖项的调用

- 关于实施和使用紧急杠杆的提示
  - 当杠杆被激活时，求少不求多
  - 保持简单，避免双模态行为
  - 定期测试您的杠杆
- 以下为非紧急杠杆的操作示例
  - 添加容量
  - 号召依赖您的服务的客户端服务所有者，要求他们降低调用
  - 更改代码并将其释放

变更管理

问题

- REL 6 如何监控工作负载资源？
- REL 7 您如何设计工作负载，以适应不断变化的需求？
- REL 8 如何实施更改？

REL 6 如何监控工作负载资源？

日志和指标是用于了解工作负载运行状况的强大工具。您可以配置工作负载以监控日志和指标，并在超出阈值或发生重大事件时发送通知。监控让您的工作负载可以发现超出低性能阈值和发生故障的情形，从而在响应中自动恢复。

最佳实践

- REL06-BP01 为工作负载监控全部组件（生成）
- REL06-BP02 定义与计算指标（聚合）
- REL06-BP03 发送通知（实时处理和报警）
- REL06-BP04 自动响应（实时处理和告警）
- REL06-BP05 分析
- REL06-BP06 定期进行审核
- REL06-BP07 对通过系统的请求进行端到端跟踪监控

REL06-BP01 为工作负载监控全部组件（生成）

使用 Amazon CloudWatch 或第三方工具监控工作负载组件。使用 AWS Health 控制面板监控 AWS 服务。

应监控您的工作负载的全部组件，包括前端、业务逻辑和存储层。定义关键指标，描述如何将其从日志中提取出来（如有必要），并且设置用于触发对应警报事件的阈值。确保这些指标与您工作负载的关键性能指标（KPI，Key Performance Indicator）相关，并使用指标和日志来确定服务性能下降的早期警告信号。例如，每分钟成功处理的订单数等与业务成果相关的指标，相比 CPU 利用率等技术指标，可以更快地指示工作负载问题。使用 AWS Health 控制面板提供 AWS 资源底层的 AWS 服务的性能和可用性的个性化视图。

云中监控创造新的机会。大多数云提供商都已经开发出可自定义的挂钩，可以提供分析洞察来帮助您监控工作负载的多个层。Amazon CloudWatch 等 AWS 服务应用统计和机器学习算法，集中分析系统与应用程序的指标，确定正常基准，并发现异常，同时最大程度地减少用户干预。异常检测算法考虑了指标的季节性和趋势变动。

AWS 提供了丰富的监控和日志信息以供使用，这些信息可用于定义特定于工作负载的指标、需求变化流程并且采用机器学习技术而无需机器学习专业知识。

此外还会监控您的所有外部端点，确保它们独立于基本实施。这种主动监控可通过合成事务（有时被称为 用户金丝雀，但不要与金丝雀部署相混淆）实现，它们会按照工作负载的客户端所执行的操作，定期执行许多常见任务。请确保这些任务的持续时间较短，并且在测试期间不要使您的工作流过载。Amazon CloudWatch Synthetics 使您能够 创建合成金丝雀 以便监控您的终端节点和 API。您还可以整合 Synthetic Canary 客户端节点和 AWS X-Ray 控制台，精确定位哪些 Synthetic Canary 遇到错误、故障，或对指定时段的速率进行限制的问题。

期望结果：

从工作负载的所有组件收集并使用关键指标，用于确保工作负载的可靠性和提供最佳用户体验。通过检测未能实现业务成果的工作负载，您可以快速发现灾难并从意外事件中恢复。

常见反模式：

- 仅监控连接到工作负载的外部接口。
- 未生成任何特定于工作负载的指标，并且只依靠工作负载所用的 AWS 服务提供给您的指标。
- 仅使用工作负载中的技术指标，不监控与工作负载所带来的非技术 KPI 相关的任何指标。
- 依靠生产流量和简单的运行状况检查来监控并评估工作负载状态。

建立此最佳实践的好处： 在工作负载的所有层级进行监控，方便您更快地预测并解决组成工作负载的组件中的问题。

未建立这种最佳实践的情况下暴露的风险等级： 高

实施指导

1. 启用日志记录功能（如适用）。 应从工作负载的所有组件获取监控数据。启用额外的日志记录，例如 S3 访问日志，并使工作负载记录特定于工作负载的数据。从 Amazon ECS、Amazon EKS、Amazon EC2、Elastic Load Balancing、AWS Auto Scaling 和 Amazon EMR 等服务收集 CPU、网络 I/O 和磁盘 I/O 平均值等指标。请参阅 发布 CloudWatch 指标的 AWS 服务 以了解将指标发布到 CloudWatch 的 AWS 服务列表。
2. 审查所有默认指标并探究任何数据收集欠缺。 每项服务都生成默认指标。通过收集默认指标，您可以更好地了解工作负载组件之间的依赖关系，以及组件的可靠性和性能如何影响工作负载。您还可以 使用 AWS CLI 或 API 创建自己的指标并发布到 CloudWatch。这将
3. 评估所有指标，以确定对于工作负载中的每个 AWS 服务，需要针对哪些指标发布警报。 您还可以选择对工作负载可靠性有重大影响的指标的子集。专注于关键指标和阈值让您可以精调 警报 的数量，并可帮助尽可能减少误报。
4. 定义警报，以及在触发警报之后工作负载的恢复流程。 通过定义警报，您可以快速通知、上报意外事件，按照必要的步骤从意外事件中恢复，并满足规定的恢复时间目标（RTO，Recovery Time Objective）。您可以使用 Amazon CloudWatch 告警 ，根据定义的阈值来调用自动工作流并启动恢复程序。
5. 探索使用合成事务来收集有关工作负载状态的相关数据。 合成监控遵循与客户相同的路线并执行相同的操作，这使得您可以持续验证客户体验，甚至在您的工作负载上没有任何客户流量时也可以。通过使用 合成事务，您可以先于客户发现问题。

资源

相关最佳实践：

- REL11-BP03 自动修复所有层

相关文档：

- 开始使用 AWS Health 控制面板 – 账户的运行状况
- 发布 CloudWatch 指标的 AWS 服务
- Network Load Balancer 的访问日志

- 应用程序负载均衡器的访问日志
- 访问 AWS Lambda 的 Amazon CloudWatch Logs
- Amazon S3 服务器访问日志记录
- 启用 Classic Load Balancer 的访问日志
- 将日志数据导出到 Amazon S3
- 在 Amazon EC2 实例上安装 CloudWatch 代理
- 发布自定义指标
- 使用 Amazon CloudWatch 控制面板
- 使用 Amazon CloudWatch 指标
- 使用金丝雀（Amazon CloudWatch Synthetics）
- 什么是 Amazon CloudWatch Logs？

用户指南：

- 创建跟踪记录
- 监控 Amazon EC2 Linux 实例的内存和磁盘指标
- 结合使用 CloudWatch Logs 与容器实例
- VPC 流日志
- 什么是 Amazon DevOps Guru？
- 什么是 AWS X-Ray？

相关博客：

- 使用 Amazon CloudWatch Synthetics 和 AWS X-Ray 进行调试

相关示例和研讨会：

- AWS Well-Architected 实验室：卓越运营 – 依赖项监控
- Amazon Builders' Library：检测分布式系统的运营可见性
- 可观测性研讨会

REL06-BP02 定义与计算指标（聚合）

存储日志数据并在必要时应用筛选条件以计算指标，例如，特定日志事件的数量，或从日志事件时间戳计算得到的延迟。

Amazon CloudWatch 和 Amazon S3 充当主要聚合层和存储层。某些服务（如 AWS Auto Scaling 和 Elastic Load Balancing）针对整个集群或实例，默认情况下为 CPU 负载或平均请求延迟提供了一些默认指标。对于流式处理服务（如 VPC 流日志和 AWS CloudTrail），事件数据将被转发给 CloudWatch Logs，您需要定义和应用指标筛选条件，才能从事件数据中提取指标。这为您提供了时间序列数据，可被输入到您定义的触发提醒的 CloudWatch 警报。

未建立此最佳实践暴露的风险等级： 高

实施指导

- 定义与计算指标（聚合）。存储日志数据并在必要时应用筛选条件以计算指标，例如，特定日志事件的数量，或从日志事件时间戳计算得到的延迟
  - 指标筛选条件定义在将日志数据发送到 CloudWatch Logs 中时所查找的术语和模式。CloudWatch Logs 使用这些指标筛选条件将日志数据转换为 CloudWatch 数字指标，您可以对这些指标绘制图形或设置警报。
    - 搜索和筛选日志数据
  - 使用受信任第三方来聚合日志。
    - 遵循第三方的说明。大多数第三方产品可以与 CloudWatch 和 Amazon S3 集成。
  - 某些 AWS 服务可以直接向 Amazon S3 发布日志。如果您的主要需求是将日志存储在 Amazon S3 中，则可以让生成日志的服务轻松将其直接发送至 Amazon S3，无需设置额外的基础设施。
    - 将日志直接发送到 Amazon S3

资源

相关文档：

- Amazon CloudWatch Logs Insights 查询示例
- 使用 Amazon CloudWatch Synthetics 和 AWS X-Ray 进行调试
- 可观测性研讨会
- 搜索和筛选日志数据
- 将日志直接发送到 Amazon S3

- Amazon Builders' Library：检测分布式系统的运营可见性

REL06-BP03 发送通知（实时处理和报警）

发生重大事件时，需要知晓的组织会收到通知。

警报可以发送到 Amazon Simple Notification Service（Amazon SNS）主题中，然后推送给任意数量的订阅者。例如，Amazon SNS 可以将提醒转发给某个电子邮件别名，以便技术人员可以回复。

常见反模式：

- 配置过低的告警阈值，导致发送过多通知。
- 未存档告警以备将来查看。

建立此最佳实践的好处： 事件通知（即使是可以响应并自动解决的事件）允许您记录事件，还可用于在将来通过其他方式处理事件。

未建立这种最佳实践的情况下暴露的风险等级： 高

实施指导

- 执行实时处理和报警。发生重大事件时，需要知晓的组织会收到通知
  - Amazon CloudWatch 控制面板是 CloudWatch 控制台中的可自定义主页，方便您通过单一视图监控您的资源，即使这些资源分布在不同区域中。
    - 使用 Amazon CloudWatch 控制面板
  - 创建指标超过限制时发出的警报。
    - 使用 Amazon CloudWatch 告警

资源

相关文档：

- 可观测性研讨会
- Amazon Builders' Library：检测分布式系统的运营可见性
- 使用 Amazon CloudWatch 告警
- 使用 Amazon CloudWatch 控制面板
- 使用 Amazon CloudWatch 指标

REL06-BP04 自动响应（实时处理和告警）

检测到事件后，利用自动化功能执行操作；例如，更换故障组件。

提醒可以触发 AWS Auto Scaling 事件，以便集群对需求的变化做出反应。警报还可以发送到 Amazon Simple Queue Service（Amazon SQS），后者可充当第三方票证系统的集成点。AWS Lambda 还可以订阅警报，为用户提供一种无服务器的异步模式，以动态方式应对更改。AWS Config 会持续监视和记录您的 AWS 资源配置，并且可以触发 AWS Systems Manager Automation 以修正问题。

Amazon DevOps Guru 可以自动监控应用程序资源的异常行为并提供针对性的建议，以缩短识别问题和进行修复所需的时间。

未建立这种最佳实践的情况下暴露的风险等级： 中

实施指导

- 使用 Amazon DevOps Guru 执行自动化操作。Amazon DevOps Guru 可以自动监控应用程序资源的异常行为并提供针对性的建议，以缩短识别问题和进行修复所需的时间。
  - 什么是 Amazon DevOps Guru？
- 使用 AWS Systems Manager 执行自动化操作。AWS Config 会持续监控和记录您的 AWS 资源配置，还可以触发 AWS Systems Manager Automation 以修正问题。
  - AWS Systems Manager Automation
    - 创建和使用 Systems Manager Automation 文档。它们可定义当自动化流程运行时，Systems Manager 对托管实例和其他 AWS 资源执行的操作。
    - 使用自动化文档（行动手册）
- Amazon CloudWatch 将警报状态更改事件发送到 Amazon EventBridge。创建 EventBridge 规则以自动做出响应。
  - 创建通过 AWS 资源中的事件触发的 EventBridge 规则
- 创建和执行自动响应计划。
  - 盘点所有警报响应程序。您必须在对任务排名之前制定警报响应计划。
  - 盘点所有必须执行特定操作的任务。大多数操作均已记录在运行手册中。您还必须具有针对意外事件的警报行动手册。
  - 检查所有可自动化操作的运行手册和行动手册。一般而言，可定义的操作很可能可以实现自动化。
  - 将容易出错或耗时的活动排在前列。这非常有助于删除错误源和缩短解决问题的时间。
  - 制定计划，完成自动化。维护有效计划，以自动执行并更新自动化。
  - 检查手动要求，寻找自动化机会。挑战手动流程，发现自动化机会。

资源

相关文档：

- AWS Systems Manager Automation
- 创建通过 AWS 资源中的事件触发的 EventBridge 规则
- 可观测性研讨会
- Amazon Builders' Library：检测分布式系统的运营可见性
- 什么是 Amazon DevOps Guru？
- 使用自动化文档（行动手册）

REL06-BP05 分析

收集日志文件和指标历史，并对其进行分析以获得更广泛的趋势和工作负载见解。

Amazon CloudWatch Logs Insights 支持 简单但强大的查询语言，您可以用它分析日志数据。Amazon CloudWatch Logs 还支持订阅，允许数据无缝流动到 Amazon S3（您可以在其中使用此类数据）或 Amazon Athena 以便对数据进行查询。它还支持查询多种格式。请参阅 支持的 SerDes 和数据格式（参见 Amazon Athena 用户指南）。针对大型日志文件集的分析，您可以运行 Amazon EMR 集群以执行 PB 级分析。

AWS 合作伙伴和第三方提供了许多用于聚合、处理、存储和分析的工具。这些工具包括 New Relic、Splunk、Loggly、Logstash、CloudHealth 和 Nagios。但是，系统和应用程序日志之外的生成对于每个云提供商，甚至每个服务来说都是独一无二的。

监控过程中常常被忽视的部分是数据管理。您需要确定数据监控的保留要求，然后相应地应用生命周期策略。Amazon S3 支持 S3 存储桶级别的生命周期管理。此生命周期管理可以通过不同的方式应用到存储桶中的不同路径。您可以在生命周期临近结束时，将数据转移到 Amazon S3 Glacier 进行长期存储，然后在保留期结束后让它们过期。S3 智能分层存储类旨在通过将数据自动移动到最具成本效益的访问层，而不会对性能或运营开销产生影响，从而实现优化成本的目的。

未建立此最佳实践暴露的风险等级： 中

实施指导

- 借助 CloudWatch Logs Insights，您可对 Amazon CloudWatch Logs 中的日志数据进行交互搜索和分析。
  - 使用 CloudWatch Logs Insights 分析日志数据

    - Amazon CloudWatch Logs Insights 查询示例
- 使用 Amazon CloudWatch Logs 将日志发送到 Amazon S3，您可以在此处使用这些日志或者使用 Amazon Athena 来查询数据。
    - 如何使用 Athena 分析我的 Amazon S3 服务器访问日志？
        - 为服务器访问日志存储桶创建 S3 生命周期策略。配置生命周期策略以定期删除日志文件。这样做可以减少 Athena 针对每次查询分析的数据量。
            - 如何为 S3 存储桶创建生命周期策略？

资源

相关文档：

- Amazon CloudWatch Logs Insights 查询示例
- 使用 CloudWatch Logs Insights 分析日志数据
- 使用 Amazon CloudWatch Synthetics 和 AWS X-Ray 进行调试
- 如何为 S3 存储桶创建生命周期策略？
- 如何使用 Athena 分析我的 Amazon S3 服务器访问日志？
- 可观测性研讨会
- Amazon Builders' Library：检测分布式系统的运营可见性

REL06-BP06 定期进行审核

经常审核工作负载监控的实施情况，并根据重大事件和变更加以更新。

关键业务指标可促进有效监控。确保随着业务优先事项的变化在您的工作负载中对这些指标进行调整。

审计监控有助于确保您了解应用程序何时达到其可用性目标。根本原因分析需要具备在出现故障时发现具体情况的能力。AWS 提供的服务让您能够在意外事件发生期间跟踪服务的状态：

- Amazon CloudWatch Logs： 您可以将日志存储在此服务中并检查日志内容。
- Amazon CloudWatch Logs Insights：是一项完全托管式服务，让您可以在数秒内分析大量日志。它为您提供快速、交互式的查询和可视化。
- AWS Config： 您可以查看在不同的时间点使用了哪些 AWS 基础设施。
- AWS CloudTrail： 您可以查看哪些委托人在什么时候调用了哪些 AWS API。

AWS 每周召开一次会议，以审查运营性能并在团队之间分享经验。因为 AWS 有很多团队，我们设置了 The Wheel 以随机挑选一个工作负载进行审查。定期开展运营性能审查和知识共享，有助于您增强帮助运营团队提高绩效的能力。

常见反模式：

- 仅收集默认指标。
- 设置监控策略后不再过问。
- 部署重大更改后不讨论监控问题。

建立此最佳实践的好处： 定期审核监控可主动预测潜在问题，而不是当预测问题真实发生后被动应对通知。

未建立此最佳实践暴露的风险等级： 中

实施指导

- 为工作负载创建多个控制面板。您必须具有顶级控制面板，其中包含关键业务指标，以及已确定与使用情况发生变化时工作负载的预期运行状况最相关的技术指标。您还应该具有可以检查各种应用程序层和依赖项的控制面板。
  - 使用 Amazon CloudWatch 控制面板
- 计划和执行工作负载控制面板常规检查。执行控制面板常规检查。您可能对检查深度具有不同的安排。
  - 检查指标中的趋势。对比指标值与历史值，了解是否有趋势表明需要调查某些情况。这种情况的示例包括：延迟增加、主要业务功能减少以及故障响应增加。
  - 检查指标中的离群值/异常值。平均值或中值会掩盖离群值和异常值。查看时间范围内的最高值和最低值，调查出现这些极值的原因。当您继续消除这些原因时，降低对极值的定义可以使您继续提高工作负载性能的一致性。
  - 查找清晰的行为变化。指标数量或方向的立即更改可能表示应用程序出现更改，或者出现了您需要添加额外指标进行跟踪外部因素。

资源

相关文档：

- Amazon CloudWatch Logs Insights 查询示例
- 使用 Amazon CloudWatch Synthetics 和 AWS X-Ray 进行调试

- 可观测性研讨会
- Amazon Builders' Library：检测分布式系统的运营可见性
- 使用 Amazon CloudWatch 控制面板

REL06-BP07 对通过系统的请求进行端到端跟踪监控

利用 AWS X-Ray 或第三方工具，开发人员可以更轻松地分析与调试分布式系统，了解他们的应用程序及其底层服务的表现。

未建立此最佳实践暴露的风险等级： 中

实施指导

- 对通过系统的请求进行端到端跟踪监控。AWS X-Ray 服务用于收集有关应用程序所服务的请求的数据，并提供工具来供您用来查看、筛选和深入了解该数据，以识别问题和优化机会。对于有关应用程序的任何跟踪请求，您将不仅可以查看有关请求和响应的详细信息，还可以查看应用程序对下游 AWS 资源、微服务、数据库和 Web API 进行调用的详细信息。
  - 什么是 AWS X-Ray？
  - 使用 Amazon CloudWatch Synthetics 和 AWS X-Ray 进行调试

资源

相关文档：

- 使用 Amazon CloudWatch Synthetics 和 AWS X-Ray 进行调试
- 可观测性研讨会
- Amazon Builders' Library：检测分布式系统的运营可见性
- 使用金丝雀（Amazon CloudWatch Synthetics）
- 什么是 AWS X-Ray？

REL 7 您如何设计工作负载，以适应不断变化的需求？

可扩展工作负载具有自动添加或移除资源的弹性，因此确保在任何时间点都能准确满足当前的需求。

最佳实践

- REL07-BP01 在获取或扩展资源时利用自动化

- REL07-BP02 在检测到对工作负载的破坏时获取资源
- REL07-BP03 当检测到某个工作负载需要更多资源时，就会获取资源
- REL07-BP04 对工作负载进行负载测试

REL07-BP01 在获取或扩展资源时利用自动化

在替换被损坏的资源或扩展您的工作负载时，通过采用托管 AWS 服务（如 Amazon S3 和 AWS Auto Scaling）对流程进行自动处理。您还可以使用第三方工具和 AWS 开发工具包自动扩展。

托管 AWS 服务包括 Amazon S3、Amazon CloudFront、AWS Auto Scaling、AWS Lambda、Amazon DynamoDB、AWS Fargate 和 Amazon Route 53。

AWS Auto Scaling 让您可以检测与替换被破坏的实例。它还可以帮助您为资源制定扩展计划，包括 Amazon EC2 实例和 Spot 队列、 Amazon ECS 任务、 Amazon DynamoDB 表和索引，以及 Amazon Aurora 副本。

在扩展 EC2 实例时，请确保您使用多个可用区（最好至少三个）并增加或减少容量以保持这些可用区之间的平衡。ECS 任务或 Kubernetes 容器组（pod）（使用 Amazon Elastic Kubernetes Service 时）也应分布在多个可用区中。

如果使用 AWS Lambda，实例会自动扩展。每次收到关于您的函数的事件通知时，AWS Lambda 会快速找到其计算队列中的可用容量，然后运行您的代码至分配的并发值。您需要确保在特定的 Lambda 上，以及在您的Service Quotas中配置必要的并发值。

Amazon S3 会自动扩展以处理较高的请求速率。例如，您的应用程序可以在存储桶中为每个前缀每秒至少发送 3500 个 PUT/COPY/POST/DELETE 或 5500 个 GET/HEAD 请求。存储桶中的前缀数量没有限制。您可以通过并行化读取提高您的读取或写入性能。例如，如果在 Amazon S3 存储桶中创建 10 个前缀以便对读取进行并行化，您可以将读取性能扩展至每秒 55000 个读取请求。

配置和使用 Amazon CloudFront 或受信任的内容分发网络（CDN，Content Delivery Network）。CDN 可以缩短最终用户的响应时间，并从缓存中为请求提供内容，从而减少扩展工作负载的请求。

常见反模式：

- 实施 Auto Scaling 组进行自动修复，但无法实施弹性。
- 使用 Auto Scaling 响应流量激增。
- 部署高状态应用程序，消除了部署弹性选项。

建立此最佳实践的好处： 自动化可以避免部署和淘汰资源时的潜在手动错误。自动化可以避免由于缓慢响应部署或淘汰需求而导致的服务超支和拒绝服务风险。

未建立此最佳实践暴露的风险等级： 高

实施指导

- 配置和使用 AWS Auto Scaling。它会监控您的应用程序，并自动调整容量来维持稳定、可预测的性能，并且成本最低。使用 AWS Auto Scaling，您可以跨多个服务为多个资源轻松设置应用程序扩展。
  - 什么是 AWS Auto Scaling？
    - 在您的 Amazon EC2 实例和竞价型实例集、Amazon ECS 任务、Amazon DynamoDB 表和索引、Amazon Aurora 副本以及 AWS Marketplace 设备上配置自动扩展（如果适用）。
      - 使用 DynamoDB Auto Scaling 自动管理吞吐能力
        - 使用服务 API 操作来指定警报、扩展策略、预热时间和冷却时间。
- 使用 Elastic Load Balancing。负载均衡器可以按路径或网络连接分配负载。
  - 什么是 Elastic Load Balancing？
    - Application Load Balancers 可以按路径分配负载。
      - 什么是 Application Load Balancer？
        - 配置 Application Load Balancer，根据域名下的路径将流量分配给不同的工作负载。
        - Application Load Balancers 可以与 AWS Auto Scaling 集成来分配负载，以便管理需求。
          - 将负载均衡器与自动扩缩组配合使用
    - 网络负载均衡器可以按连接分配负载。
      - 什么是网络负载均衡器？
        - 配置网络负载均衡器，以便使用 TCP 将流量分配给不同的工作负载，或者为工作负载指定一组恒定的 IP 地址。
        - 网络负载均衡器可以与 AWS Auto Scaling 集成来分配负载，以便管理需求。
- 使用高度可用的 DNS 提供商。使用 DNS 名称，用户可以输入名称而不是 IP 地址来访问您的工作负载，并将该信息分发到指定的范围内，通常面向全局范围内工作负载的所有用户。
  - 使用 Amazon Route 53 或可信 DNS 提供商。
    - 什么是 Amazon Route 53？
  - 使用 Route 53 管理 CloudFront 分配和负载均衡器。
    - 确定要管理的域和子域。
    - 使用 ALIAS 或 CNAME 记录来创建适当的记录集。

    - 使用记录
- 使用 AWS 全球网络可优化用户与应用程序之间的路径。AWS Global Accelerator 持续监控应用程序端点的运行状况，可在 30 秒内将流量重定向到运行状况良好的端点。
    - AWS Global Accelerator 是一项可帮助本地或全球用户提高应用程序可用性和性能的服务。它提供的静态 IP 地址可用作从单个或多个 AWS 区域 区域（例如 Application Load Balancers、网络负载均衡器或 Amazon EC2 实例）访问应用程序端点的固定入口点。
        - 什么是 AWS Global Accelerator？
- 配置和使用 Amazon CloudFront 或受信任的内容分发网络（CDN，Content Delivery Network）。内容分发网络可以缩短最终用户的响应时间，还可以对可能导致工作负载进行不必要扩展的内容请求做出响应。
    - 什么是 Amazon CloudFront？
        - 针对您的工作负载配置 Amazon CloudFront 分配，或者使用第三方 CDN。
            - 您可以通过在端点安全组或访问策略中使用 CloudFront 的 IP 范围，将对工作负载的访问限制为只能从 CloudFront 访问。

资源

相关文档：

- APN 合作伙伴：可以帮您制定自动计算解决方案的合作伙伴
- AWS Auto Scaling：扩展计划的工作原理
- AWS Marketplace：可以与 Auto Scaling 一起使用的产品
- 使用 DynamoDB Auto Scaling 自动管理吞吐能力
- 将负载均衡器与自动扩缩组配合使用
- 什么是 AWS Global Accelerator？
- 什么是 Amazon EC2 Auto Scaling？
- 什么是 AWS Auto Scaling？
- 什么是 Amazon CloudFront？
- 什么是 Amazon Route 53？
- 什么是 Elastic Load Balancing？
- 什么是网络负载均衡器？
- 什么是 Application Load Balancer？

- 使用记录

REL07-BP02 在检测到对工作负载的破坏时获取资源

如果可用性受到影响，在必要时被动扩展资源，从而还原工作负载的可用性。

首先，您必须配置运行状况检查和关于此类检查的标准，表示在什么时候可用性会因缺少资源而受到影响。然后，通知适当的人员手动扩展资源，或触发自动化以对其进行自动扩展。

可以为您的工作负载手动调整扩展，例如，通过 AWS Management Console 或 AWS CLI 更改自动扩缩组中 EC2 实例的数量，或者修改 DynamoDB 表的吞吐量来实现。不过，应在可能的情况下尽量使用自动化（请参阅 在获取或扩展资源时利用自动化）。

未建立这种最佳实践的情况下暴露的风险等级： 中

实施指导

- 在检测到对工作负载的破坏时获取资源。如果可用性受到影响，在必要时被动扩展资源，从而还原工作负载的可用性。
  - 使用扩展计划来配置指令集以用于扩展您的资源，扩展计划是 AWS Auto Scaling 的核心组件。如果使用 AWS CloudFormation 或为 AWS 资源添加标签，您可以根据应用程序为不同的资源集设置扩展计划。AWS Auto Scaling 提供了针对每个资源自定义的扩展策略建议。创建扩展计划后，AWS Auto Scaling 结合了动态扩展和预测式扩缩方法来支持扩展策略。
    - AWS Auto Scaling：扩展计划的工作原理
  - Amazon EC2 Auto Scaling 有助于确保您拥有适量的 Amazon EC2 实例，可处理您的应用程序负载。您可创建 EC2 实例集合，称为 Auto Scaling 组。您可以指定每个自动扩缩组中的最小实例数量，Amazon EC2 Auto Scaling 会确保您组中的实例绝不会低于该数量。您可以指定每个自动扩缩组中的最大实例数量，Amazon EC2 Auto Scaling 会确保您组中的实例绝不会高于该数量。
    - 什么是 Amazon EC2 Auto Scaling？
  - Amazon DynamoDB Auto Scaling 使用 AWS Application Auto Scaling 服务，代表您动态调整预置的吞吐能力，以响应实际的流量模式。这将使表或全局二级索引提高预置读取和写入容量，从而不受限制地应对流量突增。
    - 使用 DynamoDB Auto Scaling 自动管理吞吐能力

资源

相关文档：

- APN 合作伙伴：可以帮您制定自动计算解决方案的合作伙伴
- AWS Auto Scaling：扩展计划的工作原理
- AWS Marketplace：可以与 Auto Scaling 一起使用的产品
- 使用 DynamoDB Auto Scaling 自动管理吞吐能力
- 什么是 Amazon EC2 Auto Scaling？

REL07-BP03 当检测到某个工作负载需要更多资源时，就会获取资源

主动扩展资源以满足需求并避免影响可用性。

很多 AWS 服务会自动扩展以满足需求。如果使用 Amazon EC2 实例或 Amazon ECS 集群，您可以根据与您的工作负载的需求对应的使用指标，配置它们会在何时自动扩展。针对 Amazon EC2，平均 CPU 利用率、负载均衡器请求数量，或网络带宽可被用于扩展（或缩减）EC2 实例。而对于 Amazon ECS，可使用平均 CPU 利用率、负载均衡器请求数量和内存利用率横向扩展（或横向缩减）ECS 任务。在 AWS 上使用 Target Auto Scaling，Autoscaler 将扮演“家用恒温器”的角色，增加或减少资源以保持您所指定的目标值（例如，70% CPU 利用率）。

AWS Auto Scaling 还可以执行 Predictive Auto Scaling，该操作利用机器学习来分析每个资源的历史工作负载，并且定期预测未来两天的负载。

利特尔法则可帮助计算您需要多少计算实例（EC2 实例、并发 Lambda 函数，等等）。

L = λW

L = 实例数量（或系统中的平均并发值）

λ = 收到请求的平均速率（请求数量/秒）

W = 每个请求在系统中所花的平均时间（秒）

例如，假设每秒请求数为 100，若每个请求所需的处理时间为 0.5 秒，您将需要 50 个实例才能满足需求。

未建立此最佳实践暴露的风险等级： 中

实施指导

- 当检测到某个工作负载需要更多资源时，就会获取资源。主动扩展资源以满足需求并避免影响可用性。

- 计算处理给定请求速率需要多少计算资源（计算并发）。
  - 讲述与利特尔法则有关的故事
- 当您具有历史使用模式时，请为 Amazon EC2 Auto Scaling 设置计划扩展。
  - Amazon EC2 Auto Scaling 的计划扩缩
- 使用 AWS 预测式扩缩。
  - 由机器学习提供支持的 EC2 预测式扩缩

资源

相关文档：

- AWS Auto Scaling：扩展计划的工作原理
- AWS Marketplace：可以与 Auto Scaling 一起使用的产品
- 使用 DynamoDB Auto Scaling 自动管理吞吐能力
- 由机器学习提供支持的 EC2 预测式扩缩
- Amazon EC2 Auto Scaling 的计划扩缩
- 讲述与利特尔法则有关的故事
- 什么是 Amazon EC2 Auto Scaling？

REL07-BP04 对工作负载进行负载测试

采用负载测试方法来衡量扩展活动能否满足工作负载要求。

持续开展负载测试，这一点很重要。负载测试用于发现工作负载的断点并测试工作负载的性能。利用 AWS，您可以轻松设置能够模拟生产工作负载规模的临时测试环境。在云中，您可以根据需要创建一套生产规模等级的测试环境，完成测试，然后停用资源。由于测试环境只需在运行时付费，您模拟真实环境的成本仅为本地测试成本的一小部分。

生产中的负载测试还应该被视为实际试用活动的一部分，因为在客户使用量降低的那几个小时内，在场的所有员工都忙于解读结果与处理任何出现的问题，生产系统承受着很大的压力。

常见反模式：

- 对与您的生产采用不同配置的部署执行负载测试。
- 仅对单个工作负载分段（而非整个工作负载）执行负载测试。
- 使用请求子集，而不是具有代表性的实际请求集执行负载测试。

- 对超出预期负载的较小安全系数执行负载测试。

建立此最佳实践的好处： 您知道架构中哪些组件会在负载下失败，而且能够确定要监控哪些可指示您即将达到该负载的指标，从而及时解决问题，防止故障影响。

未建立这种最佳实践的情况下暴露的风险等级： 中

实施指导

- 执行负载测试，确定工作负载的哪些方面表明您必须添加或移除容量。负载测试应具有您在生产中接收的流量类似的代表性流量。增加负载，同时监视所有已检测指标，以便确定哪种指标指示何时必须添加或移除资源。
  - AWS 上的分布式负载测试：模拟数千个连接的用户
    - 确定请求组合。您可能拥有不同的请求组合，因此应当在确定流量组合时查看不同的时间范围。
    - 实施负载驱动程序。您可以使用自定义代码、开源或商用软件来实施负载驱动程序。
    - 最初使用小容量进行负载测试。通过将负载降低到较小容量（可能小到一个实例或容器），可能会有立竿见影的效果。
    - 针对更大的容量进行负载测试。分布式负载的效果会有所不同，因此您必须对尽量接近生产环境的目标进行测试。

资源

相关文档：

- AWS 上的分布式负载测试：模拟数千个连接的用户

REL 8 如何实施更改？

要部署新功能，必须对更改加以控制，以确保工作负载和操作环境正在运行已知的软件，并以可预测的方式进行修补和替换。如果此类更改不受控制，您将难以预测这些更改的影响，或难以处理由它们引发的问题。

最佳实践

- REL08-BP01 对部署等标准活动使用运行手册
- REL08-BP02 将功能测试作为部署的一部分进行集成
- REL08-BP03 将弹性测试作为部署的一部分进行集成

- REL08-BP04 使用不可变基础设施进行部署
- REL08-BP05 使用自动化功能部署更改

REL08-BP01 对部署等标准活动使用运行手册

运行手册是用来实现特定结果的预定义程序。使用运行手册执行标准活动，无论这些活动是手动还是自动执行。其中的示例包括部署工作负载，修补工作负载，或修改 DNS。

例如，实施流程以 确保部署期间安全回滚.确保您可以为客户进行部署回滚而不会出现中断，这是保证服务可靠的关键。

针对运行手册程序，从一个有效的手动流程开始，用代码进行实施，并在适当的情况下触发其自动运行。

即使是高度自动化的复杂工作负载，运行手册同样适用于 运行实际试用 或用于满足严格的报告和审计要求。

请注意，行动手册可用于对特定事件做出响应，运行手册则用来达成特定的结果。通常，运行手册适用于例行活动，而行动手册则被用于对非例行事件做出响应。

常见反模式：

- 对生产中的配置执行计划外更改。
- 跳过计划中的步骤以加快部署速度，导致部署失败。
- 在未测试反向更改的情况下做出更改。

建立此最佳实践的好处： 有效更改计划有助于您成功执行更改，因为您知道所有受影响的系统。在测试环境中验证更改能够增强您的信心。

未建立此最佳实践暴露的风险等级： 高

实施指导

- 通过在运行手册中记录程序，实现对为人熟知的事件的一致且及时的响应。
  - AWS Well-Architected Framework：概念：运行手册
- 使用基础设施即代码的原则定义您的基础设施。通过使用 AWS CloudFormation（或受信任的第三方）来定义您的基础设施，您可以使用版本控制软件对更改实施版本控制并进行跟踪。
  - 使用 AWS CloudFormation（或受信任的第三方提供商）定义您的基础设施。

    - 什么是 AWS CloudFormation？
- 使用良好的软件设计原则创建单个解耦模板。
    - 确定实施的权限、模板和责任方。
        - 使用 AWS Identity and Access Management 控制访问权限
    - 使用源代码控制（例如 AWS CodeCommit 或受信任的第三方工具）进行版本控制。
        - 什么是 AWS CodeCommit？

资源

相关文档：

- AWS 合作伙伴：可以帮助您创建自动化部署解决方案的合作伙伴
- AWS Marketplace：可用于自动实施部署的产品
- AWS Well-Architected Framework：概念：运行手册
- 什么是 AWS CloudFormation？
- 什么是 AWS CodeCommit？

相关示例：

- 使用行动手册和运行手册自动完成操作

REL08-BP02 将功能测试作为部署的一部分进行集成

功能测试作为自动化部署的一部分运行。若未满足成功条件，则相关管道会中止或回滚。

这些测试在预生产环境中运行，该环境会在管道中的生产开始前被暂存。在理想情况下，此操作是部署管道的一部分。

未建立这种最佳实践的情况下暴露的风险等级： 高

实施指导

- 将功能测试作为部署的一部分进行集成。功能测试作为自动化部署的一部分运行。若未满足成功条件，则相关管道会中止或回滚。
    - 当在 AWS CodePipeline 中建模的软件发布管道执行“测试操作”时，调用 AWS CodeBuild。此功能使您能够对代码轻松运行各种测试，例如单元测试、静态代码分析和集成测试。
        - AWS CodePipeline 增加了对通过 AWS CodeBuild 进行单位和自定义集成测试的支持

- 使用 AWS Marketplace 解决方案，将自动化测试作为软件交付管道的一部分执行。
  - 软件测试自动化

资源

相关文档：

- AWS CodePipeline 增加了对通过 AWS CodeBuild 进行单位和自定义集成测试的支持
- 软件测试自动化
- 什么是 AWS CodePipeline？

REL08-BP03 将弹性测试作为部署的一部分进行集成

将弹性测试（使用 混沌工程的原则）作为预生产环境中自动化部署管道的一部分执行。

这些测试会在预生产环境的管道中暂存并运行。它们应在生产中运行，作为 实际试用的一部分。

未建立这种最佳实践的情况下暴露的风险等级： 中

实施指导

- 将弹性测试作为部署的一部分进行集成。混沌工程是对工作负载进行试验的规范，用于建立人们对工作负载能够在生产中经受住混乱情形的信心。
  - 弹性测试会注入故障或资源降级，以此评估您的工作负载能否以预期弹性做出响应。
    - Well-Architected 实验室：第 300 级：测试 EC2 RDS 和 S3 的弹性
  - 这些测试可以在自动部署管道的预生产环境中定期执行。
  - 它们还应作为计划实际演练的一部分在生产环境中运行。
  - 使用混沌工程原则，提出有关工作负载在各种破坏情况下如何表现的假设，然后使用弹性测试验证您的假设。
    - 混沌工程的原则

资源

相关文档：

- 混沌工程的原则
- 什么是 AWS Fault Injection Simulator?

相关示例：

- Well-Architected 实验室：第 300 级：测试 EC2 RDS 和 S3 的弹性

REL08-BP04 使用不可变基础设施进行部署

不可变基础设施模式要求在生产系统上不会出现就地更新、安全补丁或配置更改。需要更改时，会在新的基础设施上构建架构，并将其部署到生产环境中。

最常被实施的不可变基础设施范式为 不可变服务器.这意味着，若服务器需要更新或修复，将部署新的服务器，而不是对使用中的服务器进行更新。因此，相对于通过 SSH 登录到服务器并更新软件版本，应用程序的每次更改都会在开始时将软件推送到代码库，如 git 推送。由于在不可变基础设施中不允许更改，您可以确定已部署系统的状态。不可变基础设施在本质上具有更稳定、可靠和可预测的特性，它们对软件开发和运行的多个方面进行了简化。

当您在不可变基础设施中部署应用程序时，使用 Canary 或蓝绿部署。

金丝雀部署 是将您的少量客户引导到新版本的做法，它通常在单个服务实例 (Canary) 上运行。然后，您可以深入检查生成的任何行为更改或错误。如果遇到了严重问题，您可以将 Canary 中的流量删除，并将用户发回到以前的版本。如果部署成功，您可以继续以期望的速度进行部署，同时监控更改以便发现错误，直到所有部署完成。AWS CodeDeploy 的部署配置可以配置为启用金丝雀部署。

蓝绿部署 与金丝雀部署类似，只是会并行部署一整套应用程序。您可以在两个堆栈（蓝和绿）之间轮流部署。同样，您可以将流量发送到新版本中，如果发现部署中存在问题，可以对其进行故障恢复，然后送回旧版本中。通常来说，所有流量会被一次性切换，但您也可以通过 Amazon Route 53 的加权 DNS 路由功能向每个版本发送部分流量，以加快采用新版本的速度。AWS CodeDeploy 和 AWS Elastic Beanstalk 的部署配置可以配置为启用蓝绿部署。

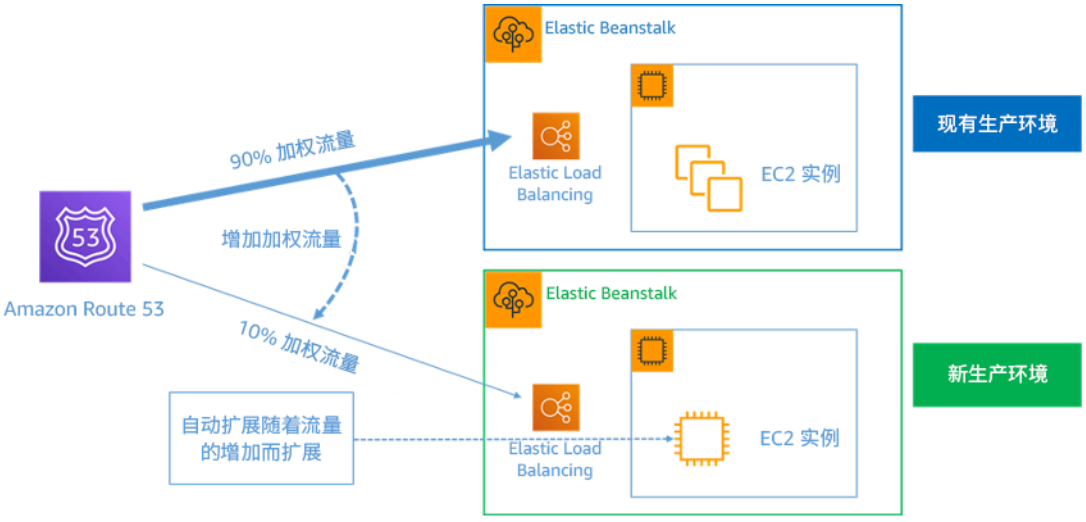


图 8：使用 AWS Elastic Beanstalk 和 Amazon Route 53 进行蓝绿部署

不可变基础设施的优点：

- 减小配置偏差： 通过从基本、已知，而且版本受控的配置频繁替换服务器，基础设施会被 重置 为已知状态，以避免配置偏差。
- 简化部署：由于无需支持升级，部署得到简化。升级即意味着新的部署。
- 可靠的原子部署： 成功完成部署，或没有任何更改。它让您更信任部署流程。
- 采用快速回滚和恢复流程的更安全部署： 由于之前运行的版本未发生更改，因此部署变得更安全。您可以在检测到错误时进行回滚。
- 一致的测试和调试环境： 由于所有服务器都使用相同的映像，因此环境之间没有任何差异。同一个版本被部署到多个环境。它还防止出现不一致的环境，并且简化测试与调试。
- 增强可扩展性： 服务器都使用一个基础映像，它们是一致、可重复的，自动扩展并不重要。
- 简化工具链：您无需采用配置管理工具对生产软件升级进行管理，因此工具链也得到简化。也不需要在服务器上安装其他工具或代理。对基础映像进行更改，然后在经过测试后实施。
- 提高安全性： 通过拒绝对服务器的所有更改，您可以在实例上禁用 SSH 并移除密钥。这样做可以减少攻击载体，改善您的组织的安全状况。

未建立此最佳实践暴露的风险等级： 中

实施指导

- 使用不可变基础设施进行部署。不可变基础设施是一个不会在生产系统上 就地 发生更新、安全修补或配置更改的模型。如果需要任何更改，则会构建架构的新版本，并将其部署到生产环境中。
  - 蓝绿部署概览
  - 逐步部署无服务器应用程序
  - 不可改变基础设施：通过不可改变特性带来的可靠性、一致性和信心
  - CanaryRelease

资源

相关文档：

- CanaryRelease
- 逐步部署无服务器应用程序
- 不可改变基础设施：通过不可改变特性带来的可靠性、一致性和信心

- 蓝绿部署概览
- Amazon Builders' Library：确保部署期间安全回滚

REL08-BP05 使用自动化功能部署更改

自动部署与修补以消除负面影响。

对许多组织来说，对生产系统进行变更是风险最大的工作之一。除了软件解决的业务问题外，我们认为部署也是亟待解决的首要问题。如今，这意味着根据实际情况在操作中使用自动化，包括测试和部署更改、添加或删除容量以及迁移数据。AWS CodePipeline 让您可以管理释放您的工作负载所需的步骤。其中包括，采用 AWS CodeDeploy 将应用程序代码自动部署到 Amazon EC2 实例、本地实例、无服务器 Lambda 函数或 Amazon ECS 服务的部署状态。

> **推荐**
>
> 虽然传统智慧告诉我们，循环中最困难的操作程序应该由人来负责，但出于相同的原因，我们建议您将最困难的程序自动化。

常见反模式：

- 手动执行更改。
- 通过紧急工作流程跳过自动化中的步骤。
- 未遵守您的计划。

建立此最佳实践的好处： 通过自动化功能部署所有更改，可消除引入人为错误的可能性，还能在更改生产之前进行测试，从而确保计划完成。

未建立这种最佳实践的情况下暴露的风险等级： 中

实施指导

- 实现部署管道的自动化。借助部署管道，您可以调用自动化测试和异常检测，并且能够在生产部署前的某个步骤停止管道，或自动回滚更改。
  - Amazon Builders' Library：确保部署期间安全回滚
  - Amazon Builders' Library：采用持续交付，加速交付进度
    - 使用 AWS CodePipeline（或受信任的第三方产品）定义和运行您的管道。
      - 将管道配置为在将更改实施到代码存储库后开始。

    - 什么是 AWS CodePipeline？
  - 使用 Amazon Simple Notification Service（Amazon SNS）和 Amazon Simple Email Service（Amazon SES）在管道中发送有关问题的通知，或者与 Amazon Chime 等团队聊天工具集成。
    - 什么是 Amazon Simple Notification Service？
    - 什么是 Amazon SES？
    - 什么是 Amazon Chime？
    - 使用 Webhook 自动发送聊天消息。

资源

相关文档：

- AWS 合作伙伴：可以帮助您创建自动化部署解决方案的合作伙伴
- AWS Marketplace：可用于自动实施部署的产品
- 使用 Webhook 自动发送聊天消息。
- Amazon Builders' Library：确保部署期间安全回滚
- Amazon Builders' Library：采用持续交付，加速交付进度
- 什么是 AWS CodePipeline？
- 什么是 CodeDeploy？
- AWS Systems Manager 补丁管理器
- 什么是 Amazon SES？
- 什么是 Amazon Simple Notification Service？

相关视频：

- 2019 年 AWS 峰会：AWS 上的 CI/CD

故障管理

问题

- REL 9 如何备份数据？
- REL 10 如何使用故障隔离来保护您的工作负载？

- REL 11 如何将您的工作负载设计为可承受组件故障的影响？
- REL 12 如何测试可靠性？
- REL 13 如何规划灾难恢复 (DR)？

REL 9 如何备份数据？

备份数据、应用程序和配置，以满足恢复时间目标（RTO）和恢复点目标（RPO）的要求。

最佳实践

- REL09-BP01 识别和备份需要备份的所有数据，或从源复制数据
- REL09-BP02 保护并加密备份
- REL09-BP03 自动执行数据备份
- REL09-BP04 定期执行数据恢复以验证备份完整性和流程

REL09-BP01 识别和备份需要备份的所有数据，或从源复制数据

所有 AWS 数据存储均提供备份功能。Amazon RDS 和 Amazon DynamoDB 等服务还额外地支持可实现时间点故障恢复（PITR，Point-In-Time Recovery）的自动备份，这使您可以将备份恢复到距当前时间不超过五分钟的任意时间点。许多 AWS 服务提供了将备份复制到其他 AWS 区域 的功能。AWS Backup 工具向您提供了跨 AWS 服务来集中实现自动化数据保护的功能。

Amazon S3 可用作自行管理数据来源和 AWS 托管数据来源的备份目标。Amazon EBS、Amazon RDS 和 Amazon DynamoDB 等 AWS 服务具有可用于创建备份的内置功能。此外，也可使用第三方备份软件。

本地数据可以备份到 AWS Cloud 中（通过使用 AWS Storage Gateway 或者 AWS DataSync）。Amazon S3 存储桶可用于在 AWS 上存储此数据。Amazon S3 提供了多种存储层，例如 Amazon S3 Glacier 或 S3 Glacier Deep Archive ，可用于降低数据存储的成本。

您可以从其他来源复制数据，以此来满足数据恢复需求。例如， Amazon Elasticache 复制节点 或者 RDS 只读副本 可用于在主来源丢失时复制数据。如果此类来源可用于满足您的 恢复点目标（RPO，Recovery Point Objective）和恢复时间目标（RTO，Recovery Time Objective），则您可能无需备份。在另一个例子中，如果使用 Amazon EMR，只要可以将数据从 S3 复制到 EMR 中， 就无需备份您的 HDFS 数据存储。

在选择备份策略时，请考虑恢复数据所用的时间。恢复数据所需的时间取决于备份的类型（在采用备份策略时）或数据复制机制的复杂性。此时间应该符合工作负载的 RTO。

期望结果：

数据来源已确定，并根据重要性进行了分类。然后，根据 RPO 为数据恢复建立了策略。此策略涉及到备份这些数据来源，或者能够从其他来源复制数据。在出现数据丢失的情况下，所实施的策略可以在定义的 RPO 和 RTO 内实现数据的恢复或再现。

云成熟度阶段： 基础

常见反模式：

- 并不了解工作负载的所有数据来源及其重要性。
- 没有对关键数据来源进行备份。
- 仅对部分数据来源进行备份，但没有考虑重要性标准。
- 没有定义 RPO，或者备份频率无法满足 RPO。
- 没有评估备份是否必需或者是否可以从其他来源复制数据。

建立此最佳实践的好处： 确定需要备份的位置并实施某种机制来创建备份，或者具备从外部来源复制数据的能力，这样可以提高在停机期间还原和恢复数据的能力。

未建立这种最佳实践的情况下暴露的风险等级： 高

实施指导

了解并使用工作负载所用的 AWS 服务和资源的备份功能。大部分 AWS 服务提供了备份工作负载数据的功能。

实施步骤：

1. 确定工作负载的所有数据来源。数据可以存储在多种资源中，例如 数据库， 卷， 文件系统， 日志记录系统和 对象存储。请参阅 资源 部分以查找 相关文档， 了解存储数据的不同 AWS 服务，以及这些服务提供的备份功能。
2. 根据重要性对数据来源进行分类。对于工作负载，不同数据集具有不同的重要程度，因此对弹性具有不同的要求。例如，一些数据可能会非常重要，要求接近于零的 RPO，而另一些数据则不那么重要，可以承受较高的 RPO 和某种程度的数据丢失。与此类似，不同数据集也可能会有不同的 RTO 要求。
3. 使用 AWS 或第三方服务来创建数据的备份。 AWS Backup 是一种托管服务，可用于为 AWS 上的各种数据来源创建备份。大部分这些服务还具有原生的创建备份功能。AWS Marketplace 有许多解决方案同样提供了这些功能。请参阅以下列出的 资源 ，了解如何从不同 AWS 服务创建数据备份的信息。

4. 对于没有备份的数据，请建立数据复制机制。您可能会出于各种原因，不对可从其他来源复制的数据进行备份。您可能会遇到一种情况，在需要时从来源复制数据的成本相比创建备份更低，因为可能会有与存储备份相关的成本。另一个例子是从备份进行还原的时间比从来源复制数据用时更长，使得备份不符合 RTO 要求。在此类情况下请做出权衡，并建立明确定义的流程，确定在需要进行恢复时如何从这些来源复制数据。例如，如果您从 Amazon S3 将数据加载到数据仓库（如 Amazon Redshift）或 MapReduce 集群（如 Amazon EMR），以便对此类数据进行分析，这就是可从其他来源复制数据的例子。只要此类分析的结果被存储在某位置，或者可重现，您不会因为数据仓库或 MapReduce 集群故障而遭遇数据丢失的情况。其他可从数据源复制数据的例子包括，缓存（如 Amazon ElastiCache）或 RDS 只读副本。
5. 建立备份数据的频率。创建数据来源的备份是一个定期执行的流程，其频率取决于 RPO。

实施计划的工作量级别： 适中

资源

相关最佳实践：

REL13-BP01 定义停机和数据丢失的恢复目标

REL13-BP02 使用定义的恢复策略来实现恢复目标

相关文档：

- 什么是 AWS Backup？
- 什么是 AWS DataSync？
- 什么是卷网关？
- AWS 合作伙伴：可以帮助进行备份的合作伙伴
- AWS Marketplace：可以用于备份的产品
- Amazon EBS 快照
- 备份 Amazon EFS
- 备份 Amazon FSx for Windows File Server
- ElastiCache for Redis 备份和还原
- 在 Neptune 中创建数据库集群快照
- 创建数据库快照
- 创建按计划触发的 EventBridge 规则

- 跨区域复制（使用 Amazon S3）
- EFS 到 EFS AWS Backup
- 将日志数据导出到 Amazon S3
- 对象生命周期管理
- DynamoDB 的按需备份和还原
- DynamoDB 的时间点恢复
- 使用 Amazon OpenSearch Service 索引快照

相关视频：

- AWS re:Invent 2021 – 使用 AWS 进行备份、灾难恢复和勒索软件防护
- AWS Backup 演示：跨账户和跨区域备份
- AWS Backup re:Invent 2019：深入了解 AWS，主讲：Rackspace（STG341）

相关示例：

- Well-Architected 实验室：为 Amazon S3 实施双向跨区域复制（CRR，Cross-Region Replication）
- Well-Architected 实验室：测试数据的备份与还原
- Well-Architected 实验室：面向分析工作负载的备份和还原（具备故障恢复功能）
- Well-Architected 实验室：灾难恢复 – 备份与还原

REL09-BP02 保护并加密备份

使用 AWS IAM 等身份验证和授权服务，控制并检测对备份的访问。使用加密功能，防止并检测备份的数据完整性是否受到损坏。

Amazon S3 支持多种对您的静态数据进行加密的方式。借助服务器端加密功能，Amazon S3 以未加密数据的形式接受您的对象，然后在存储此类数据时进行加密。若采用客户端加密，您的工作负载应用程序需要负责在将其发送到 Amazon S3 之前加密数据。这两种方式都让您可以使用 AWS Key Management Service（AWS KMS）创建并存储数据密钥，或者您也可以提供自己的密钥并自行对其负责。使用 AWS KMS，您可以通过 IAM 设置策略，决定谁可以以及谁不可以访问您的数据密钥与解密数据。

针对 Amazon RDS，如果您已选择对数据库进行加密，那么您的备份也会被加密。DynamoDB 备份则始终被加密。

常见反模式：

- 对备份和还原自动化的访问权限与对数据的访问权限相同。
- 未加密您的备份。

建立此最佳实践的好处： 保护备份安全可防止篡改数据，而加密数据可防止数据意外暴露时对其访问。

未建立此最佳实践暴露的风险等级： 高

实施指导

- 对每个数据存储使用加密功能。如果源数据已加密，则备份也将被加密。
  - 在 RDS 中启用加密功能。当您创建 RDS 实例时，可以使用 AWS Key Management Service 配置静态加密。
    - 加密 Amazon RDS 资源
  - 在 EBS 卷中启用加密。您可以配置默认加密或在创建卷时指定唯一密钥。
    - Amazon EBS 加密
  - 使用所需的 Amazon DynamoDB 加密。DynamoDB 会加密所有静态数据。您可以使用 AWS 拥有的 AWS KMS 密钥或者 AWS 托管 KMS 密钥，指定存储在您账户中的密钥。
    - DynamoDB 静态加密
    - 管理加密的表
  - 加密 Amazon EFS 中存储的数据。在创建文件系统时配置加密。
    - 在 EFS 中加密数据和元数据
  - 在源和目标区域中配置加密功能。您可以使用 KMS 中存储的密钥配置 Amazon S3 中的静态加密，但这些密钥是特定于区域的。您在配置复制时可以指定目标密钥。
    - CRR 附加配置：复制通过存储在 AWS KMS 中的加密密钥、使用服务器端加密（SSE，Server-Side Encryption）创建的对象
- 实施用于访问您的备份的最低权限。请遵循最佳实践，根据安全最佳实践来限制对备份、快照和副本的访问。
  - 安全性支柱：AWS Well-Architected

资源

相关文档：

- AWS Marketplace：可以用于备份的产品
- Amazon EBS 加密
- Amazon S3：利用加密保护数据
- CRR 附加配置：复制通过存储在 AWS KMS 中的加密密钥、使用服务器端加密（SSE，Server-Side Encryption）创建的对象
- DynamoDB 静态加密
- 加密 Amazon RDS 资源
- 在 EFS 中加密数据和元数据
- AWS 中的备份的加密
- 管理加密的表
- 安全性支柱：AWS Well-Architected

相关示例：

- Well-Architected 实验室：为 Amazon S3 实施双向跨区域复制（CRR，Cross-Region Replication）

REL09-BP03 自动执行数据备份

将备份配置为根据遵循恢复点目标（RPO，Recovery Point Objective）的定期计划自动备份，或者在数据集发生更改时自动备份。具有低数据丢失需求的关键数据资产需要频繁地自动备份，而可以接受某些丢失的较不重要数据的备份频率可以更低。

AWS Backup 可用于创建各种 AWS 数据来源的自动数据备份。Amazon RDS 实例可以按照五分钟的频率进行几乎连续的备份，Amazon S3 对象可以按照十五分钟的频率进行几乎连续的备份，提供可恢复到备份历史记录中的特定时间点的时间点故障恢复（PITR，Point-In-Time Recovery）功能。对于其他 AWS 数据来源，例如 Amazon EBS 卷、Amazon DynamoDB 表或 Amazon FSx 文件系统，AWS Backup 最快可以按每小时的频率运行自动备份。这些服务还提供了原生备份功能。以下 AWS 服务提供了具备时间点故障恢复的自动备份功能： Amazon DynamoDB， Amazon RDS和 Amazon Keyspaces（Apache Cassandra 兼容） – 这些备份可以恢复到备份历史记录中的特定时间点。大部分其他 AWS 数据存储服务提供了计划定期备份的功能，频率最快为每小时一次。

Amazon RDS 和 Amazon DynamoDB 提供支持时间点故障恢复的持续备份。一旦启用，Amazon S3 版本控制就会自动执行。Amazon Data Lifecycle Manager 可用于自动创建、复制和删除 Amazon EBS 快照。它还可以自动创建、复制、弃用和取消注册 Amazon EBS 支持的亚马逊云机器镜像（AMI，Amazon Machine Image）及其底层 Amazon EBS 快照。

针对您的备份自动化和历史的集中式视图，AWS Backup 提供完全托管的基于策略的备份解决方案。它会使用 AWS Storage Gateway 将云端和本地的多项 AWS 服务的数据备份集中在一起并自动处理。

除了版本控制，Amazon S3 还具有复制功能。整个 S3 存储桶都可自动复制到相同或不同 AWS 区域中的其他存储桶。

期望结果：

按照确定的节奏创建数据来源备份的自动流程。

常见反模式：

- 手动执行备份。
- 使用具有备份功能的资源，但不包括自动化中的备份。

建立此最佳实践的好处： 自动化备份可以确保按照 RPO 执行备份，并在未备份时发出警报。

未建立这种最佳实践的情况下暴露的风险等级： 中

实施指导

1. 确定当前 在手动备份的数据来源。请参阅 REL09-BP01 识别和备份需要备份的所有数据，或从源复制数据 以获取有关此项的指南。
2. 确定工作负载的 RPO。请参阅 REL13-BP01 定义停机和数据丢失的恢复目标 以获取有关此项的指南。
3. 使用自动化备份解决方案或托管服务。AWS Backup 是一项完全托管的服务，它简化了 跨 AWS 服务、云端以及本地以集中方式自动进行数据保护的过程。备份计划是 AWS Backup 的功能，可用于创建规则来定义要备份的资源，以及创建这些备份的频率。此频率应遵循在第 2 步中确定的 RPO。本 WA 实验室 提供有关如何使用 AWS Backup 创建自动备份的动手实践指南。大多数存储数据的 AWS 服务提供了原生备份功能。例如，可以利用 RDS 来实现支持时间点故障恢复（PITR，Point-In-Time Recovery）的自动备份。
4. 对于 自动备份解决方案或托管服务不支持的数据来源（如本地数据来源或消息队列），请考虑使用受信任的第三方解决方案来创建自动备份。或者，您可以使用 AWS CLI 或开发工具包创建自动化过程来完成此操作。您可以使用 AWS Lambda 函数或 AWS Step Functions 来定义创建数据备份中涉及的逻辑，并使用 Amazon EventBridge 按照基于 RPO 确定的频率来执行它（如第 2 步中所述）。

实施计划的工作量级别： 低

资源

相关文档：

- AWS 合作伙伴：可以帮助进行备份的合作伙伴
- AWS Marketplace：可以用于备份的产品
- 创建按计划触发的 EventBridge 规则
- 什么是 AWS Backup？
- 什么是 AWS Step Functions？

相关视频：

- AWS Backup re:Invent 2019：深入了解 AWS，主讲：Rackspace（STG341）

相关示例：

- Well-Architected 实验室：测试数据的备份与还原

REL09-BP04 定期执行数据恢复以验证备份完整性和流程

通过执行恢复测试，验证您的备份流程实施是否满足恢复时间目标（RTO）和恢复点目标（RPO）要求。

使用 AWS，您可以构建一个测试环境，还原您的备份以评估 RTO 和 RPO 功能，并且对数据的内容和完整性执行测试。

此外，Amazon RDS 和 Amazon DynamoDB 还允许时间点恢复 (PITR)。您可以使用持续备份将您的数据集还原到其在指定日期与时间所处的状态。

期望结果： 使用明确定义的机制定期从备份恢复数据，确保可以按照为工作负载确定的恢复时间目标（RTO，Recovery Time Objective）来恢复数据。验证从备份进行还原可以得到包含原始数据的资源，而不会造成数据损坏或无法访问数据，并且数据丢失在恢复点目标（RPO，Recovery Point Objective）之内。

常见反模式：

- 还原备份，但未查询或检索任何数据以确保还原操作可用。
- 假定采取了备份。

- 假定系统的备份完全正常运行，并且可从中恢复数据。
- 假定从备份还原或恢复数据的时间满足工作负载的 RTO。
- 假定备份中包含的数据满足工作负载的 RPO
- 临时进行还原，没有使用运行手册或者没有按照确定的自动程序执行。

建立此最佳实践的好处： 测试备份的恢复过程可以确保在需要时能够将数据还原，不必担心数据可能丢失或损坏，可以按照工作负载要求的 RTO 还原和恢复，并且任何数据丢失都在工作负载的 RPO 以内。

未建立此最佳实践暴露的风险等级： 中

实施指导

测试备份和还原功能可树立信心，确信能够在出现中断时执行这些操作。定期将备份还原到新的位置，并运行测试以验证数据的完整性。需要执行一些常见的测试，以检查

数据是否可用、没有损坏、是否可以访问并且任意数据丢失都符合工作负载的 RPO。此类测试还可以帮助确定恢复机制是否足够快以满足工作负载的 RTO 要求。

1. 确定当前 所备份的数据来源以及存储这些备份的位置。请参阅 REL09-BP01 识别和备份需要备份的所有数据，或从源复制数据 以获取有关如何实施此项的指导。
2. 为每个 数据来源建立数据验证标准。不同类型的数据具有不同的属性，这可能需要不同的验证机制。在将此数据用于生产之前，请考虑可以如何验证此数据。一些验证数据的常见方法包括使用数据和备份属性，例如数据类型、格式、校验和、大小，或者将这些属性与自定义的验证逻辑结合使用。例如，可以将所恢复资源的校验和值，与创建备份时数据来源的校验和值进行比较。
3. 建立 RTO 和 RPO 以根据数据重要性来还原数据。请参阅 REL13-BP01 定义停机和数据丢失的恢复目标 以获取有关如何实施此项的指导。
4. 评估数据恢复能力。检查备份和还原策略，了解它是否可以满足您的 RTO 和 RPO，并根据需要调整策略。使用 AWS Resilience Hub，您可以对工作负载运行评估。该评估根据弹性策略评估您的应用程序配置，并报告是否能够满足 RTO 和 RPO 目标。
5. 使用当前 为生产环境中数据还原所确立的流程执行测试还原。这些流程依赖于对原始数据来源进行备份的方法，备份本身的格式和存储位置，或者数据是否从其他来源复制。例如，如果您使用 AWS Backup 等托管服务，则此过程可能就是简单地将备份恢复到新的资源。如果您使用 AWS 弹性灾难恢复，则可以 启动恢复演习。。
6. 根据您之前在 第 2 步中为数据验证确立的标准，从还原后的资源（来自上一步）验证数据恢复。还原和恢复的数据是否包含备份时的最新记录/项目？ 此数据是否在工作负载的 RPO 之内？

7. 测量 还原和恢复所需的时间，并与在之前在第 3 步中确立的 RTO 进行比较。此流程是否符合工作负载的 RTO？ 例如，比较还原进程开始时的时间戳以及恢复验证完成时的时间戳，以计算此进程的用时。所有 AWS API 调用均有时间戳，此信息在 AWS CloudTrail中提供。虽然此信息可以提供还原进程何时开始的详细信息，但验证完成时的结束时间戳应该由您的验证逻辑来记录。如果使用自动化进程，则 Amazon DynamoDB 等服务可用于存储此信息。此外，许多 AWS 服务提供了事件历史记录，其中可提供发生特定操作时的时间戳信息。在 AWS Backup 中，备份和还原操作称为 作业，这些作业在其元数据中包含时间戳信息，可用于测量还原和恢复所需的时间。
8. 如果 数据验证失败，或者如果还原和恢复所需的时间超过了为工作负载设定的 RTO，则通知利益相关方。在实施自动化以完成此操作时（例如在本实验中），可以使用 Amazon Simple Notification Service（Amazon SNS）等服务将推送通知（例如电子邮件或 SMS）发送给利益相关方。 这些消息还可以发布到消息传递应用程序，例如 Amazon Chime、Slack 或 Microsoft Teams， 或者 通过使用 AWS Systems Manager OpsCenter 来创建 OpsItems 等任务。
9. 自动执行此流程以便定期运行。例如，AWS Lambda 等服务或 AWS Step Functions 中的状态机可用于自动完成还原和恢复流程，Amazon EventBridge 可用于定期触发此自动工作流，如以下架构图中所示。了解如何 使用 AWS Backup 自动完成数据恢复验证。此外， 本 Well-Architected 实验室 提供动手实践体验，可用于练习针对此处的多个步骤实现自动化的方法。

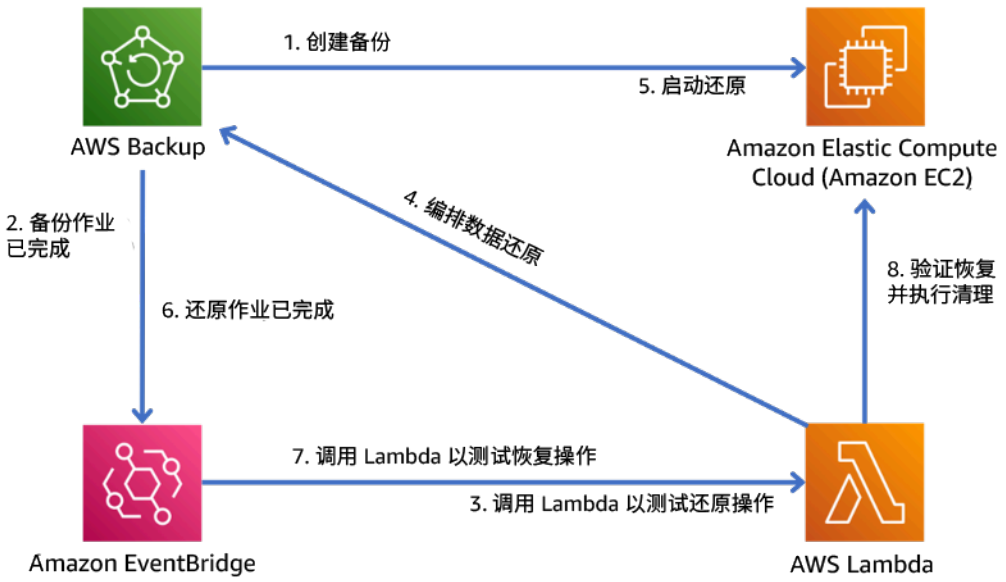


图 9.自动化的备份和还原流程

实施计划的工作量级别： 中到高，具体取决于验证条件的复杂性。

资源

相关文档：

- 使用 AWS Backup 自动完成数据恢复验证
- AWS 合作伙伴：可以帮助进行备份的合作伙伴
- AWS Marketplace：可以用于备份的产品
- 创建按计划触发的 EventBridge 规则
- DynamoDB 的按需备份和还原
- 什么是 AWS Backup？
- 什么是 AWS Step Functions？
- 什么是 AWS 弹性灾难恢复
- AWS 弹性灾难恢复

相关示例：

- Well-Architected 实验室：测试数据的备份与还原

REL 10 如何使用故障隔离来保护您的工作负载？

故障隔离边界可将一个工作负载内的故障影响限制于有限数量的组件。边界以外的组件不会受到故障的影响。使用多个故障隔离边界，您可以限制作用于您的工作负载的影响。

最佳实践

- REL10-BP01 将工作负载部署到多个位置
- REL10-BP02 为您的多位置部署选择合适的位置
- REL10-BP03 组件的自动恢复受限于单个位置
- REL10-BP04 采用隔板架构来限制影响范围

REL10-BP01 将工作负载部署到多个位置

将工作负载数据和资源分布到多个可用区，或在必要时分布到多个 AWS 区域。可通过选择不同位置满足各种需求。

在 AWS，服务设计的其中一个基本原则是避免底层物理基础设施中存在单点故障。这促使我们构建使用多个可用区并能灵活应对单个可用区故障的软件和系统。同样，系统也被构建可灵活应对单个计算节点、单个存储卷或单个数据库实例故障。构建依赖冗余组件的系统时，务必要确保组件独立运行；如果是 AWS 区域，组件应自主运行。只有实现了这一点，采用冗余组件的理论可用性计算的优点才能发挥作用。

可用区（AZ，Availability Zone）

AWS 区域由在设计上彼此相互独立的多个可用区组成。每个可用区之间都间隔相当的物理距离，以避免因环境公害（如火灾、洪水和龙卷风）导致的相互关联的故障情况。每个可用区还拥有独立的物理基础设施：专用的公用电源连接、独立备份电源、独立机械服务以及可用区内外的独立网络连接。此设计将任意这些系统的故障限制在受影响的那一个 AZ 中。尽管可用区在地理位置上相互分离，但它们位于相同的区域中，从而实现高吞吐量、低延迟的联网。整个 AWS 区域（跨多个可用区，由多个物理上独立的设计中心组成）可以视为工作负载的单个逻辑部署目标，包括同步复制数据（例如，在两个数据库之间）的能力。这样一来，您便能在主动/主动或主动/备用配置中使用可用区。

可用区是独立的，因此当工作负载采用了使用多个可用区的架构时，可以提高工作负载的可用性。一些 AWS 服务（包括 Amazon EC2 实例数据面板）作为严格的区级别服务部署，与其所在的可用区共存亡。其他 AZ 中的 Amazon EC2 实例不受影响，可以继续正常工作。与此类似，如果某个可用区中的故障导致 Amazon Aurora 数据库失败，则不受影响的 AZ 中的只读副本 Aurora 实例可以自动提升为主实例。另一方面，区域性 AWS 服务（例如 Amazon DynamoDB）在内部以主动/主动配置的形式使用多个可用区，以实现为该服务设定的可用性设计目标，而且无需您配置 AZ 置放。

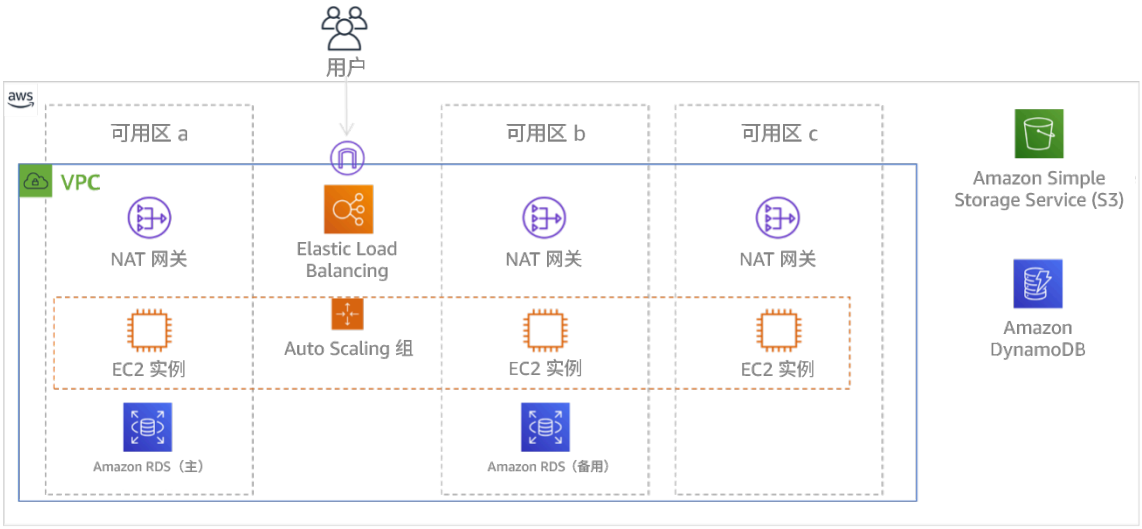


图 9：跨三个可用区部署多层架构。请注意，Amazon S3 和 Amazon DynamoDB 始终会自动部署到多个可用区。而 ELB 也会被部署到所有三个区。

虽然 AWS 控制面板通常提供在整个区域（多个可用区）内管理资源的功能，但某些控制面板（包括 Amazon EC2 和 Amazon EBS）能够将结果筛选到单个可用区。完成筛选后，请求仅在指定可用区中进行处理，从而降低其他可用区的中断风险。此 AWS CLI 示例演示仅从 us-east-2c 可用区中获取 Amazon EC2 实例信息：

```
AWS ec2 describe-instances --filters Name=availability-zone,Values=us-east-2c
```

AWS Local Zones

AWS Local Zones 在其各自的 AWS 区域内的作用与可用区相似，它们可被选择作为区级别 AWS 资源（如子网和 EC2 实例）的置放位置。特别之处在于，它们并不位于相关的 AWS 区域内，而是靠近目前还未设置 AWS 区域的人口密集的工业和 IT 中心。但是，您还是可以享有高带宽，并且能够在本地区域的本地工作负载与在 AWS 区域内运行的工作负载之间进行安全连接。您应该利用 AWS Local Zones 将工作负载尽量部署在接近用户的地方，以满足低延迟的要求。

Amazon 全球边缘网络

Amazon 全球边缘网络由全球各大城市的边缘站点组成。Amazon CloudFront 使用此网络以较低的延迟向最终用户分发内容。您可以通过 AWS Global Accelerator 在这些边缘站点创建您的工作负载端点，以便在靠近您的用户的 AWS 全球网络进行接入。利用 Amazon API Gateway，使用 CloudFront 分配的边缘优化 API 端点可以通过最近的边缘站点方便客户端访问。

AWS 区域

AWS 区域采用自主设计，因此通过多区域方法，您可以将服务的专用副本部署到每个区域。

多区域方法对于 灾难恢复 策略很常见，用于在偶发的大规模事件中满足恢复目标。请参阅 灾难恢复 (DR) 计划 以了解有关这些策略的更多信息。不过，这里我们的重点是 可用性，旨在达成长期使用中的平均正常运行时间目标。对于高可用性目标，通常可以将多区域架构设计为主动/主动模式，各个服务副本（在其各自的区域中）处于活动状态（为请求提供服务）。

> **推荐**
>
> 大多数工作负载的可用性目标都可通过在单个 AWS 区域内采用多 AZ 策略来实现。只有当工作负载具有极高的可用性要求或者其他业务目标时，才考虑多区域架构，在这些情况下需要使用多区域架构。

AWS 提供了跨区域运行服务的功能。例如，AWS 使用 Amazon Simple Storage Service（Amazon S3）复制、Amazon RDS 只读副本（包括 Aurora 只读副本）和 Amazon DynamoDB 全局表，提供了连续异步数据复制功能。通过连续复制，您的数据版本可近乎实时地供各个活动区域使用。

使用 AWS CloudFormation，您可以跨 AWS 账户和 AWS 区域定义基础设施并一致地进行部署。AWS CloudFormation StackSets 扩展了此功能，允许您通过单个操作，跨多个账户和区域创建、更新或删除 AWS CloudFormation 堆栈。对于 Amazon EC2 实例部署，亚马逊云机器镜像（AMI，Amazon Machine Image）可用于提供诸如硬件配置和已安装软件等信息。您可以实施 Amazon EC2 Image

Builder 管道来创建所需的 AMI，并将这些 AMI 复制到您的活动区域。这可以确保这些 Golden AMI 具有您需要部署的所有内容，并可在各个新区域中扩展您的工作负载。

对于路由流量，Amazon Route 53 和 AWS Global Accelerator 均可定义策略来确定哪些用户转向哪个活动的区域端点。使用 Global Accelerator，您可以设置流量转盘，控制导向各个应用程序端点的流量的百分比。Route 53 支持这种百分比方法，还有多个其他策略可用，包括基于地理位置距离和延迟的策略。Global Accelerator 自动利用 AWS 边缘服务器广泛的网络，尽可能快地将流量载入到 AWS 主干网，从而得到较低的请求延迟。

所有这些功能在执行时，都保留了各个区域的自主性。这种方法有极少的例外，包括我们提供全球边缘交付的服务（例如 Amazon CloudFront 和 Amazon Route 53），以及 AWS Identity and Access Management（IAM）服务的控制面板。大多数服务都完全在单个区域中运行。

本地数据中心

对于在本地数据中心运行的工作负载来说，尽可能打造混合体验。AWS Direct Connect 提供从您的本地到 AWS 的专用网络连接，使您可以同时在两者中运行。

另一个选项是，通过 AWS Outposts 在本地运行 AWS 基础设施和服务。AWS Outposts 是一项完全托管式服务，可将 AWS 基础设施、AWS 服务、API 和工具延伸到您的数据中心。在您的设计中心会安装与 AWS Cloud 中使用的相同硬件基础设施。然后，AWS Outposts 会连接到最近的 AWS 区域。您可以使用 AWS Outposts 支持您的低延迟工作负载，或满足本地数据处理要求。

未建立这种最佳实践的情况下暴露的风险等级： 高

实施指导

- 使用多个可用区和 AWS 区域。将工作负载数据和资源分布到多个可用区，或在必要时分布到多个 AWS 区域。可通过选择不同位置满足各种需求。
  - 区域性服务本质上是跨多个可用区部署的。
    - 这包括 Amazon S3、Amazon DynamoDB 和 AWS Lambda（未连接到 VPC 时）
  - 将容器、实例和基于功能的工作负载部署到多个可用区中。使用包括缓存在内的多可用区数据存储。使用 EC2 Auto Scaling、ECS 任务置放、AWS Lambda 函数配置（在 VPC 中运行时）和 ElastiCache 集群的功能。
    - 部署 Auto Scaling 组时，请使用单独可用区中的子网。
      - 示例：跨多个可用区分布实例
      - Amazon ECS 任务置放策略
      - 配置 AWS Lambda 函数以访问 Amazon VPC 中的资源

    - 选择区域和可用区
  - 部署 Auto Scaling 组时，请使用单独可用区中的子网。
    - 示例：跨多个可用区分布实例
  - 使用 ECS 任务置放参数，并指定数据库子网组。
    - Amazon ECS 任务置放策略
  - 配置要在 VPC 中运行的函数时，请使用多个可用区中的子网。
    - 配置 AWS Lambda 函数以访问 Amazon VPC 中的资源
  - 将多个可用区与 ElastiCache 集群配合使用。
    - 选择区域和可用区
- 如果您的工作负载必须部署到多个区域，请选择一个多区域策略。大多数可靠性需求可通过在单个 AWS 区域中使用多可用区策略来满足。可在必要时使用多区域策略来满足您的业务需求。
  - AWS re:Invent 2018：适用于多区域主动-主动应用程序的架构模式（ARC209-R2）
    - 备份到另一个 AWS 区域可以让您更加确信，数据在需要时可用。
    - 有些工作负载具有法规要求，需要使用多区域策略。
- 评估您工作负载的 AWS Outposts。如果您的工作负载需要到本地部署数据中心的较低延迟，或具有本地数据处理要求，然后使用 AWS Outposts 在本地运行 AWS 基础设施和服务
  - 什么是 AWS Outposts？
- 确定 AWS Local Zones 是否可以帮助您为用户提供服务。如果您有低延迟要求，请查看 AWS Local Zones 是否距离您的用户较近。如果是，则使用它将工作负载部署到离这些用户较近的位置。
  - AWS Local Zones 常见问题

资源

相关文档：

- AWS 全球基础设施
- AWS Local Zones 常见问题
- Amazon ECS 任务置放策略
- 选择区域和可用区
- 示例：跨多个可用区分布实例
- 全局表：使用 DynamoDB 的多区域复制
- 使用 Amazon Aurora 全局数据库

- 使用 AWS 服务创建多区域应用程序博客系列
- 什么是 AWS Outposts？

相关视频：

- AWS re:Invent 2018：适用于多区域主动-主动应用程序的架构模式（ARC209-R2）
- AWS re:Invent 2019：AWS 全球网络基础设施的创新与运营（NET339）

REL10-BP02 为您的多位置部署选择合适的位置

期望结果

要实现高可用性，请始终（在可能时）将您的工作负载组件部署到多个可用区（AZ，Availability Zone），如图 10 中所示。对于具有极高弹性要求的工作负载，请谨慎评估用于多区域架构的选项。

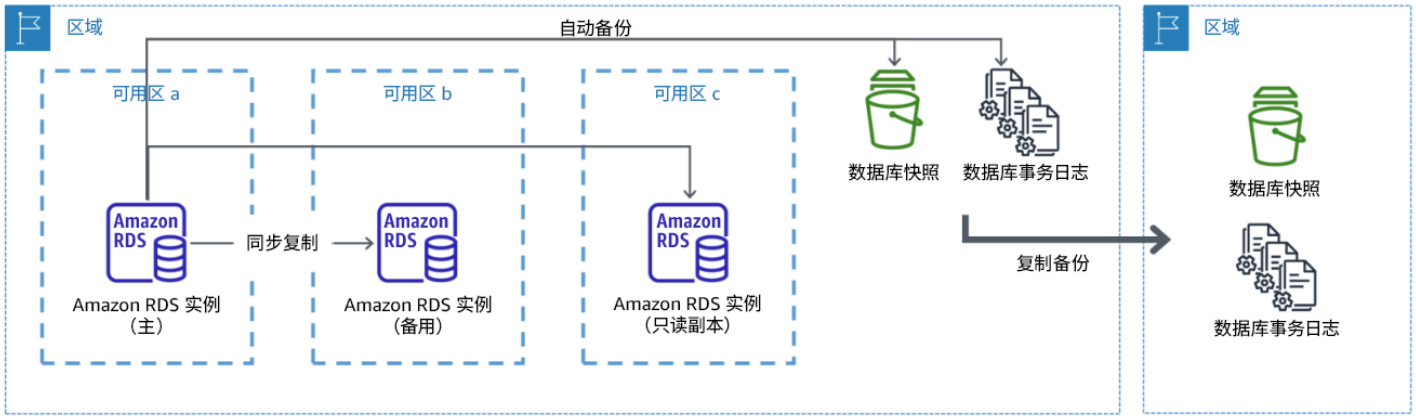


图 10：一个弹性多 AZ 数据库部署，该部署备份到另一个 AWS 区域

常见反模式

- 在多 AZ 架构可以满足需求时选择设计多区域架构。
- 当应用程序部件之间的弹性和多位置需求不同时，没有考虑它们之间的依赖关系。

建立此最佳实践的好处

要实现弹性，您应使用构建防御层的方法。其中一层使用多 AZ，通过构建高度可用的架构，防护较小规模的、更常见的中断。另一个防御层用于防御很少发生的事件，例如大范围的自然灾害和区域级别的中断。这个第二层涉及到设计应用程序的架构来跨越多个 AWS 区域。

- 99.5% 的可用性与 99.99% 的可用性相比，每个月的正常运行时间之差超过 3.5 小时。采用多 AZ 的工作负载的可用性，预期只能达到“四个九”。
- 通过在多个 AZ 中运行工作负载，您可以隔离电力、冷却和网络中的故障，以及火灾和洪水之类的大多数自然灾害。
- 为工作负载实施多区域策略，有助于防御影响到某个国家/地区中较大地理面积的大范围自然灾害，或者区域范围的技术故障。请注意，实施多区域架构会有很高的复杂性，对于大部分工作负载通常来说都是不必要的。

未建立这种最佳实践的情况下暴露的风险等级： 高

实施指导

对于一个可用区的中断或部分丢失而导致的灾难事件，在单个 AWS 区域内的多个可用区中实施高可用工作负载，有助于防范自然灾难和技术灾难。每个 AWS 区域由多个可用区组成，各个可用区之间实现了故障隔离并且间隔相当的物理距离。不过，对于可能造成间隔相当距离的多个可用区组件丢失风险的灾难事件，您应该实施灾难恢复选项，以防范整个区域的自然灾难和技术故障。对于需要极高弹性的工作负载（关键基础设施、与生命健康相关的应用程序、财务系统基础设施等），需要使用多区域策略。

实施步骤

1. 评估您的工作负载并确定需要使用多 AZ 方法（单个 AWS 区域）还是多区域方法才能够满足弹性需求。实施多区域架构来满足这些需求会引入额外的复杂性，因此请谨慎考虑您的使用场景及其需求。弹性需求几乎总是可以使用单个 AWS 区域来满足。在确定您是否需要使用多区域时，请考虑以下可能的需求：
    a. 灾难恢复（DR，Disaster Recovery）：对于一个可用区的中断或部分丢失而导致的灾难事件，在单个 AWS 区域内的多个可用区中实施高可用工作负载，有助于防范自然灾难和技术灾难。对于可能造成间隔相当距离的多个可用区组件丢失风险的灾难事件，您应该实施跨多个区域的灾难恢复，以防范整个区域的自然灾难和技术故障。
    b. 高可用性（HA，High Availability）：多区域架构（在每个区域中使用多个 AZ）可用于实现四个 9 以上（> 99.99%）的可用性。
    c. 堆栈本地化：面向全球受众部署工作负载时，您可以将本地化的堆栈部署在不同的 AWS 区域中，以便服务于这些区域中的受众。本地化可以包括语言、货币和所存储数据的类型。
    d. 靠近用户： 面向全球受众部署工作负载时，您可以通过在靠近最终用户所在位置的 AWS 区域中部署堆栈，从而减少延迟。

e. 数据驻留：一些工作负载面临着数据驻留要求，来自特定用户的数据必须保留在特定国家/地区的边界内。根据相关的法规，您可以选择将整个堆栈或者仅仅将数据部署到这些边界内的 AWS 区域中。

2. 以下是 AWS 服务提供的一些多 AZ 功能的示例：

   a. 为了使用 EC2 或 ECS 保护工作负载，请在计算资源前端部署 Elastic Load Balancer。然后，Elastic Load Balancing 提供解决方案来检测未正常运行的区中的实例，并将流量路由到正常运行的区中。

      i. 开始使用 Application Load Balancers

      ii. 开始使用网络负载均衡器

   b. 当 EC2 实例运行不支持负载均衡的现成商用软件时，您可以通过实施多 AZ 灾难恢复方法来实现某种形式的容错能力。

      i. the section called “REL13-BP02 使用定义的恢复策略来实现恢复目标”

   c. 对于 Amazon ECS 任务，将您的服务均匀地部署在三个 AZ 上以实现可用性与成本的平衡。

      i. Amazon ECS 可用性最佳实践 | 容器

   d. 对于非 Aurora Amazon RDS，您可以选择多 AZ 作为配置选项。在主数据库实例出现故障时，Amazon RDS 会自动提升备用数据库，用于接收其他可用区中的流量。还可以创建多区域只读副本来改进弹性。

      i. Amazon RDS 多可用区部署

      ii. 在不同 AWS 区域中创建只读副本

3. 以下是 AWS 服务提供的一些多区域功能的示例：

   a. 对于 Amazon S3 工作负载，当服务自动提供了多 AZ 可用性时，如果需要多区域部署，请考虑多区域接入点。

      i. Amazon S3 中的多区域接入点

   b. 对于 DynamoDB 表，此时服务自动提供了多 AZ 可用性，您可以轻松地将现有表转换为全局表来利用多区域的优势。

      i. 将单区域 Amazon DynamoDB 表转换为全局表

   c. 如果您的工作负载采用 Application Load Balancers 或网络负载均衡器作为前端，请将流量引导到包含正常运行端点的多个区域，从而使用 AWS Global Accelerator 来改进应用程序的可用性。

      i. AWS Global Accelerator 中标准加速器的端点 – AWS Global Accelerator（amazon.com）

   d. 对于利用 AWS EventBridge 的应用程序而言，请考虑使用跨区域总线来将事件转发到您选择的其他区域。

      i. 在 AWS 区域之间发送和接收 Amazon EventBridge 事件

e. 对于 Amazon Aurora 数据库，请考虑使用跨多个 AWS 区域的 Aurora 全局数据库。可以对现有集群进行修改来添加新的区域。

   i. 开始使用 Amazon Aurora 全局数据库

f. 如果您的工作负载包括 AWS Key Management Service（AWS KMS）加密密钥，请考虑多区域密钥是否适合您的应用程序。

   i. AWS KMS 中的多区域密钥

g. 对于其他 AWS 服务功能，请参阅此博客系列中的以下内容： 使用 AWS 服务创建多区域应用程序系列

实施计划的工作量级别： 中到高

资源

相关文档：

- 使用 AWS 服务创建多区域应用程序系列
- AWS 上的灾难恢复（DR，Disaster Recovery）架构，第 IV 部分：多站点主动/主动
- AWS 全球基础设施
- AWS Local Zones 常见问题
- AWS 上的灾难恢复（DR，Disaster Recovery）架构，第 I 部分：云中的恢复策略
- 云中的灾难恢复不相同
- 全局表：使用 DynamoDB 的多区域复制

相关视频：

- AWS re:Invent 2018：适用于多区域主动-主动应用程序的架构模式（ARC209-R2）
- Auth0：多区域高可用性架构，可扩展至每月 15亿+ 次登录，并具有自动故障转移功能

相关示例：

- AWS 上的灾难恢复（DR，Disaster Recovery）架构，第 I 部分：云中的恢复策略
- DTCC 实现了本地部署无法企及的弹性
- Expedia Group 使用具有专有 DNS 服务的多区域、多可用区架构来增加应用程序的弹性
- Uber：用于多区域 Kafka 的灾难恢复

- Netflix：实现多区域弹性的主动-主动架构
- 我们如何为 Atlassian Cloud 构建数据驻留
- Intuit TurboTax 跨两个区域运行

REL10-BP03 组件的自动恢复受限于单个位置

如果工作负载的组件只能在单个可用区或本地部署数据中心内运行，您必须实施相关功能，以在定义的恢复目标内彻底重建工作负载。

如果由于技术限制无法使用将工作负载部署到多个位置的最佳实践，您必须实施其他的弹性路径。在这种情况下，您必须让重建必要基础设施、重新部署应用程序和重建必要数据的操作实现自动化。

例如，Amazon EMR 会为相同可用区内的特定集群启动全部节点，因为在相同区内运行集群可以改善作业流的性能，提高数据访问速率。如果这是工作负载弹性所需的必要组件，则您必须设法重新部署集群及其数据。同样对于 Amazon EMR，您还应该通过除多可用区以外的方式对冗余进行预置。您可以预置 多个节点。使用 EMR 文件系统 (EMRFS)，EMR 中的数据可存储在 Amazon S3 内，进而可以实现跨多个可用区或 AWS 区域复制。

同样，对于 Amazon Redshift 来说，它默认会在您选择的 AWS 区域内随机选择可用区，然后对其中的集群进行预置。相同区内的全部集群节点都会被预置。

未建立这种最佳实践的情况下暴露的风险等级： 中

实施指导

- 实施自我修复。尽可能使用弹性伸缩部署实例或容器。如果不能使用弹性伸缩，则使用 EC2 实例的自动恢复功能，或者基于 Amazon EC2 或 ECS 容器生命周期事件实施自我修复自动化。
  - 将 Auto Scaling 组用于对单个实例 IP 地址、私有 IP 地址、弹性 IP 地址和实例元数据没有要求的实例和容器工作负载。
    - 什么是 EC2 Auto Scaling？
    - 服务弹性伸缩
      - 启动配置用户数据可以用于实现自动化，从而让大多数工作负载可以自我修复。
  - 将 EC2 实例的自动恢复功能用于需要单个实例 ID 地址、私有 IP 地址、弹性 IP 地址和实例元数据的工作负载。
    - 恢复您的实例。
      - 自动恢复功能会在检测到实例故障时，向 SNS 主题发送恢复状态提醒。

- 在无法使用弹性伸缩或 EC2 恢复的情况下，请使用 EC2 实例生命周期事件或 ECS 事件实现自我修复自动化。
  - EC2 Auto Scaling 生命周期钩子
  - Amazon ECS 事件
    - 使用这些事件调用自动化，该自动化将根据您需要的流程逻辑来修复组件。

资源

相关文档：

- Amazon ECS 事件
- EC2 Auto Scaling 生命周期钩子
- 恢复您的实例。
- 服务弹性伸缩
- 什么是 EC2 Auto Scaling？

REL10-BP04 采用隔板架构来限制影响范围

类似于船上的隔板，此模式确保将故障限制在一小部分请求或客户端，受损的请求数量有限，因此大部分请求可以继续执行而不会受错误影响。数据的隔板经常被称作分区，而服务的隔板称为单元格。

在 基于单元格的架构中，每个单元格都是完整、独立的服务实例，而且都有固定的最大大小。当负载增加时，工作负载会通过添加更多单元格而变大。分区键用于传入流量，以确定哪个单元格将处理请求。任何故障都会被限制在它所发生的单个单元格内，因此受损请求的数量有限，而其他单元格将继续执行而不受错误影响。确定适当的分区键，最大限度地减少跨单元格交互，以便使每个请求无需使用复杂的映射服务，这一点非常重要。需要复杂映射的服务最终只是把问题转移到映射服务上，而需要跨单元格交互的服务会在单元格之间创建依赖关系（因此这样做会减少假定的可用性改进）。

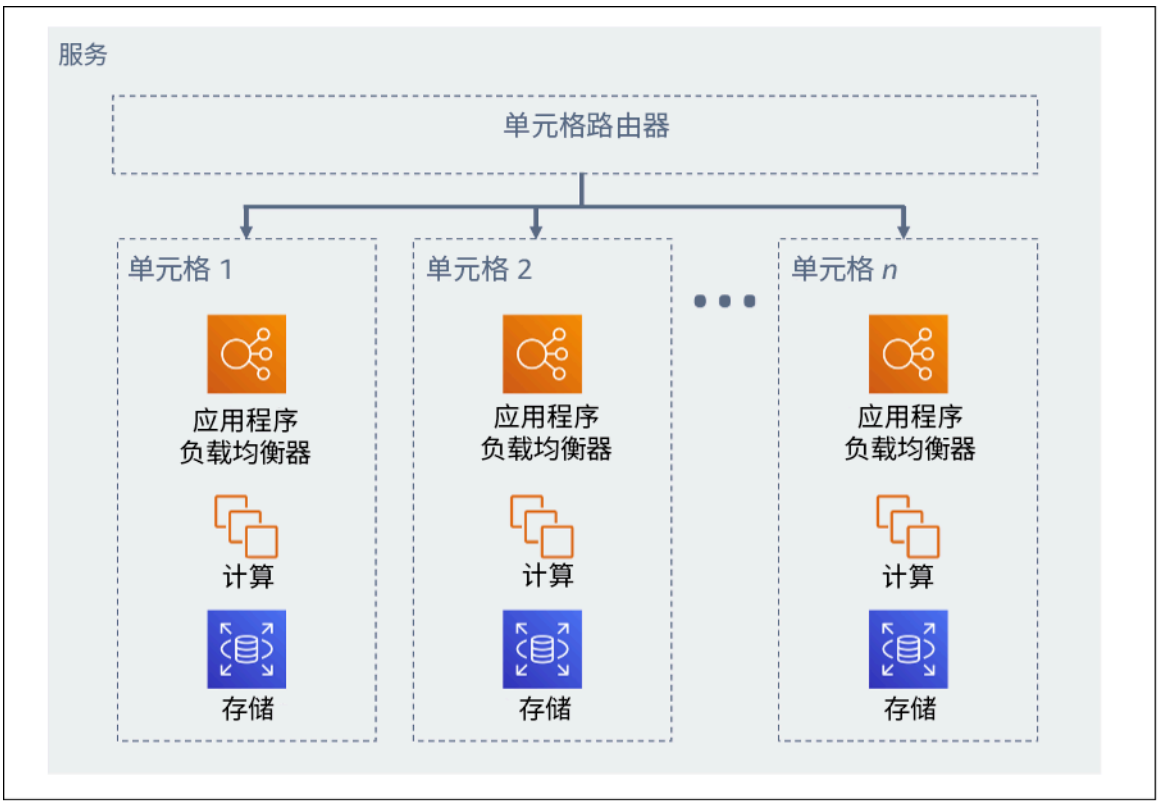


图 11：基于 Cell 的架构

Colm MacCarthaigh 在他的 AWS 博客中说明了 Amazon Route 53 如何利用 随机分区 的概念来隔离客户请求以避免影响其他分区。在此情况下，一个分区由两个或更多单元格组成。根据分区键，来自客户（或资源，或其他您想要隔离的对象）的流量会被路由至其指定的分区。若有八个单元格（每个分区中有两个单元格），而且在四个分区中划分客户，25% 的客户将在出现问题时受到影响。

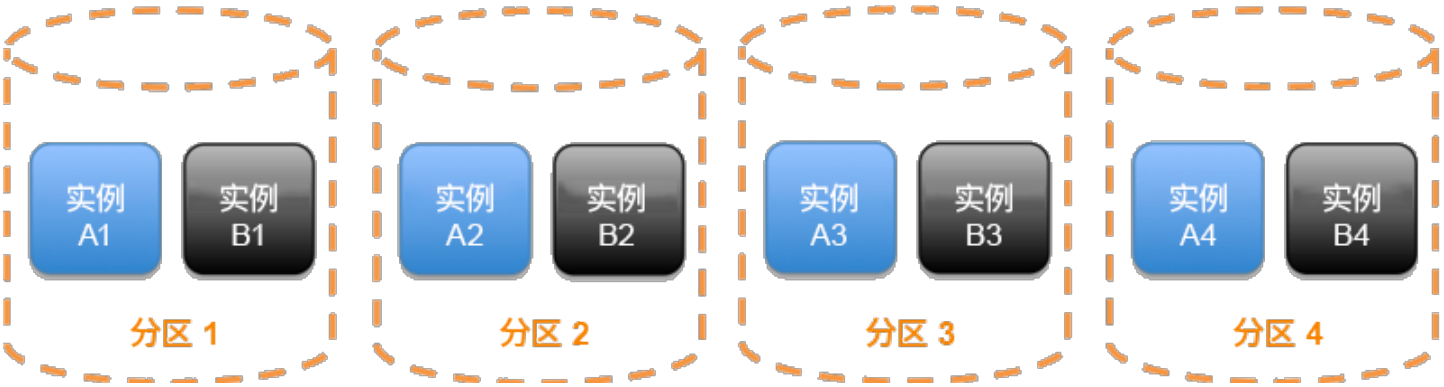


图 12：在四个传统分片内划分服务，每个分片有两个 Cell

通过随机分区，您可以创建由两个单元格组成的虚拟分区，然后将您的客户指定给其中的一个虚拟分区。当问题发生时，您还是会失去完整服务的四分之一，但分配客户或资源的方式意味着若采用随机分区，影响的范围会在很大程度上小于 25%。在八个单元格中，存在着 28 种由两个单元格组成的独特组

合，亦即有 28 种可能的随机分区（虚拟分区）。如果您有数百或数千个客户，并将每个客户指定给一个随机分区，那么问题的影响范围仅为 1/28。这比正常分区的情况好七倍。

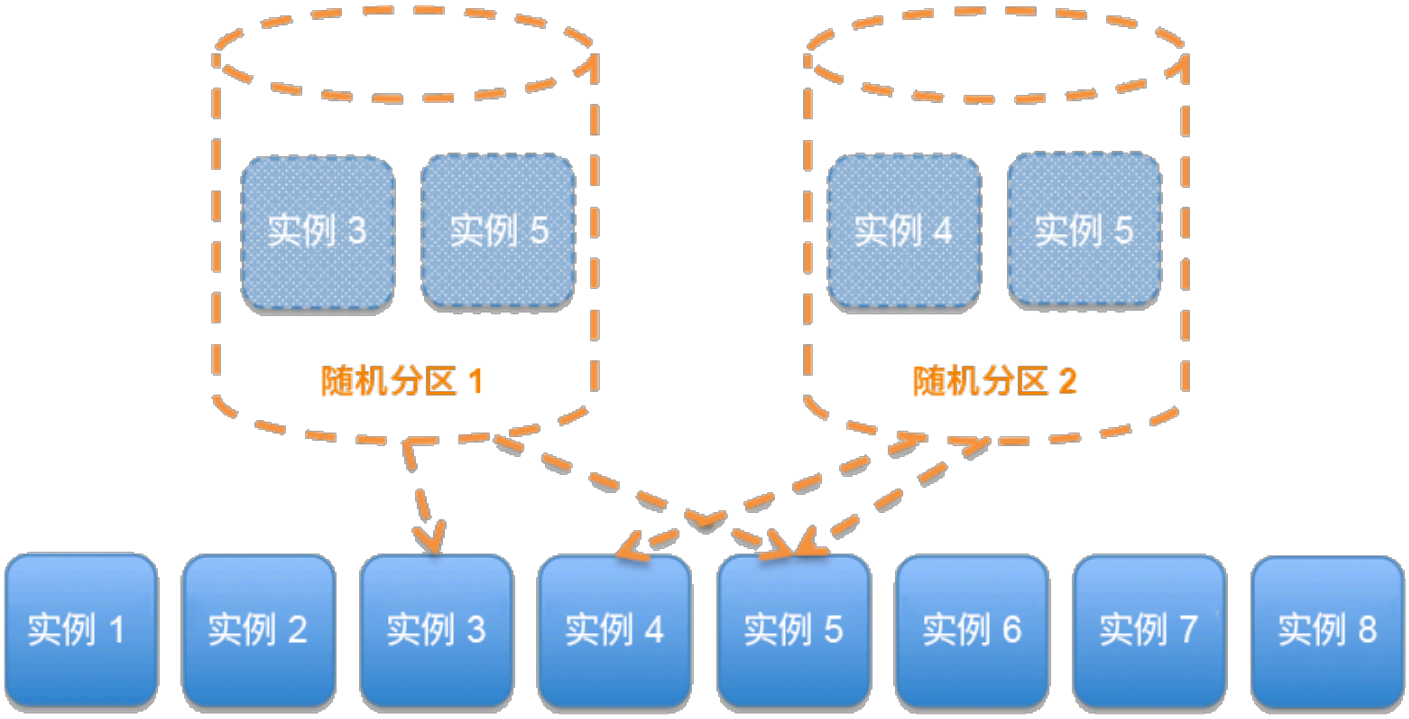


图 13：服务被划分到 28 个随机分片（虚拟分片），每个分片由两个 Cell 组成（仅显示 28 种可能中的两个随机分片）

除了单元格，分区还可用于服务器、队列或其他资源。

未建立这种最佳实践的情况下暴露的风险等级： 中

实施指导

- 采用隔板架构。类似于船上的隔板，此模式确保将故障限制在较小的请求或用户子集，受损的请求数量有限，因此大部分可以继续执行而不会受错误影响。数据的隔板经常被称作分区，而服务的隔板称为单元格。
  - Well-Architected 实验室：使用随机分区进行故障隔离
  - 随机分片：AWS re:Invent 2019：Amazon Builders' Library 简介（DOP328）
  - AWS re:Invent 2018：AWS 如何将故障的影响范围最小化（ARC338）
- 评估工作负载的基于 Cell 的架构。在基于单元格的架构中，每个单元格都是完整、独立的服务实例，而且都有固定的最大大小。当负载增加时，工作负载会通过添加更多单元格而变大。分区键用于传入流量，以确定哪个单元格将处理请求。任何故障都会被限制在它所发生的单个单元格内，因此受损请求的数量有限，而其他单元格将继续执行而不受错误影响。确定适当的分区键，最大限度地减少

跨单元格交互，以便使每个请求无需使用复杂的映射服务，这一点非常重要。需要复杂映射的服务最终只是把问题转移到映射服务上，而需要跨 Cell 交互的服务会降低 Cell 的自主性（因此，假定这样做可以提高可用性）。

- Colm MacCarthaigh 在他的 AWS 博客中说明了 Amazon Route 53 如何利用随机分片的概念来隔离客户请求以避免影响其他分片
  - 随机分区：神奇的大规模故障隔离

资源

相关文档：

- 随机分区：神奇的大规模故障隔离
- Amazon Builders' Library：采用随机分区进行工作负载隔离

相关视频：

- AWS re:Invent 2018：AWS 如何将故障的影响范围最小化（ARC338）
- 随机分片：AWS re:Invent 2019：Amazon Builders’ Library 简介（DOP328）

相关示例：

- Well-Architected 实验室：使用随机分区进行故障隔离

REL 11 如何将您的工作负载设计为可承受组件故障的影响？

在设计具有高可用性和较短平均恢复时间（MTTR）要求的工作负载时必须考虑到弹性。

最佳实践

- REL11-BP01 监控工作负载的所有组件以检测故障
- REL11-BP02 失效转移到运行状况良好的资源
- REL11-BP03 自动修复所有层
- REL11-BP04 恢复期间依赖于数据面板而不是控制面板
- REL11-BP05 使用静态稳定性来防止双模态行为
- REL11-BP06 当事件影响可用性时发出通知

REL11-BP01 监控工作负载的所有组件以检测故障

持续监控您的工作负载的运行状况，以便您和您的自动化系统立即发现任何性能下降或故障情况。监控基于商业价值的关键性能指标（KPI）。

所有恢复和修复机制必须从快速检测问题的能力入手。首先，应该检测技术故障并加以解决。不过，可用性基于您的工作负载创造商业价值的能力，因此衡量它的关键性能指标（KPI，Key Performance Indicator）需要成为您的检测和补救策略一部分。

常见反模式：

- 由于未配置警报，因此在发生中断时不会进行通知。
- 虽然存在警报，但只有在达到阈值时才会发出警报，导致没有足够的响应时间。
- 收集指标的频率不够高，无法满足恢复时间目标（RTO）。
- 只有面向客户的工作负载层才会受到主动监控。
- 只收集技术指标，不收集业务功能指标。
- 没有衡量工作负载用户体验的指标。

建立此最佳实践的好处： 如果您在所有层面都设置了适当的监控，则可以通过减少检测时间来减少恢复时间。

未建立这种最佳实践的情况下暴露的风险等级： 高

实施指导

- 根据恢复目标确定您的组件的收集间隔。
  - 您的监控间隔取决于您必须实现的恢复速度。您的恢复时间取决于恢复所需的时间，因此您在确定收集频率时，必须考虑此时间和恢复时间目标（RTO）。
- 为组件配置详细监控。
  - 确定是否需要为 EC2 实例和 Auto Scaling 配置详细监控。详细监控以 1 分钟为间隔提供指标，默认监控以 5 分钟为间隔提供指标。
    - 为您的实例启用或禁用详细监控
    - 使用 Amazon CloudWatch 监控自动扩缩组和实例
  - 确定是否需要为 RDS 设置增强监控。增强监控使用 RDS 实例上的代理来获取关于 RDS 实例上不同进程或线程的有用信息。
    - 增强监控

- 创建自定义指标来测量业务关键性能指标（KPI，Key Performance Indicator）。工作负载实现关键业务功能。这些功能应用作 KPI 来帮助在发生间接问题时确定这些问题。
  - 发布自定义指标
- 使用用户金丝雀来监控用户的故障体验。可以运行综合事务测试（又称为“金丝雀测试”，但不要和金丝雀部署相混淆）模拟客户的行为，这是最重要的测试流程之一。从不同的远程位置针对您的工作负载端点持续地运行此类测试。
  - Amazon CloudWatch Synthetics 使您能够创建用户金丝雀
- 创建跟踪用户体验的自定义指标。如果您可以衡量客户体验，就可以确定发生了客户体验下降。
  - 发布自定义指标
- 设置告警，以在检测到工作负载未正常运行时发出告警，并指示什么时候对资源进行弹性伸缩。告警可以显示在控制面板上，可通过 Amazon SNS 或电子邮件发送提醒，并使用 Auto Scaling 来纵向扩展或缩减工作负载的资源。
  - 使用 Amazon CloudWatch 告警
- 创建控制面板以可视化形式呈现指标。可以使用控制面板直观地查看趋势、离群值和表示其他潜在问题的指标，或者提供您可能需要进行调查的问题的指示。
  - 使用 CloudWatch 控制面板

资源

相关文档：

- Amazon CloudWatch Synthetics 使您能够创建用户金丝雀
- 为您的实例启用或禁用详细监控
- 增强监控
- 使用 Amazon CloudWatch 监控自动扩缩组和实例
- 发布自定义指标
- 使用 Amazon CloudWatch 告警
- 使用 CloudWatch 控制面板

相关示例：

- Well-Architected 实验室：第 300 级：实施运行状况检查和管理依赖项以提高可靠性

REL11-BP02 失效转移到运行状况良好的资源

确保如果某个资源发生故障，该运行状况良好的资源可以继续为请求提供服务。对于位置故障（如可用区或 AWS 区域），确保您拥有适当的系统以失效转移到未受损位置内运行状况良好的资源。

Elastic Load Balancing 和 AWS Auto Scaling 等 AWS 服务有助于跨资源和可用区分配负载。因此，可以通过将流量转移到运行状况良好的剩余资源，缓解单个资源（例如 EC2 实例）的故障或可用区的损坏。对于多区域工作负载就比较复杂。例如，跨区域只读副本让您可以将数据部署到多个 AWS 区域，但在发生失效转移时，您仍必须提升只读副本至主节点，并将流量指向该节点。Amazon Route 53 和 AWS Global Accelerator 可以帮助跨 AWS 区域路由流量。

如果您的工作负载使用 Amazon S3 或 Amazon DynamoDB 等 AWS 服务，则它们会自动部署到多个可用区。当发生故障时，AWS 控制面板会自动为您将流量路由至运行正常的位置。数据在多个可用区中进行冗余存储，并保持可用。针对 Amazon RDS，您必须选择多可用区作为配置选项，然后在发生故障时，AWS 会自动将流量定向至运行正常的实例。对于 Amazon EC2 实例、Amazon ECS 任务或 Amazon EKS 容器组（pod），您要选择部署到哪些可用区。然后，Elastic Load Balancing 会提供解决方案以检测运行不正常区内的实例，并将流量路由至运行正常的区。Elastic Load Balancing 甚至可以将流量路由至本地数据中心内的组件。

针对多区域方法（也可能包括本地数据中心），Amazon Route 53 会提供定义互联网域并指定路由策略的方式，而此类策略可能包含运行状况检查，以确保流量被路由至运行正常的区域。此外，AWS Global Accelerator 也可以提供静态 IP 地址作为您的应用程序的固定接入点，然后通过 AWS 全球网络而不是互联网路由至您选择的 AWS 区域内的终端节点，以提高性能和可靠性。

AWS 在设计服务时始终会考虑故障恢复功能。我们设计服务时会尽量缩短从故障恢复的时间并降低对数据的影响。我们的服务主要使用的数据存储，只有在数据持久存储在一个区域中的多个副本之后，才会确认请求。这些服务和资源包括 Amazon Aurora、Amazon Relational Database Service (Amazon RDS) 多可用区数据库实例、Amazon S3、Amazon DynamoDB、Amazon Simple Queue Service (Amazon SQS) 和 Amazon Elastic File System (Amazon EFS)。它们被构建为使用基于单元格的隔离，并使用可用区提供的故障隔离功能。我们在自己的运营过程中广泛使用自动化。我们还将替换和重新启动功能优化为可从中断快速恢复。

未建立这种最佳实践的情况下暴露的风险等级： 高

实施指导

- 失效转移到运行状况良好的资源。确保如果某个资源发生故障，该运行状况良好的资源可以继续为请求提供服务。对于位置故障（如可用区或 AWS 区域），确保您拥有适当的系统以失效转移到未受损位置内运行状况良好的资源。

- 如果您的工作负载使用 Amazon S3 或 Amazon DynamoDB 等 AWS 服务，则它们会自动部署到多个可用区。当发生故障时，AWS 控制面板会自动为您将流量路由至运行正常的位置。
- 针对 Amazon RDS，您必须选择多可用区作为配置选项，然后在发生故障时，AWS 会自动将流量定向至运行正常的实例。
  - Amazon RDS 的高可用性（多可用区）
- 对于 Amazon EC2 实例或 Amazon ECS 任务，您要选择部署到哪些可用区。然后，Elastic Load Balancing 会提供解决方案以检测运行不正常区内的实例，并将流量路由至运行正常的区。Elastic Load Balancing 甚至可以将流量路由至本地数据中心内的组件。
- 针对多区域方案（也可能包括本地部署数据中心），要确保来自正常运行位置的数据和资源可以继续用于处理请求
  - 例如，跨区域只读副本让您可以将数据部署到多个 AWS 区域，但在主要位置发生故障时，您仍必须提升只读副本至主节点，并将流量指向该节点。
    - Amazon RDS 只读副本概述
  - Amazon Route 53 提供定义互联网域并指定路由策略的方式，而此类策略可能包含运行状况检查，以确保流量被路由至运行正常的区域。此外，AWS Global Accelerator 也可以提供静态 IP 地址作为您的应用程序的固定接入点，然后通过 AWS 全球网络而不是互联网路由至您选择的 AWS 区域内的端点，以提高性能和可靠性。
    - Amazon Route 53：选择一个路由策略
    - 什么是 AWS Global Accelerator？

资源

相关文档：

- AWS 合作伙伴：可以帮助您实现容错自动化的合作伙伴
- AWS Marketplace：可以支持容错的产品
- AWS OpsWorks：使用自动修复来更换失败的实例
- Amazon Route 53：选择一个路由策略
- Amazon RDS 的高可用性（多可用区）
- Amazon RDS 只读副本概述
- Amazon ECS 任务置放策略
- 为多个可用区创建 Kubernetes 自动扩缩组
- 什么是 AWS Global Accelerator？

相关示例：

- Well-Architected 实验室：第 300 级：实施运行状况检查和管理依赖项以提高可靠性

REL11-BP03 自动修复所有层

在检测到故障时，使用自动化功能执行修复操作。

重启功能 是用于修复故障的重要工具。正如我们之前在分布式系统部分讨论过那样，最佳实践是尽可能使服务为无状态。它可以防止重启时数据丢失或可用性受损。您可以（而且在一般情况下也应该）在云中替换完整的资源（如 EC2 实例或 Lambda 函数），并将其作为重启的一部分。重启本身是从故障恢复的简单而可靠的方法。工作负载中会发生很多不同类型的故障。故障可能发生在硬件、软件、通信和操作上。将众多不同类别的故障映射到相同的恢复策略上，而不是构建新颖的机制来捕获、确定和纠正各个不同类型的故障。实例可能会因为硬件故障、操作系统错误、内存泄漏或其他原因而出现故障。系统不会针对每种情况构建自定义修复，而是会将它们全部视为实例故障。终止实例，并且允许使用 AWS Auto Scaling 进行取代。然后，在带外对故障资源进行分析。

另一个例子是重启网络请求功能。向网络超时以及依赖项返回错误的依赖性故障应用相同的恢复方法。这两个事件对系统具有类似的影响，应用类似的采用指数回退和抖动的有限重试策略，而不是尝试将各个事件当作特例进行处理。

重启功能 是面向恢复的计算和高可用性集群架构的特色恢复机制。

Amazon EventBridge 可用于监控和筛选事件，例如 CloudWatch 警报或其他 AWS 服务中的状态更改。根据事件信息，它可以触发 AWS Lambda、AWS Systems Manager Automation（或其他目标）在您的工作负载上执行自定义修复逻辑。

Amazon EC2 Auto Scaling 可以配置为对 EC2 实例的运行状况进行检查。若实例处于正在运行以外的任何状态，或系统状态受损，Amazon EC2 Auto Scaling 会认为实例的运行不正常，并且启动替换实例。如果使用 AWS OpsWorks，您可以在 OpsWorks 层级别配置 EC2 实例的自动修复。

针对大规模替换（如整个可用区受损），静态稳定性更适合高可用性，而不是立即尝试获取多个新的资源。

常见反模式：

- 在实例或容器中单独部署应用程序。
- 在不使用自动恢复的情况下，部署无法部署到多个位置的应用程序。
- 手动修复自动扩展和自动恢复无法修复的应用程序。

建立此最佳实践的好处： 自动修复，即使工作负载一次只能部署到一个位置也能减少平均恢复时间，并确保工作负载的可用性。

未建立这种最佳实践的情况下暴露的风险等级： 高

实施指导

- 使用自动扩缩组在工作负载中部署多个层。Auto Scaling 可以对无状态应用程序执行自我修复，以及添加和移除容量。
  - AWS Auto Scaling 的工作原理
- 在部署了无法部署到多个位置，但在故障后允许重新启动的应用程序的 EC2 实例上，实施自动恢复。当无法将应用程序部署到多个位置时，可以使用自动恢复来替换发生故障的硬件并重新启动实例。实例元数据和关联的 IP 地址将被保留，Amazon EBS 卷以及 Elastic File System 或 File Systems for Lustre 和 File Systems for Windows 的挂载点也是如此。
  - Amazon EC2 Automatic Recovery
  - Amazon Elastic Block Store（Amazon EBS）
  - Amazon Elastic File System（Amazon EFS）
  - 什么是 Amazon FSx for Lustre？
  - 什么是 Amazon FSx for Windows File Server？
    - 使用 AWS OpsWorks 时，您可以在层级别配置 EC2 实例的自动修复。
      - AWS OpsWorks：使用自动修复来更换失败的实例
- 当您无法使用自动扩展或自动恢复时，或者自动恢复出故障时，使用 AWS Step Functions 和 AWS Lambda 实施自动恢复。当您无法使用自动扩展，并且无法使用自动恢复或自动恢复失败时，您可以使用 AWS Step Functions 和 AWS Lambda 进行自动修复。
  - 什么是 AWS Step Functions？
  - 什么是 AWS Lambda？
    - Amazon EventBridge 可用于监控和筛选事件，例如 CloudWatch 警报或其他 AWS 服务中的状态更改。根据事件信息，它可以触发 AWS Lambda（或其他目标）在您的工作负载上运行自定义修复逻辑。
      - 什么是 Amazon EventBridge？
      - 使用 Amazon CloudWatch 告警

资源

相关文档：

- AWS 合作伙伴：可以帮助您实现容错自动化的合作伙伴
- AWS Marketplace：可以支持容错的产品
- AWS OpsWorks：使用自动修复来更换失败的实例
- Amazon EC2 Automatic Recovery
- Amazon Elastic Block Store（Amazon EBS）
- Amazon Elastic File System（Amazon EFS）
- AWS Auto Scaling 的工作原理
- 使用 Amazon CloudWatch 告警
- 什么是 Amazon EventBridge？
- 什么是 AWS Lambda？
- AWS Systems Manager Automation
- 什么是 AWS Step Functions？
- 什么是 Amazon FSx for Lustre？
- 什么是 Amazon FSx for Windows File Server？

相关视频：

- AWS 中的静态稳定性：AWS re:Invent 2019：Amazon Builders' Library 简介（DOP328）

相关示例：

- Well-Architected 实验室：第 300 级：实施运行状况检查和管理依赖项以提高可靠性

REL11-BP04 恢复期间依赖于数据面板而不是控制面板

控制面板用于配置资源，数据面板可提供服务。数据面板通常比控制面板具有更高的可用性设计目标，并且通常不太复杂。在对可能影响弹性的事件实施恢复或缓解响应时，使用控制面板操作会降低架构的整体弹性。例如，您可以依靠 Amazon Route 53 数据面板，根据运行状况检查可靠地路由 DNS 查询，但更新 Route 53 路由策略时使用控制面板，因此不要依赖它进行恢复。

Route 53 数据面板回复 DNS 查询，并执行和评估运行状况检查。它们分布在全球各地，专为 100% 可用性的服务等级协议（SLA，Service Level Agreement）而设计。 您用于创建、更新和删除 Route 53 资源的 Route 53 管理 API 和控制台是在控制面板上运行的，而这些控制面板设计用于优先考虑

您在管理 DNS 时所需的强一致性和持久性。为了实现这一点，控制面板位于单个区域 US East (N. Virginia) 中。虽然这两个系统都非常可靠，但控制面板不包含在 SLA 中。在极少数情况下，数据面板的弹性设计允许它保持可用性，而控制面板做不到。对于灾难恢复和失效转移机制，使用数据面板功能可提供尽可能好的可靠性。

有关数据面板、控制面板以及 AWS 如何构建服务以满足高可用性目标的更多信息，请参阅 使用可用区的静态稳定性 文章以及 Amazon Builders' Library。

未建立此最佳实践暴露的风险等级： 高

实施指导

- 使用 Amazon Route 53 进行灾难恢复时依赖数据面板而不是控制面板。Route 53 Application Recovery Controller 使用就绪性检查和路由控制来帮助您管理和协调失效转移。这些功能持续监控您的应用程序从故障中恢复的能力，并使您能够跨多个 AWS 区域、可用区和本地部署控制您的应用程序恢复。
  - 什么是 Route 53 Application Recovery Controller
  - 使用 Amazon Route 53 创建灾难恢复机制
  - 使用 Amazon Route 53 Application Recovery Controller 构建高弹性应用程序，第 1 部分：单区域堆栈
  - 使用 Amazon Route 53 Application Recovery Controller 构建高弹性应用程序，第 2 部分：多区域堆栈
- 了解哪些操作位于数据面板以及哪些位于控制面板。
  - Amazon Builders' Library：通过控制较小的服务来避免分布式系统中出现过载
  - Amazon DynamoDB API（控制面板和数据面板）
  - AWS Lambda 执行（拆分为控制面板和数据面板）
  - AWS Lambda 执行（拆分为控制面板和数据面板）

资源

相关文档：

- AWS 合作伙伴：可以帮助您实现容错自动化的合作伙伴
- AWS Marketplace：可以支持容错的产品
- Amazon Builders' Library：通过控制较小的服务来避免分布式系统中出现过载
- Amazon DynamoDB API（控制面板和数据面板）

- AWS Lambda 执行（拆分为控制面板和数据面板）
- AWS Elemental MediaStore 数据面板
- 使用 Amazon Route 53 Application Recovery Controller 构建高弹性应用程序，第 1 部分：单区域堆栈
- 使用 Amazon Route 53 Application Recovery Controller 构建高弹性应用程序，第 2 部分：多区域堆栈
- 使用 Amazon Route 53 创建灾难恢复机制
- 什么是 Route 53 Application Recovery Controller

相关示例：

- Amazon Route 53 Application Recovery Controller 简介

REL11-BP05 使用静态稳定性来防止双模态行为

双模态行为是指您的工作负载在正常和故障模式下展现出不同的行为，例如，可用区发生故障时依赖于启动新的实例。您应该构建静态稳定的工作负载，并且仅在一个模式下运行。在这种情况下，如果删除了一个可用区，要在每个可用区内预置足够的实例来处理工作负载，然后再使用 Elastic Load Balancing 或 Amazon Route 53 运行状况检查将负载从受损实例中转出。

适用于计算部署（如 EC2 实例或容器）的静态稳定性将提供最高水平的可靠性。您必须在稳定性水平和成本之间认真权衡。预置较小的计算容量，并在发生故障时依赖启动新实例，其成本较低。但对于大规模故障（如可用区故障）来说，此方法的效果较差，因为它依赖于对发生的损坏做出反应，而不会在损坏发生前做好准备。您选择的解决方案应在工作负载的可用性和成本需求之间做出取舍。若使用更多可用区，静态稳定性所需的额外计算量就会减少。

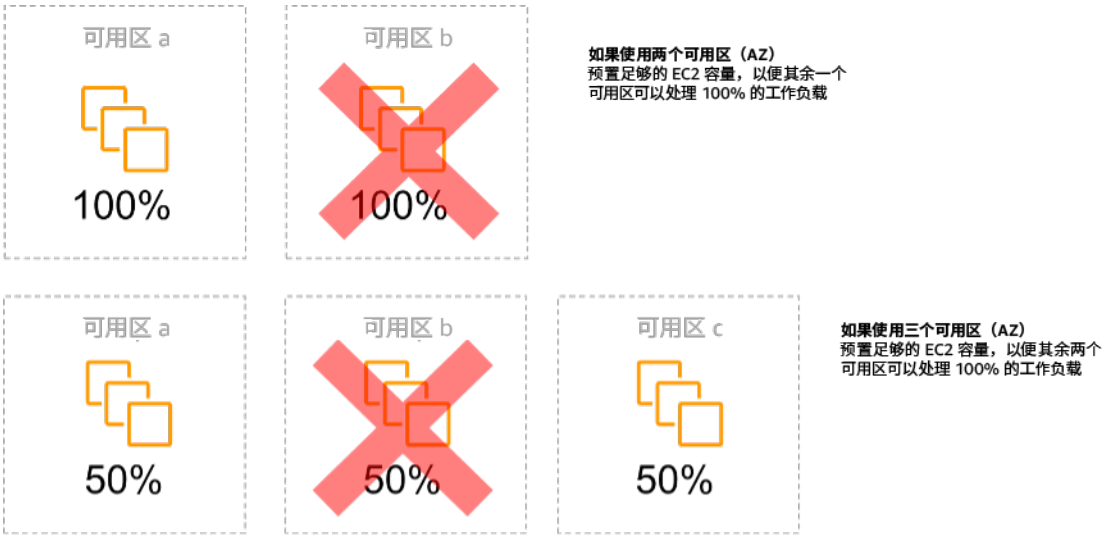


图 14：跨可用区的 EC2 实例的静态稳定性

在流量转移以后，使用 AWS Auto Scaling 异步替换故障区的实例，并且在运行正常的区内启动。

双模态行为的另一个例子是，网络超时可能会导致系统尝试刷新整个系统的配置状态。这会向另一个组件添加意外负载，进而可能导致该组件出现故障，触发其他意外后果。此负面的反馈环路会影响您的工作负载的可用性。您应该构建静态稳定的系统，并且仅在一个模式下运行。静态稳定的设计会持续工作，并且始终定期刷新配置状态。当调用失败时，工作负载会使用之前的缓存值，并触发警报。

双模态行为的另一个示例是允许客户端在故障发生时绕过您的工作负载缓存。这看起来似乎是可以满足客户端需求的解决方案但却不应该被允许，因为它会明显改变您的工作负载的需求，而且很有可能导致故障。

未建立此最佳实践暴露的风险等级： 中

实施指导

- 使用静态稳定性来防止双模态行为。双模态行为是指您的工作负载在正常和故障模式下展现出不同的行为，例如，可用区发生故障时依赖于启动新的实例。
  - 尽可能减小灾难恢复计划中的依赖关系
  - Amazon Builders' Library：使用可用区的静态稳定性
  - AWS 中的静态稳定性：AWS re:Invent 2019：Amazon Builders’ Library 简介（DOP328）
    - 您应该构建静态稳定的系统，并且仅在一个模式下运行。在这种情况下，在每个区内预置足够的实例来处理删除了一个工作区时的工作负载，然后再使用 Elastic Load Balancing 或 Amazon Route 53 运行状况检查将负载从受损实例中转出。
    - 双模态行为的另一个示例是允许客户端在故障发生时绕过您的工作负载缓存。这看起来似乎是可以满足客户端需求的解决方案，但却不应该被允许，因为它会明显改变您的工作负载的需求，而且很有可能导致故障。

资源

相关文档：

- 尽可能减小灾难恢复计划中的依赖关系
- Amazon Builders' Library：使用可用区的静态稳定性

相关视频：

- AWS 中的静态稳定性：AWS re:Invent 2019：Amazon Builders' Library 简介（DOP328）

REL11-BP06 当事件影响可用性时发出通知

在检测到重大事件时发送通知，即使由事件引发的问题已经自动解决。

自动修复使您的工作负载变得可靠。不过，它也可能会掩盖需要处理的潜在问题。实施适当的监控和措施，以便检测问题的模式，包括那些被自动修复的问题，从而从根本上解决问题。Amazon CloudWatch 警报会基于发生的故障触发。它们还可能由于执行自动修复操作而被触发。CloudWatch 警报可被配置为发送电子邮件，或使用 Amazon SNS 集成将事件记录到第三方事件跟踪系统。

常见反模式：

- 发出不需要有人采取措施的告警。
- 执行自动修复，但不通知需要进行该修复。

建立此最佳实践的好处： 恢复事件通知将确保您不会忽略不经常发生的问题。

未建立此最佳实践暴露的风险等级： 中

实施指导

- 在业务关键性能指标超出低阈值时发出警报：收到关于您的业务 KPI 的低阈值告警，可帮助您及时了解工作负载不可用或未正常工作的情况。
  - 基于静态阈值创建 CloudWatch 告警
- 针对调用自动修复的事件发出告警：您可以使用任何已创建的自动化功能直接调用 SNS API 来发送通知。
  - 什么是 Amazon Simple Notification Service？

资源

相关文档：

- 基于静态阈值创建 CloudWatch 告警
- 什么是 Amazon EventBridge？
- 什么是 Amazon Simple Notification Service？

REL 12 如何测试可靠性？

在为您的工作负载采用弹性设计以应对生产压力以后，测试是确保其按设计预期运行，并且提供您所预期弹性的唯一方式。

最佳实践

- REL12-BP01 使用行动手册调查故障
- REL12-BP02 在意外事件发生后执行分析
- REL12-BP03 测试功能要求
- REL12-BP04 测试扩展和性能要求
- REL12-BP05 使用混沌工程测试弹性
- REL12-BP06 定期进行实际试用

REL12-BP01 使用行动手册调查故障

通过在行动手册中记录调查流程，实现对并不十分了解的故障场景做出一致且及时的响应。行动手册是在确定哪些因素导致故障场景时要执行的预定义步骤。所有流程步骤的结果都将用于确定要采取的后续步骤，直到问题得到确定或上报。

行动手册是您必须要执行的主动计划，以便有效采取响应措施。当在生产中遇到行动手册未涉及的故障场景时，首先要解决问题（灭火）。然后回过头来思考您在解决问题时采取的措施，并将这些措施作为新条目添加到行动手册中。

请注意，行动手册可用于对特定事件做出响应，运行手册则用来达成特定的结果。通常，运行手册适用于例行活动，而行动手册则被用于对非例行事件做出响应。

常见反模式：

- 计划在以下情况下部署工作负载：不清楚诊断问题或响应意外事件的流程。
- 关于在对事件进行调查时从哪些系统收集日志和指标的计划外的决定。
- 指标和事件保留的时间不够长，无法检索到数据。

建立此最佳实践的好处： 使用行动手册可确保始终如一地遵循程序。编写行动手册可以减少手动操作导致的错误。通过实现行动手册自动化，可以消除团队成员干预的需要，或者在他们开始干预时便向他们提供更多信息，从而缩短事件响应时间。

未建立此最佳实践暴露的风险等级： 高

实施指导

- 使用行动手册来发现问题。管理手册是用于调查问题的书面程序。在行动手册中记录流程，实现对故障场景的一致而及时的响应。行动手册必须包含所需的信息和指导，让足够熟练的员工能够收集适用信息、确定故障的潜在来源、隔离故障，并确定成因（在意外事件发生后执行分析）。
  - 以代码形式实施行动手册。为行动手册编写脚本，以代码形式执行运营，以确保一致性并减少由手动流程引起的错误。行动手册可以由代表不同步骤的多个脚本组成，这些步骤可能是确定问题成因所必需的。系统可能会在运行手册活动过程中触发或执行行动手册活动，也可能针对响应发现的事件而提示执行行动手册活动。
    - 使用 AWS Systems Manager 自动执行您的运营手册
    - AWS Systems Manager Run Command
    - AWS Systems Manager Automation
    - 什么是 AWS Lambda？
    - 什么是 Amazon EventBridge？
    - 使用 Amazon CloudWatch 告警

资源

相关文档：

- AWS Systems Manager Automation
- AWS Systems Manager Run Command
- 使用 AWS Systems Manager 自动执行您的运营手册
- 使用 Amazon CloudWatch 告警
- 使用金丝雀（Amazon CloudWatch Synthetics）
- 什么是 Amazon EventBridge？
- 什么是 AWS Lambda？

相关示例：

- 使用行动手册和运行手册自动完成操作

REL12-BP02 在意外事件发生后执行分析

审核影响客户的事件，确定这些事件的成因和预防措施。利用这些信息来制定缓解措施，以限制或防止再次发生同类事件。制定程序以迅速有效地做出响应。根据目标受众，适当传达事件成因和纠正措施。如果需要，可将这些原因告知他人。

评估为什么现有测试找不到问题。如果还没有，增设测试。

常见反模式：

- 查找事件成因，但不继续深入探究其他潜在问题和缓解问题的方法。
- 只找出人为错误原因，但不提供任何培训或可防止人为错误的自动化功能。

建立此最佳实践的好处： 如果其他工作负载实施了相同的故障因素，那么在意外事件发生后执行分析并共享分析结果可帮助缓解这些工作负载的故障风险，并使它们能够在意外事件发生之前实施缓解或自动恢复措施。

未建立这种最佳实践的情况下暴露的风险等级： 高

实施指导

- 制定事后分析标准。有效的事后分析让您有机会针对系统中其他地方使用的架构模式存在的问题提出常见的解决方案。
  - 确保在提出事件成因时秉承诚实原则并且不苛责。
  - 如果您不记录问题，就无法予以纠正。
    - 确保事后分析不带苛责，这样您便可以冷静地看待建议的纠正措施，并在您的应用程序团队中促进诚实的自我评估和协作。
- 通过流程来确定事件成因。设置流程来确定和记录事件成因，以便制定缓解措施来限制或阻止事件再次发生，并且您还可以据此制定及时有效的应对措施。根据目标受众，适当传达成因。
  - 什么是日志分析？

资源

相关文档：

- 什么是日志分析？
- 为什么您应该制定更正错误（COE，Correction of Error）措施

REL12-BP03 测试功能要求

使用的技术包括用于验证所需功能的单元测试和集成测试。

如果这些测试作为构建和部署措施的一部分自动运行，则您可以获得最佳的结果。例如，使用 AWS CodePipeline，开发人员会在 CodePipeline 自动检测到变更时提交对源存储库的更改。执行更改，然后加以测试。在测试完成以后，构建的代码会被部署到用于测试的暂存服务器。CodePipeline 会从暂存服务器运行更多测试，如集成或负载测试等。在成功完成此类测试以后，CodePipeline 会将经过测试并获得批准的代码部署到生产实例。

此外，过去的经验告诉我们，可运行合成事务测试（又被称作 金丝雀测试，但不要和金丝雀部署相混淆）模拟用户行为，这是最重要的测试流程之一。从不同的远程位置针对您的工作负载端点持续地运行此类测试。Amazon CloudWatch Synthetics 使您能够 创建 Canary 以便监控您的终端节点和 API。

未建立此最佳实践暴露的风险等级： 高

实施指导

- 测试功能要求。这包括用于验证所需功能的单元测试和集成测试。
  - 将 CodePipeline 与 AWS CodeBuild 结合使用以测试代码和运行构建
  - AWS CodePipeline 增加了对通过 AWS CodeBuild 进行单位和自定义集成测试的支持
  - 持续交付和持续集成
  - 使用金丝雀（Amazon CloudWatch Synthetics）
  - 软件测试自动化

资源

相关文档：

- AWS 合作伙伴：可帮助实施持续集成管道的合作伙伴
- AWS CodePipeline 增加了对通过 AWS CodeBuild 进行单位和自定义集成测试的支持
- AWS Marketplace：可用于实现持续集成的产品
- 持续交付和持续集成
- 软件测试自动化
- 将 CodePipeline 与 AWS CodeBuild 结合使用以测试代码和运行构建
- 使用金丝雀（Amazon CloudWatch Synthetics）

REL12-BP04 测试扩展和性能要求

使用的技术包括负载测试以验证工作负载是否满足扩展和性能要求。

在云中，您可以按需为您的工作负载创建生产规模环境。如果在缩减的基础设施上运行这些测试，您必须根据您认为在生产中将会发生的情况扩展您观察到的结果。如果不想影响实际用户，您可以在生产中开展负载和性能测试，并且对您的测试数据进行标记，以避免它与真实的用户数据、损坏的使用情况统计或生产报告混在一起。

通过测试确保您的基础资源、扩展设置、服务限额和弹性设计能够在负载之下如预期运行。

未建立这种最佳实践的情况下暴露的风险等级： 高

实施指导

- 测试扩展和性能要求。执行负载测试以验证工作负载是否满足扩展和性能要求。
  - AWS 上的分布式负载测试：模拟数千个连接的用户
  - Apache JMeter
    - 将您的应用程序部署在与生产环境相同的环境中，然后执行负载测试。
      - 使用基础设施即代码概念，以创建尽可能类似于生产环境的环境。

资源

相关文档：

- AWS 上的分布式负载测试：模拟数千个连接的用户
- Apache JMeter

REL12-BP05 使用混沌工程测试弹性

在处于或尽可能接近生产的环境中定期运行混沌试验，以了解系统如何应对不利条件。

期望结果：

除了在事件期间验证已知预期工作负载行为的弹性测试之外，还可以通过以故障注入实验或注入意外负载的形式应用混沌工程，定期验证工作负载的弹性。将混沌工程和弹性测试结合起来，您可以提升信心，相信工作负载能够经受组件故障，并可从意外中断中恢复，而影响极小甚至没有影响。

常见反模式：

- 进行弹性设计，但不验证故障发生时工作负载如何作为一个整体运行。
- 从不在真实环境和预期负载下进行试验。
- 不将实验视为代码，也不在整个开发周期中维护它们。
- 不将混沌实验作为 CI/CD 管道的一部分，也不在部署之外运行。
- 在确定要对哪些故障进行试验时，没有想到使用过去的意外事件后分析。

建立此最佳实践的好处： 注入故障来验证工作负载的弹性，这可以让您提升信心，相信您的弹性设计的恢复程序将在真正发生故障的情况下能够发挥作用。

未建立这种最佳实践的情况下暴露的风险等级： 中

实施指导

利用混沌工程，您的团队能够在服务提供商、基础设施、工作负载和组件级别，以可控的方式不断注入真实世界的干扰（模拟），而对客户的影响极小甚至没有影响。它使您的团队能够从故障中学习，观察、测量和提高工作负载的弹性，并验证在发生事件时，系统会发出警报并通知团队。

当持续执行时，混沌工程会突出工作负载中的缺陷，这些缺陷若不加以解决，可能会对可用性和运营产生负面影响。

> **Note**
>
> 混沌工程是对系统进行试验以让人们确信系统能够在生产中经受住混乱情形的规范。– 混沌工程的原则

如果系统能够经受住这些干扰，那么应将混沌实验作为自动回归测试来加以维护。这样一来，应将混沌实验作为系统开发生命周期（SDLC）的一部分，以及作为 CI/CD 管道的一部分来执行。

为了确保您的工作负载能够承受组件故障，请在实验中注入实际事件。例如，对 Amazon EC2 实例的丢失或主 Amazon RDS 数据库实例的失效转移进行试验，并验证您的工作负载没有受到影响（或影响极小）。使用组件故障的组合来模拟可能因可用区中断而引起的事件。

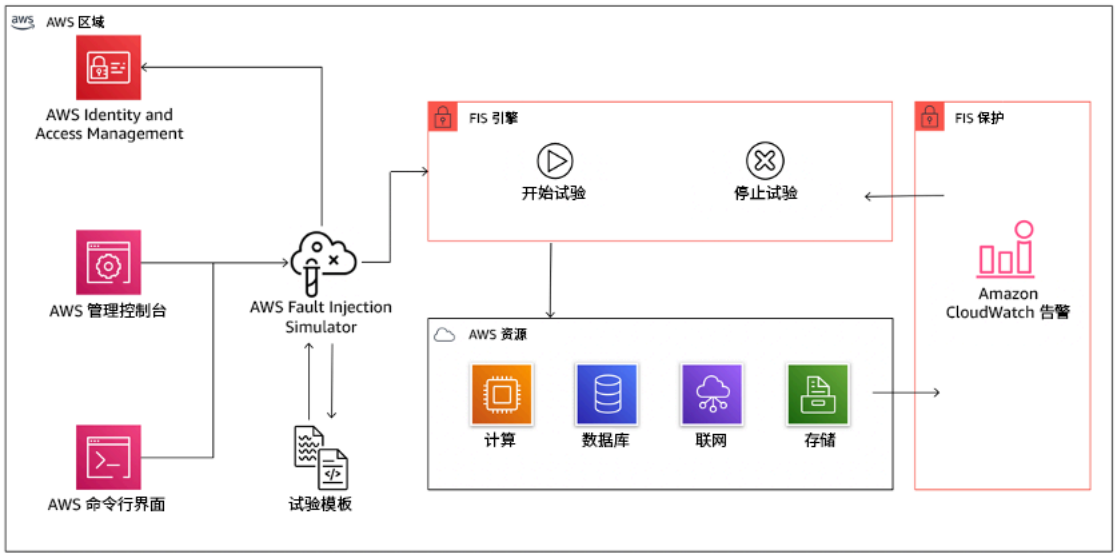
对于应用程序级故障（如崩溃），您可以从内存和 CPU 耗尽等压力源开始。

为了验证因间歇性网络中断而引发的外部依赖项的 回退或失效转移机制 ，您的组件应通过在指定时间段（从几秒到几小时不等）内阻止对第三方提供商的访问来模拟此类事件。

其他降级模式可能会影响功能的使用并降低响应速度，这通常会导致服务中断。性能下降的常见原因是，关键服务的延迟增加以及网络通信不可靠（丢包）。对于这些故障（包括延迟、丢弃的消息和 DNS 故障等网络效应）的实验可能包括无法解析名称、无法访问 DNS 服务或无法建立与依赖服务的连接。

混沌工程工具：

AWS Fault Injection Service（AWS FIS）是一项完全托管式服务，用于运行故障注入实验，而这些实验可用作 CD 管道的一部分，或在管道之外使用。AWS FIS 是在混沌工程实际试用期间使用的一个不错选择。它支持在不同类型的资源中同时引入故障，包括 Amazon EC2、Amazon Elastic Container Service (Amazon ECS)、Amazon Elastic Kubernetes Service（Amazon EKS）和 Amazon RDS 等资源。这些故障包括终止资源、强制失效转移、对 CPU 或内存施加压力、节流、延迟和数据包丢失。由于它与 Amazon CloudWatch 警报集成，因此您可以设置停止条件作为防护机制，以在实验导致意外影响时回滚。



AWS Fault Injection Service 与 AWS 资源集成，使您能够为您的工作负载运行故障注入实验。

故障注入实验也有多种第三方选项。其中包括开源工具，例如 Chaos ToolKit、Chaos Mesh和 Litmus Chaos以及商业选项，如 Gremlin。为了扩大可在 AWS 上注入的故障范围，AWS FIS 与 Chaos Mesh 和 Litmus Chaos 集成，使您能够在多个工具之间协调故障注入工作流。例如，您可以使用 Chaos Mesh 或 Litmus 故障对容器组（pod）的 CPU 运行压力测试，同时使用 AWS FIS 故障操作终止随机选择的集群节点百分比。

实施步骤

- 确定哪些故障要用于实验。

评估工作负载的设计是否具有弹性。这种设计（使用 Well-Architected Framework的最佳实践创建）考虑到了基于关键依赖关系、过去的事件、已知问题和合规性要求的风险。列出每个旨在保持弹性的设计元素及其旨在缓解的故障。有关创建此类列表的更多信息，请参阅 《运营准备就绪情况审核》白皮书 ，该白皮书指导您如何创建流程来防止以前的事件再次发生。故障模式与影响分析（FMEA）流程为您提供了一个框架，用于对故障及其对工作负载的影响执行组件级分析。Adrian Cockcroft 在 《Failure Modes and Continuous Resilience》中更详细地概述了 FMEA。

- 为每个故障指定一个优先级。

  先进行粗略的分类，如高、中或低。要评估优先级，请考虑故障的频率和故障对整体工作负载的影响。

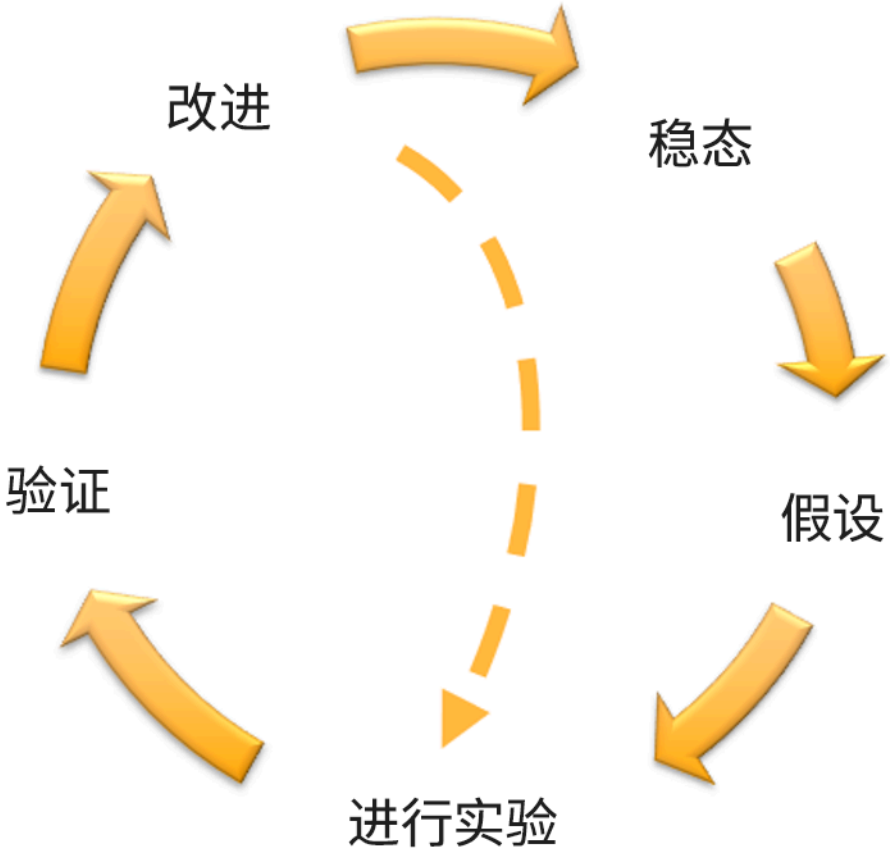
  考虑给定故障的频率时，请分析此工作负载的以往数据（如有）。如果没有以往数据，则使用在类似环境中运行的其他工作负载的数据。

  考虑给定故障的影响时，故障的范围越大，一般来说影响也越大。还要考虑工作负载设计和目的。例如，访问源数据存储的能力对于进行数据转换和分析的工作负载至关重要。在这种情况下，您将确定关于访问故障以及节流访问和延迟插入的实验的优先级。

  意外事件后分析是了解故障模式的频率和影响的良好数据来源。

  使用指定的优先级来确定首先用哪些故障进行实验，以及开发新的故障注入实验的顺序。

- 对于您执行的每项实验，请遵循混沌工程和连续弹性飞轮。



Adrian Hornsby 采用科学方法制作的混沌工程和连续弹性飞轮。

- 将稳态定义为指示正常行为的工作负载的一些可测量输出。

  如果工作负载运行可靠且符合预期，则显示为稳态。因此，定义稳态之前，请验证您的工作负载正常运行。稳态并不一定意味着故障发生时对工作负载没有影响，因为一定百分比的故障可能在可接受的范围内。稳态是您在实验期间将观察到的基线，如果您在下一步中定义的假设结果不符合预期，它将突出显示异常。

  例如，可以将某个支付系统的稳态定义为处理 300TPS，成功率为 99%，且往返时间为 500ms。

- 形成一个关于工作负载如何应对故障的假设。

  一个好的假设是基于工作负载预计如何缓解故障以保持稳态。该假设指出，如果发生特定类型的故障，系统或工作负载将继续保持稳态，因为该工作负载在设计时就有特定的缓解措施。应在假设中具体说明特定的故障类型和缓解措施。

  假设可以使用以下模板（但其他措辞也可以接受）：

> **Note**
>
> 如果发生 #### ，则 ###### 工作负载将 ######## 维持 #########。

例如：

- 如果 Amazon EKS 节点组中 20% 的节点出现故障，则 Transaction Create API 将在不到 100ms 的时间内继续处理 99% 的请求（稳态）。Amazon EKS 节点将在五分钟内恢复，容器组（pod）将在实验开始后八分钟内得到调度并处理流量。警报将在三分钟内将发出。
- 如果发生单个 Amazon EC2 实例故障，订单系统的 Elastic Load Balancing 运行状况检查将导致 Elastic Load Balancing 仅向剩余的运行状况良好的实例发送请求，而 Amazon EC2 Auto Scaling 将替换故障实例，从而保持服务器端（5xx）错误增长率低于 0.01%（稳态）。
- 如果主 Amazon RDS 数据库实例发生故障，则供应链数据收集工作负载将失效转移并连接到备用 Amazon RDS 数据库实例，以保持不到 1 分钟的数据库读写错误（稳态）。

- 通过注入故障来进行实验。

默认情况下，实验应具有故障保护机制，可承受工作负载。如果您知道工作负载将发生故障，则不要进行实验。混沌工程应该用于寻找已知的不确定因素或未知的不确定因素。已知的不确定因素是您知道但不完全理解的东西，而 未知的不确定因素 是您既不知道也不完全理解的东西。对您知道已经发生故障的工作负载进行试验不会为您提供新的见解。您应该对实验仔细规划，明确一个影响范围，并提供一种可在出现意外动荡时应用的回滚机制。如果尽职调查表明您的工作负载应该能经受住实验，那就继续这项实验。有几种注入故障的选项。对于 AWS 上的工作负载，AWS FIS 提供了许多称为 操作的预定义故障模拟。您还可以定义在 AWS FIS 中运行的自定义操作（使用 AWS Systems Manager 文档）。

我们不鼓励使用自定义脚本进行混沌实验，除非这些脚本能够了解工作负载的当前状态，能够发出日志，并在可能的情况下提供回滚和停止条件的机制。

支持混沌工程的有效框架或工具集应跟踪实验的当前状态，发出日志，并提供回滚机制以支持实验的受控执行。从 AWS FIS 这样的成熟服务开始，该服务支持您在明确定义的范围内和安全机制下进行实验，如果实验引入了意外的动荡，则可以回滚实验。要了解更多使用 AWS FIS 的实验，另请参阅 “通过混沌工程构建弹性且架构完善的应用程序”实验室。此外， AWS Resilience Hub 将分析您的工作负载，并创建您可以选择在 AWS FIS 中实施和运行的实验。

> **Note**
>
> 对于每项实验，要清楚地了解其范围及影响。我们建议首先在非生产环境中模拟故障，然后再在生产环境中运行。

应使用实际负载，通过 金丝雀部署 在生产环境中进行实验，尽可能同时启动控制和实验系统部署。在非高峰时间进行实验是一种很好的做法，可以减小首次在生产环境中试验时的潜在影响。此外，如果使用实际的客户流量会带来太大的风险，您可以在生产基础设施上针对控制和实验部署使用合成流量进行实验。当不能使用生产环境时，在尽可能接近生产环境的预生产环境中进行实验。

您必须建立和监控防护机制，确保实验对生产流量或其他系统的影响不会超过可接受的限度。建立停止条件，以便在实验达到您定义的防护机制指标的阈值时停止实验。这应该包括工作负载的稳态指标，以及针对您要注入故障的组件的指标。A 合成监控器（也称为用户金丝雀）是一个通常应作为用户代理包含的指标。AWS FIS 的停止条件 应纳入实验模板中，每个模板最多可以有五个停止条件。

混沌的原则之一是尽量缩小实验范围并减小其影响：

虽然必须考虑到一些短期负面影响，但混沌工程师有责任和义务确保实验产生的影响极小且可控。

验证范围和潜在影响的一种方法是首先在非生产环境中进行实验（验证停止条件的阈值在实验期间按预期激活，并且可观测性到位以捕获异常），而不是直接在生产环境中进行实验。

运行故障注入实验时，确保所有责任方均知情。与适当的团队（如运营团队、服务可靠性团队和客户支持团队）沟通，让他们知道实验将在何时运行以及预期会发生什么。为这些团队提供沟通工具，以便在他们看到任何不利影响时通知进行实验的人员。

必须将工作负载及其底层系统恢复到最初的已知良好状态。通常，工作负载的弹性设计会自我修复。但一些故障设计或失败的实验可能会使您的工作负载处于意外的失败状态。在实验结束时，您必须意识到这一点，并恢复工作负载和系统。使用 AWS FIS，您可以在操作参数中设置回滚配置（也称为后期操作）。后期操作将目标返回到运行该操作之前的状态。无论是自动执行（如使用 AWS FIS）还是手动执行，这些后期操作都应包含在描述如何检测和处理故障的行动手册中。

- 验证假设。

  混沌工程的原则 为如何验证工作负载的稳态提供了以下指导：

关注系统的可测量输出，而不是系统的内部属性。短时间内对该输出的测量构成了系统稳态的代理。整个系统的吞吐量、错误率和延迟百分比都可以是代表稳态行为的相关指标。通过关注实验过程中的系统行为模式，混沌工程验证系统确实在工作，而不是试图验证它如何工作。

在之前的两个示例中，我们包括了服务器端（5xx）错误增长率低于 0.01% 和数据库读写错误持续时间不到 1 分钟的稳态指标。

5xx 错误是一个很好的指标，因为它们是工作负载客户端将直接经历的故障模式的结果。数据库错误测量适合作为故障的直接结果，但是还应补充一个客户端影响测量，例如失败的客户请求或向客户端显示的错误。此外，在工作负载客户端直接访问的任何 API 或 URI 上包括一个合成监控器（也称为用户金丝雀）。

- 改进工作负载设计，以提高弹性。

  如果未保持稳态，则调查如何改进工作负载设计以缓解故障，应用 AWS Well-Architected 可靠性支柱的最佳实践。可以在 AWS Builder's Library中找到其他指导和资源，其中包含有关如何 改进运行状况检查 或 在应用程序代码中结合采用重试与回退的文章，等等。

  实施这些更改后，再次进行实验（如混沌工程飞轮中的虚线所示），以确定其有效性。如果验证步骤表明假设成立，那么工作负载将处于稳态，循环将继续。

- 定期进行实验。

  混沌实验是一个循环，作为混沌工程的一部分，应定期进行实验。在工作负载满足实验的假设后，实验应实现自动化，作为 CI/CD 管道的回归部分持续运行。要了解如何做到这一点，请参阅关于 如何使用 AWS CodePipeline 进行 AWS FIS 实验的博客。这个关于反复 在 CI/CD 管道中进行 AWS FIS 实验 的实验室使您能够动手实践。

  故障注入实验也是实际试用的一部分（请参阅 REL12-BP06 定期进行实际试用）。实际试用会模拟故障或事件，以便验证系统、流程和团队的响应。其目的是实际执行团队在发生意外事件时会执行的操作。

- 捕获和存储实验结果。

  必须捕获并持久保存故障注入实验的结果。包括所有必要的数据（如时间、工作负载和条件），以便以后能够分析实验结果和趋势。结果示例可能包括控制面板的屏幕截图、从指标数据库进行的 CSV 转储，或实验中事件和观察结果的手写记录。使用 AWS FIS 进行实验记录 可作为这种数据捕获的一部分。

资源

相关最佳实践：

- REL08-BP03 将弹性测试作为部署的一部分进行集成
- REL13-BP03 测试灾难恢复实施以验证实施效果

相关文档：

- 什么是 AWS Fault Injection Service？
- 什么是 AWS Resilience Hub？
- 混沌工程的原则
- 混沌工程：规划您的首次实验
- 弹性工程：学会接受故障
- 混沌工程案例
- 避免在分布式系统中回退
- 用于混沌实验的金丝雀部署

相关视频：

- AWS re:Invent 2020：使用混沌工程测试弹性（ARC316）
- AWS re:Invent 2019：通过混沌工程提高弹性（DOP309-R1）
- AWS re:Invent 2019：在无服务器世界中执行混沌工程（CMY301）

相关示例：

- Well-Architected 实验室：第 300 级：测试 Amazon EC2、Amazon RDS 和 Amazon S3 的弹性
- “混沌工程在 AWS 上的应用”实验室
- “通过混沌工程构建弹性且架构完善的应用程序”实验室
- “无服务器混沌”实验室
- “使用 AWS Resilience Hub 测量和提高应用程序弹性”实验室

相关工具：

- AWS Fault Injection Service
- AWS Marketplace： Gremlin 混沌工程平台
- Chaos ToolKit
- Chaos Mesh
- Litmus

REL12-BP06 定期进行实际试用

利用实际试用活动，在尽可能接近生产环境的环境中（包括在生产环境中），与将参与实际故障情景的人员一起为应对事件和故障而练习如何使用您的程序。实际试用会强制执行相关措施，以确保生产事件不会影响用户。

实际试用会模拟故障或事件，以便测试系统、流程和团队的响应。其目的是实际执行团队在发生意外事件时会执行的操作。这将帮助您了解可以从哪些方面作出改进，并有助于培养组织处理各种事件的经验。这些操作应该定期进行，让团队建立起关于响应方式的 肌肉记忆 。

在非生产环境中对您的弹性设计进行测试以后，可通过 Game Day 确保生产中的一切按照计划运行。Game Day，尤其如果是首次开展，是所有人员都应该参加的活动，工程师和运营团队都会得到关于开展时间以及活动内容的信息。运行手册准备就绪。以规定的方式在生产系统中执行模拟事件（包括可能出现的故障事件），并评估影响。如果所有系统如设计运行，检测和自我修复不会产生或只会产生非常轻微的影响。但如果观察到负面影响，测试将会回滚，并且（使用运行手册）修复问题，在必要时手动修复。由于实际试用经常在生产中进行，所以应采取全部预防措施，以确保不会对客户造成可用性影响。

常见反模式：

- 记录您的程序，但不要执行。
- 不要让业务决策者参与测试练习。

建立此最佳实践的好处： 定期执行实际试用可确保在发生实际事件时，所有员工都遵守策略和程序，并且能够验证这些策略和程序是否合适。

未建立这种最佳实践的情况下暴露的风险等级： 中

实施指导

- 安排实际试用，以定期运用运行手册和行动手册。生产事件中涉及的所有人员均需参与实际试用：业务负责人、开发人员、运营人员和事件响应团队。

- 运行负载或性能测试，然后运行故障注入。
- 寻找运行手册中的异常，并利用这些异常机会练习使用行动手册。
  - 如果您违反运行手册，请完善运行手册或纠正相应行为。如果练习使用行动手册，请确定应使用的运行手册，或者创建一个新运行手册。

资源

相关文档：

- 什么是 AWS 实际试用？

相关视频：

- AWS re:Invent 2019：通过混沌工程提高弹性（DOP309-R1）

相关示例：

- AWS Well-Architected 实验室 – 测试弹性

REL 13 如何规划灾难恢复 (DR)？

拥有适当的备份和冗余工作负载组件是您的 DR 策略的开始。RTO 和 RPO 是您恢复工作负载的目标。根据业务需求设置这些目标。通过实施策略来实现这些目标，同时考虑工作负载资源和数据的位置和功能。中断概率和恢复成本也是关键因素，有助于了解为工作负载提供灾难恢复的商业价值。

最佳实践

- REL13-BP01 定义停机和数据丢失的恢复目标
- REL13-BP02 使用定义的恢复策略来实现恢复目标
- REL13-BP03 测试灾难恢复实施以验证实施效果
- REL13-BP04 管理 DR 站点或区域的配置偏差
- REL13-BP05 自动执行恢复

REL13-BP01 定义停机和数据丢失的恢复目标

工作负载具有恢复时间目标（RTO）和恢复点目标（RPO）。

恢复时间目标 (RTO) 是指服务中断和服务恢复之间的最大可接受延迟。这可以确定在服务不可用时被视为可接受的时间窗口。

恢复点目标 (RPO) 是指自上一个数据恢复点以来的最大可接受时间。这可以确定在上一个恢复点和服务中断之间可接受的数据丢失程度。

在为您的工作负载选择合适的灾难恢复（DR，Disaster Recovery）策略时，RTO 和 RPO 值是重要的考虑因素。这些目标由业务部门确定，然后由技术团队用来选择和实施 DR 策略。

期望结果：

每个工作负载都有一个根据业务影响定义的指定 RTO 和 RPO。工作负载被分配到一个预定义的层，该层定义服务可用性和可接受的数据丢失，以及关联的 RTO 和 RPO。如果无法进行这样的分层，那么可以为每个工作负载分配定制的分层，用于以后创建层。在为工作负载选择灾难恢复策略实施时，使用 RTO 和 RPO 作为主要考虑因素之一。在选择 DR 策略时还要考虑成本约束、工作负载依赖关系和运维需求。

对于 RTO，了解基于中断持续时间的影响。是线性的还是非线性的影响？（例如，四小时后，您关闭一条生产线，直到下一班开始）。

如下所示的灾难恢复矩阵可以帮助您了解工作负载的重要性与恢复目标之间的关系。（请注意，X 轴和 Y 轴的实际值应根据您组织的需求进行定制）。

灾难恢复矩阵						
		恢复点目标				
		少于 1 分钟	少于 1 小时	少于 6 小时	少于 1 天	多于 1 天
恢复时间目标	少于 10 分钟	严重	严重	高	中	中
	少于 2 小时	严重	高	中	中	低
	少于 8 小时	高	中	中	低	低
	少于 24 小时	中	中	低	低	低
	多于 24 小时	中	低	低	低	低

图 16：灾难恢复矩阵

常见反模式：

- 未定义恢复目标。

- 选择任意恢复目标。
- 选择过于宽松并且不符合业务目标的恢复目标。
- 不了解停机和数据丢失的影响。
- 选择不切实际的恢复目标，如零恢复时间和零数据丢失，这对于您的工作负载配置可能无法实现。
- 选择比实际业务目标更严格的恢复目标。这将强制实施比工作负载所需的成本更高并且更复杂的 DR。
- 选择与所依赖工作负载的恢复目标不兼容的恢复目标。
- 您的恢复目标没有考虑法规合规性要求。
- 为工作负载定义了 RTO 和 RPO，但从未测试过。

建立此最佳实践的好处： 在指导您的 DR 实施时，需要您的恢复时间目标和数据丢失恢复目标。

未建立此最佳实践暴露的风险等级： 高

实施指导

对于给定的工作负载，您必须了解停机和数据丢失对业务的影响。随着停机时间或数据丢失的增加，影响通常会越来越大，但这种增长的形式可能会因工作负载类型而异。例如，您也许可以容忍长达一小时的停机时间而没有多大影响，但在一小时之后影响会迅速上升。对业务的影响表现为多种形式，包括货币成本（如收入损失）、客户信任（以及对声誉的影响）、运维问题（如错过工资发放或生产力下降）和监管风险。使用以下步骤了解这些影响，并为您的工作负载设置 RTO 和 RPO。

实施步骤

1. 确定此工作负载的业务利益相关者，并与他们一起实施这些步骤。工作负载的恢复目标是一项业务决策。然后，技术团队与业务利益相关者合作，使用这些目标来选择 DR 策略。

   > **Note**
   >
   > 对于步骤 2 和 3，您可以使用 the section called “实施工作表”.

2. 通过回答以下问题，收集必要的信息来做出决策。
3. 在组织中，您是否对工作负载影响的重要性进行了分类或分级？
   a. 如果有，请将此工作负载分配到一个类别
   b. 如果没有，则建立这些类别。创建不超过五个类别，并细化每个类别的恢复时间目标范围。类别示例包括：关键、高、中、低。要了解工作负载如何映射到类别，请考虑工作负载是任务关键型、业务重要型还是非业务驱动型。

   c. 根据类别设置工作负载 RTO 和 RPO。始终选择比进入此步骤时计算的原始值更严格的类别（更低的 RTO 和 RPO）。如果这导致值发生了不适当的较大改变，那么考虑创建一个新类别。

4. 根据这些答案，为工作负载分配 RTO 和 RPO 值。这可以直接完成，也可以通过将工作负载分配给预定义的服务层来完成。
5. 在工作负载团队和利益相关者可访问的位置，记录此工作负载的灾难恢复计划（DRP，disaster recovery plan），此计划是组织的 业务连续性计划（BCP，Business Continuity Plan）的一部分
   a. 记录 RTO 和 RPO，以及用于确定这些值的信息。包括用于评估工作负载对业务影响的策略
   b. 除 RTO 和 RPO 之外，记录您根据灾难恢复目标正在跟踪或计划跟踪的其他指标
   c. 在进行创建时，您将 DR 策略和运行手册的详细信息添加到此计划中。
6. 通过在如图 15 所示的矩阵中查找工作负载的重要性，您可以开始建立为组织定义的预定义服务层。
7. 根据 实施 DR 策略（或 DR 策略的概念验证）之后，the section called “REL13-BP02 使用定义的恢复策略来实现恢复目标”测试此策略以确定工作负载的实际 RTC（Recovery Time Capability，恢复时间能力）和 RPC（Recovery Point Capability，恢复点能力）。如果这些能力没有达到所预期的恢复目标，那么，要么与您的业务利益相关者一起调整这些目标，要么对 DR 策略进行更改以便实现预期的目标。

主要问题

1. 在对业务产生严重影响之前，工作负载可以停止的最长时间是多少
   a. 确定在工作负载中断时，每分钟业务的货币成本（直接财务影响）。
   b. 请注意，影响并不总是线性的。影响可能在一开始是有限的，然后在超过一个关键时间点后迅速增加。
2. 在对业务造成严重影响之前，可以丢失的最大数据量是多少
   a. 对于最关键的数据存储，请考虑此值。确定其他数据存储的各自关键性。
   b. 如果工作负载数据丢失，是否可以重新创建？ 如果这在操作上比备份和还原更容易，那么根据用于重新创建工作负载数据的源数据的重要性来选择 RPO。
3. 此工作负载所依赖的工作负载（下游）或依赖于此工作负载的工作负载（上游）的恢复目标和可用性期望是什么？
   a. 选择使此工作负载能够满足上游依赖项要求的恢复目标
   b. 根据下游依赖项的恢复能力，选择可实现的恢复目标。非关键的下游依赖项（您可以“绕过”它们）可以排除。或者，处理关键的下游依赖项，在必要时提高其恢复能力。

其他问题

考虑以下问题，以及它们如何应用于此工作负载：

4. 根据中断类型（区域与可用区等），您是否有不同的 RTO 和 RPO？
5. 您的 RTO/RPO 是否会在特定时间（季节性、销售活动、产品发布）发生变化？ 如果是这样，不同的测量和时间边界是什么？
6. 如果工作负载中断，会有多少客户受到影响？
7. 如果工作负载中断，对声誉有何影响？
8. 如果工作负载中断，可能会产生哪些其他运营影响？ 例如，如果电子邮件系统不可用或工资单系统无法提交事务，则会影响员工的工作效率。
9. 工作负载 RTO 和 RPO 如何与业务线和组织 DR 策略保持一致？
10. 是否存在提供服务的内部合同义务？ 不履行这些义务会受到处罚吗？
11. 数据的监管或合规性约束是什么？

实施工作表

您可以将此工作表用于实施步骤 2 和 3。您可以调整此工作表以满足您的特定需求，例如添加其他问题。

步骤 2：主要问题	适用于工作负载？	工作负载 RTO	工作负载 RPO	RTO 调整。	RPO 调整。	说明
[1] 工作负载可以停止的最长时间						以从中断开始到恢复的时间进行衡量
[2] 可以丢失的最大数据量						以从最后一个已知的可恢复数据集算起的时间进行衡量
[3a] 上游依赖关系						输入最严格的上游恢复目标
[3b] 下游依赖关系						输入最不严格的下游恢复目标
[3a] 协调后的上游依赖关系						如果上游值小于当前值，而下游值大于当前值，则使用依赖关系进行协调，
[3b] 协调后的下游依赖关系						并在此处输入协调后的值
[3] 依赖关系						降低值以满足上游依赖关系，或者根据下游依赖关系的能力提高值
步骤 2：其他问题						指出问题是否适用。如果问题不适用，则跳过
基本 RTO/RPO						将 RTO 和 RPO 值从上方向下移到此处
[4] 中断类型	[ ]Y / [ ]N					输入要求最严格的事件类型的恢复目标
[5] 基于时间的具体目标	[ ]Y / [ ]N					输入要求最严格的恢复时间目标
[6] 客户中断	[ ]Y / [ ]N					根据停机时间或数据丢失情况绘制受影响客户的图表。 根据客户影响输入允许的最大的 RTO 和 RPO
[7] 声誉影响	[ ]Y / [ ]N					与业务部门合作，根据对声誉的影响确定最大的 RTO 和 RPO
[8] 运营影响	[ ]Y / [ ]N					根据运营影响输入最大的 RTO 和 RPO
[9] 组织一致性	[ ]Y / [ ]N					根据 LOB 和组织要求，输入此类工作负载的最大 RTO 和 RPO
[10] 合同义务	[ ]Y / [ ]N					根据合同义务输入最大的 RTO 和 RPO
[11] 监管合规性	[ ]Y / [ ]N					根据适用的监管合规性输入最大的 RTO 和 RPO
基于其他问题的目标						从问题 4-11 中取最小值（更严格的值）并在此处输入
调整后的目标						如果无法满足上述目标，请与利益相关者一起放松约束， 并在此处输入新的最小值
调整后的 RTO/RPO						输入基本 RPO/RTO 值或调整后的目标值，以较低者为准
步骤 3						
映射到预定义的类别或层						将这两个值向下调整（更严格），以符合最接近的定义层

工作表

实施计划的工作量级别： 低

资源

相关最佳实践：

- the section called “REL09-BP04 定期执行数据恢复以验证备份完整性和流程”
- the section called “REL13-BP02 使用定义的恢复策略来实现恢复目标”
- the section called “REL13-BP03 测试灾难恢复实施以验证实施效果”

相关文档：

- AWS 架构博客：灾难恢复系列
- AWS 上工作负载的灾难恢复：云中的恢复（AWS 白皮书）
- 使用 AWS Resilience Hub 管理弹性策略
- AWS 合作伙伴：可以帮助进行灾难恢复的合作伙伴
- AWS Marketplace：可以用于灾难恢复的产品

相关视频：

- AWS re:Invent 2018：适用于多区域主动-主动应用程序的架构模式（ARC209-R2）
- AWS 上工作负载的灾难恢复

REL13-BP02 使用定义的恢复策略来实现恢复目标

定义满足工作负载恢复目标的灾难恢复（DR, disaster recovery）策略。选择一种策略，例如：备份和还原；备用（主动/被动）；或主动/主动。

DR 策略依赖于在主位置无法运行工作负载的情况下，在恢复站点中支持工作负载的能力。最常见的恢复目标是 RTO 和 RPO，相关讨论内容位于 REL13-BP01 定义停机和数据丢失的恢复目标.

跨单个 AWS 区域 内的多个可用区（AZ）的 DR 策略可以缓解火灾、洪水和重大停电等灾难事件。如果需要实施保护措施，为工作负载无法在给定 AWS 区域 中运行这种不太可能发生的事件提供保护，您可以使用跨多个区域的 DR 策略。

在跨多个区域构建 DR 策略时，您应该选择以下策略之一。这些策略按成本和复杂性升序排列，按 RTO 和 RPO 降序排列。 恢复区域 指的是 AWS 区域，而不是用于工作负载的主要区域。

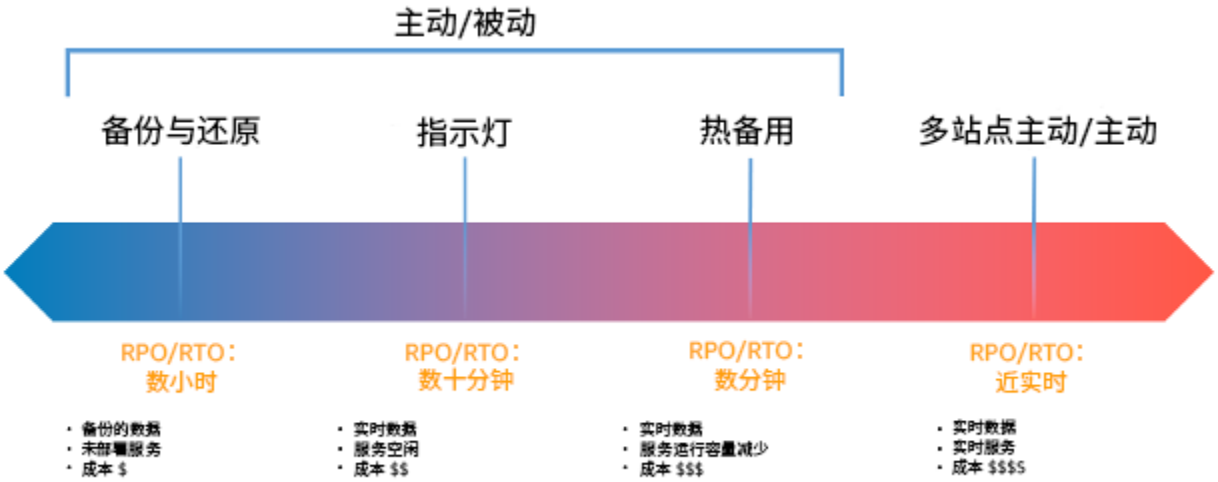


图 17：灾难恢复（DR）策略

- 备份和还原 （RPO 以小时为单位，RTO 为 24 小时或以内）：将您的数据和应用程序备份到恢复区域。使用自动或连续备份可以实现时间点故障恢复，在某些情况下，可以将 RPO 降低到 5 分钟。在发生灾难的情况下，您将部署基础设施（使用基础设施即代码来减少 RTO）、部署代码并还原备份的数据，以便在恢复区域从灾难中恢复。
- 指示灯 （RPO 以分钟为单位，RTO 为数十分钟）：在恢复区域中预置核心工作负载基础设施的副本。将您的数据复制到恢复区域并在那里创建数据备份。支持数据复制和备份所需的资源（如数据库和对象存储）始终处于启用状态。其他元素（如应用程序服务器或无服务器计算）未部署，但可以在需要时使用必要的配置和应用程序代码创建。
- 热备用 （RPO 以秒为单位，RTO 以分钟为单位）：保证在恢复区域中始终运行缩减但功能齐全版本的工作负载。业务关键型系统是完全重复，而且始终可用的系统，只是其队列的规模经过缩减。数据在恢复区域中复制并留存。在需要恢复时，系统会快速扩展以处理生产负载。热备用系统的规模越大，RTO 和控制面板依赖度就越低。当完全扩展时，这称为 热备用服务器。
- 多区域（多站点）主动-主动 （RPO 接近于零，RTO 可能为零）：您的工作负载被部署到多个 AWS 区域，并且主动处理来自这些区域的流量。此策略要求您跨区域同步数据。必须避免或处理在两个不同区域副本中写入同一记录可能引起的冲突，这会很复杂。数据复制对于数据同步非常有用，并且可以防止某些类型的灾难，但是它不能防止数据损坏或破坏，除非您的解决方案还包含时间点故障恢复选项。

> **Note**
>
> 指示灯和热备用之间的差异有时难以区分。两者都在恢复区域中包含一个环境，其中具有主区域资产的副本。区别在于，如果不先采取额外措施，指示灯无法处理请求，而热备用可以立即处理流量（容量级别降低）。指示灯将要求您启用服务器，可能需要部署额外的（非核心）基础设施并纵向扩展，而热备用只需要您纵向扩展（所有内容都已部署并运行）。根据您的 RTO 和 RPO 需求在两者之间进行选择。

期望结果：

对于每个工作负载，都有一个已定义和实施的 DR 策略，使该工作负载能够实现 DR 目标。工作负载之间的 DR 策略利用可重用模式（如前面描述的策略）。

常见反模式：

- 为具有类似 DR 目标的工作负载实施不一致的恢复过程。
- 在发生灾难时临时实施 DR 策略。
- 没有 DR 计划。
- 恢复期间依赖于控制面板操作。

建立此最佳实践的好处：

- 通过定义恢复策略，您可以使用常用工具和测试步骤。
- 通过使用定义的恢复策略，可以在团队之间更高效地共享知识，并更容易地在他们自己的工作负载上实施 DR。

未建立此最佳实践暴露的风险等级： 高

- 若没有经过计划、实施和测试的 DR 策略，在发生灾难时不太可能实现恢复目标。

实施指导

对于每个步骤，请参见下面的详细信息。

1. 确定将满足此工作负载恢复要求的 DR 策略。
2. 查看如何实施所选 DR 策略的模式。

3. 评估工作负载的资源，以及失效转移之前（正常操作期间）恢复区域中的资源配置。
4. 确定并实施措施，让恢复区域在需要时（在灾难事件期间）可以进行失效转移。
5. 确定并实施措施，以在需要时（在灾难事件期间）可以重新路由流量进行失效转移。
6. 设计工作负载的故障恢复计划。

实施步骤

1. 确定将满足此工作负载恢复要求的 DR 策略。

选择 DR 策略是在减少停机时间和数据丢失（RTO 和 RPO）与策略实施的成本和复杂性之间进行权衡。您应该避免实施比所需策略更严格的策略，因为这会产生不必要的成本。

例如，在下图中，企业已经确定了他们允许的最大 RTO 以及他们可以在服务恢复策略上花费的费用限额。鉴于企业目标，指示灯或热备用这样的 DR 策略将同时满足 RTO 和成本标准。

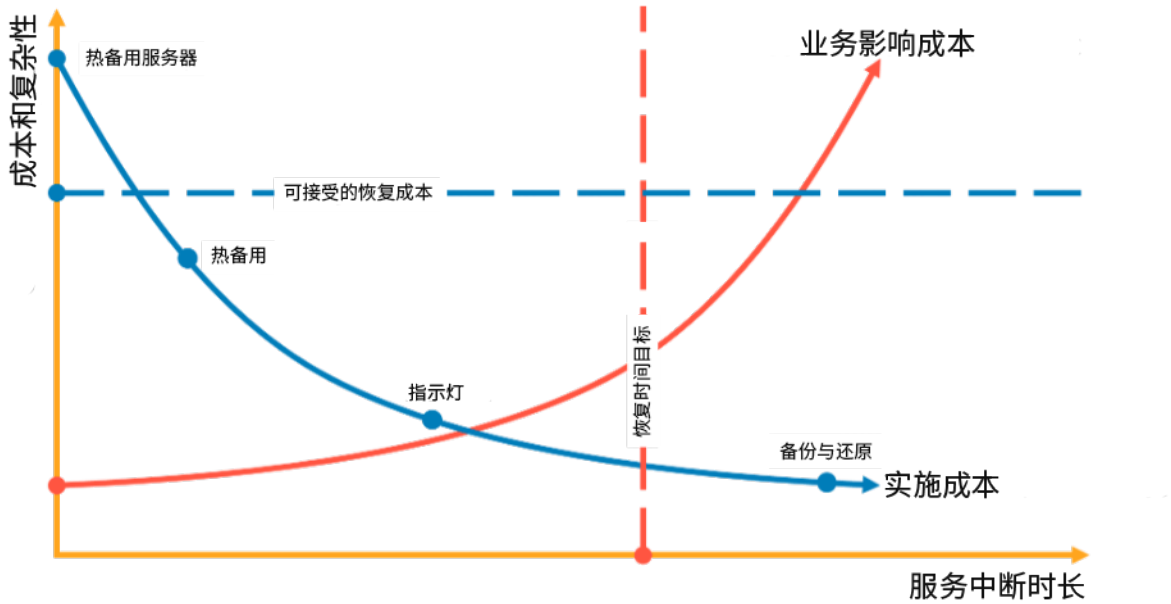


图 18：根据 RTO 和成本选择 DR 策略

如需了解更多信息，请参阅 业务连续性计划（BCP，Business Continuity Plan）。

2. 查看如何实施所选 DR 策略的模式。

这一步是了解如何实施所选策略。这些策略可以解释为使用多个 AWS 区域 作为主要站点和恢复站点。不过，您也可以选择使用单个区域内的多个可用区作为 DR 策略，这将利用多个策略的元素。

在这一步之后的后续步骤中，您将对特定的工作负载应用策略。

备份和还原

备份和还原 是实施起来最简单的策略，但需要更多时间和工作来恢复工作负载，从而导致更高的 RTO 和 RPO。最好的做法是，始终备份数据并将数据备份复制到另一个站点（如另一个 AWS 区域）。

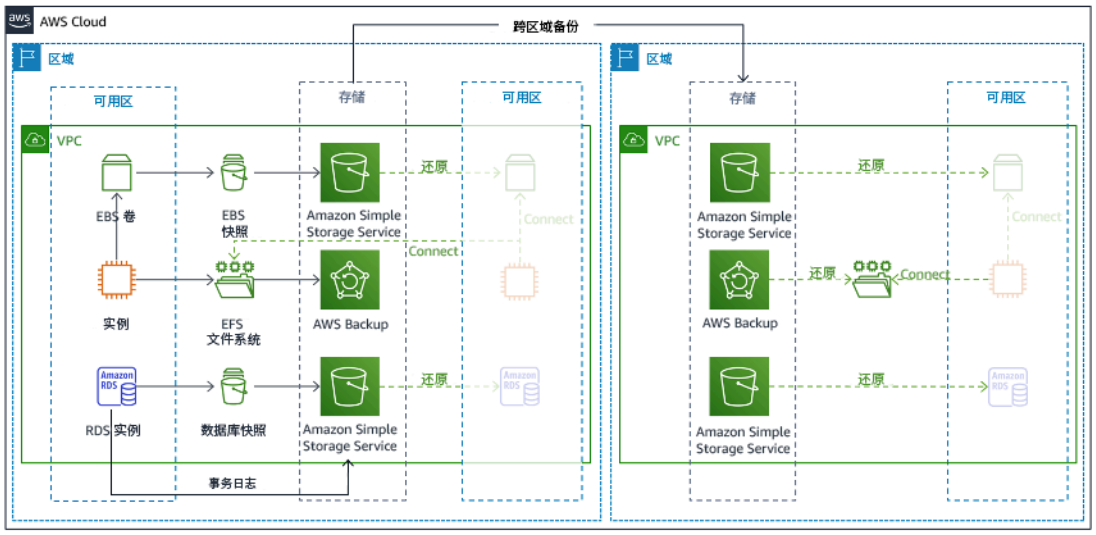


图 19：备份和还原架构

有关此策略的更多详细信息，请参阅 AWS 上的灾难恢复（DR）架构，第 II 部分：使用快速恢复功能的备份与还原。

指示灯

利用 指示灯 方法，您可以将数据从主要区域复制到恢复区域。用于工作负载基础设施的核心资源部署在恢复区域中，但仍需要额外的资源和所有依赖项才能使此恢复区域成为功能堆栈。例如，在图 20 中，没有部署计算实例。

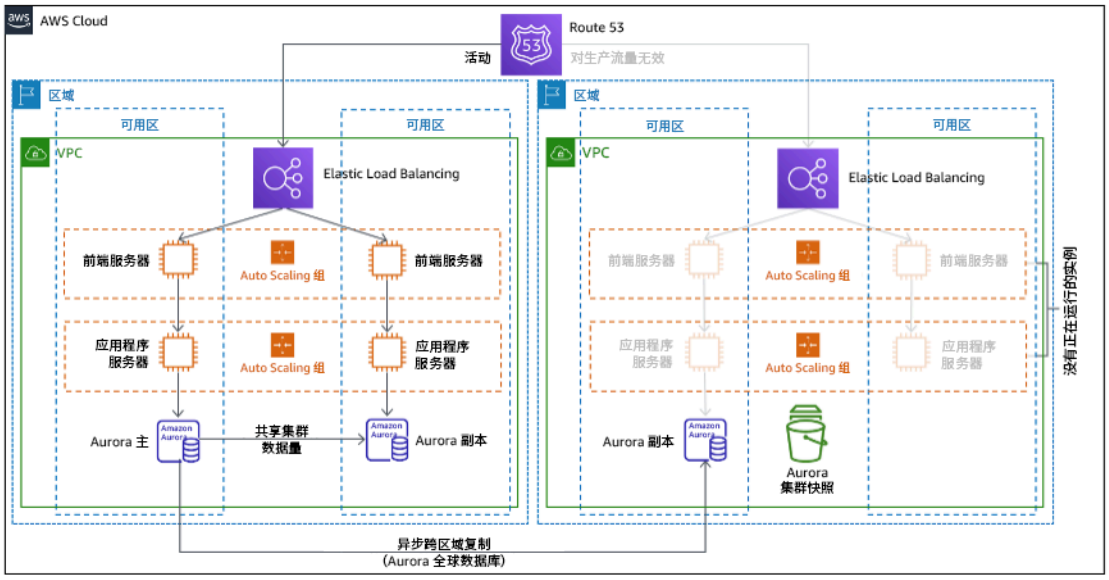


图 20：指示灯架构

有关此策略的更多详细信息，请参阅 AWS 上的灾难恢复（DR）架构，第 III 部分：指示灯和热备用。

热备用

热 备用 方法涉及到确保在另一个区域中存在生产环境的规模缩减但功能齐全的副本。这种方法扩展了指示灯概念并减少了恢复时间，因为您的工作负载始终在另一个区域中运行。如果恢复区域以满容量部署，那么这种方式称为 热备用服务器。

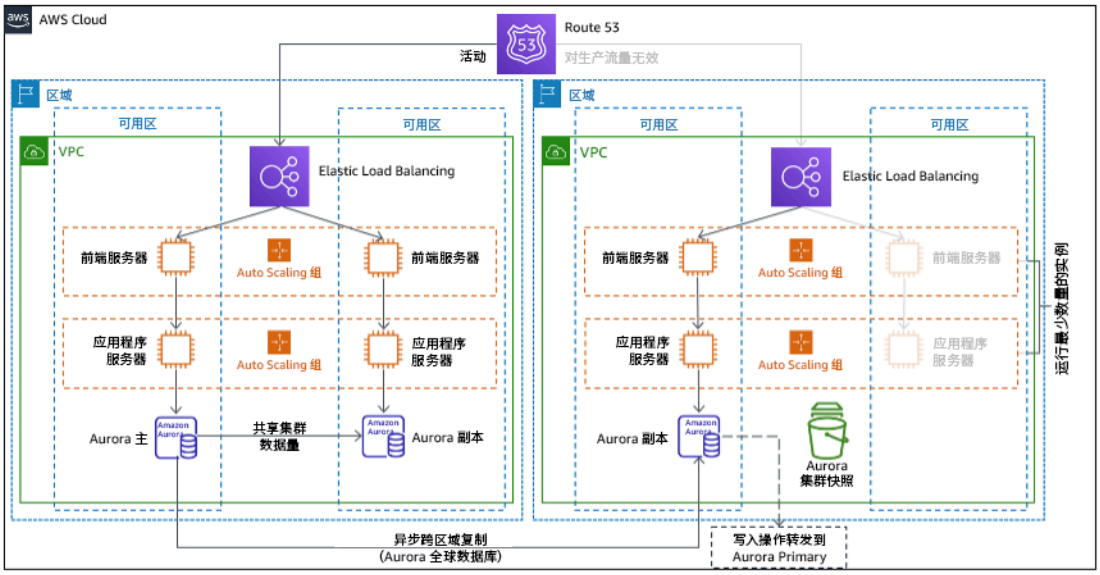


图 21：热备用架构

使用热备用或指示灯需要扩展恢复区域中的资源。为确保在需要时有可用的容量，请考虑使用 EC2 实例的 容量预留 。如果使用 AWS Lambda，那么 预置并发 可以确保执行环境，以便它们准备好立即响应函数的调用。

有关此策略的更多详细信息，请参阅 AWS 上的灾难恢复（DR）架构，第 III 部分：指示灯和热备用。

多站点主动/主动

作为 多站点主动/主动 策略的一部分，您可以在多个区域中同时运行工作负载。多站点主动/主动策略处理来自其部署到的所有区域的流量。客户可能会出于 DR 以外的原因选择此策略。此策略可以用于提高可用性，或者在向全球受众部署工作负载时（使端点更靠近用户和/或部署针对该区域受众的本地化堆栈）使用此策略。作为一种 DR 策略，如果工作负载在部署此策略的某个 AWS 区域 中不能得到支持，那么该区域将被撤出，使用其余区域维护可用性。多站点主动/主动策略是 DR 策略中操作最复杂的策略，只有在业务需求时才应选择它。

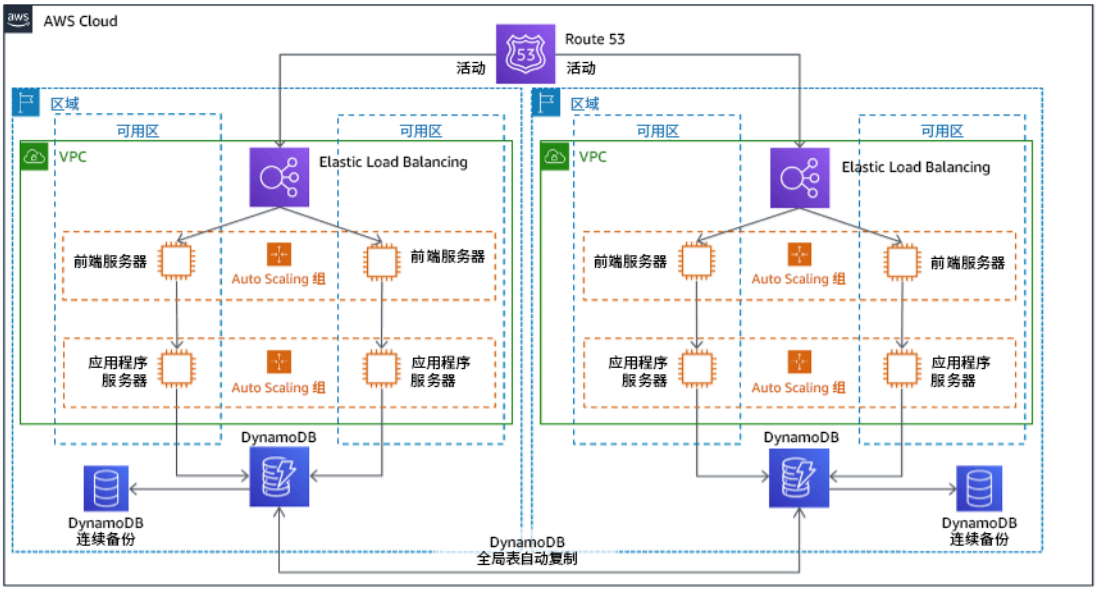


图 22：多站点主动/主动架构

有关此策略的更多详细信息，请参阅 AWS 上的灾难恢复（DR，Disaster Recovery）架构，第 IV 部分：多站点主动/主动。

其他保护数据的实践

对于所有这些策略，您还必须减轻数据灾难的影响。持续的数据复制可以防止某些类型的灾难，但它可能无法防止数据损坏或破坏，除非您的策略还包括存储数据的版本控制或用于时间点故障恢复的选项。除了副本之外，您还必须备份恢复站点中的复制数据以创建时间点备份。

使用单个 AWS 区域 内的多个可用区（AZ）

使用单个区域内的多个 AZ 时，您的 DR 实施会使用上述策略的多个元素。首先，您必须使用多个 AZ 创建一个高可用性（HA，High Availability）架构，如图 23 所示。此架构使用多站点主动/主动方法，因为 Amazon EC2 实例 和 Elastic Load Balancer 在多个 AZ 中部署了资源，主动处理请求。此架构还演示了热备用服务器方法，如果主 Amazon RDS 实例出现故障（或 AZ 本身出现故障），则备用实例将提升为主实例。

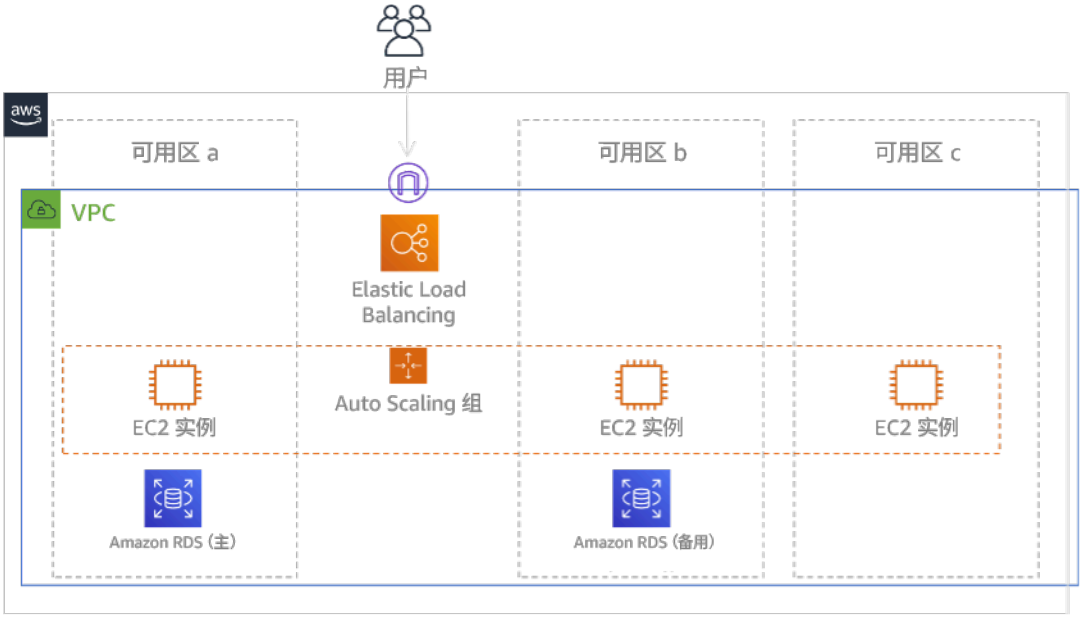


图 23：多可用区架构

除了这种 HA 架构之外，您还需要添加运行工作负载所需的所有数据的备份。这对于限制在单个区的数据尤其重要，例如 Amazon EBS 卷 或者 Amazon Redshift 集群。如果一个 AZ 发生故障，您需要将这些数据恢复到另一个 AZ。如果可能，您还应该将数据备份复制到另一个 AWS 区域，作为额外的保护层。

下面的博客文章中介绍了一种不太常见的单区域多可用区 DR 的替代方法： 使用 Amazon Route 53 Application Recovery Controller 构建高弹性应用程序，第 1 部分：单区域堆栈。这里的策略是尽可能保持 AZ 之间的隔离，就像区域的运作方式一样。使用这种替代策略，您可以选择主动/主动或主动/被动方法。

注意：某些工作负载具有数据驻留法规要求。如果这适用于当前只有一个 AWS 区域的位置的工作负载，那么多区域将不适合您的业务需求。多可用区策略可以很好地抵御大多数灾难。

3. 评估工作负载的资源，以及失效转移之前（正常操作期间）恢复区域中的资源配置。

对于基础设施和 AWS 资源，使用基础设施即代码功能（如 AWS CloudFormation ）或第三方工具（如 Hashicorp Terraform）。要使用单个操作跨多个账户和区域部署，您可以使用 AWS

CloudFormation StackSets。对于多站点主动/主动和热备用服务器策略，恢复区域中部署的基础设施具有与主区域相同的资源。对于指示灯和热备用策略，部署的基础设施将需要额外的操作才可用于生产。使用 CloudFormation 参数 和 条件逻辑，您可以通过单个模板控制部署的堆栈是活动的还是备用的。此 CloudFormation 模板示例见 这篇博客文章。

所有 DR 策略都要求在 AWS 区域 内备份数据源，然后将这些备份复制到恢复区域。AWS Backup 提供了一个集中视图，您可以在其中配置、调度和监控这些资源的备份。对于指示灯、热备用和多站点主动/主动方法，您还应该将数据从主区域复制到恢复区域中的数据资源，例如 Amazon Relational Database Service（Amazon RDS）数据库实例或 Amazon DynamoDB 表。因此，这些数据资源处于活动状态，可以随时处理恢复区域中的请求。

要了解更多关于 AWS 服务如何跨区域运行的信息，请参阅以下博客系列： 使用 AWS 服务创建多区域应用程序。

4. 确定并实施措施，让恢复区域在需要时（在灾难事件期间）可以进行失效转移。

对于多站点主动/主动策略，失效转移意味着撤离一个区域，并依赖剩余的活动区域。通常，这些区域已准备好接受流量。对于指示灯和热备用策略，恢复操作将需要部署缺失的资源（如图 20 中的 EC2 实例），以及任何其他缺失的资源。

对于上述所有策略，您可能需要将数据库的只读实例提升为主读/写实例。

对于备份和还原，从备份中还原数据时会为该数据创建资源，例如 EBS 卷、RDS 数据库实例和 DynamoDB 表。您还需要还原基础设施并部署代码。您可以使用 AWS Backup 来还原恢复区域中的数据。请参阅 REL09-BP01 识别和备份需要备份的所有数据，或从源复制数据 了解更多详细信息。重建基础设施包括创建资源，例如，EC2 实例以及所需的 Amazon Virtual Private Cloud（Amazon VPC）、子网和安全组。您可以自动执行大部分还原过程。要了解具体方法，请参阅 这篇博客文章。

5. 确定并实施措施，以在需要时（在灾难事件期间）可以重新路由流量进行失效转移。

此失效转移操作可以自动或手动启动。应谨慎使用基于运行状况检查或警报自动启动的失效转移，因为不必要的失效转移（误报）会产生不可用和数据丢失等成本。因此，通常会手动启动的失效转移。在这种情况下，您仍然应该自动执行失效转移步骤，这样手动启动就像按一下按钮一样简单。

在使用 AWS 服务时，需要考虑几个流量管理选项。一种选项是使用 Amazon Route 53。使用 Amazon Route 53，您可以将一个或多个 AWS 区域 中的多个 IP 端点与一个 Route 53 域名相关联。要实施手动启动的失效转移，您可以使用 Amazon Route 53 Application Recovery Controller，它提供高度可用的数据面板 API 以将流量重新路由到恢复区域。实施失效转移时，使用数据面板操作并避免控制面板操作，如 REL11-BP04 恢复期间依赖于数据面板而不是控制面板.

要了解有关此选项和其他选项的更多信息，请参阅 灾难恢复白皮书的这一部分。

6. 设计工作负载的故障恢复计划。

故障恢复是指在灾难事件消除后将工作负载操作返回主区域。向主区域预置基础设施和代码通常遵循最初使用的相同步骤，依赖于基础设施即代码和代码部署管道。故障恢复的挑战是还原数据存储，并确保它们与运行中的恢复区域保持一致。

在失效转移状态下，恢复区域中的数据库处于活动状态，并且具有最新数据。然后，目标是从恢复区域重新同步到主区域，确保主区域是最新的。

某些 AWS 服务会自动执行此操作。如果使用 Amazon DynamoDB 全局表，即使主区域中的表不可用，当它重新联机时，DynamoDB 也会继续传播任何挂起的写操作。如果使用 Amazon Aurora 全局数据库 并使用 托管的计划失效转移，则维护 Aurora 全局数据库的现有复制拓扑。因此，主区域中以前的读/写实例将成为副本，并从恢复区域接收更新。

如果这不是自动执行的，您将需要在主区域中重新建立数据库，作为恢复区域中数据库的副本。在许多情况下，这将涉及删除旧的主数据库，然后创建新的副本。例如，有关如何使用 Amazon Aurora 全局数据库对 计划外 失效转移执行此操作的说明，请参阅下面的实验： 全局数据库的故障恢复。

失效转移后，如果您可以继续在恢复区域中运行，请考虑将此区域设为新的主区域。您仍然需要执行上述所有步骤，将以前的主区域变成恢复区域。有些组织会进行定期轮换，定期交换其主区域和恢复区域（例如每三个月一次）。

失效转移和故障恢复所需的所有步骤都应保存在行动手册且可供所有团队成员使用，并定期进行审查。

实施计划的工作量级别：高

资源

相关最佳实践：

- the section called “REL09-BP01 识别和备份需要备份的所有数据，或从源复制数据”
- the section called “REL11-BP04 恢复期间依赖于数据面板而不是控制面板”
- the section called “REL13-BP01 定义停机和数据丢失的恢复目标”

相关文档：

- AWS 架构博客：灾难恢复系列

- AWS 上工作负载的灾难恢复：云中的恢复（AWS 白皮书）
- 云中的灾难恢复选项
- 在一小时内构建无服务器多区域、主动-主动后端解决方案
- 多区域无服务器后端 – 重新加载
- RDS：跨区域复制只读副本
- Route 53：配置 DNS 故障转移
- S3：跨区域复制
- 什么是 AWS Backup？
- 什么是 Route 53 Application Recovery Controller？
- AWS 弹性灾难恢复
- HashiCorp Terraform：入门 – AWS
- AWS 合作伙伴：可以帮助进行灾难恢复的合作伙伴
- AWS Marketplace：可以用于灾难恢复的产品

相关视频：

- AWS 上工作负载的灾难恢复
- AWS re:Invent 2018：适用于多区域主动-主动应用程序的架构模式（ARC209-R2）
- 开始使用 AWS 弹性灾难恢复 | Amazon Web Services

相关示例：

- AWS Well-Architected 实验 – 灾难恢复 – 说明 DR 策略的系列研讨会

REL13-BP03 测试灾难恢复实施以验证实施效果

定期测试到恢复站点的失效转移，以确保正常运行，并满足 RTO 和 RPO。

要避免的模式是制定了恢复路径但很少测试。例如，您可能有一个用于只读查询的辅助数据存储。当您写入某个数据存储，却发现主存储故障时，您可能希望将故障转移到辅助数据存储。如果您不经常测试此故障转移，可能会发现您关于辅助数据存储容量的假设是错误的。辅助数据存储容量在您上次测试时可能是足够的，但可能无法再容纳这次情况下的负载。我们的经验表明，唯一有效的错误恢复是您经常测试的路径。因此，最好只开发几条恢复路径。您可以建立恢复模式并定期对其进行测试。如果恢复路

径比较复杂或至关重要，您仍需定期在生产环境中测试该故障，确保恢复路径有效。在我们刚才讨论的示例中，您应该定期将故障转移到备用存储，无论是否有需要。

常见反模式：

- 从不在生产环境中测试失效转移。

建立此最佳实践的好处： 定期测试您的灾难恢复计划，确保该计划在需要时能够正常发挥作用，并且您的团队知道如何执行该策略。

未建立此最佳实践暴露的风险等级： 高

实施指导

- 为灾难恢复设计工作负载。定期测试恢复路径：面向恢复的计算可识别系统中能够增强恢复功能的特性。这些特性包括：隔离和冗余，系统范围回滚更改的能力，监控并确定运行状况的能力，提供诊断、自动恢复、模块化设计的能力，以及重启的能力。练习恢复路径，以确保您可以在指定时间内恢复到指定状态。在此恢复过程中使用运行手册来记录问题，并在下一次测试之前找到问题的解决方案。
  - 加州大学伯克利分校/斯坦福大学的面向恢复的计算项目
- 使用 CloudEndure Disaster Recovery 来实施和测试您的 DR 策略。
  - 使用 CloudEndure 测试灾难恢复解决方案
  - CloudEndure Disaster Recovery
  - AWS 的 CloudEndure Disaster Recovery

资源

相关文档：

- AWS 合作伙伴：可以帮助进行灾难恢复的合作伙伴
- AWS 架构博客：灾难恢复系列
- AWS Marketplace：可以用于灾难恢复的产品
- CloudEndure Disaster Recovery
- AWS 上工作负载的灾难恢复：云中的恢复（AWS 白皮书）
- 使用 CloudEndure 测试灾难恢复解决方案
- 加州大学伯克利分校/斯坦福大学的面向恢复的计算项目

- 什么是 AWS Fault Injection Simulator？

相关视频：

- AWS re:Invent 2018：适用于多区域主动-主动应用程序的架构模式（ARC209-R2）
- AWS re:Invent 2019：AWS 的备份与还原，以及灾难恢复解决方案（STG208）

相关示例：

- AWS Well-Architected 实验 – 测试弹性

REL13-BP04 管理 DR 站点或区域的配置偏差

确保 DR 站点或区域的基础设施、数据和配置满足需求。例如，检查 AMI 和服务限额是否为最新。

AWS Config 会持续监控和记录 AWS 资源配置。它可以检测到偏差并触发 AWS Systems Manager Automation 进行修复和发出警报。AWS CloudFormation 还可以在您已部署的堆栈中检测到偏差。

常见反模式：

- 在主位置进行配置或基础设施更改时，未能在恢复位置进行更新。
- 不考虑主位置和恢复位置的潜在限制（如服务区别）。

建立此最佳实践的好处： 确保您的 DR 环境与现有环境一致，可保证完整恢复。

未建立这种最佳实践的情况下暴露的风险等级： 中

实施指导

- 确保您的交付管道可交付到主站点和备份站点。用于将应用程序部署到生产中的交付管道必须分布到所有指定的灾难恢复策略位置，包括开发和测试环境。
- 启用 AWS Config 来跟踪潜在偏差位置。使用 AWS Config 规则来创建可强制实施灾难恢复策略并在检测到偏差时生成提醒的系统。
  - 按照 AWS Config 规则 修正不合规 AWS 资源
  - AWS Systems Manager Automation
- 使用 AWS CloudFormation 部署基础设施。AWS CloudFormation 可以检测 CloudFormation 模板指定的内容和实际部署内容之间的偏差。

- AWS CloudFormation：在整个 CloudFormation 堆栈上检测偏差

资源

相关文档：

- AWS 合作伙伴：可以帮助进行灾难恢复的合作伙伴
- AWS 架构博客：灾难恢复系列
- AWS CloudFormation：在整个 CloudFormation 堆栈上检测偏差
- AWS Marketplace：可以用于灾难恢复的产品
- AWS Systems Manager Automation
- AWS 上工作负载的灾难恢复：云中的恢复（AWS 白皮书）
- 如何在 AWS 上实施基础设施配置管理解决方案？
- 按照 AWS Config 规则 修正不合规 AWS 资源

相关视频：

- AWS re:Invent 2018：适用于多区域主动-主动应用程序的架构模式（ARC209-R2）

REL13-BP05 自动执行恢复

利用 AWS 或第三方工具自动进行系统恢复，并将流量路由至 DR 站点或区域。

根据已配置的运行状况检查，Elastic Load Balancing 和 AWS Auto Scaling 等 AWS 服务可将负载分配到运行正常的可用区，而 Amazon Route 53 和 AWS Global Accelerator 等服务则可将负载路由到运行正常的 AWS 区域。Amazon Route 53 Application Recovery Controller 可帮助您使用就绪检查和路由控制功能来管理和协调失效转移操作。这些功能持续监控您的应用程序从故障中恢复的能力，因此您可以跨多个 AWS 区域、可用区和本地部署控制您的应用程序恢复。

对于现有的物理或虚拟数据中心或私有云上的工作负载，AWS 弹性灾难恢复（通过 AWS Marketplace 提供）使组织能够设置自动向 AWS 进行灾难恢复的策略。CloudEndure 还支持 AWS 中的跨区域/跨可用区灾难恢复。

常见反模式：

- 实施相同的自动故障转移和故障恢复可能会导致在故障时发生摆动。

建立此最佳实践的好处： 自动恢复通过消除发生手动错误的可能性来缩短恢复时间。

未建立这种最佳实践的情况下暴露的风险等级： 中

实施指导

- 恢复路径自动化。如果恢复时间很短，人工判断和操作无法用于可用性非常高的场景。在这种情况下，系统每次必须自动进行恢复。
  - 使用 CloudEndure Disaster Recovery 自动执行失效转移和故障恢复操作。CloudEndure Disaster Recovery 可持续将您的计算机（包括操作系统、系统状态配置、数据库、应用程序和文件）复制到目标 AWS 账户和首选区域中的低成本暂存区域。在发生灾难时，您可以指示 CloudEndure Disaster Recovery 在几分钟内自动启动数千台处于完全预置状态的计算机。
    - 执行灾难恢复故障转移和故障恢复
    - CloudEndure Disaster Recovery

资源

相关文档：

- AWS 合作伙伴：可以帮助进行灾难恢复的合作伙伴
- AWS 架构博客：灾难恢复系列
- AWS Marketplace：可以用于灾难恢复的产品
- AWS Systems Manager Automation
- AWS 的 CloudEndure Disaster Recovery
- AWS 上工作负载的灾难恢复：云中的恢复（AWS 白皮书）

相关视频：

- AWS re:Invent 2018：适用于多区域主动-主动应用程序的架构模式（ARC209-R2）

性能效率

主题

- 选择
- 审核

- 监控
- 权衡

选择

问题

- PERF 1 如何选择性能最好的架构？
- PERF 2 如何选择计算解决方案？
- PERF 3 如何选择存储解决方案？
- PERF 4 如何选择数据库解决方案？
- PERF 5 如何配置联网解决方案？

PERF 1 如何选择性能最好的架构？

一个工作负载通常需要采用多种方法才能实现最佳性能。架构完善的系统会使用多种解决方案和功能来提高性能。

最佳实践

- PERF01-BP01 了解可用的服务和资源
- PERF01-BP02 制定架构选择流程
- PERF01-BP03 在制定决策时考虑成本要求
- PERF01-BP04 使用策略或参考架构
- PERF01-BP05 使用云提供商或相关合作伙伴提供的指南
- PERF01-BP06 对现有工作负载进行基准测试
- PERF01-BP07 对工作负载进行负载测试

PERF01-BP01 了解可用的服务和资源

了解云中提供的各种服务和资源。识别与您的工作负载相关的服务和配置选项，并了解如何实现最佳的性能。

如果要评估现有工作负载，您必须生成评估所需使用的各种服务资源的清单。这份清单可帮助您评估可以用托管服务和较新技术替换的组件。

常见反模式：

- 您可以将云用作联合数据中心。
- 您可以使用共享存储来存储所有需要持久性存储的内容。
- 请勿使用 Automatic Scaling。
- 您应使用最符合您当前标准的实例类型，但应根据需要使用较大的实例。
- 您可以部署和管理作为托管服务提供的技术。

建立此最佳实践的好处： 通过考量您可能不熟悉的服务，您也许能够大大降低基础设施的成本和维护服务所需的工作量。通过部署新服务和功能，您也许能够缩短上市时间。

未建立此最佳实践暴露的风险等级： 高

实施指导

盘点相关服务的工作负载软件和架构：收集工作负载清单，并确定要详细了解哪类产品。确定可以用托管服务替换的工作负载组件，以提高性能并降低运维复杂性。

资源

相关文档：

- AWS Architecture Center
- AWS Partner Network
- AWS 解决方案库
- AWS 知识中心

相关视频：

- Amazon Builders' Library 简介 (DOP328)
- 这就是我的架构

相关示例：

- AWS 示例
- AWS 开发工具包示例

PERF01-BP02 制定架构选择流程

使用关于云的内部经验和知识或外部资源（例如，已发布的使用案例、相关文档或白皮书），制定资源和服务选择流程。您应该制定一个流程，以鼓励对可能会用于工作负载的不同服务进行试验和基准测试。

针对架构编写重要用户案例时，您应该纳入性能要求，例如，指定每个重要案例应以多快速度运行。对于这些重要案例，您应该实施额外的脚本化用户体验，以确保您可以深入了解这些案例如何根据您的要求执行。

常见反模式：

- 您可以假设当前的架构将为静态并且不会随着时间的推移而更新。
- 您可以随着时间的推移对架构进行更改，而无需提供理由。

建立此最佳实践的好处： 制定架构更改流程后，您可以允许使用所收集的数据来影响以后的工作负载设计。

未建立这种最佳实践的情况下暴露的风险等级： 高

实施指导

选择架构方法：确定满足性能要求的架构类型。确定限制因素，例如，交付媒介（桌面、Web、移动设备、IoT）、传统要求和集成。确定重用（包括重构）的机会。咨询其他团队，查阅构架图和其他资源（例如，AWS 解决方案架构师、AWS 参考架构和 AWS 合作伙伴），以帮助您选择架构。

定义性能要求：根据客户体验来确定最重要的指标。确定每个指标的目标、衡量方式和优先程度。定义客户体验。记录客户所需的性能体验，包括客户如何判断工作负载的性能。优先考虑重要用户案例的体验问题。包括性能要求和实施脚本化的用户历程，以确保您知道如何根据您的要求执行用户案例。

资源

相关文档：

- AWS Architecture Center
- AWS Partner Network
- AWS 解决方案库
- AWS 知识中心

相关视频：

- Amazon Builders' Library 简介 (DOP328)
- 这就是我的架构

相关示例：

- AWS 示例
- AWS 开发工具包示例

PERF01-BP03 在制定决策时考虑成本要求

工作负载通常具有运营成本要求。根据预测的资源需求，使用内部成本控制机制来选择资源类型和规模。

确定可以将哪些工作负载组件替换为完全托管式服务，例如托管数据库、内存缓存和 ETL 服务。减少运营工作负载让您可以将资源集中到取得业务成果上。

有关成本要求最佳实践，请参阅 具有成本效益的资源 成本优化支柱白皮书 部分.

常见反模式：

- 您只应使用一个系列的实例。
- 您没有对授予许可解决方案与开源解决方案进行评估
- 您只应使用数据块存储。
- 您可以在 EC2 实例和 Amazon EBS 或临时卷上部署作为托管服务提供的常用软件。

建立此最佳实践的好处： 在制定决策时考虑成本将使您能够进行其他投资。

未建立这种最佳实践的情况下暴露的风险等级： 中

实施指导

优化工作负载组件以降低成本：设置大小合适的工作负载组件并实现弹性，可降低成本并最大程度提高组件效率。确定哪些工作负载组件可在适当情况下由托管服务替代，例如，托管数据库、内存缓存和反向代理。

资源

相关文档：

- AWS Architecture Center
- AWS Partner Network
- AWS 解决方案库
- AWS 知识中心
- AWS Compute Optimizer

相关视频：

- Amazon Builders' Library 简介 (DOP328)
- 这就是我的架构
- 优化 AWS 计算的性能和成本（CMP323-R1）

相关示例：

- AWS 示例
- AWS 开发工具包示例
- 在启用 Compute Optimizer 和内存利用率的情况下合理调整大小
- AWS Compute Optimizer 演示代码

PERF01-BP04 使用策略或参考架构

通过评估内部策略和现有参考架构，以及使用分析为工作负载选择服务和配置，来最大程度提高性能和效率。

常见反模式：

- 您应该允许广泛使用可能会影响公司管理开销的各种技术。

建立此最佳实践的好处： 制定架构、技术和供应商选择策略将有助于快速做出决策。

未建立此最佳实践暴露的风险等级： 中

实施指导

使用现有策略或参考架构部署工作负载：将服务集成到您的云部署中，然后使用性能测试来确保您可以继续满足性能要求。

资源

相关文档：

- AWS Architecture Center
- AWS Partner Network
- AWS 解决方案库
- AWS 知识中心

相关视频：

- Amazon Builders' Library 简介 (DOP328)
- 这就是我的架构

相关示例：

- AWS 示例
- AWS 开发工具包示例

PERF01-BP05 使用云提供商或相关合作伙伴提供的指南

使用云公司提供的资源，例如，解决方案架构师、专业服务或适当的合作伙伴来指导您的决策。这些资源可帮助进行审核，并改进您的架构，从而实现最佳性能。

如需其他指导或产品信息，请联系 AWS 以获取帮助。AWS 解决方案架构师和 AWS 专业服务 提供解决方案实施指导。 AWS 合作伙伴 提供 AWS 专业知识，可帮助您实现业务敏捷性和创新能力。

常见反模式：

- 您使用 AWS 作为普通数据中心提供商。
- 您没有按 AWS 服务的既定用途使用这些服务。

建立此最佳实践的好处： 咨询您的提供商或合作伙伴将使您在决策中充满信心。

未建立此最佳实践暴露的风险等级： 中

实施指导

联系 AWS 资源以获得帮助：AWS 解决方案架构师和专业服务提供解决方案实施指导。APN 合作伙伴提供 AWS 专业知识，可帮助您实现业务敏捷性和创新能力。

资源

相关文档：

- AWS Architecture Center
- AWS Partner Network
- AWS 解决方案库
- AWS 知识中心

相关视频：

- Amazon Builders' Library 简介 (DOP328)
- 这就是我的架构

相关示例：

- AWS 示例
- AWS 开发工具包示例

PERF01-BP06 对现有工作负载进行基准测试

对现有工作负载的性能进行基准测试，以了解工作负载在云上的运行情况。使用从基准测试中收集的数据来推动架构决策。

结合使用基准测试与综合测试和真实用户监控，生成有关工作负载组件性能的数据。相比负载测试，基准测试通常可以更快速地设置，适用于评估特定组件的技术。基准测试通常在新项目开始时进行，因为此时您还没有用于进行负载测试的完整解决方案。

您可以构建您自己的自定义基准测试，或者您可以使用行业标准的测试，例如 TPC-DS （对您的数据仓库工作负载进行基准测试）。行业基准适用于比较不同的环境。对于架构中的特定操作类型，自定义基准十分有用。

进行基准测试时，为了确保获得有效结果，预热您的测试环境尤为重要。多次运行同一基准测试，确保捕获在一段时间内的差异信息。

由于基准测试运行速度通常比负载测试快，它们可以在部署管道的早期使用，并能更快地提供有关性能偏差的反馈。当您评估一个组件或服务的重要更改时，您可以使用基准快速了解您是否有合理的理由来执行更改。结合使用基准测试与负载测试这一点很重要，因为负载测试会告诉您工作负载在生产环境中的表现如何。

常见反模式：

- 您可以依赖于不表示工作负载特性的常见基准。
- 您依赖客户反馈和看法，将其作为唯一的基准。

建立此最佳实践的好处： 对您的当前实施进行基准测试，以便衡量性能改进情况。

未建立这种最佳实践的情况下暴露的风险等级： 中

实施指导

在开发期间监控性能：实施可以让您在工作负载的发展期间了解其性能的流程。

集成到您的交付管道：在您的交付管道中自动运行负载测试。将测试结果与预先定义的关键性能指标 (KPI) 和阈值进行比较，以确保您继续满足性能要求。

测试用户体验：使用合成或净化版本的生产数据（删除敏感信息或身份识别信息）进行负载测试。在应用程序中大规模使用重演或预先编程的用户体验，从而演练整个架构。

真实用户监控：使用 CloudWatch RUM 帮助您收集和查看有关应用程序性能的客户端数据。使用这些数据来帮助建立您的真实用户性能基准。

资源

相关文档：

- AWS Architecture Center
- AWS Partner Network
- AWS 解决方案库
- AWS 知识中心
- Amazon CloudWatch RUM
- Amazon CloudWatch Synthetics

相关视频：

- Amazon Builders' Library 简介 (DOP328)
- 这就是我的架构
- 通过 Amazon CloudWatch RUM 优化应用程序
- Amazon CloudWatch Synthetics 演示

相关示例：

- AWS 示例
- AWS 开发工具包示例
- 分布式负载测试
- 使用 Amazon CloudWatch Synthetics 测量页面加载时间
- Amazon CloudWatch RUM Web 客户端

PERF01-BP07 对工作负载进行负载测试

使用不同的资源类型和大小在云上部署最新的工作负载架构。监控部署情况，捕获用于识别性能瓶颈或容量过剩的性能指标。使用此性能信息来设计或改进您的架构和资源选择。

负载测试使用 您的实际 工作负载，以便您可以了解解决方案在生产环境中的表现。负载测试必须使用生产数据的合成或净化版本（删除敏感信息或身份识别信息）运行。大规模使用重演或预设的工作负载用户旅程，演练整个架构。作为交付管道的一部分，自动执行负载测试，并将结果与预定义的 KPI 和阈值进行比较。这可以确保您持续获得所需的性能。

常见反模式：

- 您可以对工作负载的各个部分进行单独负载测试，而不必测试整个工作负载。
- 您可以在与生产环境不同的基础设施上进行负载测试。
- 您只能对预期负载，而不能对其他负载进行负载测试，以帮助预测未来可能会出现问题的方面。
- 在不通知 AWS Support 的情况下执行负载测试，并让您的测试就像拒绝服务事件那样失败。

建立此最佳实践的好处： 通过负载测试来衡量您的性能，可向您说明随着负载的增加，您将在哪些方面受到影响。这样您便可以在更改影响您的工作负载之前，对所需进行的更改进行预测。

未建立此最佳实践暴露的风险等级： 低

实施指导

利用负载测试来验证方法：对概念验证方案进行负载测试，以确定您是否满足性能要求。您可以使用 AWS 服务来运行生产规模的环境，以测试您的架构。由于您只需在需要时为测试环境付费，因此，执行全面测试的成本远远低于使用本地环境的成本。

监控指标：Amazon CloudWatch 可以收集架构中各种资源的指标。您也可以收集和发布自定义指标，用于显示业务指标或派生指标。使用 CloudWatch 或第三方解决方案来设置指示超出阈值的警报。

大规模测试：负载测试时使用您的实际工作负载，以便您可以了解解决方案在生产环境中的表现。您可以使用 AWS 服务来运行生产规模的环境，以测试您的架构。由于您只需为所需的测试环境付费，因此，执行全面测试的成本要低于使用本地环境的成本。利用 AWS Cloud 测试您的工作负载，以发现工作负载的哪些部分无法扩展或者是否以非线性方式扩展。例如，您可以使用 Spot 实例以很低的成本生成负载，并在投入生产前发现瓶颈。

资源

相关文档：

- AWS CloudFormation
- 使用 CloudFormer 构建 AWS CloudFormation 模板
- Amazon CloudWatch RUM
- Amazon CloudWatch Synthetics
- AWS 上的分布式负载测试

相关视频：

- Amazon Builders' Library 简介 (DOP328)
- 通过 Amazon CloudWatch RUM 优化应用程序
- Amazon CloudWatch Synthetics 演示

相关示例：

- AWS 上的分布式负载测试

PERF 2 如何选择计算解决方案？

适合工作负载的最佳计算解决方案会根据应用程序设计、使用模式和配置设置而有所不同。架构可以使用不同的计算解决方案来支持各种组件，并且可以实现各种不同的功能来提高性能。为架构选择错误的计算解决方案可能会降低性能效率。

最佳实践

- PERF02-BP01 评估可用的计算方案
- PERF02-BP02 了解可用的计算配置选项
- PERF02-BP03 收集与计算相关的指标
- PERF02-BP04 通过合理调整大小来确定需要的配置
- PERF02-BP05 利用可用的资源弹性
- PERF02-BP06 根据指标重新评估计算需求

PERF02-BP01 评估可用的计算方案

了解您的工作负载如何从使用不同的计算方案（例如实例、容器和函数）中受益。

期望结果： 通过了解所有可用的计算方案，您可以发现提高性能、降低不必要的基础设施成本和减少维护工作负载所需的运营工作量的机会。部署新服务和功能后，您还能缩短上市时间。

常见反模式：

- 在迁移后工作负载中，使用与本地使用的相同的计算解决方案。
- 缺乏对云计算解决方案以及这些解决方案可如何提高计算性能的认识。
- 为了满足扩展或性能需求，现有计算解决方案采用了过大的规模，而使用替代计算解决方案可以更准确地满足您的工作负载特性需求。

建立此最佳实践的好处： 通过确定计算需求和评估可用的计算解决方案，业务利益相关者和工程团队将了解使用所选计算解决方案的好处和局限性。所选计算解决方案应符合工作负载性能标准。关键标准包括：处理需求、流量模式、数据访问模式、扩展需求和延迟要求。

未建立这种最佳实践的情况下暴露的风险等级： 高

实施指导

了解虚拟化、容器化和管理解决方案，这些解决方案可以为您的工作负载带来好处并满足性能要求。一个工作负载可以包含多种类型的计算解决方案。每种计算解决方案都有不同的特征。根据您的工作负载

规模和计算要求，可以选择和配置计算解决方案以满足您的需求。云架构师应该了解实例、容器和函数的优缺点。以下步骤将帮助您了解如何选择计算解决方案，以符合您的工作负载特性和性能要求。

类型	服务器	容器	函数
AWS 服务	Amazon Elastic Compute Cloud (Amazon EC2)	Amazon Elastic Container Service（Amazon ECS）、Amazon Elastic Kubernetes Service（Amazon EKS）	AWS Lambda
主要特征	具有面向硬件许可要求的专用选项、放置选项，以及基于计算指标的大量不同实例系列选择	易于部署、一致的环境、在 EC2 实例之上运行、可扩展	运行时间短（15 分钟或更短），最大内存和 CPU 不如其他服务高，托管硬件层，可扩展到数百万并发请求
常见使用案例	直接迁移、整体式应用程序、混合环境、企业应用程序	微服务、混合环境、	微服务、事件驱动的应用程序

实施步骤：

1. 通过评估选择计算解决方案必须驻留的位置 the section called “PERF05-BP06 根据网络要求选择工作负载的位置”。此位置将限制可供您使用的计算解决方案的类型。
2. 确定符合位置要求和应用程序要求的计算解决方案类型
    a. Amazon Elastic Compute Cloud (Amazon EC2) 虚拟服务器实例具有各种不同的系列和规模。它们提供各种功能，包括固态硬盘（SSD，Solid State Drive）和图形处理单元（GPU，Graphics Processing Unit）。EC2 实例在实例选择方面提供了最大的灵活性。启动 EC2 实例时，您指定的实例类型决定了实例的硬件。每种实例类型都提供不同的计算、内存和存储功能。我们按照这些功能把实例分组到实例系列。典型的使用案例包括：运行企业应用程序、高性能计算（HPC，High Performance Computing）、训练和部署机器学习应用程序以及运行云原生应用程序。

b. Amazon Elastic Container Service (Amazon ECS) 是一项完全托管的容器编排服务，通过此服务，您可以使用 AWS Fargate 在 EC2 实例或无服务器实例集群上自动运行和管理容器。您可以结合使用 Amazon ECS 与其他服务，如 Amazon Route 53、Secrets Manager、AWS Identity and Access Management（IAM）和 Amazon CloudWatch。如果您的应用程序是容器化的并且工程团队首选 Docker 容器，则建议使用 Amazon ECS。

c. Amazon Elastic Kubernetes Service（Amazon EKS） 是一项完全托管的 Kubernetes 服务。您可以选择使用 AWS Fargate 运行 EKS 集群，而无需预置和管理服务器。由于与 AWS 服务（如 Amazon CloudWatch、自动扩缩组、AWS Identity and Access Management（IAM）和 Amazon Virtual Private Cloud（VPC））集成，Amazon EKS 的管理得到了简化。使用容器时，必须使用计算指标为您的工作负载选择最佳类型，类似于使用计算指标选择 EC2 或 AWS Fargate 实例类型的方式。如果您的应用程序是容器化的并且工程团队首选 Kubernetes 容器而不是 Docker 容器，则建议使用 Amazon EKS。

d. 您可以使用 AWS Lambda 运行支持允许的运行时、内存和 CPU 选项的代码。您只需上传代码，AWS Lambda 就会处理运行和扩展代码所需的一切工作。您可以将代码设置为从其他 AWS 服务自动触发或直接调用它。对于为云开发的短时间运行的微服务架构，建议使用 Lambda。

3. 在试用新的计算解决方案后，规划迁移并验证性能指标。这是一个持续的过程，请参阅 the section called “PERF02-BP04 通过合理调整大小来确定需要的配置”.

实施计划的工作量级别： 如果工作负载从一种计算解决方案转移到另一种计算解决方案，则重构应用程序可能需要 中等 工作量。

资源

相关文档：

- 使用 AWS 进行云计算
- EC2 实例类型
- EC2 实例的处理器状态控制
- EKS 容器：EKS Worker 节点
- Amazon ECS 容器：Amazon ECS 容器实例
- 函数：Lambda 函数配置
- 容器规范性指南
- 无服务器规范性指南

相关视频：

- 如何为初创公司选择计算方案
- 优化 AWS 计算的性能和成本（CMP323-R1）
- Amazon EC2 foundations (CMP211-R2)
- 推动下一代 Amazon EC2：深入了解 Nitro 系统
- 使用 AWS Inferentia 提供高性能的 ML 推理（CMP324-R1）
- 更好、更快、更便宜的计算：Amazon EC2 成本优化（CMP202-R1）

相关示例：

- 将 Web 应用程序迁移到容器
- 运行无服务器 Hello World

PERF02-BP02 了解可用的计算配置选项

每种计算解决方案都有可供您使用的选项和配置，以支持您的工作负载特性。了解各种选项如何补充您的工作负载，以及哪些配置选项最适合您的应用程序。这些选项的示例包括实例系列、规模、功能（GPU、I/O）、突增、超时、函数大小、容器实例和并发度。

期望结果： 包括 CPU、内存、网络吞吐量、GPU、IOPS、流量模式和数据访问模式在内的工作负载特性将整理在案，用于配置计算解决方案以匹配工作负载特性。这些指标加上特定于工作负载的自定义指标都会被记录并监控，然后用于优化计算配置以最好地满足要求。

常见反模式：

- 使用与本地使用的相同的计算解决方案。
- 不审核计算方案或实例系列以匹配工作负载特性。
- 扩大计算规模以确保突增能力。
- 您可以为同一工作负载使用多个计算管理平台。

建立此最佳实践的好处： 熟悉 AWS 计算产品/服务，以便为每个工作负载确定合适的解决方案。为工作负载选择计算产品/服务后，您可以快速试用这些计算产品/服务，以确定它们在多大程度上满足您的工作负载需求。为满足您的工作负载特性而优化的计算解决方案将会提高性能、降低成本并提高可靠性。

未建立这种最佳实践的情况下暴露的风险等级： 高

实施指导

如果您的工作负载已经使用相同的计算方案超过四周，并且您预计这些特征在未来将保持不变，那么您可以使用 AWS Compute Optimizer 根据您的计算特征向您提供建议。如果由于缺乏指标、实例类型不受支持或预计特征会发生变化而无法选择使用 AWS Compute Optimizer，那么您必须 根据负载测试和实验来预测您的指标。

实施步骤：

1. 您是否在 EC2 实例或具有 EC2 启动类型的容器上运行？
    a. 您的工作负载能否使用 GPU 来提高性能？
        i. 加速计算型 实例是基于 GPU 的实例，可为机器学习训练、推理和高性能计算提供最高性能。
    b. 您的工作负载是否运行机器学习推理应用程序？
        i. AWS Inferentia（Inf1） – Inf1 实例旨在支持机器学习推理应用程序。通过使用 Inf1 实例，客户可以运行大规模机器学习推理应用程序，例如图像识别、语音识别、自然语言处理、个性化和欺诈检测。您可以在 TensorFlow、PyTorch 或 MXNet 等流行的机器学习框架中构建模型，并使用 GPU 实例来训练模型。在对机器学习模型进行训练以满足要求之后，您可以使用 AWS Neuron在 Inf1 实例上部署模型，AWS Neuron 是一种专门的软件开发工具包（SDK），它由编译器、运行时和分析工具组成，可优化 Inferentia 芯片的机器学习推理性能。
    c. 您的工作负载是否与底层硬件集成以提高性能？
        i. 现场可编程门阵列 (FPGA) – 使用 FPGA，您可以通过为要求最苛刻的工作负载定制硬件加速执行来优化工作负载。您可以利用受支持的通用编程语言（例如 C 语言或 Go 语言）或面向硬件的语言（例如 Verilog 语言或 VHDL 语言）来定义算法。
    d. 您是否有至少四周的指标，并且可以预测您的流量模式和指标在未来将保持不变？
        i. 使用 Compute Optimizer 获得关于哪种计算配置最符合您的计算特征的机器学习建议。
    e. 您的工作负载性能是否受到 CPU 指标的限制？
        i. 计算优化型 实例非常适合需要高性能处理器的工作负载。
    f. 您的工作负载性能是否受到内存指标的限制？
        i. 内存优化型 实例提供大量内存以支持内存密集型工作负载。
    g. 您的工作负载性能是否受到 IOPS 的限制？
        i. 存储优化型 实例专为需要对本地存储进行大量顺序读写访问（IOPS）的工作负载而设计。
    h. 您的工作负载特性是否表示需要在所有指标之间取得平衡？
        i. 您的工作负载 CPU 是否需要突增以处理流量峰值？

A. 可突增性能 实例类似于计算优化型实例，不同之处在于它们提供了功能，可以突破计算优化型实例中确定的固定 CPU 基线。

ii. 通用型 实例平衡了所有特性以支持各种工作负载。

i. 您的计算实例是否在 Linux 上运行并受到网络接口卡上的网络吞吐量的限制？

i. 查看 性能问题 5，最佳实践 2：评估可用的联网功能， 找到合适的实例类型和系列来满足您的性能需求。

j. 您的工作负载是否在特定可用区中需要一致、可预测的实例且您可以承诺一年的使用？

i. 预留实例 确保特定可用区中的容量预留。预留实例是在特定可用区中提供所需计算能力的理想选择。

k. 您的工作负载是否具有需要专用硬件的许可证？

i. 专用主机 支持现有的软件许可证，并帮助您满足合规性要求。

l. 您的计算解决方案是否会出现突增并需要同步处理？

i. 按需实例 让您可以按小时或按秒使用计算容量，而无需做出长期承诺。这些实例非常适合超出性能基线需求的突增情况。

m. 您的计算解决方案是无状态、具备容错能力和异步的吗？

i. 竞价型实例 让您可以将未使用的实例容量用于无状态的容错工作负载。

2. 您是否在 Fargate上运行容器？

a. 您的任务性能是否受到内存或 CPU 的限制？

i. 使用 任务大小 调整内存或 CPU。

b. 性能是否受到流量模式突增的影响？

i. 使用 Auto Scaling 配置以匹配您的流量模式。

3. 您的计算解决方案是否位于 Lambda？

a. 您是否有至少四周的指标，并且可以预测您的流量模式和指标在未来将保持不变？

i. 使用 Compute Optimizer 获得关于哪种计算配置最符合您的计算特征的机器学习建议。

b. 您是否没有足够的指标来使用 AWS Compute Optimizer？

i. 如果您没有可用的指标来使用 Compute Optimizer，请使用 AWS Lambda Power Tuning 帮助选择最佳配置。

c. 您的函数性能是否受到内存或 CPU 的限制？

i. 配置 Lambda 内存 以满足您的性能需求指标。

d. 您的函数在执行时是否超时？

i. 更改 超时设置

e. 您的函数性能是否受到突发活动和并发性的限制？

   i. 配置 并发设置 以满足您的性能要求。

f. 您的函数是否异步执行并且在重试时失败？

   i. 在 异步配置 设置中配置事件的最大期限和最大重试次数限制。

实施计划的工作量级别：

要建立此最佳实践，您必须了解当前的计算特征和指标。收集这些指标，建立基线，然后使用这些指标来确定理想的计算方案，这需要 低 到 中等 工作量。这最好通过负载测试和实验来验证。

资源

相关文档：

- 使用 AWS 进行云计算
- AWS Compute Optimizer
- EC2 实例类型
- EC2 实例的处理器状态控制
- EKS 容器：EKS Worker 节点
- Amazon ECS 容器：Amazon ECS 容器实例
- 函数：Lambda 函数配置

相关视频：

- Amazon EC2 foundations (CMP211-R2)
- 推动下一代 Amazon EC2：深入了解 Nitro 系统
- 优化 AWS 计算的性能和成本（CMP323-R1）

相关示例：

- 在启用 Compute Optimizer 和内存利用率的情况下合理调整大小
- AWS Compute Optimizer 演示代码

PERF02-BP03 收集与计算相关的指标

要了解计算资源的性能，您必须记录和跟踪各种系统的利用率。此数据可用于更准确地确定资源需求。

工作负载会生成大量数据，例如指标、日志和事件。确定您现有的存储、监控和可观察性服务是否可以管理生成的数据。确定反映资源利用率并且可以在单个平台上收集、聚合和关联的指标。这些指标应该代表您的所有工作负载资源、应用程序和服务，以便您可以轻松获得系统范围的可见性，并快速识别性能改进机会和问题。

期望结果： 在单个平台上，识别、收集、聚合和关联涉及到计算相关资源的所有指标，并进行保留以支持成本和运营目标。

常见反模式：

- 您只能手动搜索日志文件来查找指标。
- 您只能将指标发布到内部工具。
- 您只使用所选监控软件记录的默认指标。
- 您只在出现问题时检查指标。

建立此最佳实践的好处： 要监控工作负载的性能，必须记录一段时间的多项性能指标。您可以利用这些指标来检测性能异常。这些指标还有助于根据业务指标衡量性能，以确保满足工作负载需求。

未建立这种最佳实践的情况下暴露的风险等级： 高

实施指导

识别、收集、聚合和关联与计算相关的指标。使用 Amazon CloudWatch 之类的服务可以使实施速度更快并更易于维护。除了记录的默认指标外，还可以识别和跟踪工作负载中的其他系统级指标。记录 CPU 利用率、内存、磁盘 I/O 和网络入站和出站指标等数据，以深入了解利用率水平或瓶颈。这些数据对于了解工作负载的性能以及计算解决方案的使用方式至关重要。将这些指标用作数据驱动方法的一部分，以便主动调整和优化工作负载的资源。

实施步骤：

1. 必须跟踪哪些计算解决方案指标？
    a. EC2 默认指标
    b. Amazon ECS 默认指标
    c. EKS 默认指标

   d. Lambda 默认指标
   e. EC2 内存和磁盘指标
2. 我目前是否有经过批准的日志记录和监控解决方案？
   a. Amazon CloudWatch
   b. 适用于 OpenTelemetry 的 AWS Distro
   c. Amazon Managed Service for Prometheus
3. 我是否确定并配置了数据留存策略，以符合我的安全和运营目标？
   a. CloudWatch 指标的默认数据留存
   b. CloudWatch Logs 的默认数据留存
4. 您如何部署指标和日志聚合代理？
   a. AWS Systems Manager Automation
   b. OpenTelemetry Collector

实施计划的工作量级别： 从所有计算资源中识别、跟踪、收集、聚合和关联指标所需的工作量为 中 。

资源

相关文档：

- Amazon CloudWatch 文档
- 使用 CloudWatch 代理从 Amazon EC2 实例和本地服务器收集指标和日志
- 访问 AWS Lambda 的 Amazon CloudWatch Logs
- 结合使用 CloudWatch Logs 与容器实例
- 发布自定义指标
- AWS Answers：集中式日志记录
- 发布 CloudWatch 指标的 AWS 服务
- 在 AWS Fargate 上监控 Amazon EKS

相关视频：

- AWS 上的应用程序性能管理
- 制定监控计划

相关示例：

- 第 100 级：使用 CloudWatch 控制面板进行监控
- 第 100 级：使用 CloudWatch 控制面板监控 Windows EC2 实例
- 第 100 级：使用 CloudWatch 控制面板监控 Amazon Linux EC2 实例

PERF02-BP04 通过合理调整大小来确定需要的配置

分析您的工作负载的各种性能特性，以及这些特性与内存、网络和 CPU 使用率之间的关系。根据这些数据选择最适合您的工作负载配置文件的资源。例如，实例的 r 系列可以最好地处理内存密集型工作负载（例如数据库）。但是，弹性容器系统可为突增的工作负载提供更多优势。

常见反模式：

- 您应选择可用于所有工作负载的最大的实例。
- 您应将所有实例类型标准化为一种类型，以便于管理。

建立此最佳实践的好处： 熟悉 AWS 计算产品/服务可帮助您确定适用于各种工作负载的合适解决方案。为工作负载选择各种计算产品/服务后，您可以快速灵活地试用这些计算产品/服务，以确定哪些产品/服务满足您的工作负载需求。

未建立此最佳实践暴露的风险等级： 中

实施指导

通过合理调整大小来修改工作负载配置：要优化性能和整体效率，请确定工作负载需要哪些资源。对于对内存的要求比对 CPU 的要求更高的系统，选择内存优化型实例，对于执行非内存密集型数据处理的组件，选择计算优化型实例。合理调整大小可让您的工作负载能够在只使用所需资源的情况下，尽可能高性能地运行。

资源

相关文档：

- AWS Compute Optimizer
- 使用 AWS 进行云计算
- EC2 实例类型
- ECS 容器：Amazon ECS 容器实例
- EKS 容器：EKS Worker 节点

- 函数：Lambda 函数配置
- EC2 实例的处理器状态控制

相关视频：

- Amazon EC2 foundations (CMP211-R2)
- 更好、更快、更便宜的计算：Amazon EC2 成本优化（CMP202-R1）
- 使用 AWS Inferentia 提供高性能的 ML 推理（CMP324-R1）
- 优化 AWS 计算的性能和成本（CMP323-R1）
- 推动下一代 Amazon EC2：深入了解 Nitro 系统
- 如何为初创公司选择计算方案
- 优化 AWS 计算的性能和成本（CMP323-R1）

相关示例：

- 在启用 Compute Optimizer 和内存利用率的情况下合理调整大小
- AWS Compute Optimizer 演示代码

PERF02-BP05 利用可用的资源弹性

云让您能够通过各种机制灵活地动态扩展或缩减资源，以便满足不断变化的需求。结合与计算相关的指标，工作负载可以自动响应这些变化，并利用一系列最优的资源来实现其目标。

实现最佳供需匹配能够尽可能降低工作负载成本，但您也必须准备充足的供应，以便应对预置时间问题和单个资源的故障。需求可以是固定的，也可以是变化的，所以需要通过指标和自动化来确保管理本身不会成为一种负担，而且不会产生不成比例的高成本。

借助 AWS，您可以使用大量不同方法以实现供需匹配。《成本优化支柱》白皮书描述了如何使用以下方法进行成本优化：

- 基于需求的方法
- 基于缓冲区的方法
- 基于时间的方法

您必须确保工作负载部署可以处理扩展和缩减事件。创建缩减事件的测试方案，以确保工作负载按预期方式运行。

常见反模式：

- 您通过手动增加容量来对警报做出反应。
- 在扩展事件之后，您将保留增加的容量，而不是缩减容量。

建立此最佳实践的好处： 配置和测试工作负载弹性将有助于节省资金，维护性能基准，并在流量变化时提高可靠性。大多数非生产实例在不使用时都应该停止。尽管可以手动关闭未使用的实例，但在规模较大时这是无法实现的。您也可以利用基于卷的弹性，此功能通过在需求激增时自动增加计算资源数量，并在需求减少时减小容量，从而能够优化性能并降低成本。

未建立此最佳实践暴露的风险等级： 中

实施指导

利用弹性：弹性可根据您对资源的需求来向您提供这些资源。实例、容器和函数都能够与自动扩展功能相结合或作为此服务的一项功能来提供可实现弹性的机制。在您的架构中利用弹性，可确保您有足够的容量来满足所有使用规模的性能要求。确保衡量扩展或缩减弹性资源的指标已根据所部署的工作负载类型进行了验证。如果您正在部署一个视频转码应用程序，CPU 利用率预计为 100%，并且不应将此作为您的主要指标。或者，您也可以衡量等待缩放您的实例类型的转码作业的队列深度。确保工作负载部署可以处理扩展事件和缩减事件。安全地缩减工作负载组件，与在需要时扩展资源同样重要。创建缩减事件的测试方案，以确保工作负载按预期方式运行。

资源

相关文档：

- 使用 AWS 进行云计算
- EC2 实例类型
- ECS 容器：Amazon ECS 容器实例
- EKS 容器：EKS Worker 节点
- 函数：Lambda 函数配置
- EC2 实例的处理器状态控制

相关视频：

- Amazon EC2 foundations (CMP211-R2)
- 更好、更快、更便宜的计算：Amazon EC2 成本优化（CMP202-R1）

- 使用 AWS Inferentia 提供高性能的 ML 推理（CMP324-R1）
- 优化 AWS 计算的性能和成本（CMP323-R1）
- 推动下一代 Amazon EC2：深入了解 Nitro 系统

相关示例：

- Amazon EC2 Auto Scaling 组示例
- Amazon EFS 教程

PERF02-BP06 根据指标重新评估计算需求

使用系统级指标来确定工作负载的行为和要求。通过比较可用资源和这些要求来评估工作负载的需求，并对计算环境进行更改以实现与您的工作负载配置文件的最佳匹配。例如，随着时间的推移，系统可能比最初认为的要更频繁地使用内存，所以转为使用其他实例系列或调整实例大小可能会提高性能和效率。

常见反模式：

- 您只需监控系统级指标，即可深入了解您的工作负载。
- 您需要为峰值工作负载要求设计您的计算需求。
- 为了满足扩展或性能需求，现有计算解决方案采用了过大的规模，而迁移到新的计算解决方案即可满足您的工作负载特性需求。

建立此最佳实践的好处： 要优化性能和提高资源利用率，您需要一个统一的运营视图、实时粒度数据和历史参考。您可以创建自动控制面板来显示这些数据并执行指标计算，以进行运营和利用率分析。

未建立这种最佳实践的情况下暴露的风险等级： 低

实施指导

使用数据驱动的方法来优化资源：要实现最高性能和效率，请使用一段时间内从工作负载中收集的数据来调整和优化您的资源。查看工作负载对当前资源的使用趋势，并确定可以在哪些方面做出更改，以便更好地满足您的工作负载需求。当资源被过度使用时，系统性能会降低，而资源没有得到充分利用会导致资源使用效率较低并且成本较高。

资源

相关文档：

- 使用 AWS 进行云计算
- AWS Compute Optimizer
- 使用 AWS 进行云计算
- EC2 实例类型
- ECS 容器：Amazon ECS 容器实例
- EKS 容器：EKS Worker 节点
- 函数：Lambda 函数配置
- EC2 实例的处理器状态控制

相关视频：

- Amazon EC2 foundations (CMP211-R2)
- 更好、更快、更便宜的计算：Amazon EC2 成本优化（CMP202-R1）
- 使用 AWS Inferentia 提供高性能的 ML 推理（CMP324-R1）
- 优化 AWS 计算的性能和成本（CMP323-R1）
- 推动下一代 Amazon EC2：深入了解 Nitro 系统

相关示例：

- 在启用 Compute Optimizer 和内存利用率的情况下合理调整大小
- AWS Compute Optimizer 演示代码

PERF 3 如何选择存储解决方案？

针对特定系统的最佳存储解决方案往往取决于访问类型（块、文件或者对象存储）、访问模式（随机或者连续）、数据吞吐量要求、访问频率（在线、离线、归档）、更新频度（WORM、动态）以及可用性与持久性限制等因素。架构良好的系统使用多种解决方案，并且可以实现各种不同的功能来提高性能。

最佳实践

- PERF03-BP01 了解存储特征和要求
- PERF03-BP02 评估可用的配置选项
- PERF03-BP03 根据访问模式和指标做出决策

PERF03-BP01 了解存储特征和要求

确定和记录工作负载存储需求，并定义每个位置的存储特征。存储特征示例包括：可共享访问、文件大小、增长率、吞吐量、IOPS、延迟、访问模式和数据持久性。使用这些特征来评估数据块、文件、对象或实例存储服务是否是满足您的存储需求的最有效解决方案。

期望结果： 根据存储要求确定并记录存储要求，并评估可用的存储解决方案。基于关键存储特征，您的团队将了解所选存储服务将如何提高您的工作负载性能。关键标准包括数据访问模式、增长率、扩展需求和延迟要求。

常见反模式：

- 对于所有工作负载，您都只使用一种存储类型，例如 Amazon Elastic Block Store（Amazon EBS）。
- 您可以假设所有工作负载都具有相似的存储访问性能要求。

建立此最佳实践的好处： 根据已确定和所需的特征选择存储解决方案将有助于提高工作负载性能，降低成本并减少维护工作负载的运营工作量。您的工作负载性能将受益于存储服务的解决方案、配置和位置。

未建立这种最佳实践的情况下暴露的风险等级： 高

实施指导

确定您的工作负载最重要的存储性能指标，并使用基准测试或负载测试，将其作为数据驱动型方法的一部分来实施改进。使用这些数据确定存储解决方案受限的方面，并检查可以改进解决方案的配置选项。确定工作负载的预期增长率，然后选择满足这些增长率的存储解决方案。研究 AWS 存储产品以确定适合您的各种工作负载需求的正确存储解决方案。通过在 AWS 中预置存储解决方案，您有更多机会测试存储产品并确定它们是否适合您的工作负载需求。

AWS 服务	主要特征	常见使用场景
Amazon S3	持久性高达 99.999999999%，无限增长，可从任何地方访问，多种基于访问和弹性的成本模式	云原生应用程序数据、数据存档和备份、分析、数据湖、静态网站托管、IoT 数据
Amazon S3 Glacier	几秒钟到几小时的延迟，无限增长，极低成本，长期存储	数据存档，媒体存档，长期备份保留。

AWS 服务	主要特征	常见使用场景
Amazon EBS	存储大小需要管理和监控，低延迟，持久性存储，99.8% 至 99.9% 的持久性，大多数卷类型只能从一个 EC2 实例访问。	COTS 应用程序，I/O 密集型应用程序，关系型和 NoSQL 数据库，备份和恢复
EC2 Instance Store	预先确定的存储大小，极低延迟，不持久，只能从一个 EC2 实例访问	COTS 应用程序，I/O 密集型应用程序，内存中数据存储
Amazon EFS	持久性高达 99.999999999%，无限增长，可由多项计算服务访问	现代化应用程序跨多项计算服务共享文件，文件存储用于扩展内容管理系统
Amazon FSx	支持四个文件系统（NetApp、OpenZFS、Windows File Server 和 Amazon FSx for Lustre），每个文件系统的可用存储空间不同，可由多项计算服务访问	云原生工作负载，私有云爆发，需要特定文件系统的迁移工作负载，VMC，ERP 系统，本地文件存储和备份
Snow Family	便携式设备，256 位加密，NFS 端点，机载计算，TB 级存储	将数据迁移到云端，存储，以及在极端的本地条件下的计算，灾难恢复，远程数据收集
AWS Storage Gateway	提供对云支持存储的低延迟本地访问，完全托管本地缓存	本地数据到云的迁移，从本地数据源填充云数据湖，现代化的文件共享。

实施步骤：

1. 使用基准测试或负载测试来收集您的存储需求的主要特征。主要特征包括：
    a. 可共享（什么组件可以访问这个存储）
    b. 增长率
    c. 吞吐量
    d. 延迟

e. I/O 大小

f. 持久性

g. 访问模式（读写、频率、峰值或一致）

2. 确定支持您的存储特征的存储解决方案的类型。

a. Amazon S3 是一项对象存储服务，具有无限的可扩展性、高可用性和多种可访问性选项。在 Amazon S3 内外传输和访问对象可以使用诸如 Transfer Acceleration 或 Access Points 之类的服务，来支持您的位置、安全需求和访问模式。使用 Amazon S3 的性能准则 来帮助您优化 Amazon S3 配置，以满足工作负载性能需求。

b. Amazon S3 Glacier 是 Amazon S3 的一个存储类，用于数据存档。有三种存档解决方案可供您选择，访问时间从几毫秒到 5-12 小时不等，具有不同的成本和安全选项。Amazon S3 Glacier 通过实施支持业务需求和数据特征的数据生命周期，可以帮助您满足性能需求。

c. Amazon Elastic Block Store（Amazon EBS） 是一项专用于 Amazon Elastic Compute Cloud（Amazon EC2）的高性能数据块存储服务。您可以选择 基于 SSD 或 HDD 的解决方案，这些解决方案具有不同的特征，并对 IOPS 或 吞吐量划分了优先级。EBS 卷非常适合高性能工作负载，是文件系统、数据库或只能访问附加阶段系统的应用程序的主存储。

d. Amazon EC2 实例存储 与 Amazon EBS 类似，因为它附加到 Amazon EC2 实例，但是，该实例存储只是临时存储，最好是作为缓冲区、缓存或其他临时内容使用。如果实例关闭，则无法分离实例存储，并且所有数据都会丢失。实例存储可用于高 I/O 性能和低延迟使用场景，在这些使用场景中，数据不需要持续存在。

e. Amazon Elastic File System（Amazon EFS） 是可由多种类型的计算解决方案访问的可挂载文件系统。Amazon EFS 会自动增大和缩小存储，并进行性能优化以提供一致的低延迟。EFS 有 两种性能配置模式：通用和最大 I/O。通用模式具有亚毫秒级读取延迟和几毫秒的写入延迟。最大 I/O 模式可以支持成千上万个需要共享文件系统的计算实例。Amazon EFS 支持 两种吞吐量模式：突增和预置。经历峰值访问模式的工作负载将受益于突增吞吐量模式，而一个持续较高的工作负载会在预置吞吐量模式下表现得很好。

f. Amazon FSx 基于全新 AWS 计算解决方案而构建，支持四种常用文件系统：NetApp ONTAP、OpenZFS、Windows 文件服务器和 Lustre。Amazon FSx 延迟、吞吐量和 IOPS 因文件系统而不同，因此，在为您的工作负载需求选择合适的文件系统时应考虑这些因素。

g. AWS Snow Family 是存储和计算设备，支持在线和离线数据迁移到云端，以及本地数据存储和计算。AWS Snow 设备支持收集大量本地数据，对数据进行处理，并将数据迁移到云端。在文件数量、文件大小和压缩方面，有几种 记录在案的性能最佳实践 。

h. AWS Storage Gateway 为本地应用程序提供对基于云的存储的访问。AWS Storage Gateway 支持多种云存储服务，包括 Amazon S3、Amazon S3 Glacier、Amazon FSx 和 Amazon EBS。它

支持多种协议，如 iSCSI、SMB 和 NFS。它通过在本地缓存频繁访问的数据来提供低延迟性能，并且只向 AWS 发送更改的数据和压缩数据。

3. 在试用新的存储解决方案并确定最佳配置后，规划迁移并验证性能指标。这是一个持续的过程，当主要特征更改或者可用服务或选项更改时，应重新对该过程进行评估。

实施计划的工作量级别： 如果工作负载从一种存储解决方案转移到另一种存储解决方案，则重构应用程序可能需要 适中 工作量。

资源

相关文档：

- Amazon EBS 卷类型
- Amazon EC2 存储
- Amazon EFS：Amazon EFS 性能
- Amazon FSx for Lustre 性能
- Amazon FSx for Windows File Server 性能
- Amazon FSx for NetApp ONTAP 性能
- Amazon FSx for OpenZFS 性能
- Amazon S3 Glacier：Amazon S3 Glacier 文档
- Amazon S3：请求速率和性能注意事项
- 使用 AWS 进行云存储
- AWS Snow Family
- EBS I/O 特征

相关视频：

- 深入讨论 Amazon EBS（STG303-R1）
- 利用 Amazon S3 优化存储性能（STG343）Amazon S3

相关示例：

- Amazon EFS CSI 驱动程序
- Amazon EBS CSI 驱动程序

- Amazon EFS 实用程序
- Amazon EBS 自动扩展
- Amazon S3 示例
- Amazon FSx for Lustre Container Storage Interface（CSI）驱动程序

PERF03-BP02 评估可用的配置选项

评估各种特性和配置选项以及它们与存储的关系。了解在何处以及如何使用预置 IOPS、SSD、磁性存储、对象存储、存档存储或短暂存储来针对工作负载优化存储空间和性能。

Amazon EBS 提供了一系列选项，让您能够优化存储性能和工作负载成本。这些选项分为两大类：用于事务型工作负载、由 SSD 提供支持的存储，例如数据库和启动卷（性能主要取决于 IOPS）；用于吞吐量密集型工作负载、由 HDD 提供支持的存储，例如 MapReduce 和日志处理（性能主要取决于传输速度）。

SSD 支持的卷包括：具有最高性能的预调配 IOPS SSD 卷，适用于有低延迟要求的事务型工作负载；通用型 SSD 卷，可以针对各种事务数据实现价格和性能的平衡。

Amazon S3 transfer acceleration 可以在您的客户端与 S3 存储桶之间实现快速的远距离文件传输。Transfer Acceleration 利用 Amazon CloudFront 遍布全球的边缘站点，通过优化的网络路径来路由数据。对于 S3 存储桶中具有密集 GET 请求的工作负载，可结合使用 Amazon S3 与 CloudFront。上传大型文件时，使用分段上传同时上传多个部分，以便尽可能提高网络吞吐量。

Amazon Elastic File System（Amazon EFS） 提供了一个简单、可扩展、完全托管式弹性 NFS 文件系统，可配合 AWS Cloud 服务和本地资源使用。为了支持各种云存储工作负载，Amazon EFS 提供了两种性能模式：通用性能模式和最大 I/O 性能模式。对于文件系统，还有两种吞吐量模式可供选择：突增吞吐量和预置吞吐量。要确定对工作负载使用哪种设置，请参阅 Amazon EFS 用户指南。

Amazon FSx 提供了四个文件系统供您选择： Amazon FSx for Windows File Server （适合于企业工作负载）、 Amazon FSx for Lustre （适合于高性能工作负载）、 Amazon FSx for NetApp ONTAP （适合于 NetApp 流行的 ONTAP 文件系统），以及 Amazon FSx for OpenZFS （适合于基于 Linux 的文件服务器）。FSx 由 SSD 提供支持，旨在提供快速、可预测、可扩展且稳定的性能。Amazon FSx 文件系统提供持续的高读写速度和稳定的低延迟数据访问。您可以选择所需的吞吐量级别来满足工作负载需求。

常见反模式：

- 对于所有工作负载，您都只使用一种存储类型，例如 Amazon EBS。
- 您对所有工作负载都使用预调配 IOPS，而没有对所有存储层进行真实测试。

- 您可以假设所有工作负载都具有相似的存储访问性能要求。

建立此最佳实践的好处： 评估所有存储服务选项可以降低基础设施的成本和维护您的工作负载所需的工作量。这样可能会缩短您的上市时间，从而部署新服务和功能。

未建立这种最佳实践的情况下暴露的风险等级： 中

实施指导

确定存储特征：评估存储解决方案时，请确定需要哪些存储特征，例如共享能力、文件大小、缓存大小、延迟、吞吐量和数据持久性。然后，使用最符合您的需求的 AWS 服务来满足您的要求。

资源

相关文档：

- 使用 AWS 进行云存储
- Amazon EBS 卷类型
- Amazon EC2 存储
- Amazon EFS：Amazon EFS 性能
- Amazon FSx for Lustre 性能
- Amazon FSx for Windows File Server 性能
- Amazon Glacier：Amazon Glacier 文档
- Amazon S3：请求速率和性能注意事项
- 使用 AWS 进行云存储
- 使用 AWS 进行云存储
- EBS I/O 特征

相关视频：

- 深入讨论 Amazon EBS (STG303-R1)
- 利用 Amazon S3 优化存储性能 (STG343)Amazon S3

相关示例：

- Amazon EFS CSI 驱动程序

- Amazon EBS CSI 驱动程序
- Amazon EFS 实用程序
- Amazon EBS 自动扩展
- Amazon S3 示例

PERF03-BP03 根据访问模式和指标做出决策

根据工作负载的访问模式选择存储系统，并通过确定工作负载访问数据的方式对其进行配置。通过选择对象存储而不是数据块存储来提高存储效率。按照您的数据访问模式，配置您选择的存储选项。

访问数据的方式将影响存储解决方案的效果。选择最适合您的访问模式的存储解决方案，或者考虑根据存储解决方案更改访问模式，以便尽可能提高性能。

通过创建 RAID 0 阵列，与在单个卷上进行预置相比，您可以实现更高的文件系统性能。当 I/O 性能比容错能力更重要时，请考虑使用 RAID 0。例如，您可以将其用于已经单独设置了数据复制的常用数据库。

在工作负载使用的所有存储选项中，为您的工作负载选择合适的存储指标。当利用使用突增积分的文件系统时，创建警报，以便系统在您即将达到积分限额时通知您。您必须创建存储控制面板以显示工作负载存储的总体运行情况。

对于固定大小的存储系统（例如 Amazon EBS 或 Amazon FSx），请确保您正在监控使用的存储量与总体存储量大小之间的关系，如果可以请创建自动化流程，以便在达到阈值时增加存储大小

常见反模式：

- 如果客户没有提出意见，您可以认为存储性能足够高。
- 如果所有工作负载都位于一个存储层，您只应使用该存储层。

建立此最佳实践的好处： 要优化性能和提高资源利用率，您需要一个统一的运营视图、实时粒度数据和历史参考。您可以创建自动控制面板，使用粒度为一秒的数据来对您的数据执行指标计算，并生成对您的存储需求的运维和利用率见解。

未建立这种最佳实践的情况下暴露的风险等级： 低

实施指导

优化存储使用情况和访问模式：根据工作负载的访问模式和可用存储选项的特征选择存储系统。确定存储数据的最佳位置，确保在减少开销的同时满足您的要求。根据存储特性配置数据并与其进行交互时，使用性能优化和访问模式（例如卷条带化或将数据分区）。

为存储选项选择适当的指标：确保您为工作负载选择适当的存储指标。每个存储选项都提供了各种指标，用于跟踪您的工作负载随着时间的推移的性能情况。确保您在测量任何存储突增指标（例如，监控 Amazon EFS 的突增点数）。对于固定大小的存储系统（例如，Amazon Elastic Block Store 或 Amazon FSx），请确保您正在监控所使用的存储量与总存储大小。在可能的情况下创建自动化流程，以在快要达到阈值时增加存储大小。

监控指标：Amazon CloudWatch 可以收集架构中各种资源的指标。您也可以收集和发布自定义指标，用于显示业务指标或派生指标。使用 CloudWatch 或第三方解决方案来设置指示超出阈值的警报。

资源

相关文档：

- Amazon EBS 卷类型
- Amazon EC2 存储
- Amazon EFS：Amazon EFS 性能
- Amazon FSx for Lustre 性能
- Amazon FSx for Windows File Server 性能
- Amazon Glacier：Amazon Glacier 文档
- Amazon S3：请求速率和性能注意事项
- 使用 AWS 进行云存储
- EBS I/O 特征
- 使用 Amazon CloudWatch 监控和了解 Amazon EBS 性能

相关视频：

- 深入讨论 Amazon EBS (STG303-R1)
- 利用 Amazon S3 优化存储性能 (STG343)Amazon S3

相关示例：

- Amazon EFS CSI 驱动程序
- Amazon EBS CSI 驱动程序
- Amazon EFS 实用程序

- Amazon EBS 自动扩展
- Amazon S3 示例

PERF 4 如何选择数据库解决方案？

针对特定系统的最优数据库解决方案取决于您的具体需求，包括可用性、一致性、分区容错性、延迟、持久性、可扩展性以及查询能力等等。许多系统会使用多种不同的数据库解决方案满足其各子系统的实际需要，并启用不同的功能来提高性能。为系统选择错误的数据库解决方案和功能可能会导致性能效率降低。

最佳实践

- PERF04-BP01 了解数据特征
- PERF04-BP02 评估可用的选项
- PERF04-BP03 收集和记录数据库性能指标
- PERF04-BP04 根据访问模式选择数据存储
- PERF04-BP05 根据访问模式和指标优化数据存储

PERF04-BP01 了解数据特征

选择数据管理解决方案，以最佳地匹配工作负载数据集的特征、访问模式和要求。在选择和实施数据管理解决方案时，您必须确保查询、扩展和存储特征支持工作负载数据要求。了解各种数据库选项如何匹配您的数据模型，以及哪些配置选项最适合您的使用案例。

AWS 提供了多种数据库引擎，包括关系、键值、文档、内存、图形、时间序列和分类账数据库。每种数据管理解决方案都有可供您使用的选项和配置，以支持您的使用案例和数据模型。根据数据特征，您的工作负载也许能够使用多种不同的数据库解决方案。通过选择针对特定问题的最佳数据库解决方案，您可以摆脱整体式数据库的束缚（整体式数据库采用具有限制性的一刀切方法），专注于管理数据以满足客户的需求。

期望结果： 工作负载数据特征的记录足够详细，可以帮助选择和配置支持的数据库解决方案，并深入了解潜在的替代方案。

常见反模式：

- 没有考虑将大型数据集分割成具有相似特征的较小数据集合的方法，导致失去使用更符合数据和增长特征的专用数据库的机会。

- 没有预先识别数据访问模式，导致以后进行成本高昂且复杂的重复工作。
- 使用的数据存储策略无法按需求快速扩展，限制了增长
- 为所有工作负载选择一个数据库类型和供应商。
- 由于员工拥有某种特定类型的数据库解决方案的经验和知识，坚持使用该数据库解决方案。
- 保持一种数据库解决方案，因为它在本地环境中运行良好。

建立此最佳实践的好处： 熟悉所有的 AWS 数据库解决方案，以便为各种工作负载确定正确的数据库解决方案。为您的工作负载选择合适的数据库解决方案后，您可以快速试用每种数据库产品/服务，以确定它们是否继续满足您的工作负载需求。

未建立这种最佳实践的情况下暴露的风险等级： 高

- 可能无法确定潜在的成本节约机会。
- 数据的保护级别可能达不到要求。
- 数据访问和存储性能可能不是最佳的。

实施指导

定义工作负载的数据特征和访问模式。查看所有可用的数据库解决方案，以确定哪种解决方案支持您的数据需求。对于给定的工作负载，可以选择多个数据库。评估每个服务或每组服务，并单独进行评估。如果为部分或全部数据确定了潜在的替代数据管理解决方案，那么可以试用替代实施方案，以获得成本、安全性、性能和可靠性方面的好处。如采用新的数据管理方法，需要更新现有文档。

类型	AWS 服务	主要特征	常见使用案例
关系	Amazon RDS、Amazon Aurora	参照完整性、ACID 事务、写时模式	ERP、CRM、商用现货软件
键值	Amazon DynamoDB	高吞吐量、低延迟、近乎无限的可扩展性	购物车（电子商务）、产品目录、聊天应用程序
文档	Amazon DocumentDB	存储 JSON 文档并查询任何属性	内容管理（CMS）、客户资料、移动应用程序

类型	AWS 服务	主要特征	常见使用案例
内存	Amazon ElastiCache、Amazon MemoryDB	微秒级延迟	缓存、游戏排行榜
图形	Amazon Neptune	高度相关的数据，其数据之间的关系具有意义	社交网络、个性化引擎、欺诈检测
时间序列	Amazon Timestream	以时间为主维度的数据	DevOps、IoT、监控
宽列	Amazon Keyspaces	Cassandra 工作负载。	工业设备维护、路线优化
分类账	Amazon QLDB	不可变且可加密验证的变更分类账	记录系统、医疗保健、供应链、金融机构

实施步骤

1. 数据结构如何？（例如，非结构化、键值、半结构化、关系型）
   a. 如果数据是非结构化的，请考虑使用对象存储，例如 Amazon S3 或 NoSQL 数据库，如 Amazon DocumentDB。
   b. 对于键值数据，请考虑使用 DynamoDB、 ElastiCache for Redis 或者 MemoryDB。
   c. 如果数据具有关系结构，那么需要什么级别的参照完整性？
      i. 对于外键约束，关系数据库（如 Amazon RDS 和 Aurora ）可以提供这种级别的完整性。
      ii. 通常，在 NoSQL 数据模型中，您可以将数据去规范化到单个文档或文档集合，以便在单个请求中进行检索，而不是跨各文档或各表联接。
2. 是否要求符合 ACID（原子性、一致性、隔离性、持久性）？
   a. 如果需要与关系数据库关联的 ACID 属性，请考虑使用关系数据库，例如 Amazon RDS 和 Aurora。
3. 需要什么样的一致性模型？
   a. 如果您的应用程序可以容许最终一致性，请考虑使用 NoSQL 实施。查看其他特征，以帮助选择最合适的 NoSQL 数据库 。

   b. 如果需要强一致性，您可以使用 DynamoDB 强一致性读取，或者使用关系数据库，如 Amazon RDS。

4. 必须支持哪些查询和结果格式？（例如，SQL、CSV、Parque、Avro、JSON 等）
5. 存在哪些数据类型、字段大小和总体数量？（例如，文本、数字、空间、时间序列计算、二进制或 BLOB、文档）
6. 存储需求将如何随时间变化？ 这对可扩展性有何影响？
   a. 无服务器数据库（如 DynamoDB 和 Amazon Quantum Ledger Database ）将动态扩展至近乎无限的存储空间。
   b. 关系数据库的预置存储空间设有上限，一旦达到这些限制，通常必须通过分片等机制进行水平分区。
7. 读查询与写查询的比例是多少？ 缓存有可能提高性能吗？
   a. 包含大量读操作的工作负载可以受益于缓存层，如 ElastiCache 或者 DAX （如果数据库是 DynamoDB）。
   b. 读操作也可以通过关系数据库（如 Amazon RDS）分流到只读副本上。
8. 存储和修改（OLTP – Online Transaction Processing，联机事务处理）还是检索和报告（OLAP – Online Analytical Processing，联机分析处理）具有更高的优先级？
   a. 对于高吞吐量事务处理，请考虑使用 NoSQL 数据库，如 DynamoDB 或 Amazon DocumentDB。
   b. 对于分析查询，请考虑使用列存数据库（如 Amazon Redshift ），或者将数据导出到 Amazon S3 并使用 Athena 或者 QuickSight 执行分析。
9. 这些数据有多敏感，需要什么级别的保护和加密？
   a. 所有的 Amazon RDS 和 Aurora 引擎都支持使用 AWS KMS 进行静态数据加密。Microsoft SQL Server 和 Oracle 在使用 Amazon RDS 时也支持本机透明数据加密（TDE，Transparent Data Encryption）。
   b. 对于 DynamoDB，您可以使用 IAM 的精细访问控制功能，在关键字级别控制谁可以访问哪些数据。
10. 数据需要什么级别的持久性？
   a. Aurora 自动在一个区域内的三个可用区复制您的数据，这意味着您的数据具有高度的持久性，数据丢失的可能性较小。
   b. DynamoDB 自动跨多个可用区复制，提供高可用性和数据持久性。
   c. Amazon S3 提供 11 个 9 的持久性。许多数据库服务（如 Amazon RDS 和 DynamoDB）支持将数据导出到 Amazon S3 以进行长期保留和归档。

11.恢复 时间目标（RTO）或恢复点目标（RPO） 要求是否影响解决方案？

a. Amazon RDS、Aurora、DynamoDB、Amazon DocumentDB 和 Neptune 全部支持时间点恢复以及按需备份和还原。

b. 对于高可用性要求，可以全局复制 DynamoDB 表（使用 全局表 功能），并且可以使用全局数据库功能跨多个区域复制 Aurora 集群。此外，可以使用跨区域复制功能，跨 AWS 区域复制 S3 存储桶。

12.是否希望摆脱商用数据库引擎/许可成本？

a. 考虑使用 Amazon RDS 或 Aurora 上的开源引擎，如 PostgreSQL 和 MySQL

b. 利用 AWS DMS 和 AWS SCT 执行从商用数据库引擎到开源引擎的迁移

13.对数据库的运维有什么期望？ 迁移到托管服务是主要的关注点吗？

a. 利用 Amazon RDS 而不是 Amazon EC2，以及利用 DynamoDB 或 Amazon DocumentDB 而不是自行托管的 NoSQL 数据库可以减少运维开销。

14.当前如何访问数据库？ 是只有应用程序访问，还是有商业智能（BI，Business Intelligence）用户和其他互联的现成应用程序？

a. 如果您依赖于外部工具，那么您可能必须保持与它们支持的数据库的兼容性。Amazon RDS 完全兼容其支持的不同引擎版本，包括 Microsoft SQL Server、Oracle、MySQL 和 PostgreSQL。

15.下面列出了潜在的数据管理服务，以及这些服务的最佳使用位置：

a. 关系数据库通过预定义 schema 及其之间的关系存储数据。这些数据库旨在支持 ACID（原子性、一致性、隔离性、持久性）事务，并保持参照完整性和数据强一致性。许多传统应用程序、企业资源规划（ERP, enterprise resource planning）、客户关系管理（CRM, customer relationship management）和电子商务都使用关系数据库来存储其数据。您可以在 Amazon EC2 上运行许多这些数据库引擎，或者从以下 AWS 托管数据库服务中进行选择： Amazon Aurora， Amazon RDS和 Amazon Redshift.

b. 键值数据库已针对常见的访问模式进行优化，通常用于存储和检索大量数据。这些数据库即使在出现大量并发请求的情况下也能实现快速响应。键值数据库的典型使用案例包括高流量 Web 应用程序，电子商务系统和游戏应用程序。在 AWS 中，您可以利用 Amazon DynamoDB数据库，这是一个完全托管的多区域、多主表持久数据库，具有适用于互联网规模的应用程序的内置安全性、备份和还原以及内存中的缓存。

c. 内存数据库用于需要实时访问数据、最低延迟和最高吞吐量的应用程序。对于毫秒级延迟不足以满足需求的应用程序，这些数据库通过直接将数据存储在内存中来提供微秒级延迟。您可以将内存数据库用于应用程序缓存、会话管理、游戏排行榜和地理空间应用程序。 Amazon ElastiCache 是一种完全托管的内存数据存储，兼容 Redis 或者 Memcached。如果应用程序还有更高的持久性要求，可以结合 适用于 Redis 的 Amazon MemoryDB 来提供持久的内存数据库服务，以实现超快的性能。

d. 文档数据库旨在将半结构化数据存储为类似 JSON 的文档。这些数据库可帮助开发人员快速构建和更新应用程序，例如内容管理、目录和用户配置文件。 Amazon DocumentDB 是一种快速、可扩展、高度可用且完全托管的文档数据库服务，支持 MongoDB 工作负载。

e. 宽列存储是 NoSQL 数据库的一种类型。它使用表、行和列，但是与关系数据库不同的是，同一个表中各行的列名称和格式可能会有所不同。您通常会看到一个宽列存储在大规模工业应用程序中，用于设备维护、队列管理和路线优化。 Amazon Keyspaces（Apache Cassandra 兼容） 是一种宽列可扩展、高度可用且兼容 Apache Cassandra 的托管数据库服务。

f. 图形数据库适用于需要大规模以毫秒延迟在高度连接的图形数据集之间浏览和查询数百万关系的应用程序。许多公司将图形数据库用于欺诈检测、社交网络和推荐引擎。 Amazon Neptune 是一种快速、可靠、完全托管的图数据库服务，便于用户能轻松构建并运行适用于高度互连数据集的应用程序。

g. 时间序列数据库可以高效收集、合成数据，并从不断变化的数据中获得见解。IoT 应用程序、开发运营和工业遥测可以利用时间序列数据库。 Amazon Timestream 是适用于 IoT 和运营应用程序的快速、可扩展、完全托管的时间序列数据库服务，可用于轻松存储和分析每天数以万亿计的事件。

h. 分类账数据库提供可信中央机构，以维护每个应用程序的可扩展、不可变和允许以加密方式进行验证的交易记录。我们看到分类账数据库用于记录系统、供应链、注册甚至银行交易。 Amazon Quantum Ledger Database (Amazon QLDB) 是一种完全托管的分类账数据库，提供可信中央机构拥有的透明、不可变和允许以加密方式进行验证的交易日志。Amazon QLDB 跟踪每个应用程序数据更改，并持续维护完整且可验证的更改历史记录。

实施计划的工作量级别： 如果工作负载从一种数据库解决方案转移到另一种计算解决方案，则重构数据和应用程序可能需要 高 工作量。

资源

相关文档：

- AWS 云数据库
- AWS 数据库缓存
- Amazon DynamoDB Accelerator
- Amazon Aurora 最佳实践
- Amazon Redshift 性能
- Amazon Athena 10 大性能提示
- Amazon Redshift Spectrum 最佳实践

- Amazon DynamoDB 最佳实践
- 在 EC2 和 Amazon RDS 之间进行选择
- 实施 Amazon ElastiCache 的最佳实践

相关视频：

- AWS 专用数据库（DAT209-L）
- Amazon Aurora 存储揭秘：工作原理（DAT309-R）
- Amazon DynamoDB 深入研究：高级设计模式 (DAT403-R1)

相关示例：

- 使用 Amazon Redshift 数据共享优化数据模式
- 数据库迁移
- MS SQL Server – AWS Database Migration Service（DMS）复制演示
- 数据库现代化动手实践研讨会
- Amazon Neptune 示例

PERF04-BP02 评估可用的选项

在选择数据管理解决方案之前，需要了解可用的数据库选项及其如何优化性能。使用负载测试确定与您的工作负载相关的重要数据库指标。在研究数据库选项时，要考虑各种方面，如参数组、存储选项、内存、计算、只读副本、最终一致性、连接池和缓存选项。尝试使用这些不同的配置选项来改进指标。

期望结果： 基于数据类型，工作负载可以使用一个或多个数据库解决方案。数据库功能和优势与数据特征、访问模式和工作负载要求完美匹配。要优化您的数据库性能和成本，您必须评估数据访问模式以确定适当的数据库选项。评估可接受的查询时间，以确保选定的数据库选项可以满足要求。

常见反模式：

- 未识别数据访问模式。
- 不了解所选数据管理解决方案的配置选项。
- 仅依赖于增加实例大小，而不考虑其他可用的配置选项。
- 不测试所选解决方案的扩展特征。

建立此最佳实践的好处： 通过探索和试用数据库选项，您也许能够降低基础设施成本，提高性能和可扩展性，并减少维护工作负载所需的工作量。

未建立这种最佳实践的情况下暴露的风险等级： 高

- 必须针对 一刀切类型的 数据库进行优化意味着做出不必要的妥协。
- 由于没有配置数据库解决方案以匹配流量模式，导致成本增加。
- 扩展问题可能会导致运维问题。
- 数据的保护级别可能达不到要求。

实施指导

了解您的工作负载数据特征，以便配置数据库选项。运行负载测试以确定您的关键性能指标和瓶颈。使用这些特征和指标来评估数据库选项并尝试使用不同的配置。

AWS 服务	Amazon RDS、Amazon Aurora	Amazon DynamoDB	Amazon DocumentDB	Amazon ElastiCache	Amazon Neptune	Amazon Timestream	Amazon Keyspaces	Amazon QLDB
扩展计算	增加实例大小，Aurora 无服务器实例自动扩展以响应负载变化	按需容量模式下的自动读/写扩展，或预置容量模式下的预置读/写容量自动扩展	增加实例大小	增加实例大小，将节点添加到集群	增加实例大小	自动扩展以调整容量	按需容量模式下的自动读/写扩展，或预置容量模式下的预置读/写容量自动扩展	自动扩展以调整容量
横向扩展读取	所有引擎都支持只读副	增加预置的读取容量单位	只读副本	只读副本	只读副本。支持只读副本实	自动扩展	增加预置的读取容量单位	自动纵向扩展到规定

AWS 服务	Amazon RDS、Amazon Aurora	Amazon DynamoDB	Amazon DocumentDB	Amazon ElastiCache	Amazon Neptune	Amazon Timestream	Amazon Keyspaces	Amazon QLDB
	本。Aurora 支持只读副本实例的自动扩展				例的自动扩展			的并发限制
横向扩展写操作	增加实例大小，批处理应用程序中的写操作，或在数据库前面添加队列。通过跨多个实例的应用程序级分片进行横向扩展	增加预置的写入容量单位。确保最佳分区键，以防止分区级写操作节流	增加主实例大小	在集群模式下使用 Redis 跨分片分布写操作	增加实例大小	扩展时，写请求可能会受到限制。如果遇到节流异常，请继续以相同（或更高）吞吐量发送数据，以自动扩展。批量写入以减少并发写入请求	增加预置的写入容量单位。确保最佳分区键，以防止分区级写操作节流	自动纵向扩展到规定的并发限制
引擎配置	参数组	不适用	参数组	参数组	参数组	不适用	不适用	不适用

AWS 服务	Amazon RDS、Amazon Aurora	Amazon DynamoDB	Amazon DocumentDB	Amazon ElastiCache	Amazon Neptune	Amazon Timestream	Amazon Keyspaces	Amazon QLDB
缓存	内存中的缓存，可通过参数组进行配置。与 ElastiCache for Redis 等专用缓存结合使用，分流对经常访问项的请求	DAX 完全托管式缓存可用	内存中的缓存。（可选）与 ElastiCache for Redis 等专用缓存结合使用，分流对经常访问项的请求	主要功能是缓存	使用查询结果缓存来缓存只读查询的结果	Timestream 有两个存储层；其中之一是高性能内存中存储层	部署单独的专用缓存（如 ElastiCache for Redis），分流对经常访问项的请求	不适用

AWS 服务	Amazon RDS、Amazon Aurora	Amazon DynamoDB	Amazon DocumentDB	Amazon ElastiCache	Amazon Neptune	Amazon Timestream	Amazon Keyspaces	Amazon QLDB
高可用性/灾难恢复	对于生产工作负载，推荐的配置是在第二个可用区中运行备用实例，以在一个区域内提供弹性。 对于跨区域的弹性，可以使用 Aurora 全球数据库	在一个区域内高度可用。可以使用 DynamoDB 全局表跨区域复制表	跨可用区创建多个实例以实现可用性。 快照可以跨区域共享，集群可以使用 DMS 进行复制，用于提供跨区域复制/灾难恢复	对于生产集群，推荐的配置是在备用可用区中至少创建一个节点。 ElastiCache 全局数据存储可用于跨区域复制集群。	其他可用区中的只读副本用作失效转移目标。 快照可以跨区域共享，集群可以使用 Neptune 流进行复制，用于在两个不同区域的两个集群之间复制数据。	在一个区域内高度可用。跨区域复制需要使用 Timestream SDK 进行自定义应用程序开发	在一个区域内高度可用。 跨区域复制需要自定义应用程序逻辑或第三方工具	在一个区域内高度可用。 要跨区域复制，请将 Amazon QLDB 日志的内容导出到 S3 存储桶，并配置该存储桶以进行跨区域复制。

实施步骤

1. 哪些配置选项可用于选定的数据库？
   a. 利用 Amazon RDS 和 Aurora 的参数组，您可以调整常见的数据库引擎级别设置（例如为缓存分配的内存），或调整数据库的时区
   b. 对于预置的数据库服务（如 Amazon RDS、Aurora、Neptune、Amazon DocumentDB）以及在 Amazon EC2 上部署的数据库服务，您可以更改实例类型、预置存储和添加只读副本。

c. DynamoDB 允许您指定两种容量模式：按需和预置。考虑到不同的工作负载，您可以在这两种模式之间进行更改，并在预置模式下随时增加所分配的容量。

2. 工作负载是否包含大量的读取或写入操作？

   a. 哪些解决方案可用于分流读取操作（只读副本、缓存等）？

      i. 对于 DynamoDB 表，您可以使用 DAX 缓存功能来分流读取操作。

      ii. 对于关系数据库，您可以创建一个 ElastiCache for Redis 集群，并将应用程序配置为首先从缓存中读取，并在请求的项目不存在时返回到数据库。

      iii. 关系数据库（如 Amazon RDS 和 Aurora）以及预置的 NoSQL 数据库（如 Neptune 和 Amazon DocumentDB）全部支持添加只读副本，以分流工作负载的读取部分。

      iv. DynamoDB 等无服务器数据库将自动扩展。确保您预置了足够的读取容量单位（RCU，Read Capacity Unit）来处理工作负载。

   b. 哪些解决方案可用于扩展写入操作（分区键分片、引入队列等）？

      i. 对于关系数据库，您可以增加实例的大小以适应增加的工作负载，或增加预调配 IOPS 以增加底层存储的吞吐量。

         - 您还可以在数据库前面引入队列，而不是直接写入数据库。此模式允许您将摄取操作与数据库解耦，并控制流量，这样数据库就不会过载。
         - 对写入请求进行批处理，而不是创建许多短期事务，这样有助于提高有大量写入的关系数据库的吞吐量。

      ii. 像 DynamoDB 这样的无服务器数据库可以自动扩展写入吞吐量，也可以根据容量模式调整预置的写入容量单位（WCU，Write Capacity Unit）。

         - 但是，当达到给定分区键的吞吐量限制时，仍然会遇到 热 分区问题。这可以通过选择更均匀分布的分区键或对分区键进行写分片来缓解。

3. 当前或预期的每秒事务数（TPS）峰值是多少？ 使用此流量和此流量 +X% 进行测试，以了解扩展特征。

   a. 适用于 PostgreSQL 的 pg_bench 等原生工具可用于对数据库进行压力测试，以了解瓶颈和扩展特征。

   b. 应该捕获类似生产的流量，以便重放这些流量，从而在合成工作负载之外模拟真实世界的情况。

4. 如果使用无服务器或弹性可扩展计算，请测试此扩展对数据库的影响。如果合适，引入连接管理或池技术以降低对数据库的影响。

   a. RDS 代理可与 Amazon RDS 和 Aurora 结合使用，以管理与数据库的连接。

   b. DynamoDB 等无服务器数据库没有与之关联的连接，但会考虑预置容量和自动扩展策略来处理负载峰值。

5. 负载是否可预测，是否会出现负载峰值和不活动时段？
    a. 如果有一段时间处于不活动状态，请考虑在这段时间内缩减预置的容量或实例大小。Aurora Serverless V2 将根据负载自动纵向扩展和缩减。
    b. 对于非生产实例，请考虑在非工作时间暂停或停止这些实例。
6. 您是否需要根据访问模式和数据特征对数据模型进行分段和拆分？
    a. 考虑使用 AWS DMS 或 AWS SCT 将您的数据移动到其他服务。

实施计划的工作量级别：

要建立此最佳实践，您必须了解当前的数据特征和指标。收集这些指标，建立基线，然后使用这些指标来确定理想的数据库配置选项，这需要 低 到 中等 工作量。这最好通过负载测试和实验来验证。

资源

相关文档：

- AWS 云数据库
- AWS 数据库缓存
- Amazon DynamoDB Accelerator
- Amazon Aurora 最佳实践
- Amazon Redshift 性能
- Amazon Athena 10 大性能提示
- Amazon Redshift Spectrum 最佳实践
- Amazon DynamoDB 最佳实践

相关视频：

- AWS 专用数据库（DAT209-L）
- Amazon Aurora 存储揭秘：工作原理（DAT309-R）
- Amazon DynamoDB 深入研究：高级设计模式 (DAT403-R1)

相关示例：

- Amazon DynamoDB 示例

- AWS 数据库迁移示例
- 数据库现代化研讨会
- 使用 Amazon RDS for Postgress DB 上的参数

PERF04-BP03 收集和记录数据库性能指标

要了解数据管理系统的运行情况，跟踪相关指标非常重要。这些指标将帮助您优化数据管理资源，确保满足您的工作负载需求，并确保您清楚地了解工作负载的运行情况。使用各种工具、库和系统来记录与数据库性能相关的性能测量值。

有些指标与数据库所在的系统有关（例如，CPU、存储、内存、IOPS），有些指标与访问数据本身有关（例如，每秒事务数、查询速率、响应时间、错误）。这些指标应便于任何支持或操作人员访问，并具有足够的历史记录，以便能够识别趋势、异常和瓶颈。

期望结果： 为了监控数据库工作负载的性能，您必须记录一段时间内的多个性能指标。这样您便可以检测异常并根据业务指标衡量性能，确保满足您的工作负载需求。

常见反模式：

- 您只能手动搜索日志文件来查找指标。
- 您只将指标发布到团队使用的内部工具，而没有全面了解您的工作负载。
- 您只使用所选监控软件记录的默认指标。
- 您只在出现问题时检查指标。
- 您只监控系统级指标，而不捕获数据访问或使用情况指标。

建立此最佳实践的好处： 建立性能基准有助于了解工作负载的正常行为和需求。可以更快地识别和调试异常模式，从而提高数据库的性能和可靠性。可以配置数据库容量，以确保在不影响性能的情况下实现最佳成本。

未建立这种最佳实践的情况下暴露的风险等级： 高

- 无法区分异常与正常的性能水平会给问题识别和决策带来困难。
- 可能无法确定潜在的成本节约机会。
- 无法识别增长模式，这可能导致可靠性或性能下降。

实施指导

识别、收集、聚合和关联与数据库相关的指标。指标应包括支持数据库的底层系统指标和数据库指标。底层系统指标可包括 CPU 利用率、内存、可用磁盘存储、磁盘 I/O 和网络入站和出站指标，而数据库指标可包括每秒事务数、最多的查询、平均查询速率、响应时间、索引使用情况、表锁定、查询超时和打开的连接数。这些数据对于了解工作负载的性能以及数据库解决方案的使用方式至关重要。将这些指标用作数据驱动方法的一部分，以便调整和优化工作负载的资源。

实施步骤：

1. 必须跟踪哪些数据库指标？
    a. 监控 Amazon RDS 的指标
    b. 使用 Performance Insights 进行监控
    c. 增强监控
    d. DynamoDB 指标
    e. 监控 DynamoDB DAX
    f. 监控 MemoryDB
    g. 监控 Amazon Redshift
    h. 时间序列指标和维度
    i. Aurora 的集群级指标
    j. 监控 Amazon Keyspaces
    k. 监控 Amazon Neptune
2. 数据库监控是否会受益于检测操作异常和性能问题的机器学习解决方案？
    a. Amazon DevOps Guru for Amazon RDS 会显示性能问题，并提出纠正措施的建议。
3. 您是否需要有关 SQL 使用情况的应用程序级详细信息？
    a. AWS X-Ray 可以签入到应用程序中以获得见解，并为单个查询封装所有数据点。
4. 您目前是否有经过批准的日志记录和监控解决方案？
    a. Amazon CloudWatch 可以收集架构中各种资源的指标。您也可以收集和发布自定义指标，用于显示业务指标或派生指标。使用 CloudWatch 或第三方解决方案来设置指示超出阈值的警报。
5. 您是否确定并配置了数据留存策略以匹配我的安全和运营目标？
    a. CloudWatch 指标的默认数据留存
    b. CloudWatch Logs 的默认数据留存

实施计划的工作量级别： 从所有数据库资源中识别、跟踪、收集、聚合和关联指标所需的工作量为 中 。

资源

相关文档：

- AWS 数据库缓存
- Amazon Athena 10 大性能提示
- Amazon Aurora 最佳实践
- Amazon DynamoDB Accelerator
- Amazon DynamoDB 最佳实践
- Amazon Redshift Spectrum 最佳实践
- Amazon Redshift 性能
- AWS 云数据库
- Amazon RDS Performance Insights

相关视频：

- AWS 专用数据库（DAT209-L）
- Amazon Aurora 存储揭秘：工作原理（DAT309-R）
- Amazon DynamoDB 深入研究：高级设计模式 (DAT403-R1)

相关示例：

- 第 100 级：使用 CloudWatch 控制面板进行监控
- AWS 数据集摄取指标收集框架
- Amazon RDS 监控研讨会

PERF04-BP04 根据访问模式选择数据存储

根据工作负载的访问模式来确定要使用的服务和技术。在性能和规模等非功能性要求外，访问模式还会很大程度影响数据库和存储解决方案的选择。第一个方面是对事务、ACID 合规性和一致性读取的需求。并非每个数据库都支持这些需求，大多数 NoSQL 数据库都提供最终一致性模型。第二个重要方面

是写入和读取操作在时间和空间上的分布。全球分布式应用程序需要考虑流量模式、延迟和访问要求，以便确定最佳存储解决方案。第三个需要选择的关键方面是查询模式灵活性、随机访问模式和一次性查询。还必须考虑针对文本和自然语言处理、时间序列和图形的高度专业化查询功能。

期望结果： 根据已识别和记录的数据访问模式选择数据存储。这可包括最常见的读取、写入和删除查询，对临时计算和聚合的需求，数据的复杂性，数据的相互依赖关系以及所要求的一致性需求。

常见反模式：

- 您只能选择一个数据库供应商来简化运营管理。
- 您可以假设数据访问模式会随着时间的推移保持一致。
- 您在应用程序中实施复杂的事务、回滚和一致性逻辑。
- 数据库配置为支持可能出现的高流量突增，这导致数据库资源大部分时间保持空闲状态。
- 使用共享数据库进行事务处理和分析。

建立此最佳实践的好处： 基于访问模式选择和优化数据存储将有助于降低开发复杂性并优化性能。了解何时使用只读副本、全局表、数据分区和缓存将帮助您减少运维开销，并根据您的工作负载需求进行扩展。

未建立这种最佳实践的情况下暴露的风险等级： 中

实施指导

识别和评估数据访问模式，以选择正确的存储配置。每个数据库解决方案都有配置和优化存储解决方案的选项。使用收集的指标和日志，并尝试使用各种选项以找到最佳配置。使用下表查看每个数据库服务的存储选项。

AWS 服务	Amazon RDS、Amazon Aurora	Amazon DynamoDB	Amazon DocumentDB	Amazon ElastiCache	Amazon Neptune	Amazon Timestream	Amazon Keyspaces	Amazon QLDB
扩展存储	当利用预调配 IOPS 存储类型时，用	自动扩展。表的大小不受限制。	存储自动扩展选项可用于扩	存储在内存中，绑定到实	存储自动扩展选项可用于自动扩展	配置内存层和磁介质层的保留	自动扩展和缩减表存储	自动扩展。表的大小不受限制。

AWS 服务	Amazon RDS、An Aurora	Amazon DynamoD	Amazon Document B	Amazon ElastiCache	Amazon Neptune	Amazon Timestrea m	Amazon Keyspace	Amazon QLDB
	于自动扩展预调配存储 IOPS 的存储自动扩展选项，也可以独立于预调配的存储进行扩展		展预置存储	例类型或计数	预置存储	期（以天为单位）		

实施步骤：

1. 确定并记录数据和流量的预期增长。
   a. Amazon RDS 和 Aurora 支持存储自动扩展到规定的限制。除此之外，可以考虑将旧数据转移到 Amazon S3 进行归档，聚合历史数据进行分析，或通过分片进行横向扩展。
   b. DynamoDB 和 Amazon S3 将自动扩展到接近无限的存储量。
   c. 在 EC2 上运行的 Amazon RDS 实例和数据库的大小可以手动调整，并且 EC2 实例可以在以后添加新的 EBS 卷以增加存储空间。
   d. 实例类型可以根据活动的变化而改变。例如，您可以在测试时从较小的实例开始，然后在服务开始接收生产流量时扩展实例。Aurora Serverless V2 缩放以响应负载的变化。

1. 记录有关正常和峰值下的性能（每秒事务数 TPS 和每秒查询数 QPS）及一致性（ACID 和最终一致性）要求。
2. 记录解决方案部署方面和数据库访问要求（全局、多可用区、读取复制、多个写入节点）

实施计划的工作量级别： 如果您未记录数据管理解决方案的日志或指标，那么您需要在识别和记录数据访问模式之前完成这项工作。一旦了解了数据访问模式，选择和配置数据存储的工作量就会比较 低 工作量。

资源

相关文档：

- AWS 数据库缓存
- Amazon Athena 10 大性能提示
- Amazon Aurora 最佳实践
- Amazon DynamoDB Accelerator
- Amazon DynamoDB 最佳实践
- Amazon Redshift Spectrum 最佳实践
- Amazon Redshift 性能
- AWS 云数据库
- Amazon RDS 存储类型

相关视频：

- AWS 专用数据库（DAT209-L）
- Amazon Aurora 存储揭秘：工作原理（DAT309-R）
- Amazon DynamoDB 深入研究：高级设计模式 (DAT403-R1)

相关示例：

- 使用 AWS 分布式负载测试进行试验和测试

PERF04-BP05 根据访问模式和指标优化数据存储

使用性能特性和访问模式来优化数据的存储和查询方式，以便实现最佳性能。衡量索引、键分配、数据仓库设计或缓存策略等优化对系统性能或整体效率的影响。

常见反模式：

- 您只能手动搜索日志文件来查找指标。

- 您只能将指标发布到内部工具。

建立此最佳实践的好处： 为了确保满足工作负载的指标要求，您必须监控与读写操作相关的数据库性能指标。您可以根据这些数据向数据存储层添加新的读写优化功能。

未建立这种最佳实践的情况下暴露的风险等级： 低

实施指导

根据指标和模式优化数据存储：使用报告的指标来识别您的工作负载中任何性能欠佳的方面，并优化您的数据库组件。对于每个数据库系统，您都需要评估不同的性能相关特性，例如为数据建立索引的方式、缓存数据的方式，以及在多个系统中分配数据的方式。衡量优化所带来的影响。

资源

相关文档：

- AWS 数据库缓存
- Amazon Athena 10 大性能提示
- Amazon Aurora 最佳实践
- Amazon DynamoDB Accelerator
- Amazon DynamoDB 最佳实践
- Amazon Redshift Spectrum 最佳实践
- Amazon Redshift 性能
- AWS 云数据库
- 使用 DevOps Guru for RDS 分析性能异常
- DynamoDB 的读/写容量模式

相关视频：

- AWS 专用数据库（DAT209-L）
- Amazon Aurora 存储揭秘：工作原理（DAT309-R）
- Amazon DynamoDB 深入研究：高级设计模式 (DAT403-R1)

相关示例：

- Amazon DynamoDB 动手实验

PERF 5 如何配置联网解决方案？

适合某个工作负载的最佳网络解决方案会因延迟、吞吐量要求、抖动和带宽而有所不同。物理限制（例如用户资源或本地资源）决定位置选项。这些限制可以通过边缘站点或资源置放来抵消。

最佳实践

- PERF05-BP01 了解联网对性能的影响
- PERF05-BP02 评估可用的联网功能
- PERF05-BP03 为混合工作负载选择适当大小的专用连接或 VPN
- PERF05-BP04 利用负载均衡和加密卸载
- PERF05-BP05 选择网络协议以提高性能
- PERF05-BP06 根据网络要求选择工作负载的位置
- PERF05-BP07 根据各项指标优化网络配置

PERF05-BP01 了解联网对性能的影响

分析并了解与网络相关的决策对工作负载性能的影响。网络负责应用程序组件、云服务、边缘网络和本地数据之间的连接，因此，它会极大地影响工作负载性能。除了工作负载性能之外，用户体验还受网络延迟、带宽、协议、位置、网络拥塞、抖动、吞吐量和路由规则的影响。

期望结果： 清楚记录工作负载的联网要求列表，包括延迟、数据包大小、路由规则、协议和支持的流量模式。查看可用的联网解决方案，并确定哪种服务与您的工作负载联网特性相符。基于云的网络可以快速重建，因此有必要随着时间的推移改进网络架构，以提高性能效率。

常见反模式：

- 所有流量都会流经您现有的数据中心。
- 您不了解实际使用情况要求，建立了过多的 Direct Connect 会话。
- 在确立联网解决方案时，您未考虑工作负载特性和加密开销。
- 您将本地概念和策略用于云中的联网解决方案。

建立此最佳实践的好处： 通过了解联网如何影响工作负载性能，可帮助您识别潜在的瓶颈、改善用户体验、提高可靠性并在工作负载发生变化时减少运营维护。

未建立这种最佳实践的情况下暴露的风险等级： 高

实施指导

确定工作负载的重要网络性能指标并捕获其联网特性。使用基准测试或负载测试，在数据驱动的方法中定义和记录需求。使用此数据确定网络解决方案受限的方面，并查看可以改进工作负载的配置选项。从需求出发，了解可用的云原生联网功能和选项，以及它们如何影响工作负载性能。每项联网功能均有优缺点，可以根据您的需求配置此功能，从而匹配工作负载特性和规模。

实施步骤：

1. 定义和记录联网性能需求：
    a. 包括网络延迟、带宽、协议、位置、流量模式（峰值和频率）、吞吐量、加密、检查和路由规则等指标
2. 捕获您的基本联网特性：
    a. VPC 流日志
    b. AWS Transit Gateway 指标
    c. AWS PrivateLink 指标
3. 捕获您的应用程序联网特性：
    a. Elastic Network Adaptor
    b. AWS App Mesh 指标
    c. Amazon API Gateway 指标
4. 捕获您的边缘联网特性：
    a. Amazon CloudFront 指标
    b. Amazon Route 53 指标
    c. AWS Global Accelerator 指标
5. 捕获您的混合联网特性：
    a. Direct Connect 指标
    b. AWS Site-to-Site VPN 指标
    c. AWS Client VPN 指标
    d. AWS Cloud WAN 指标
6. 捕获您的安全联网特性：
    a. AWS Shield、WAF 和 Network Firewall 指标
7. 使用跟踪工具捕获端到端性能指标：

a. AWS X-Ray

b. Amazon CloudWatch RUM

8. 对网络性能进行基准测试和测试：

a. 对网络 吞吐量进行基准测试：当实例位于同一 VPC 中时，一些因素可能会影响 EC2 网络性能。测量同一 VPC 中的 EC2 Linux 实例之间的网络带宽。

b. 执行 负载测试 以试用各种联网解决方案和选项

实施计划的工作量级别： 记录工作负载联网要求、选项和可用的解决方案所需的工作量为 中 。

资源

相关文档：

- Application Load Balancer
- Linux 上的 EC2 增强联网
- Windows 上的 EC2 增强联网
- EC2 置放群组
- 在 Linux 实例上启用 Elastic Network Adapter (ENA) 增强联网
- Network Load Balancer
- AWS 联网产品
- Transit Gateway
- 过渡到 Amazon Route 53 中基于延迟的路由
- VPC 终端节点
- VPC 流日志

相关视频：

- 连接 AWS 和混合 AWS 网络架构（NET317-R1）
- 优化 Amazon EC2 实例的网络性能 (CMP308-R1)
- 提高应用程序的全球网络性能
- EC2 实例和性能优化最佳实践
- 优化 Amazon EC2 实例的网络性能
- 使用 Well-Architected Framework 进行联网的最佳实践和技巧

- 大规模迁移中的 AWS 联网最佳实践

相关示例：

- AWS Transit Gateway 和可扩展的安全解决方案
- AWS 联网研讨会

PERF05-BP02 评估可用的联网功能

评估云中可能提高性能的联网功能。借助测试、指标和分析来衡量这些功能的影响。例如，利用可用的网络级功能来减少延迟、数据包丢失或抖动。

许多服务的创建旨在提高性能，而其他服务通常提供优化网络性能的功能。AWS Global Accelerator 和 Amazon CloudFront 等服务旨在提高性能，而大多数其他服务具有优化网络流量的产品功能。查看服务功能来提高工作负载性能，如 EC2 实例网络功能、增强联网实例类型、Amazon EBS 优化实例、Amazon S3 Transfer Acceleration 以及 CloudFront。

期望结果： 您已经记录了工作负载中的组件清单，并确定了每个组件的哪些网络配置将有助于满足性能需求。评估网络功能之后，您已经对性能指标进行了试验和测量，以确定如何使用可用的功能。

常见反模式：

- 您将所有工作负载都放在离总部最近的 AWS 区域，而不是放在接近终端用户的 AWS 区域。
- 未能对您的工作负载性能进行基准测试，并根据该基准不断评估您的工作负载性能。
- 您不查看服务配置以获得性能改进选项。

建立此最佳实践的好处： 评估所有服务功能和选项可以提高您的工作负载性能，降低基础设施的成本，减少维护工作负载所需的工作量，并提升您的整体安全状况。您可以利用 AWS 的全球骨干网，确保为客户提供出色的联网体验。

未建立这种最佳实践的情况下暴露的风险等级： 高

实施指导

查看您可以使用哪些与网络相关的配置选项，以及这些配置选项对您的工作负载的影响。了解这些选项如何与架构进行交互，以及这些选项对实际测量的性能和用户感知到的性能的影响，对于性能优化至关重要。

实施步骤：

1. 创建工作负载组件列表。
   a. 使用 AWS Cloud WAN构建、管理和监控您的组织网络。
   b. 使用 Network Manager查看您的网络。使用现有的配置管理数据库（CMDB）工具或 AWS Config 等工具创建工作负载清单及其配置方式。
2. 如果这是一个现有的工作负载，请确定并记录性能指标的基准，重点关注瓶颈和需要改进之处。基于业务要求和工作负载特征，与性能相关的网络指标将因工作负载而异。首先，对于您的工作负载，检查带宽、延迟、数据包丢失、抖动和重传等指标可能很重要。
3. 如果这是一个新的工作负载，请执行 负载测试 以识别性能瓶颈。
4. 对于识别的性能瓶颈，请查看解决方案的配置选项，以确定性能改进机会。
5. 如果您不知道网络路径或路由，请使用 Network Access Analyzer 来识别它们。
6. 查看您的网络协议，以进一步减少延迟。
   - PERF05-BP05 选择网络协议以提高性能
7. 如果您在多个位置使用 AWS Site-to-Site VPN 连接到 AWS 区域，请查看 加速的 Site-to-Site VPN 连接， 以获得提高联网性能的机会。
8. 当工作负载流量分散在多个账户中时，请评估您的网络拓扑结构和服务以减少延迟。
   - 当连接多个账户时，请评估 VPC 对等 和 AWS Transit Gateway 之间的运营和性能权衡。AWS Transit Gateway 支持 AWS Site-to-Site VPN 吞吐量，通过使用多路径扩展到超过单一 IPsec 最大限制 。Amazon VPC 和 AWS Transit Gateway 之间的流量保持在专用 AWS 网络上，而不会暴露在互联网上。AWS Transit Gateway 简化了您互连所有 VPC 的方式，这些 VPC 可以跨越数千个 AWS 账户并进入本地网络。在多个账户之间共享您的 AWS Transit Gateway（通过使用 Resource Access Manager）。要查看您的全球网络流量，请使用 Network Manager 集中了解您的网络指标情况。
9. 查看您的用户位置，并尽量缩短用户与工作负载之间的距离。
   a. AWS Global Accelerator 是一项网络服务，使用 Amazon Web Services 全球网络基础设施，可将用户流量的性能提高多达 60%。当互联网拥塞时，AWS Global Accelerator 会优化通往您的应用程序的路径，以始终保持较低的数据包丢失、抖动和延迟。它还提供了静态 IP 地址，可简化在可用区或 AWS 区域之间移动端点的过程，而无需更新 DNS 配置或更改面向客户端的应用程序。
   b. Amazon CloudFront 可在全球范围内提高工作负载内容交付性能并减少延迟。CloudFront 拥有超过 410 个分散在全球各地的入网点，可以缓存您的内容并减少终端用户的延迟。
   c. Amazon Route 53 提供 基于延迟的路由、地理位置路由、地理位置临近度路由和 基于 IP 的路由 选项，以帮助您提高面向全球受众的工作负载性能。通过检查工作负载流量和用户位置，确定哪个路由选项将优化您的工作负载性能。
10. 评估其他 Amazon S3 功能以改进存储 IOPS。

a. Amazon S3 Transfer Acceleration 是一项功能，借助该功能，外部用户在向 Amazon S3 传输数据时可以通过 CloudFront 的网络优化获益。这就提高了将大量数据从没有专用连接的远程位置传输到 AWS Cloud 的能力。

b. Amazon S3 多区域接入点 将内容复制到多个区域，并通过提供一个接入点简化了工作负载。使用多区域接入点时，您可以使用标识最低延迟桶的服务向 Amazon S3 请求或写入数据。

11查看您的计算资源网络带宽。

a. EC2 实例、容器和 Lambda 函数使用的弹性网络接口（ENI）按流进行限制。查看您的置放群组以优化 EC2 网络吞吐量。为避免在每个流的基础上出现瓶颈，请将应用程序设计为使用多个流。要监控和查看与计算相关的网络指标，请使用 CloudWatch Metrics 和 `ethtool`。`ethtool` 包含在 ENA 驱动程序中，并公开了其他与网络相关的指标，这些指标可作为 自定义指标 发布到 CloudWatch。

b. 较新的 EC2 实例可以利用增强联网。N 系列的 EC2 实例（例如 `M5n` 和 `M5dn`）利用第四代定制 Nitro 卡为单个实例提供高达 100Gbps 的网络吞吐量。与基础 `M5` 实例相比，这些实例提供了 4 倍的网络带宽和数据包处理能力，是网络密集型应用程序的理想选择。

c. Amazon Elastic Network Adapter （ENA）通过为 集群放置组中的实例提供更好的吞吐量来提供进一步优化。

d. Elastic Fabric Adapter （EFA）是 Amazon EC2 实例的网络接口，使您能够在 AWS 上大规模运行需要高级别节点间通信的工作负载。借助 EFA，使用消息传递接口（MPI）的高性能计算（HPC）应用程序和使用 NVIDIA Collective Communications Library（NCCL）的机器学习（ML）应用程序可以扩展到数千个 CPU 或 GPU。

e. Amazon EBS 优化 实例使用经过优化的配置堆栈，可以提供额外的专用容量来提高 Amazon EBS I/O。这种优化通过最小化您的 Amazon EBS I/O 与实例的其他流量之间的争用，来为 Amazon EBS 卷提供最佳性能。

实施计划的工作量级别：

要建立这种最佳实践，您必须了解目前影响网络性能的工作负载组件选项。收集组件、评估网络改进选项、试验、实施和记录这些改进的工作量为 低 到 适中 。

资源

相关文档：

- Amazon EBS - 优化实例
- Application Load Balancer
- Amazon EC2 实例网络带宽

- Linux 上的 EC2 增强联网
- Windows 上的 EC2 增强联网
- EC2 置放群组
- 在 Linux 实例上启用 Elastic Network Adapter (ENA) 增强联网
- Network Load Balancer
- AWS 联网产品
- AWS Transit Gateway
- 过渡到 Amazon Route 53 中基于延迟的路由
- VPC 终端节点
- VPC 流日志
- 构建云 CMDB
- 使用 AWS Transit Gateway 扩展 VPN 吞吐量

相关视频：

- 连接 AWS 和混合 AWS 网络架构（NET317-R1）
- 优化 Amazon EC2 实例的网络性能（CMP308-R1）
- AWS Global Accelerator

相关示例：

- AWS Transit Gateway 和可扩展的安全解决方案
- AWS 联网研讨会

PERF05-BP03 为混合工作负载选择适当大小的专用连接或 VPN

当需要使用公用网络连接 AWS 中的本地和云资源时，请确保您的带宽足以满足性能要求。估算混合工作负载的带宽和延迟要求。这些数字将确定 AWS Direct Connect 或您的 VPN 终端节点的大小要求。

期望结果： 当部署需要混合网络连接的工作负载时，您有多个连接配置选项，例如托管和非托管 VPN 或 Direct Connect。为每个工作负载选择适当的连接类型，并确保在您的位置和云之间设置适当的带宽和加密要求。

常见反模式：

- 您仅根据网络加密要求评估 VPN 解决方案。
- 您不会评估备份或并行连接选项。
- 您使用路由器、隧道和 BGP 会话的默认配置。
- 您无法理解或识别所有工作负载要求（加密、协议、带宽和流量需求）。

建立此最佳实践的好处： 通过选择并配置适当大小的混合网络解决方案，可以提高工作负载的可靠性并最大限度地增加性能提高机会。通过确定工作负载要求、提前规划和评估混合解决方案，您将最大限度地减少昂贵的物理网络变更和运营开销，并加快上市速度。

未建立这种最佳实践的情况下暴露的风险等级： 高

实施指导

根据您的带宽要求开发混合联网架构：估算混合应用程序的带宽和延迟要求。根据您的带宽要求，单个 VPN 或 Direct Connect 连接可能不够，您必须构建混合设置以实现多个连接之间的流量负载平衡。可能需要使用 Direct Connect，因为它的专用网络连接能够提供可预测性更高且更一致的性能。它非常适合需要一致的延迟和几乎零抖动的生产工作负载。

AWS Direct Connect 提供了到 AWS 环境的专用连接，速率从 50 Mbps 到 10 Gbps 不等。这样一来，延迟得到管理和控制，并且拥有预置带宽，让您的工作负载能够以轻松且高性能的方式连接到其他环境。使用 AWS Direct Connect 合作伙伴之一，您可以拥有多个环境的端到端连接，从而提供性能一致的扩展网络。

AWS Site-to-Site VPN 是 VPC 的托管 VPN 服务。建立 VPN 连接后，AWS 将提供到两个不同的 VPN 端点的隧道。借助 AWS Transit Gateway，您可以简化多个 VPC 之间的连接，还可以通过单个 VPN 连接来连接到与 AWS Transit Gateway 连接的任何 VPC。AWS Transit Gateway 还可以通过在多个 VPN 隧道上启用等价多路径（ECMP，Equal Cost Multi-Path）路由支持，使您扩展到 1.25 Gbps IPsec VPN 吞吐量限制之外。

实施计划的工作量级别： 评估混合网络的工作负载需求和实施混合联网解决方案所需的工作量为 高 。

资源

相关文档：

- Network Load Balancer
- AWS 联网产品
- Transit Gateway

- 过渡到 Amazon Route 53 中基于延迟的路由
- VPC 终端节点
- VPC 流日志
- Site-to-Site VPN
- 构建可扩展且安全的多 VPC AWS 网络基础设施
- Direct Connect
- Client VPN

相关视频：

- 连接 AWS 和混合 AWS 网络架构（NET317-R1）
- 优化 Amazon EC2 实例的网络性能 (CMP308-R1)
- AWS Global Accelerator
- Direct Connect
- Transit Gateway Connect
- VPN 解决方案
- VPN 解决方案的安全性

相关示例：

- AWS Transit Gateway 和可扩展的安全解决方案
- AWS 联网研讨会

PERF05-BP04 利用负载均衡和加密卸载

跨多个资源或服务分配流量，以便让工作负载能够利用云提供的弹性。您也可以使用负载均衡机制来卸载加密终端，以便提高性能并有效管理和路由流量。

在实施想要在其中针对服务内容使用多个实例的横向扩展架构时，您可以利用 Amazon VPC 内部的负载均衡器。AWS 为 ELB 服务中的应用程序提供了多个模型。Application Load Balancer 最适合 HTTP 和 HTTPS 流量的负载均衡，面向交付包括微服务和容器在内的现代化应用程序架构，提供高级请求路由功能。

若要对需要极高性能的 TCP 流量进行负载均衡，Network Load Balancer 是最佳选择。网络负载均衡器每秒能够处理数百万请求，同时能保持超低延迟，还针对处理突发和不稳定的流量模式进行了优化。

Elastic Load Balancing 提供集成的证书管理和 SSL/TLS 解密，使您可以灵活地集中管理负载均衡器的 SSL 设置，并从工作负载中卸载占用大量 CPU 的工作。

常见反模式：

- 您可以通过现有负载均衡器来路由所有互联网流量。
- 您可以使用通用 TCP 负载均衡，并让每个计算节点处理 SSL 加密。

建立此最佳实践的好处： 负载均衡器可在单个可用区内或多个可用区之间处理您的应用程序的不断变化的流量负载。负载均衡器具有高可用性和自动扩展功能，并且具有强大的安全性，可让您的应用程序具有容错能力。

未建立这种最佳实践的情况下暴露的风险等级： 高

实施指导

使用适当的负载均衡器来处理工作负载：为您的工作负载选择适当的负载均衡器。如果您必须对 HTTP 请求进行负载均衡，我们建议您使用 Application Load Balancer。对于网络和传输协议（第 4 层 – TCP、UDP）负载均衡，以及极高性能和低延迟的应用程序，我们建议使用网络负载均衡器。Application Load Balancers 支持 HTTPS，网络负载均衡器支持 TLS 加密卸载。

启用卸载 HTTPS 或 TLS 加密：Elastic Load Balancing 包含集成化证书管理、用户身份验证和 SSL/TLS 解密功能。使用它可以灵活、集中地管理 TLS 设置，并且能够从您的应用程序中卸载 CPU 密集型工作负载。在部署负载均衡器的过程中加密所有 HTTPS 流量。

资源

相关文档：

- Amazon EBS – 优化实例
- Application Load Balancer
- Linux 上的 EC2 增强联网
- Windows 上的 EC2 增强联网
- EC2 置放群组
- 在 Linux 实例上启用 Elastic Network Adapter (ENA) 增强联网
- Network Load Balancer
- AWS 联网产品

- Transit Gateway
- 过渡到 Amazon Route 53 中基于延迟的路由
- VPC 终端节点
- VPC 流日志

相关视频：

- 连接 AWS 和混合 AWS 网络架构（NET317-R1）
- 优化 Amazon EC2 实例的网络性能 (CMP308-R1)

相关示例：

- AWS Transit Gateway 和可扩展的安全解决方案
- AWS 联网研讨会

PERF05-BP05 选择网络协议以提高性能

根据对工作负载性能的影响，做出有关系统与网络之间的通信协议的决策。

延迟和带宽之间的关系可以实现高吞吐量。如果文件传输使用 TCP 协议，则延迟越高，整体吞吐量越低。有一些方法可以使用 TCP 调整和优化的传输协议来解决此问题，有些方法则使用 UDP 协议。

常见反模式：

- 无论有怎样的性能要求，您都可以为所有工作负载使用 TCP。

建立此最佳实践的好处： 为工作负载组件之间的通信选择适当的协议，可确保您获得该工作负载的最佳性能。无连接 UDP 虽然允许较高速度，但不提供重新传输或高可靠性。TCP 虽然是一个功能全面的协议，但它在处理这些数据包时需要较高的开销。

未建立这种最佳实践的情况下暴露的风险等级： 中

实施指导

优化网络流量：选择适当的协议来优化您的工作负载的性能。延迟和带宽之间的关系可以实现高吞吐量。如果文件传输使用 TCP，则延迟越高，整体吞吐量就越低。有一些方法可以使用 TCP 调整和优化的传输协议来解决延迟问题，有些方法使用 UDP。

资源

相关文档：

- Amazon EBS – 优化实例
- Application Load Balancer
- Linux 上的 EC2 增强联网
- Windows 上的 EC2 增强联网
- EC2 置放群组
- 在 Linux 实例上启用 Elastic Network Adapter (ENA) 增强联网
- Network Load Balancer
- AWS 联网产品
- Transit Gateway
- 过渡到 Amazon Route 53 中基于延迟的路由
- VPC 终端节点
- VPC 流日志

相关视频：

- 连接 AWS 和混合 AWS 网络架构（NET317-R1）
- 优化 Amazon EC2 实例的网络性能 (CMP308-R1)

相关示例：

- AWS Transit Gateway 和可扩展的安全解决方案
- AWS 联网研讨会

PERF05-BP06 根据网络要求选择工作负载的位置

使用可用的云位置选项来降低网络延迟或提高吞吐量。利用 AWS 区域、可用区、置放组和边缘站点（例如 AWS Outposts、AWS Local Zones 和 AWS Wavelength）来降低网络延迟或提高吞吐量。

AWS Cloud 基础设施围绕区域和可用区构建。区域是指全球范围内的某个物理位置，每个区域有多个可用区。

可用区由一个或多个分散的数据中心组成，每个都拥有独立的配套设施，其中包括冗余电源、联网和连接。可用区能够提高生产应用程序和数据库的运行效率，使其具备比单个数据中心更强的可用性、容错能力以及可扩展性

请根据以下关键元素，为您的部署选择一个或多个合适的区域：

- 用户所在位置：选择一个接近您的工作负载用户的区域，确保他们在使用工作负载时延迟较低。
- 数据所在位置：对于数据密集型应用程序，延迟方面的主要瓶颈是数据传输。应用程序代码的执行应尽量接近数据。
- 其他制约：考虑安全性和合规性等制约。

Amazon EC2 为联网提供置放群组。置放组是实例的逻辑分组，可以减少延迟或提高可靠性。使用具有支持的实例类型和 Elastic Network Adapter (ENA) 的置放群组，可使工作负载参与低延迟的 25 Gbps 网络。建议将置放群组用于可受益于低网络延迟和/或高网络吞吐量的工作负载。使用置放群组有降低网络通信抖动的优势。

延迟敏感型服务是使用全球边缘站点网络在边缘交付的。这些边缘站点通常提供内容分发网络 (CDN) 和域名系统 (DNS) 等服务。通过在边缘交付这些服务，工作负载可以低延迟响应内容或 DNS 解析请求。这些服务还提供地理定位服务，例如内容地理定位（基于最终用户位置提供不同内容），或基于延迟的路由，用于将最终用户引导至最近的区域（最小延迟）。

Amazon CloudFront 是一个全球性内容分发网络 (CDN)，可用于加速静态内容（如图像、脚本和视频）以及动态内容（如 API 或 Web 应用程序）。它依赖于全球边缘站点网络，可以缓存内容并为您的用户提供高性能的网络连接。CloudFront 也加快了其他许多功能，如内容上传和动态应用程序，从而使通过互联网提供流量的所有应用程序的性能有所提高。 Lambda@Edge 是 Amazon CloudFront 的一项功能，使您可以更接近工作负载用户运行代码，从而提高性能并减少延迟。

Amazon Route 53 是一种高度可用且可扩展的云 DNS Web 服务。它的目的是为开发人员和企业提供一种非常可靠且经济高效的方式，将名称（如 www.example.com）转换为计算机用于互相连接的数字 IP 地址（如 192.168.2.1），从而将最终用户路由到互联网应用程序。Route 53 与 IPv6 完全兼容。

AWS Outposts 专为因延迟要求而需要保留在本地的工作负载而设计，此时您希望该工作负载与 AWS 中的其他工作负载一起无缝运行。AWS Outposts 是完全托管且可配置的计算和存储机架，这些机架使用 AWS 设计的硬件构建，可让您在本地运行计算和存储，同时无缝连接到云中 AWS 的广泛服务。

AWS Local Zones 设计用于运行需要几毫秒延迟的工作负载，例如视频渲染和图形密集型虚拟桌面应用程序。本地扩展区使您可以获得使计算和存储资源更接近最终用户的所有优势。

AWS Wavelength 通过将 AWS 基础设施、服务、API 和工具扩展到 5G 网络，旨在向 5G 设备提供超低延迟应用程序。Wavelength 将存储和计算嵌入电信运营商 5G 网络内部，以在您的 5G 工作负载需要几毫秒延迟时提供帮助，例如 IoT 设备、游戏流、自动驾驶汽车和实时媒体制作。

可使用边缘服务来减少延迟并启用内容缓存。请确保您为 DNS 和 HTTP/HTTPS 正确配置了缓存控制，以便通过这些方式获得最大优势。

常见反模式：

- 您可以将所有工作负载资源整合到一个地理位置中。
- 您选择的是离您的位置最近的区域，而不是离工作负载最终用户最近的区域。

建立此最佳实践的好处： 您必须确保无论您在哪里希望联系客户时，您的网络均可用。使用 AWS 的专用全球网络，通过将工作负载部署到离他们最近的位置，可以确保您的客户获得最低的延迟体验。

未建立此最佳实践暴露的风险等级： 中

实施指导

通过选择正确的位置减少延迟：确定用户和数据的位置。利用 AWS 区域、可用区、置放组和边缘站点来降低延迟。

资源

相关文档：

- Amazon EBS – 优化实例
- Application Load Balancer
- Linux 上的 EC2 增强联网
- Windows 上的 EC2 增强联网
- EC2 置放群组
- 在 Linux 实例上启用 Elastic Network Adapter (ENA) 增强联网
- Network Load Balancer
- AWS 联网产品
- Transit Gateway
- 过渡到 Amazon Route 53 中基于延迟的路由
- VPC 终端节点
- VPC 流日志

相关视频：

- 连接 AWS 和混合 AWS 网络架构（NET317-R1）
- 优化 Amazon EC2 实例的网络性能 (CMP308-R1)

相关示例：

- AWS Transit Gateway 和可扩展的安全解决方案
- AWS 联网研讨会

PERF05-BP07 根据各项指标优化网络配置

使用收集和分析的数据做出有关优化网络配置的明智决策。衡量更改带来的影响，并根据衡量结果来做出进一步决策。

为您的工作负载使用的所有 VPC 网络启用 VPC 流日志。VPC 流日志功能使您能够进一步捕获有关传入和传出您的 VPC 中网络接口的 IP 流量的信息。VPC 流日志可帮助您完成许多任务，例如解决为什么特定流量无法到达实例的问题，进而帮助您诊断过于严格的安全组规则。您可以使用流日志作为安全工具来监控到达实例的流量，以分析网络流量并查找异常的流量行为。

使用网络指标来随着工作负载的发展对网络配置进行更改。基于云的网络可以快速重建，因此有必要随着时间的推移改进网络架构，以保持性能效率。

常见反模式：

- 您应认为所有性能相关的问题都与应用程序有关。
- 您只需从距离已部署工作负载很近的位置测试您的网络性能。

建立此最佳实践的好处：为了确保您满足工作负载所需的指标，您必须监控网络性能指标。您可以捕获有关传入和传出您的 VPC 中网络接口的 IP 流量的信息，并使用这些数据为新的地理区域添加新的优化项目或部署您的工作负载。

未建立这种最佳实践的情况下暴露的风险等级： 低

实施指导

启用 VPC 流日志：使用 VPC 流日志，您可以捕获有关传入和传出您的 VPC 中网络接口的 IP 流量的信息。VPC 流日志可帮助您完成许多任务，例如解决为什么特定流量无法到达实例的问题，进而帮助

您诊断过于严格的安全组规则。您可以使用流日志作为安全工具来监控到达实例的流量，以分析网络流量并查找异常的流量行为。

为网络选项启用适当的指标：确保您为工作负载选择适当的网络指标。您可以启用 VPC NAT 网关、Transit Gateway 和 VPN 隧道的指标。

资源

相关文档：

- Amazon EBS – 优化实例
- Application Load Balancer
- Linux 上的 EC2 增强联网
- Windows 上的 EC2 增强联网
- EC2 置放群组
- 在 Linux 实例上启用 Elastic Network Adapter (ENA) 增强联网
- Network Load Balancer
- AWS 联网产品
- Transit Gateway
- 过渡到 Amazon Route 53 中基于延迟的路由
- VPC 终端节点
- VPC 流日志
- 使用 Amazon Cloudwatch 指标监控您的全球和核心网络
- 持续监控网络流量和资源

相关视频：

- 连接 AWS 和混合 AWS 网络架构（NET317-R1）
- 优化 Amazon EC2 实例的网络性能 (CMP308-R1)
- 监控网络流量并排查问题
- 使用 Amazon VPC Traffic Mirroring 简化流量监控并提供可见性

相关示例：

- AWS Transit Gateway 和可扩展的安全解决方案
- AWS 联网研讨会
- AWS 网络监控

审核

问题

- PERF 6 如何改进工作负载以便利用新的版本？

PERF 6 如何改进工作负载以便利用新的版本？

在最初构建解决方案时，您可能会从有限的方案选项中进行选择。但是随着时间的推移，可提升工作负载性能的新技术和方法会不断涌现。

最佳实践

- PERF06-BP01 及时了解最新资源和服务
- PERF06-BP02 制定流程来提高工作负载性能
- PERF06-BP03 随着时间的推移提高工作负载性能

PERF06-BP01 及时了解最新资源和服务

当新的服务、设计模式或产品问世时，评估可以提高性能的方法。通过评估、内部讨论或外部分析来确定哪些方法可以提高工作负载的性能或效率。

制定相应流程，评估与工作负载相关的更新、新功能和服务。例如，使用新技术构建概念验证或咨询内部团队。在尝试新想法或新服务时，运行性能测试，以衡量这些新想法或新服务对工作负载性能的影响。使用基础设施即代码（IaC）和 DevOps 文化，以最少的成本或风险，运用这些功能来频繁测试新的想法或技术。

期望的结果： 您记录了组件清单、设计模式以及工作负载特性。使用这些文档创建订阅列表，用于通知您的团队有关服务更新、功能和新产品的信息。您确定了组件利益相关者，他们将评估新发布的内容并提供有关业务影响力和优先级的推荐。

常见反模式：

- 仅当工作负载未达到性能要求时审查新选项和服务。

- 您可以假设所有新产品都不会对您的工作负载有帮助。
- 在改进工作负载时，您总是选择自行构建而不是购买服务。

建立此最佳实践的好处： 通过考虑采用新服务或产品方案，您可以提高工作负载的性能和效率，降低基础设施的成本，并减少维护服务所需的工作量。

未建立这种最佳实践的情况下暴露的风险等级： 高

实施指导

制定相应流程，评估 AWS 推出的更新、新功能和新服务。例如，构建使用新技术的概念验证。在尝试新想法或新服务时，运行性能测试，以衡量这些新想法或新服务对工作负载的效率或性能的影响。利用您在 AWS 上获得的灵活性，经常对新想法或新技术进行测试，以尽量降低成本或风险。

实施步骤

1. 记录您的工作负载解决方案。使用您的配置管理数据库（CMDB，Configuration Management DataBase）解决方案来记录清单，并对服务和依赖关系进行分类。使用 AWS Config 等工具来获取工作负载使用的所有 AWS 服务的列表。
2. 使用 标记策略 记录各个工作负载组件和类别的负责人。例如，如果您当前使用 Amazon RDS 作为数据库解决方案，请让数据库管理员（DBA）分配并记录负责人，以便评估和研究新服务及更新。
3. 确定与您工作负载组件相关的新闻和更新来源。在之前提到的 Amazon RDS 示例中，类别负责人应该订阅与其工作负载组件相符的产品的 AWS 新增功能博客 。您可以订阅 RSS 源或管理您的 电子邮件订阅。了解您使用的 Amazon RDS 数据库的升级、推出的功能、发布的实例以及 Amazon Aurora Serverless 等新产品。查看行业博客、产品以及组件所依赖的供应商。
4. 记录评估更新和新服务的流程。为类别负责人提供所需的时间和空间来研究、测试、试验和验证更新及新服务。回顾记录的业务需求和 KPI，帮助优先确定哪些更新可以带来积极的业务影响。

实施计划的工作量级别： 要建立此最佳实践，您必须了解现有的工作负载组件，确定类别负责人并确定服务更新的来源。启动这一流程所需的工作量较少，但这是个长期过程，会随着时间不断演变和改进。

资源

相关文档：

- AWS 博客
- AWS 新增功能

相关视频：

- AWS 事件 YouTube 频道
- AWS 在线技术讲座 YouTube 频道
- Amazon Web Services YouTube 频道

相关示例：

- AWS Github
- AWS Skill Builder

PERF06-BP02 制定流程来提高工作负载性能

制定相应流程，以在新的服务、设计模式、资源类型和配置推出后，对它们进行评估。例如，对新实例产品运行现有性能测试，以确定它们改进工作负载的潜力。

工作负载的性能会面临一些关键约束。记录这些约束，以便您了解哪些创新可以改进工作负载的性能。当您知道有新的服务或技术推出时，借助这些信息来确定消除约束或瓶颈的方法。

常见反模式：

- 您可以假设当前的架构将为静态并且不会随着时间的推移而更新。
- 您可以随着时间的推移对架构进行更改，而无需提供任何指标方面的依据。

建立此最佳实践的好处： 通过制定架构更改流程，您可以允许使用所收集的数据来影响以后的工作负载设计。

未建立这种最佳实践的情况下暴露的风险等级： 中

实施指导

确定工作负载的关键性能约束：记录您的工作负载的性能约束，以便您了解哪类创新可以提高工作负载的性能。

资源

相关文档：

- AWS Blog

- AWS 的新增功能

相关视频：

- AWS 事件 YouTube 频道
- AWS 在线技术讲座 YouTube 频道
- Amazon Web Services YouTube 频道

相关示例：

- AWS Github
- AWS Skill Builder

PERF06-BP03 随着时间的推移提高工作负载性能

组织需要使用在评估流程中收集的信息，积极推动对新推出的服务或资源的采用。

利用评估新服务或新技术时收集的信息来推动变革。随着您的业务或工作负载发生改变，性能需求也会改变。使用从工作负载指标中收集的数据来评估在哪些方面可以获得最大的效率或性能提升，并且积极采用新服务和新技术来紧跟需求。

常见反模式：

- 您可以假设当前的架构将为静态并且不会随着时间的推移而更新。
- 您可以随着时间的推移对架构进行更改，而无需提供任何指标方面的依据。
- 您可以仅仅因为行业中所有其他人都在使用架构而对架构进行更改。

建立此最佳实践的好处： 要优化您的工作负载的性能和成本，您必须评估所有可用的软件和服务，以确定适合您的工作负载的软件和服务。

未建立这种最佳实践的情况下暴露的风险等级： 低

实施指导

随着时间的推移提高工作负载性能：利用评估新服务或新技术时收集的信息来推动变革。随着您的业务或工作负载发生改变，性能需求也会改变。使用从工作负载指标中收集的数据来评估在哪些方面可以获得最大的效率或性能提升，并且积极采用新服务和新技术来满足不断变化的需求。

资源

相关文档：

- AWS Blog
- AWS 的新增功能

相关视频：

- AWS 事件 YouTube 频道
- AWS 在线技术讲座 YouTube 频道
- Amazon Web Services YouTube 频道

相关示例：

- AWS Github
- AWS Skill Builder

监控

问题

- PERF 7 如何监控资源以确保其性能？

PERF 7 如何监控资源以确保其性能？

系统性能会随着时间的推移而降低。监控系统性能，以发现性能降低的情况，并针对内部或外部因素（例如操作系统或应用程序负载）采取修复措施。

最佳实践

- PERF07-BP01 记录与性能相关的指标
- PERF07-BP02 在发生事件或意外事件时分析各项指标
- PERF07-BP03 建立关键性能指标（KPI）来衡量工作负载性能
- PERF07-BP04 借助监控来生成基于警报的通知
- PERF07-BP05 定期检查指标：
- PERF07-BP06 主动监控和警报

PERF07-BP01 记录与性能相关的指标

使用监控和可观察性服务来记录性能相关的指标。指标示例包括记录数据库事务、速度缓慢的查询、I/O 延迟、HTTP 请求吞吐量、服务延迟或其他关键数据。

确定对工作负载至关重要的性能指标并记录下来。这些数据对于确定影响工作负载整体性能或效率的组件非常重要。

回顾客户体验，确定至关重要的指标。确定每个指标的目标、衡量方式和优先程度。根据这些信息创建警报和通知，以主动解决与性能相关的问题。

常见反模式：

- 您只需监控操作系统级别的指标，即可深入了解您的工作负载。
- 您需要为峰值工作负载要求设计您的计算需求。

建立此最佳实践的好处： 要优化性能和提高资源利用率，您需要一个关于关键性能指标的统一运营视图。您可以创建控制面板并对数据执行指标计算，以获得运营和利用率见解。

未建立这种最佳实践的情况下暴露的风险等级： 高

实施指导

确定与工作负载相关的性能指标并记录下来。这些数据可以帮助确定哪些组件会影响工作负载的整体性能或效率。

确定性能指标：根据客户体验来确定最重要的指标。确定每个指标的目标、衡量方式和优先程度。根据这些数据创建警报和通知，以主动解决与性能相关的问题。

资源

相关文档：

- CloudWatch 文档
- 使用 CloudWatch 代理从 Amazon EC2 实例和本地服务器收集指标和日志
- 发布自定义指标
- 监控、日志记录和性能 APN 合作伙伴
- X-Ray 文档

- Amazon CloudWatch RUM

相关视频：

- 摆脱混乱：获得运营可见性和见解 (MGT301-R1)
- AWS 上的应用程序性能管理
- 制定监控计划

相关示例：

- 第 100 级：使用 CloudWatch 控制面板进行监控
- 第 100 级：使用 CloudWatch 控制面板监控 Windows EC2 实例
- 第 100 级：使用 CloudWatch 控制面板监控 Amazon Linux EC2 实例

PERF07-BP02 在发生事件或意外事件时分析各项指标

在某个事件或意外事件发生后（或发生过程中），使用监控控制面板或报告来了解和诊断影响。这些视图可让您了解工作负载哪些部分的性能没有达到预期。

针对架构编写重要用户案例时，请纳入性能要求，例如指定每个重要案例应以多快速度执行。对于这些重要案例，实施额外的脚本化用户体验，以便确保您知道这些案例是如何根据您的要求执行的。

常见反模式：

- 您可以假设性能事件是一次性问题，并且只与异常有关。
- 对性能事件进行响应时，只需评估现有性能指标。

建立此最佳实践的好处： 要确定您的工作负载是否按预期运行，您必须通过收集其他指标数据进行分析，从而对性能事件做出响应。这些数据用于了解性能事件的影响，并建议更改来提高工作负载的性能。

未建立这种最佳实践的情况下暴露的风险等级： 高

实施指导

优先考虑重要用户案例的体验问题：针对架构编写重要用户案例时，请纳入性能要求，例如指定每个重要案例应以多快速度实施。对于这些重要案例，实施额外的脚本用户历程，以确保您知道这些用户案例如何根据您的要求执行。

资源

相关文档：

- CloudWatch 文档
- Amazon CloudWatch Synthetics
- 监控、日志记录和性能 APN 合作伙伴
- X-Ray 文档

相关视频：

- 摆脱混乱：获得运营可见性和见解 (MGT301-R1)
- 通过 Amazon CloudWatch RUM 优化应用程序
- Amazon CloudWatch Synthetics 演示

相关示例：

- 使用 Amazon CloudWatch Synthetics 测量页面加载时间
- Amazon CloudWatch RUM Web 客户端

PERF07-BP03 建立关键性能指标（KPI）来衡量工作负载性能

确定定量和定性地衡量工作负载性能的 KPI。KPI 有助于衡量与业务目标相关的工作负载的运行状况。利用 KPI，业务和工程团队可在衡量目标和战略以及如何将二者结合来取得业务成果方面保持一致。当业务目标、战略或最终用户需求发生变化时，应重访 KPI。

例如，网站工作负载可能会将页面加载时间用作总体性能指示。该指标是用来衡量最终用户体验的多个数据点之一。除了确定页面加载时间阈值之外，您还应记录未达到性能要求时的预期成果或业务风险。较长的页面加载时间会直接影响最终用户的体验，降低他们的用户体验评分，并可能导致客户流失。在定义 KPI 阈值时，请结合考虑行业基准和最终用户期望。例如，如果当前行业基准是两秒内加载网页，而您的最终用户希望网页在一秒内加载，那么您在建立 KPI 时应考虑这两个数据点。KPI 的另一个示例可能侧重于满足内部绩效需求。在生成生产数据后的一个工作日内，在生成销售报告时可以确立 KPI 阈值。这些报告可能会直接影响日常决策和业务成果。

期望结果： 确立 KPI 涉及不同的部门和利益相关者。您的团队必须使用实时细粒度数据和历史数据作为参考来评估工作负载 KPI，并创建控制面板来对 KPI 数据执行指标计算，以获得运营和利用率见解。应记录 KPI，这可以说明议定的 KPI 和阈值，用于支持业务目标和战略，并且与所监控的指标对

应起来。KPI 确定了绩效要求，所有团队应专门审查并经常分享和了解这些指标。清楚地确定风险和权衡机制，并了解未达到 KPI 阈值将产生的业务影响。

常见反模式：

- 您仅监控系统级指标以获得工作负载见解，而不了解业务对这些指标产生的影响。
- 您可以假设您的 KPI 已作为标准指标数据发布和共享。
- 定义 KPI，但未与所有团队共享。
- 未定义量化的、可衡量的 KPI。
- 未使 KPI 与业务目标或战略保持一致。

建立此最佳实践的好处： 通过确定代表工作负载运行状况的具体指标，有助于使团队在其优先事项上保持一致和定义业务成果成功的标准。与所有部门共享这些指标可让所有人了解并一致认可阈值、期望值和业务影响。

未建立这种最佳实践的情况下暴露的风险等级： 高

实施指导

所有受工作负载运行状况影响的部门和业务团队应共同努力确立 KPI。由专人负责推动与组织 KPI 相关的协作、时间表、文档和信息。此单线负责人会经常分享业务目标和战略，并向业务利益相关者分配任务，以在各自的部门创建 KPI。在定义 KPI 后，运维团队通常会帮助定义指标，用于支持达成不同的 KPI 并通知成功情况。只有支持工作负载的所有团队成员都了解 KPI 时，KPI 才会有效。

实施步骤

1. 确定并记录业务利益相关者。
2. 确定公司目标和战略。
3. 审查符合公司目标和战略的常见行业 KPI。
4. 审查最终用户对您工作负载的期望。
5. 定义和记录支持公司目标和战略的 KPI。
6. 确定并记录为实现 KPI 而批准的权衡策略。
7. 确定并记录可提供 KPI 信息的指标。
8. 确定并记录严重性或警报级别的 KPI 阈值。
9. 确定并记录未满足 KPI 时带来的风险和影响。
10. 确定每个 KPI 的审查频率。

11.与所有支持工作负载的团队交流 KPI 文档内容。

实施指导的工作量级别： 定义和交流 KPI 所需的工作量为 低 。通常，可以在几周内与业务利益相关者会面，并审查目标、战略和工作负载指标来完成这项工作。

资源

相关文档：

- CloudWatch 文档
- 监控、日志记录和性能 APN 合作伙伴
- X-Ray 文档
- 使用 Amazon CloudWatch 控制面板
- Amazon QuickSight KPI

相关视频：

- AWS re:Invent 2019：扩展到第一个 1000 万用户（ARC211-R）
- 摆脱混乱：获得运营可见性和见解 (MGT301-R1)
- 制定监控计划

相关示例：

- 使用 Amazon QuickSight 创建控制面板

PERF07-BP04 借助监控来生成基于警报的通知

根据您定义的与性能相关的关键性能指标 (KPI)，使用当测量值超出预期范围时能够自动生成警报的监控系统。

Amazon CloudWatch 可以收集架构中各种资源的指标。您也可以收集和发布自定义指标，用于显示业务指标或派生指标。使用 CloudWatch 或第三方监控服务设置表明超出阈值的警报；警报表明某个指标超出预期范围。

常见反模式：

- 您可以依靠工作人员来观察指标，并在他们发现问题时做出响应。

- 您仅依赖于运维手册，但可以触发无服务器工作流来完成相同的任务。

建立此最佳实践的好处： 您可以根据预定义的阈值，或根据可识别您的指标中的异常行为的机器学习算法，设置警报并自动执行操作。这些警报还可以触发无服务器工作流，从而修改工作负载的性能特性（例如，增加计算容量、更改数据库配置）。

未建立这种最佳实践的情况下暴露的风险等级： 中

实施指导

监控指标：Amazon CloudWatch 可以收集架构中各种资源的指标。您可以收集和发布自定义指标，用于显示业务指标或派生指标。可以使用 CloudWatch 或第三方监控服务来设置指示超出阈值的警报。

资源

相关文档：

- CloudWatch 文档
- 监控、日志记录和性能 APN 合作伙伴
- X-Ray 文档
- 在 CloudWatch 中使用警报和警报操作

相关视频：

- AWS re:Invent 2019：扩展到第一个 1000 万用户（ARC211-R）
- 摆脱混乱：获得运营可见性和见解 (MGT301-R1)
- 制定监控计划
- 将 AWS Lambda 与 Amazon CloudWatch Events 配合使用

相关示例：

- Cloudwatch Logs 自定义警报

PERF07-BP05 定期检查指标：

在例行维护时，或者事件或意外事件发生后，检查收集到了哪些指标。通过这些检查，找出哪些指标对于解决问题至关重要，以及跟踪哪些其他指标会有助于发现、解决问题或预防问题发生。

在响应意外事件或事件的过程中，评估哪些指标有助于解决问题、哪些目前没有跟踪的指标会有助于解决问题。这样，您可以提高收集的指标的质量，从而预防或更快速地解决未来发生的意外事件。

常见反模式：

- 您可以允许指标保持警报状态较长时间。
- 您可以创建自动化系统无法操作的警报。

建立此最佳实践的好处： 不断检查收集的指标，以确保它们能够帮助正确地发现、解决问题或预防问题发生。如果您让指标保持警报状态过长时间，这些指标也会过时。

未建立此最佳实践暴露的风险等级： 中

实施指导

不断改进指标收集和监控：在响应意外事件或事件的过程中，评估哪些指标有助于解决问题、哪些目前没有跟踪的指标会有帮助。通过这种方法，您可以提高收集的指标的质量，从而预防或更快速地解决未来发生的意外事件。

资源

相关文档：

- CloudWatch 文档
- 使用 CloudWatch 代理从 Amazon EC2 实例和本地服务器收集指标和日志
- 监控、日志记录和性能 APN 合作伙伴
- X-Ray 文档

相关视频：

- 摆脱混乱：获得运营可见性和见解 (MGT301-R1)
- AWS 上的应用程序性能管理
- 制定监控计划

相关示例：

- 使用 Amazon QuickSight 创建控制面板

- 第 100 级：使用 CloudWatch 控制面板进行监控

PERF07-BP06 主动监控和警报

使用关键性能指标 (KPI) 并结合监控和警报系统，主动解决与性能相关的问题。使用警报触发自动操作，以便在可能的情况下修复问题。如果无法实现自动响应，则将警报上报给能够响应的人员。例如，您的系统在关键性能指标 (KPI) 超出特定阈值时，能够预测预期 KPI 值并发出警报；或者您的工具在 KPI 超出预期值时，能够自动停止或回滚部署。

实施相应流程，让您在工作负载运行期间了解其性能。构建监控控制面板并确定预期性能基准，以确定工作负载的性能是否达到最佳。

常见反模式：

- 您可以只允许运营人员对工作负载进行运营更改。
- 您可以通过设置筛选器将所有没有主动修复行为的警报发送给运营团队。

建立此最佳实践的好处： 主动修复警报行为使支持人员能够集中精力处理那些无法自动完成的工作。这可确保运营人员不需要花费精力处理所有警报，而是能够集中精力处理重要警报。

未建立此最佳实践暴露的风险等级： 低

实施指导

在运维期间监控性能：实施相应流程，让您在工作负载运行期间了解其性能。构建监控控制面板并建立性能预期基准。

资源

相关文档：

- CloudWatch 文档
- 监控、日志记录和性能 APN 合作伙伴
- X-Ray 文档
- 在 CloudWatch 中使用警报和警报操作

相关视频：

- 摆脱混乱：获得运营可见性和见解 (MGT301-R1)
- AWS 上的应用程序性能管理
- 制定监控计划
- 将 AWS Lambda 与 Amazon CloudWatch Events 配合使用

相关示例：

- Cloudwatch Logs 自定义警报

权衡

问题

- PERF 8 如何使用权衡机制来提高性能？

PERF 8 如何使用权衡机制来提高性能？

在构建解决方案时，确定权衡机制可以帮助您选出最佳方法。通常，您可以牺牲一致性、持久性和空间来换取缩短时间和延迟，从而提高性能。

最佳实践

- PERF08-BP01 了解在哪些领域性能最为重要
- PERF08-BP02 了解设计模式和服务
- PERF08-BP03 确定权衡机制对客户和效率的影响
- PERF08-BP04 衡量性能提高产生的影响
- PERF08-BP05 使用各种与性能相关的策略

PERF08-BP01 了解在哪些领域性能最为重要

了解并确定在哪些方面提高工作负载性能，会对效率或客户体验产生积极的影响。例如，拥有大量客户交互的网站会因为使用边缘服务在距离客户更近的位置向客户分发内容而受益。

期望的结果： 通过了解架构、流量模式和数据访问模式，提高性能效率，并确定延迟和处理时间。确定随着工作负载增长可能会影响客户体验的潜在瓶颈。在确定这些领域时，查看可以通过部署哪项解决方案来解决相关的性能问题。

常见反模式：

- 您认为 CPUUtilization 或内存压力等标准计算指标足以捕获性能问题。
- 您只使用由自己选定的监控软件记录的默认指标。
- 您只在出现问题时审查指标。

建立此最佳实践的好处： 了解关键性能领域可以帮助工作负载负责人监控 KPI 并确定具有高影响力的优先改进。

未建立这种最佳实践的情况下暴露的风险等级： 高

实施指导

设置端到端的跟踪，用于确定流量模式、延迟和关键性能领域。针对速度缓慢的查询或性能欠佳的碎片和分区数据，监控数据访问模式。使用负载测试或监控来确定受约束的工作负载领域。

实施步骤

1. 设置端到端的监控，用于收集所有工作负载组件和指标。
   - 使用 Amazon CloudWatch 真实用户监控（RUM，Real-User Monitoring） 来收集真实用户客户端和前端会话的应用程序性能指标。
   - 设置 AWS X-Ray 以通过应用程序层跟踪流量，并确定组件间的延迟以及依赖关系。使用 X-Ray 服务地图查看工作负载组件之间的关系和延迟。
   - 使用 Amazon Relational Database Service Performance Insights 查看数据库性能指标并确定性能改进机会。
   - 使用 Amazon RDS 增强监控 查看数据库 OS 性能指标。
   - 收集每个工作负载组件和服务的 CloudWatch 指标 ，确定哪些指标影响性能效率。
   - 设置 Amazon DevOps Guru 以获取额外的性能见解和推荐方案
2. 执行测试以生成指标，确定流量模式、瓶颈和关键性能领域。
   - 设置 CloudWatch Synthetic Canary 以使用 cron 作业或速率表达式，通过编程方式模拟浏览器端的用户活动，从而生成一段时间内的稳定指标。
   - 使用 AWS 分布式负载测试 解决方案生成峰值流量，或者在预期增长速率下测试工作负载。
3. 评估指标和遥测数据，确定您的关键性能领域。与团队一起审查这些领域，讨论监控和解决方案以避免瓶颈。
4. 试验性能改进，并利用数据来衡量这些更改。
   - 使用 CloudWatch Evidently 测试新的改进以及对工作负载的性能影响。

实施计划的工作量级别： 要建立这种最佳实践，您必须审查端到端指标并密切关注当前的工作负载性能。设置端到端监控和确定关键性能领域所需的工作量为中等。

资源

相关文档：

- Amazon Builders' Library
- X-Ray 文档
- Amazon CloudWatch RUM
- Amazon DevOps Guru
- CloudWatch RUM 和 X-Ray

相关视频：

- 亚马逊开发构建者资料库简介（DOP328）
- Amazon CloudWatch Synthetics 演示

相关示例：

- 使用 Amazon CloudWatch Synthetics 测量页面加载时间
- Amazon CloudWatch RUM Web 客户端
- 适用于 Node.js 的 X-Ray 开发工具包
- 适用于 Python 的 X-Ray 开发工具包
- 适用于 Java 的 X-Ray 开发工具包
- 适用于 .Net 的 X-Ray 开发工具包
- 适用于 Ruby 的 X-Ray 开发工具包
- X-Ray 进程守护程序
- AWS 上的分布式负载测试

PERF08-BP02 了解设计模式和服务

研究和理解有助于提高工作负载性能的各种设计模式和服务。在分析的过程中，确定您需要牺牲哪些方面来获得更高的性能。例如，使用缓存服务有助于减少数据库系统上的负载。然而，缓存会带来最终一致性问题，这就需要在业务要求和客户期望的范围内进行工程设计。

期望结果： 通过研究设计模式，您可以选择将支持性能卓越系统的架构设计。了解您可以使用哪些性能配置选项以及这些配置选项对工作负载的影响。优化工作负载性能依赖于对以下内容的了解：这些选项如何与架构进行交互，以及这些选项对实际测量的性能和终端用户感知到的性能的影响。

常见反模式：

- 您可以假设所有传统 IT 工作负载性能策略最适合云工作负载。
- 您可以构建和管理缓存解决方案，而不使用托管服务。
- 您对所有工作负载都使用相同的设计模式，而不评估哪种模式会提高工作负载性能。

建立此最佳实践的好处： 通过为您的工作负载选择正确的设计模式和服务，您将优化性能，实现卓越运营并提高可靠性。正确的设计模式将满足您当前的工作负载特征，并帮助您扩展以适应未来的增长或变更。

未建立这种最佳实践的情况下暴露的风险等级： 高

实施指导

了解哪些性能配置选项可用，以及这些配置选项对工作负载的影响。优化工作负载性能依赖于对以下内容的了解：这些选项如何与架构进行交互，以及这些选项对实际测量的性能和用户感知到的性能的影响。

实施步骤：

1. 评估和审核可以提高工作负载性能的设计模式。
    a. 如示例所示， Amazon Builders' Library 为您提供了有关亚马逊如何构建和运营技术的详细说明。这些文章均由亚马逊的高级工程师撰写，其中涵盖架构、软件交付和运营等诸多主题。
    b. AWS 解决方案库 是一个随时可部署的解决方案集合，汇集了服务、代码和配置。这些解决方案是由 AWS 和 AWS 合作伙伴基于按行业或工作负载类型分组的常见使用场景和设计模式创建而成。例如，您可以为工作负载设置 分布式负载测试解决方案 。
    c. AWS Architecture Center 提供按设计模式、内容类型和技术进行分组的参考架构图。
    d. AWS 示例 是一个包含大量实践示例的 GitHub 存储库，可帮助您探索常见的架构模式、解决方案和服务。它经常更新，提供最新的服务和示例。
2. 改进您的工作负载，以对所选的设计模式建模，并使用服务和服务配置选项来提高您的工作负载性能。
    a. 利用 AWS Skills Guild提供的资源对您的内部团队进行培训。
    b. 使用 AWS Partner Network 快速提供专业知识，并增强自己作出改进的能力。

实施计划的工作量级别： 要建立这种最佳实践，您必须了解有助于提高工作负载性能的设计模式和服务。对设计模式进行评估后，实施设计模式的工作量比较 大 。

资源

相关文档：

- AWS Architecture Center
- AWS Partner Network
- AWS 解决方案库
- AWS Knowledge Center
- Amazon Builders' Library
- 使用负载脱落来避免过载
- 缓存挑战和策略

相关视频：

- Amazon Builders' Library 简介（DOP328）
- 这是我的架构

相关示例：

- AWS 示例
- AWS 开发工具包示例

PERF08-BP03 确定权衡机制对客户和效率的影响

在评估与性能相关的改进时，确定哪些选择会对客户和工作负载效率产生影响。例如，如果使用键值数据存储可以提高系统性能，那么评估它的最终一致性将对客户的影响就非常重要。

通过指标和监控确定系统中性能不佳的方面。确定如何提高性能、性能提高带来的利弊，并确定性能提高对系统和用户体验的影响。例如，缓存数据有助于大幅提高性能，但需要就如何以及何时更新缓存的数据或使其变得无效而制定明确的策略，以防止产生不正确的系统行为。

常见反模式：

- 您可以假设所有性能收益都应实现，即使有一些权衡机制要实施，例如，最终一致性。

- 在性能问题已经非常严重时，您只需评估对工作负载的更改。

建立此最佳实践的好处： 当您评估潜在性能相关的改进时，必须决定更改时所采用的权衡机制是否符合工作负载要求。在某些情况下，您可能需要实施额外的控制来补偿权衡机制。

未建立此最佳实践暴露的风险等级： 高

实施指导

确定权衡机制：通过指标和监控确定系统中性能不佳的方面。确定如何进行改进，以及权衡机制将如何影响系统和用户体验。例如，实施缓存数据有助于大幅提高性能，但需要就如何以及何时更新缓存的数据或使其作废而制定明确的策略，以防止产生不正确的系统行为。

资源

相关文档：

- Amazon Builders' Library
- Amazon QuickSight KPI
- Amazon CloudWatch RUM
- X-Ray 文档

相关视频：

- Amazon Builders' Library 简介 (DOP328)
- 制定监控计划
- 通过 Amazon CloudWatch RUM 优化应用程序
- Amazon CloudWatch Synthetics 演示

相关示例：

- 使用 Amazon CloudWatch Synthetics 测量页面加载时间
- Amazon CloudWatch RUM Web 客户端

PERF08-BP04 衡量性能提高产生的影响

在进行更改以提高性能时，对收集的指标和数据进行评估。使用这些信息来确定性能提高对工作负载、工作负载组件和客户的影响。这种衡量可让您了解采用权衡机制后实现的性能提高，还可以帮助确定性能提高是否产生了任何不利的副作用。

架构完善的系统会使用各种与性能相关的策略。确定哪种策略会对给定的热点或瓶颈产生最大的积极影响。例如，对多个关系数据库系统中的数据进行分片可以提高整体吞吐量并保持对事务的支持，而且在每个分片内进行缓存有助于降低负载。

常见反模式：

- 您可以手动部署和管理作为托管服务提供的技术。
- 当有多个组件可用于提高工作负载的性能时，您可以只专注于一个组件，如联网。
- 您依赖客户反馈和看法，将其作为唯一的基准。

建立此最佳实践的好处： 要实施性能策略，您必须选择多个服务和功能相结合的方式，以满足工作负载的性能要求。

未建立这种最佳实践的情况下暴露的风险等级： 中

实施指导

架构完善的系统会结合使用各种与性能相关的策略。确定哪种策略会对给定的热点或瓶颈产生最大的积极影响。例如，对多个关系数据库系统中的数据进行分片可以提高整体吞吐量并保持对事务的支持，而且在每个分片内进行缓存有助于降低负载。

资源

相关文档：

- Amazon Builders' Library
- Amazon CloudWatch RUM
- Amazon CloudWatch Synthetics
- AWS 上的分布式负载测试

相关视频：

- Amazon Builders' Library 简介 (DOP328)

- 通过 Amazon CloudWatch RUM 优化应用程序
- Amazon CloudWatch Synthetics 演示

相关示例：

- 使用 Amazon CloudWatch Synthetics 测量页面加载时间
- Amazon CloudWatch RUM Web 客户端
- AWS 上的分布式负载测试

PERF08-BP05 使用各种与性能相关的策略

如果合适，使用多种策略来提高性能。例如，可以使用缓存数据等策略来防止出现过多的网络或数据库调用；使用数据库引擎的只读副本来提高读取速度；尽可能对数据进行分片或压缩以减少数据卷；在数据可用时进行缓冲和流式处理，避免拥堵。

对工作负载进行更改时，需要收集并评估各项指标，以确定更改产生的影响。衡量对系统和最终用户的影响，以便了解权衡机制如何影响工作负载。使用负载测试等系统的方法来确定权衡机制是否可以提高性能。

常见反模式：

- 如果客户没有提出意见，您可以认为工作负载性能足够高。
- 在进行性能相关的更改后，您只需收集关于性能的数据。

建立此最佳实践的好处： 要优化性能和提高资源利用率，您需要一个统一的运营视图、实时精细数据和历史参考。您可以创建控制面板并对数据执行指标计算，以便在工作负载随着时间的推移而变化时，获得工作负载的运营和利用率见解。

未建立这种最佳实践的情况下暴露的风险等级： 低

实施指导

使用数据驱动型方法来改进架构：对工作负载进行更改时，需要收集并评估各项指标，以确定更改产生的影响。衡量对系统和最终用户的影响，以便了解权衡机制如何影响工作负载。使用负载测试等系统的方法来确定权衡机制是否可以提高性能。

资源

相关文档：

- Amazon Builders' Library
- 实施 Amazon ElastiCache 的最佳实践
- AWS 数据库缓存
- Amazon CloudWatch RUM
- AWS 上的分布式负载测试

相关视频：

- Amazon Builders' Library 简介 (DOP328)
- AWS 专用数据库（DAT209-L）
- 通过 Amazon CloudWatch RUM 优化应用程序

相关示例：

- 使用 Amazon CloudWatch Synthetics 测量页面加载时间
- Amazon CloudWatch RUM Web 客户端
- AWS 上的分布式负载测试

成本优化

主题

- 践行云财务管理
- 支出和使用情况意识
- 具有成本效益的资源
- 管理需求和供应资源
- 随着时间的推移不断优化

践行云财务管理

问题

- COST 1 如何实施云财务管理？

COST 1 如何实施云财务管理？

实施云财务管理后，组织可以在 AWS 上优化成本和使用情况并进行扩展，从而实现商业价值和财务成功。

最佳实践

- COST01-BP01 建立成本优化部门
- COST01-BP02 在财务和技术人员之间建立合作关系
- COST01-BP03 建立云预算和预测
- COST01-BP04 在组织流程中落实成本意识
- COST01-BP05 报告和通知成本优化
- COST01-BP06 主动监控成本
- COST01-BP07 及时了解新发布的服务

COST01-BP01 建立成本优化部门

创建一个团队（云业务办公室或云卓越中心），负责在整个组织内建立并维护成本意识。该团队需要整个组织内担任财务、技术和业务角色的人员加入。

未建立这种最佳实践的情况下暴露的风险等级： 高

实施指导

创建一个云业务办公室（CBO）或云卓越中心（CCOE）团队，负责建立并维护一种对云计算成本敏感的文化。它可以是一个现有的个人、组织内的一个团队，也可以是整个组织中由关键财务、技术和组织利益相关者组成的新团队。

此部门（个人或团队）会排定成本管理和成本优化活动的优先级，并根据需要为这些活动投入一定比率的时间。相对于较大型企业中的全职部门，小型组织的这一部门在此方面花费的时间可能更少。

此部门需要采取多学科方法，并具备项目管理、数据科学、财务分析和软件或基础设施开发的能力。此部门可通过在三个不同的责任归属范围内执行成本优化来提高工作负载的效率：

- 集中式： 通过指定的团队（如财务运营、成本优化、CBO 或 CCOE），客户可以设计并实施治理机制，并在全公司范围内推动最佳实践。
- 分散式： 对技术团队施加影响来执行优化。

- 混合式： 集中式和分散式团队可以合作执行成本优化。

可以对照成本优化目标（例如工作负载效率指标）来衡量此部门的执行和交付能力。

您必须为此部门获得高管支持才能促成改变，这是取得成功的一个关键因素。支持者即低成本云消费理念的倡导者，他们会为此部门提供升级支持，确保按组织确定的优先级开展成本优化活动。否则，相关方面会忽视指导意见，并且不会优先考虑节省成本的机会。此部门及其支持者会共同确保组织在有效利用云资源，并持续创造业务价值。

如果您制定了商业、企业入门或企业支持计划，并需要帮助来建立此团队或部门，请通过您的客户团队联系云财务管理（CFM）专家。

实施步骤

- 定义主要成员： 您需要确保组织的所有相关组成部分都参与到成本管理中。组织中的常见团队通常包括：财务、应用程序或产品负责人、管理和技术团队（DevOps）。有些是全职工作（财务、技术），有些是根据需要定期工作。执行 CFM 的个人或团队通常需要掌握以下技能：
  - 软件开发技能 - 在构建脚本和自动化的情况下。
  - 基础设施工程技能 - 部署脚本或自动化，并了解如何预置服务或资源。
  - 运营敏锐性 - CFM 的宗旨是通过测量、监控、修改、规划和扩展对云的有效使用，在云上高效地运行。
- 定义目标和指标： 该部门需要以不同的方式为组织创造价值。相关目标已确定，并将随着组织的发展而不断完善。常见活动包括：在整个组织内创建并执行关于成本优化的培训计划、制定组织范围标准，例如监控和报告成本优化，以及设置关于优化的工作负载目标。该部门还需要定期向组织报告组织的成本优化能力。

  您可以定义基于价值的关键性能指标（KPI）。KPI 可以基于成本，也可以基于价值。定义 KPI 时，可以根据效率和预期业务成果计算预期成本。基于价值的 KPI 将成本和使用情况指标与业务价值驱动因素联系起来，并帮助我们合理调整 AWS 支出。获得基于价值的 KPI 的第一步是跨组织协作，选择并商定一组标准 KPI。
- 建立定期沟通机制： 该小组（财务、技术和业务团队）需要定期聚在一起，审核目标和指标。典型的定期沟通机制包括审核组织的状态，审核当前运行的任何计划，审核整体财务和优化指标。然后，更详细地报告关键工作负载。

  在这些定期会议上，您可以审核工作负载效率（成本）和业务成果。例如，工作负载的成本增加 20% 可能与客户使用量的增加相一致。在这种情况下，这 20% 的成本增加可以理解为一项投资。这些定期沟通通话可以帮助团队识别基于价值的 KPI，为整个组织带来意义。

资源

相关文档：

- AWS CCOE 博客
- 创建云业务办公室
- CCOE - 云卓越中心

相关视频：

- Vanguard CCOE 成功案例

相关示例：

- 使用云卓越中心（CCoE）实现整个企业转型
- 构建 CCOE 以实现整个企业转型
- 构建 CCOE 时应避免的 7 个陷阱

COST01-BP02 在财务和技术人员之间建立合作关系

在云之旅的所有阶段，都让财务和技术团队参与成本和使用情况的讨论。团队定期开会，讨论组织目标、成本和使用情况的当前状态以及财务和会计实务等主题。

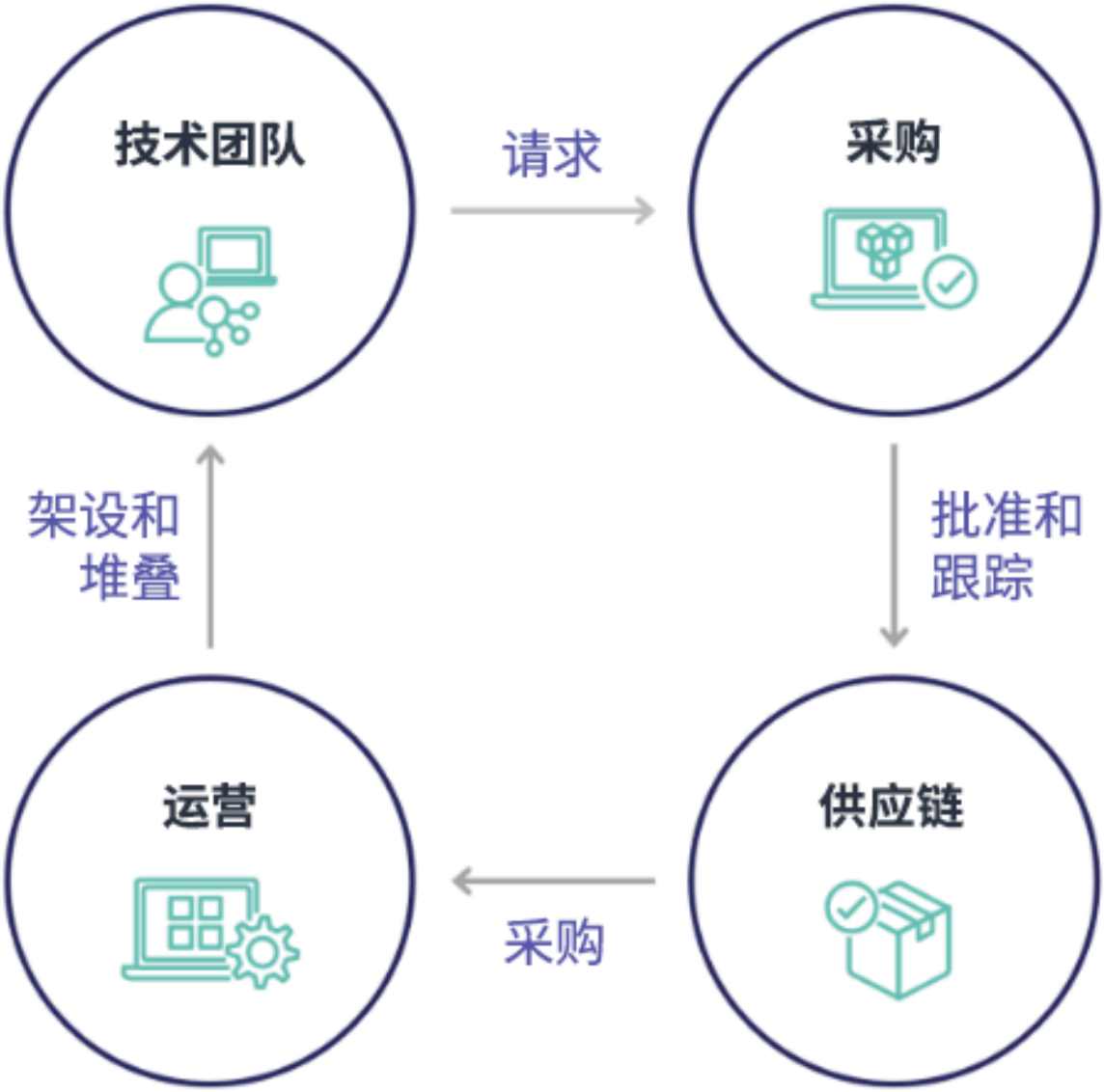
未建立这种最佳实践的情况下暴露的风险等级： 高

实施指导

由于缩短了审批、采购和基础设施部署周期，技术团队在云端的创新速度更快。这可能是对财务组织的一种调整，以前，他们习惯于执行耗时的资源密集型流程，以便在数据中心和本地环境中获取和部署资金，并且只在项目批准时进行成本分配。

从财务和采购组织的角度来看，资本预算编制、资本请求、审批、采购和安装物理基础设施这一流程是我们几十年来一直在学习和标准化的流程：

- 工程团队或 IT 团队通常是请求者
- 不同的财务团队充当审批者和采购者
- 运营团队架设、堆叠并移交可供使用的基础设施



随着云的采用，基础设施的采购和消费不再受制于一连串的依赖关系。在云模式下，技术和产品团队不再仅仅是构建者，还是产品的运营者和负责人，他们负责过去与财务和运营团队有关的大部分活动，包括采购和部署。

预置云资源真正需要的只是一个用户账户和一组正确的权限。这也有助于降低 IT 和财务风险，因为团队只需点击几下鼠标或进行几次 API 调用，就可以终止空闲或不必要的云资源。这也使技术团队能够更快创新，因为他们获得了启动实验以及之后拆除实验的敏捷性和能力。虽然从资本预算编制和预测的角度来看，云消费的可变性质可能会影响可预测性，但云为组织提供了降低过度预置成本的能力，以及降低与保守预置不足相关的机会成本的能力。

在关键的财务和技术利益相关者之间建立合作关系，让他们就组织目标达成共识，并开发在云计算的可变支出模型中获得财务成功的机制。组织内的相关团队必须在云之旅的各个阶段参与成本和使用量讨论，包括：

- 财务领导： 首席财务官、财务总监、财务规划师、业务分析师、采购、供应商开发人员和应付账款负责人必须了解消费、采购选项和每月开票流程的云模型。财务部门需要与技术团队合作创建有关 IT 价值的沟通内容并广泛传播，帮助业务团队了解技术支出与业务成果之间的联系。这样，技术支出就不会被视为成本，而会被视为投资。由于云运营（如使用量的变化速率、即付即用的定价、分级定价、定价模型以及详细的计费和使用信息）与本地运营之间存在根本差异，财务组织必须了解云的使用对业务方面的影响，包括采购流程、激励跟踪、成本分配和财务报表。
- 技术领导： 技术领导（包括产品和应用程序负责人）必须了解财务要求（如预算限制）和业务要求（如服务水平协议）。如此才能实施工作负载并实现组织的预期目标。

财务与技术人员的合作可带来以下好处：

- 财务和技术团队几乎可以实时看到成本和使用量。
- 财务和技术团队建立了标准的操作程序来处理云支出差异。
- 在如何使用资本购买承诺折扣（例如预留实例或 AWS Savings Plans）以及如何使用云来发展组织方面，财务利益相关者充当战略顾问。
- 将现有的应付账款和采购流程用于云部署。
- 财务和技术团队协作预测未来的 AWS 成本和使用量，以调整和建立组织预算。
- 通过共通的语言以及对财务概念的一致理解，更好地进行跨组织沟通。

组织中应参与成本和使用量讨论的其他利益相关者包括：

- 业务部门负责人： 业务部门负责人必须了解云业务模式，从而为业务部门和整个公司提供指导。在需要预测增长和工作负载使用情况，以及评估不同购买选项（例如预留实例或 Savings Plans）时，这方面的云知识至关重要。
- 工程团队： 在财务和技术团队之间建立合作关系对于建立对成本敏感的文化至关重要，这可以鼓励工程师在云财务管理（CFM）上采取行动。CFM 或财务运营从业者和财务团队的一个常见问题是让工程师了解云上的整个业务，遵循最佳实践，并采取建议的行动。
- 第三方： 如果您的组织使用第三方（例如顾问或工具），请确保他们与您的财务目标一致，并且可以通过参与模式和投资回报（ROI）证明一致性。第三方通常会帮助报告和分析其管理的任何工作负载，并且提供他们设计的任何工作负载的成本分析。

要想实施 CFM 并取得成功，需要财务、技术和业务团队之间彼此协作，并在整个组织内就如何转变云支出进行沟通和评估。将工程团队包括在内，让他们在所有阶段参与这些成本和使用情况的讨论；还要鼓励他们遵循最佳实践并采取相应的商定行动。

实施步骤

- 定义主要成员： 确认财务和技术团队的所有相关成员都参与合作。相关财务成员将是与云账单进行交互的人员。通常是首席财务官、财务总监、财务规划师、业务分析师和采购员。技术成员通常是产品和应用程序负责人、技术经理和所有在云上执行构建的团队的代表。其他成员可能包括业务部门负责人（例如影响产品使用的市场营销部门）和第三方（例如顾问），以确保与您的目标和机制保持一致，并协助进行报告。
- 定义讨论主题： 定义团队之间的共同主题，或者需要达成共识的主题。从生成成本之时跟踪成本，直到账单已付为止。请注意所涉及的任何成员，以及需要应用的组织流程。了解经过的每个步骤或流程以及相关信息，例如可用的定价模式、分级定价、折扣模型、预算和财务要求。

- 建立定期沟通机制： 为了让财务和技术人员展开合作，建立定期沟通机制，以促进并保持一致。该小组需要定期聚在一起，讨论目标和指标。典型的定期沟通机制包括审核组织的状态，审核当前运行的任何计划，审核整体财务和优化指标。然后，更详细地报告关键工作负载。

资源

相关文档：

- AWS 新闻博客

COST01-BP03 建立云预算和预测

调整现有的组织预算和预测流程，使之适应云成本和使用情况的易变特性。流程必须是动态的，可以使用基于趋势或基于业务驱动因素的算法，也可以将两者结合使用。

未建立这种最佳实践的情况下暴露的风险等级： 高

实施指导

客户使用云来提高效率、速度和敏捷性，这导致成本和使用量的变化速度极快。随着工作负载效率的提高或者新工作负载和功能的部署，成本可以降低。当工作负载效率提高或部署新的工作负载和功能时，成本也可能增加。或者，工作负载将扩展以服务更多的客户，这会增加云的使用量和成本。现在比以前更容易获得资源。云的弹性也使成本和预测变得具有弹性。必须修改现有的组织预算流程，将这种变化因素考虑在内。

使用基于趋势（将历史成本用作输入）或者基于业务驱动因素（例如新产品发布或区域扩张）的算法，或者将趋势和业务驱动因素相结合，调整现有的预算和预测流程，使其更为灵活。

使用 AWS Budgets 设置精细的自定义预算，通过指定时间段、重复发生或金额（固定或可变），以及添加筛选条件（例如服务、AWS 区域和标签）来实现。为了随时了解现有预算的执行情况，您可以创建 AWS Budgets 报告 并安排好时间表，定期以电子邮件的形式发送给您和您的利益相关者。您还可以创建 AWS Budgets 警报， 该警报可以根据实际成本（本质上是被动的）创建或根据预测成本（从而留出时间缓解潜在的成本超支情况）创建。您的成本或使用量超出或预计将超出预算金额时，系统会向您发送警报。

AWS 可以帮助您灵活构建动态预测和预算制定流程，因此您可以随时了解成本是否达到或超出预算限制。

使用 AWS Cost Explorer ，根据历史支出预测所定义的未来时间范围内的成本。AWS Cost Explorer 的预测引擎会根据付费类型（例如，预留实例）对您的历史数据进行细分，并结合使用机器学习和基于

规则的模型来分别预测所有付费类型的支出。使用 AWS Cost Explorer ，基于应用至历史成本（基于趋势）的机器学习算法，预测每日（最多 3 个月）或每月（最多 12 个月）的云成本。

使用 Cost Explorer 确定了基于趋势的预测后，请使用 AWS Pricing Calculator ，根据预期使用情况（流量、每秒请求数、所需 Amazon Elastic Compute Cloud（Amazon EC2）实例等）估计 AWS 使用场景和未来成本。您也可以用它来帮助您计划支出方式，找到节省成本的机会，并在使用 AWS 时做出明智的决定。

使用 AWS Cost Anomaly Detection 防止或减少意外成本，加强控制，同时不放慢创新速度。AWS Cost Anomaly Detection 利用先进的机器学习技术来识别异常支出并找出根本原因，使您能够快速采取行动。只需简单三步，您就可以创建自己的情境化监控器，在检测到任何异常支出时接收警报。让生成器专门负责生成，让 AWS Cost Anomaly Detection 监控您的支出并降低账单意外的风险。

如 “Well-Architected 成本优化支柱之财务与技术人员合作” 部分所述，在 IT 部门、财务部门和其他利益相关者之间建立合作关系和沟通机制非常重要，可以确保他们都使用相同的工具或流程来保持一致性。在预算可能需要更改的情况下，增加沟通机制接触点有助于更快地应对这些更改。

实施步骤

- 更新现有预算和预测流程： 在预算和预测流程中实施基于趋势或基于业务驱动因素的方法，或者两种方法结合应用。
- 配置警报和通知： 使用 AWS Budgets 警报和 Cost Anomaly Detection。
- 与主要利益相关者定期审核： 例如，与 IT、财务、平台部门和其他业务领域的利益相关者进行定期审核，以与业务方向和使用方面的变化保持一致。

资源

相关文档：

- AWS Cost Explorer
- AWS Budgets
- AWS Pricing Calculator
- AWS Cost Anomaly Detection
- AWS License Manager

相关示例：

- 发布：AWS Cost Explorer 中现在提供基于使用情况的预测

- AWS Well-Architected 实验室 - 成本和使用情况治理

COST01-BP04 在组织流程中落实成本意识

在影响使用量的新流程或现有流程中落实成本意识、创建成本透明度和成本问责制，并利用现有流程提高成本意识。在员工培训中贯彻成本意识。

未建立这种最佳实践的情况下暴露的风险等级： 高

实施指导

必须在新的和现有的组织流程中建立成本意识。它是其他最佳实践的基础性先决条件之一。建议在可能的情况下重用和修改现有流程，这可以更大限度地减少对敏捷性和速度的影响。向技术团队以及业务和财务团队的决策者报告云成本，以提高成本意识，并为财务和业务利益相关者建立效率关键性能指标（KPI）。以下建议有助您在工作负载中建立成本意识：

- 验证变更管理包含成本度量，以量化变更对财务的影响。这有助于主动解决与成本相关的问题，并强调成本节省。
- 验证成本优化是您运营能力的核心组成部分。例如，您可以利用现有的意外事件管理流程来调查和确定成本及使用量异常或成本超支的根本原因。
- 通过自动化或工具加快节省成本和实现业务价值。在考虑实施成本时，请在对话中加入投资回报（ROI）信息，以证明投入时间或资金的合理性。
- 通过对云支出（包括在基于承诺的购买选项、共享服务和市场购买上的支出）实施 showback 或 chargeback 来分配云成本，以推动极具成本意识的云消费。
- 扩展现有的培训和发展计划，在整个组织中开展成本意识培训。建议在其中加入持续的培训和认证。这样有助建立一个能够自我管理成本和使用量的组织。
- 利用免费的 AWS 原生工具，比如 AWS Cost Anomaly Detection，AWS Budgets和 AWS Budgets 报告.

当组织持续采用 云财务管理 （CFM）实践时，这些行为就会在工作和决策过程中落地生根。最终带来的是在从架构新的云原生应用程序的开发人员，到分析这些新的云投资的 ROI 的财务经理之间建立起更加注重成本的企业文化。

实施步骤

- 识别相关的组织流程： 每个组织单位审核自己的流程，并确定影响成本和使用情况的流程。任何导致资源创建或终止的流程都需要进行审核。查找能够在企业中支持成本意识的流程，例如事件管理和培训。

- 建立自我维持的成本意识文化： 确保所有相关的利益相关者都认同变更原因和影响是一种成本，这样他们就能理解云成本。这将使贵组织能够在创新方面建立一种自我维持的成本意识文化。
- 更新流程，增加成本意识： 每个流程都进行修改，将成本意识纳入其中。流程可能需要执行额外的预检查，例如评估成本的影响，或执行后期检查，以验证成本和使用情况是否发生了预期变化。可以扩展支持流程（如培训和事件管理），以包括成本和使用情况项目。

要获得帮助，请通过您的客户团队与 CFM 专家联系，或浏览以下资源和相关文档。

资源

相关文档：

- AWS 云财务管理

相关示例：

- 高效云成本管理战略
- 成本控制博客系列 3：如何应对成本冲击
- AWS Cost Management 初学者指南

COST01-BP05 报告和通知成本优化

配置 AWS Budgets 和 AWS Cost Anomaly Detection，以便对照目标提供成本和使用情况通知。定期举行会议来分析工作负载的成本效益并推广对成本敏感的文化。

未建立这种最佳实践的情况下暴露的风险等级： 低

实施指导

必须定期报告组织内成本和使用量的优化情况。可以设定专门的成本优化环节，或者将成本优化纳入工作负载的常规运营报告周期。使用服务和工具来识别和实施节省成本的机会。AWS Cost Explorer 提供控制面板和报告。可以通过 AWS Budgets 报告.

使用 AWS Budgets 设置自定义预算以跟踪成本和使用情况，并在您超过阈值时快速响应从电子邮件或 Amazon Simple Notification Service（Amazon SNS）通知收到的警报。将首选预算 期设置为每日、每月、每季度或每年，并创建具体的预算限制，以随时了解实际或预测成本和使用量相对于预算阈值的情况。您还可以将 警报 和 针对这些警报的操作 设置为自动运行，或在超出预算目标时通过审批流程运行。

启用关于成本和使用情况的通知，以确保在成本和使用情况发生意外变化时能够迅速采取行动。AWS Cost Anomaly Detection 使您能够减少意外成本，加强控制，同时不放慢创新速度。AWS Cost Anomaly Detection 可识别异常支出并找出根本原因，这有助于降低账单意外的风险。只需简单三步，您就可以创建自己的情境化监控器，在检测到任何异常支出时接收警报。

您也可以使用 Amazon QuickSight 与 AWS 成本和使用情况报告（CUR）数据，以提供包含更精细数据的高度定制的报告。利用 Amazon QuickSight，您可以安排报告，并定期收到关于历史成本和使用情况或成本节省机会的成本报告电子邮件。

使用 AWS Trusted Advisor，它可提供指导，以验证预置的资源是否符合 AWS 的成本优化最佳实践。

定期创建报告，其中包含来自 AWS Cost Explorer 的 Savings Plans、预留实例和 Amazon Elastic Compute Cloud（Amazon EC2）合理调整大小建议的提要，以开始降低与稳态工作负载、空闲和未充分利用的资源相关的成本。识别并收回与已部署资源的云浪费有关的支出。当创建的资源大小不正确，或者观察到不同于预期的使用模式时，就会发生云浪费。遵循 AWS 最佳实践，以减少浪费，优化并节省 云成本。

定期生成报告，为您的资源提供更好的购买选项，以降低工作负载的单位成本。诸如 Savings Plans、预留实例或 Amazon EC2 竞价型实例等购买选项可为容错工作负载节省大量成本，并使利益相关者（业务负责人、财务和技术团队）能够参与有关这些承诺的讨论。

分享包含可能有助于降低云总拥有成本（TCO）的机会或新发布公告的报告。采用新的服务、区域、功能、解决方案或新的方式来进一步削减成本。

实施步骤

- 配置 AWS Budgets： 在所有账户中为您的工作负载配置 AWS Budgets。通过使用标签设置账户总支出预算和工作负载预算。
  - Well-Architected 实验室：成本和治理使用情况
- 报告成本优化： 定期讨论和分析工作负载的效率。使用已确立的指标，报告实现的指标和实现成本。找出任何负面趋势，加以解决，并确定可以在整个组织中推广的正面趋势。报告应该包括应用程序团队和负责人、财务团队和管理团队的代表。
  - Well-Architected 实验室：可视化

资源

相关文档：

- AWS Cost Explorer

- AWS Trusted Advisor
- AWS Budgets
- AWS Budgets 最佳实践
- Amazon CloudWatch
- AWS CloudTrail
- Amazon S3 分析
- AWS 成本和使用情况报告

相关示例：

- Well-Architected 实验室：成本和治理使用情况
- Well-Architected 实验室：可视化
- 开始优化 AWS 云成本的关键方法

COST01-BP06 主动监控成本

利用工具和控制面板主动监控工作负载的成本。定期用已配置的工具或开箱即用的工具审核成本，不要只在收到通知时才查看成本和类别。主动监控和分析成本有助于识别积极趋势，使您能够在整个组织中推广这些趋势。

未建立这种最佳实践的情况下暴露的风险等级： 低

实施指导

建议在组织内部主动监控成本和使用量，而不仅仅是在出现异常或意外时。在整个办公室或工作环境中，一目了然的仪表板能确保关键人员可以访问所需信息，并凸显组织对成本优化的重视程度。通过可见的控制面板，您可以积极推动成功的结果，并在整个组织中加以实施。

创建一个每日或经常性的例程，以使用 AWS Cost Explorer 或任何其他控制面板（如 Amazon QuickSight ）来查看成本并主动分析。通过分组和筛选，在 AWS 账户级、工作负载级或特定 AWS 服务级分析 AWS 服务的使用情况和成本，并验证它们是否符合预期。使用小时级和资源级粒度和标签来筛选和识别主要资源所产生的成本。您还可以使用 Cost Intelligence Dashboard（一种 Amazon QuickSight 解决方案，由 AWS 解决方案架构师构建）构建自己的报告，并将预算与实际成本和使用情况进行比较。

实施步骤

- 报告成本优化： 定期讨论和分析工作负载的效率。使用已确立的指标，报告实现的指标和实现成本。找出任何负面趋势，加以解决，并确定积极趋势，以便在整个组织中推广。报告应该包括应用程序团队和负责人、财务团队和管理团队的代表。
- 为成本和使用情况创建并启用每日粒度的 AWS Budgets ，以便及时采取措施防止任何潜在的成本超支情况： 您可以使用 AWS Budgets 配置警报通知，以便在任何预算类型超出预配置阈值时，都会得到通知。利用 AWS Budgets 的最佳方式是将预期成本和使用量设置为您的限值，这样一来，任何超出预算的情况均视为超支。
- 为成本监控器创建 AWS Cost Anomaly Detection： AWS Cost Anomaly Detection 使用先进的机器学习技术来识别异常支出和根本原因，以便您快速采取行动。您可以利用它来配置成本监控器，以定义您想要评估的支出部分（例如，各项 AWS 服务、成员账户、成本分配标签和成本类别），并设置何时、何地以及如何接收警报通知。对于每个监控器，为业务负责人和技术团队附加多个警报订阅，包括每个订阅的名称、成本影响阈值和警报频率（单个警报、每日总结、每周总结）。
- 使用 AWS Cost Explorer 或将您的 AWS 成本和使用情况报告（CUR）数据与 Amazon QuickSight 控制面板集成，使贵组织的成本可视化： AWS Cost Explorer 有一个易于使用的界面，使您能够可视化、理解和管理随时间变化的 AWS 成本和使用情况。使用 Cost Intelligence Dashboard ，这是一个可定制且可访问的控制面板，可帮助创建您自己的成本管理和优化工具的基础。

资源

相关文档：

- AWS Budgets
- AWS Cost Explorer
- 每日成本和使用情况预算
- AWS Cost Anomaly Detection

相关示例：

- Well-Architected 实验室：可视化
- Well-Architected 实验室：高级可视化
- Well-Architected 实验室：云智能控制面板
- Well-Architected 实验室：成本可视化
- AWS Cost Anomaly Detection 警报与 Slack 集成

COST01-BP07 及时了解新发布的服务

定期咨询专家或 AWS 合作伙伴，以便确定哪些服务和功能的成本更低。查看 AWS 博客和其他信息源。

未建立这种最佳实践的情况下暴露的风险等级： 低

实施指导

AWS 不断地添加新的功能，以便您可以利用前沿技术来更快地进行试验和创新。您或许可以实施新的 AWS 服务和功能，以提高工作负载的成本效率。定期查看 AWS 成本管理、 AWS 新闻博客、 AWS 成本管理博客和 AWS 新增功能 以了解有关新服务和功能发布的信息。新增功能博客文章简要概述了 AWS 发布的所有服务、功能和区域扩展公告。

实施步骤

- 订阅博客： 转到 AWS 博客页面，订阅新增功能博客和其他相关博客。您可以在 通信偏好 页面上用您的电子邮件地址进行注册。
- 订阅 AWS 新闻： 定期查看 AWS 新闻博客 和 AWS 新增功能 以了解有关新服务和功能发布的信息。订阅 RSS 源，或通过电子邮件关注公告和发布的信息。
- 关注 AWS 降价： 我们会定期对各项服务进行降价，这是 AWS 将我们的规模所带来的经济效益传递给客户的一种标准方式。截至 2022 年 4 月，AWS 自 2006 年推出以来已经降价 115 次。如果您有任何因价格问题而悬而未决的业务决策，可在降价和新服务整合后再次考虑这些决策。您可以了解以前的降价情况，包括 Amazon Elastic Compute Cloud（Amazon EC2）实例 （在 AWS 新闻博客的降价类别中了解这些情况）。
- AWS 活动和交流会： 参加当地的 AWS 峰会，以及与当地其他组织的任何当地交流会。如果您不能亲自参加，请尝试参加虚拟活动，从 AWS 专家和其他客户的业务案例中了解更多信息。
- 与客户团队交流： 定期与您的客户团队交流，讨论行业发展趋势和 AWS 服务。与您的客户经理、解决方案架构师和支持团队沟通。

资源

相关文档：

- AWS 成本管理
- AWS 新增功能
- AWS 新闻博客

相关示例：

- Amazon EC2 – 15 年来为您优化和节省 IT 成本
- AWS 新闻博客 - 降价

支出和使用情况意识

问题

- COST 2 您如何管理使用情况？
- COST 3 如何监控使用情况和成本？
- COST 4 您如何停用资源？

COST 2 您如何管理使用情况？

制定各种策略和机制，确保花费适当的成本来达到目标。采用制约与平衡方法，您可以在不超支的情况下进行创新。

最佳实践

- COST02-BP01 根据组织的要求制定各种策略
- COST02-BP02 制定方向性目标和执行性目标
- COST02-BP03 实施账户结构
- COST02-BP04 实施组和角色
- COST02-BP05 实施成本控制
- COST02-BP06 跟踪项目生命周期

COST02-BP01 根据组织的要求制定各种策略

制定策略，规定您的组织应该如何管理资源。策略应该涵盖资源和工作负载的成本，包括在资源生命周期内创建、修改和停用。

未建立此最佳实践暴露的风险等级： 高

实施指导

了解组织的成本和驱动因素对于有效管理成本和使用量以及识别降低成本的机会至关重要。在组织中，通常会有多个团队运行多个工作负载。这些团队可能在不同的部门，每个部门都有其自己的收入来源。

将资源成本分摊到工作负载、各个组织或产品拥有者可以推动更高效的资源使用行为，减少浪费。准确的成本和使用量监控能够帮助您了解各部门和产品如何盈利，并让您能够针对组织内的资源分配做出更明智的决策。组织中各层级的人员都了解使用量是推动变化的关键，因为使用量变化会导致成本变化。考虑采用多元方法来了解您的使用量和支出情况。

执行治理的第一步是按照组织要求来针对云的使用制定策略。这些策略定义组织如何使用云以及如何管理资源。策略应涵盖与成本或使用量有关的资源和工作负载的所有方面，包括资源生命周期内的创建、修改和停用。

策略应该简单易懂，能够在整个组织中有效实施。从广泛的、高层级的策略开始，例如允许在哪个地理区域使用，或者一天中应该运行资源的时间。逐步为各组织部门和工作负载细化策略。常见策略包括可以使用哪些服务和功能（例如，测试或开发环境中性能较低的存储区），以及哪些类型的资源可供不同团队使用（例如，开发账户中最大的资源规模是中等）。

实施步骤

- 与团队成员会面： 要制定策略，请召集组织中的所有团队成员，详细说明他们的要求并相应地编制成档。采用迭代方法，首先大致进行，然后在每一步中不断细化到最小单元。团队成员包括与工作负载切身相关的人员（例如组织单位或应用程序负责人）以及支持小组（例如安全和财务团队）。
- 定义工作负载的位置： 定义工作负载的运行位置，包括国家/地区以及国家/地区中的区域。此信息用于映射到 AWS 区域和可用区。
- 定义和分组服务和资源： 定义工作负载所需的服务。对于每项服务，指定类型、大小和所需资源数量。按职能定义资源组，如应用程序服务器或数据库存储。资源可属于多个组。
- 按职能定义和分组用户： 定义与工作负载交互的用户，侧重于用户的工作范畴及其使用工作负载的方式，而不是侧重于他们的身份或其在组织中的职位。将类似用户或职能分组在一起。您可以使用 AWS 托管策略作为指南。
- 定义操作： 使用前面确定的位置、资源和用户，定义每项在其生命周期（开发、运行和停用）内实现工作负载成果所需的操作。根据每个位置的组（而不是组中的个别元素）确定操作。首先广泛读写，然后细化到每项服务的具体操作。
- 定义审核期： 工作负载和组织要求可能会随时间而变化。定义工作负载审核计划，以确保其与组织重点保持一致。
- 将策略编制成文档： 确保已定义的策略可按组织的要求访问。这些策略用于实施、维护和审计对环境的访问。

资源

相关文档：

- AWS 针对工作职能的托管策略
- AWS 多账户计费策略
- AWS 服务的操作、资源和条件键
- 云产品
- 使用 IAM 策略控制对 AWS 区域的访问
- 全球基础设施区域和可用区

COST02-BP02 制定方向性目标和执行性目标

制定工作负载的成本和使用量目标。方向性目标为组织在成本和使用情况方面指明了方向，执行性目标则为工作负载提供了可衡量的结果。

未建立这种最佳实践的情况下暴露的风险等级： 高

实施指导

为组织制定成本和使用量的方向性目标及执行性目标。方向性目标为组织提供有关预期结果的指引和方向。执行性目标则提供要实现的具体可衡量的结果。方向性目标的一个示例是：在略微（非线性）增加成本的情况下，显著提升平台使用量。执行性目标的一个示例是：在成本增长不到 5% 的情况下，将平台使用量提升 20%。另一个常见的方向性目标是每 6 个月提高一次工作负载的效率。与之相关的执行性目标是，工作负载的每项输出成本每 6 个月降低 5%。

云工作负载的一个常见方向性目标是提高工作负载效率，即随着时间的推移降低工作负载每项业务成果的成本。建议为所有工作负载实施此目标，并设定执行性目标，例如每 6 至 12 个月将效率提高 5%。通过在成本优化中增强能力以及发布新服务和服务功能，可以在云中实现这一目标。

实施步骤

- 定义预期使用量水平： 首先专注于使用量水平。与应用程序负责人、市场营销团队和更大的业务团队交流，了解工作负载的预期使用量水平。客户需求如何随时间而变化？是否会因季节性增长或营销活动而发生变化？
- 定义工作负载资源和成本： 定义使用量水平后，量化满足这些使用量水平所需的工作负载资源变化。您可能需要增加工作负载组件的资源大小或数量，增加数据传输，或者在特定级别将工作负载组件更改为不同的服务。详细说明每项要点的成本，以及当使用量发生变化时成本的变化。
- 定义业务目标： 从预期的使用量和成本变化中获取输出，将其与预期的技术变化或正在运行的任何计划相结合，制定工作负载目标。目标必须阐明使用量、成本和两者之间的关系。确认制定有组织的计划，例如培训和教育等能力培养项目，以防成本呈预期变化，而使用量无变化。

- 定义执行性目标： 对于定义的每个方向性目标，指定一个可衡量的执行性目标。如果方向性目标是提高工作负载的效率，则执行性目标将量化改进量，通常为每一美元支出的业务产出及获益时间。

资源

相关文档：

- AWS 针对工作职能的托管策略
- AWS 多账户计费策略
- 使用 IAM 策略控制对 AWS 区域 的访问

COST02-BP03 实施账户结构

实施与您的组织对应的账户结构。这有助于在整个组织内分摊和管理成本。

未建立这种最佳实践的情况下暴露的风险等级： 高

实施指导

AWS 拥有一个父级对多个子级的账户结构，通常称为管理账户（父级，之前称为付款人）账户-成员（子级，之前称为关联）账户。最佳实践是，无论组织规模或使用情况如何，始终至少有一个管理账户和一个成员账户。所有工作负载资源应仅驻留在成员账户内。

对于您应该拥有多少 AWS 账户这一问题，没有标准答案。评估当前和未来的运营和成本模型，以确保您的 AWS 账户结构反映了组织的目标。有些公司出于业务原因会创建多个 AWS 账户，例如：

- 需要在组织部门、成本中心或特定工作负载之间实施管理和/或财务和计费隔离。
- AWS Service Limits 设置为特定于特殊工作负载。
- 工作负载和资源之间必须进行隔离和分离。

在 AWS Organizations中，整合账单 会在一个或多个成员账户与管理账户之间创建结构。通过成员账户，您可以按团队隔离和区分成本和使用量。常见做法是每个组织部门（如财务、营销和销售）、每个环境生命周期（如开发、测试和生产）或每个工作负载（工作负载 a、b 和 c）具有单独的成员账户，然后使用整合账单将这些关联账户汇总在一起。

通过整合账单，您可以将多个成员 AWS 账户的付款整合至一个管理账户下，同时仍可查看每个关联账户的活动。由于成本和使用量在管理账户中汇总，因此，您可以最大限度地提高服务量折扣，并最大限度地利用承诺折扣（Savings Plans 和预留实例）来获得最高折扣。

AWS Control Tower 可以快速设置和配置多个 AWS 账户，确保治理符合您组织的要求。

实施步骤

- 定义分离要求： 分离要求涉及多项因素，包括安全性、可靠性和财务结构。按顺序阐明每项因素，并详细说明工作负载或工作负载环境是否应与其他工作负载分开。安全性可确保遵守访问和数据要求。可靠性可确保对限制进行管理，以便环境和工作负载不会影响其他项。财务结构可确保严格实施财务分离和问责制。常见分离示例有生产和测试工作负载在不同的账户中运行，或者使用单独的账户以便可以将发票和账单数据提供给第三方组织。
- 定义分组要求： 分组要求并不覆盖分离要求，而是用于协助管理。将无需分离的类似环境或工作负载分组在一起。例如，将一个或多个工作负载的多个测试或开发环境分组在一起。
- 定义账户结构： 使用这些分离和分组，为每个组指定一个账户，并确保持续满足分离要求。这些账户有成员账户或关联账户。通过将这些成员账户分组到一个管理账户或付款人账户下，可以合并使用量，从而可以跨所有账户享有更大的批量折扣，并为所有账户提供一个账单。可以分离账单数据，并为每个成员账户提供其账单数据的单独视图。如果成员账户不能让任何其他账户看到自己的使用情况或账单数据，或者，如果需要 AWS 提供单独的账单，请定义多个管理账户或付款人账户。在这种情况下，每个成员账户都有自己的管理账户或付款人账户。资源应始终放置在成员账户或关联账户中。管理账户或付款人账户应只用于管理。

资源

相关文档：

- AWS 针对工作职能的托管策略
- AWS 多账户计费策略
- 使用 IAM 策略控制对 AWS 区域的访问
- AWS Control Tower
- AWS Organizations
- 整合账单

相关示例：

- 拆分 CUR 和共享访问

COST02-BP04 实施组和角色

实施与策略一致的组和角色，控制每个组中谁可以创建、修改或停用实例和资源。例如，实施开发组、测试组和生产组。这适用于 AWS 服务和第三方解决方案。

未建立这种最佳实践的情况下暴露的风险等级： 低

实施指导

制定策略后，可以在组织内创建用户的逻辑组和角色。这样，您就可以分配权限并控制使用量。从高层级的人员分组开始，这通常与组织部门和岗位角色（例如，IT 部门的系统管理员或财务主管）相一致。这些组将执行相似任务并需要相似访问权限的人员集结在一起。角色定义组必须做什么。例如，IT 部门的系统管理员需要创建所有资源的权限，而分析团队成员仅需要创建分析资源。

实施步骤

- 实施组： 如有必要，请使用组织策略中定义的用户组实施相应的组。有关用户、组和身份验证的最佳实践，请参阅安全性支柱。
- 实施角色和策略： 使用组织策略中定义的操作，创建所需的角色和访问策略。有关角色和策略的最佳实践，请参阅安全性支柱。

资源

相关文档：

- AWS 针对工作职能的托管策略
- AWS 多账户计费策略
- 使用 IAM 策略控制对 AWS 区域 的访问
- Well-Architected 安全性支柱

相关示例：

- Well-Architected 实验室：基本身份和访问权限

COST02-BP05 实施成本控制

根据组织策略以及定义的组和角色来实施控制。这样可以确保成本只根据组织要求的规定产生，例如，使用 AWS Identity and Access Management（IAM）策略控制用户对区域或资源类型的访问。

未建立此最佳实践暴露的风险等级： 低

实施指导

实施成本控制的第一步通常是进行相关设置，以便在发生成本或使用量超出组织策略的事件时触发通知。这样，您就可以迅速采取行动，并验证是否需要采取纠正措施，而不会限制工作负载或新活动或对它们产生负面影响。了解工作负载和环境限制后，可以强制实施治理。在 AWS 中，通知是通过 AWS Budgets 执行的，因此您可以定义 AWS 成本、使用量和承诺折扣（Savings Plans 和预留实例）的月度预算。可以在总成本级别（如所有成本）创建预算，也可以在更细粒度的级别创建预算，其中只包含特定的维度，如关联的账户、服务、标记或可用区。

在第二步中，您可以通过 AWS Identity and Access Management （IAM）和 AWS Organizations 服务控制策略 (SCP)，在 AWS 中强制实施治理策略。借助 IAM，您可以安全地管理对 AWS 服务和资源的访问。您可以使用 IAM 控制谁能创建和管理 AWS 资源、可创建的资源类型以及可在何处创建。这样可以最大限度地减少创建不必要的资源。使用先前创建的角色和组，并分配 IAM 策略 以强制实施正确的使用量。SCP 用于集中管控组织中所有账户的最大可用权限，以确保您的账户始终在访问控制准则允许的范围内。SCP 仅在启用了所有功能的组织中可用，并且您可以将 SCP 配置为默认情况下拒绝或允许对成员账户执行操作。请参阅 《Well-Architected 安全性支柱》白皮书 以了解有关实施访问管理的更多详细信息。

还可以通过管理服务限额来实施治理。通过确保为服务配额设置最低开销并进行准确维护，您可以最大限度地减少组织要求以外的资源创建。要实现这一点，您必须了解要求的改变速度、了解正在进行的项目（资源的创建和停用），以及影响可以实施的限额更改速度的因素。服务限额 可用于在必要时增加配额。

实施步骤

- 实施支出通知： 使用定义的组织策略，制定 AWS 预算，当支出超出策略时发出通知。配置多个成本预算，每个账户一个，这样可通知您账户总支出。然后，在每个账户中，为该账户内的较小单元配置额外的成本预算。这些单元因您的账户结构而异。一些常见示例有 AWS 区域、工作负载（使用标签）或 AWS 服务。确保将电子邮件通讯组列表配置为通知的收件人，而不是个人的电子邮件账户。可以为超出金额的情况配置实际预算，或者使用预测预算通知预测使用量。
- 实施使用量控制： 使用定义的组织策略，实施 IAM 策略和角色，指定用户可执行的操作和无法执行的操作。一个 AWS 策略中可能包含多个组织策略。采用定义策略时所用的方式，首先大致进行，然后在每一步施加更细粒度的控制。服务限制也是一种有效的使用量控制措施。对所有账户实施正确的服务限制。

资源

相关文档：

- 针对工作职能的 AWS 托管策略
- AWS 多账户计费策略
- 使用 IAM 策略控制对 AWS 区域的访问

相关示例：

- Well-Architected 实验室：成本和使用情况治理
- Well-Architected 实验室：成本和使用情况治理

COST02-BP06 跟踪项目生命周期

跟踪、衡量并审计项目、团队和环境的生命周期，以避免使用不必要的资源并为此付费。

未建立此最佳实践暴露的风险等级： 低

实施指导

确保跟踪工作负载的整个生命周期。这样可以确保在不再需要工作负载或工作负载组件时，可以将其停用或对其进行修改。这在发布新服务或功能时尤其有用。现有的工作负载和组件看起来仍在使用中，但是应该停用以将客户重定向到新服务。注意工作负载的先前阶段 – 在工作负载进入生产之后，可以停用以前的环境或大幅降低其容量，直到再次需要它们为止。

AWS 提供了许多可用于实体生命周期跟踪的管理和治理服务。您可以使用 AWS Config 或 AWS Systems Manager 提供一份详尽的 AWS 资源和配置清单。建议集成现有项目或资产管理系统来跟踪组织内的活动项目和产品。将当前系统与 AWS 提供的丰富事件集和指标结合起来，您就可以构建大量生命周期事件的视图并主动管理资源，以减少不必要的成本。

有关 Web 应用程序后端方面的建议， 《Well-Architected 卓越运营支柱》白皮书 以了解有关实施实体生命周期跟踪的更多详细信息。

实施步骤

- 执行工作负载审核： 按照组织策略的规定，审计现有项目。在审计方面投入的工作量应与组织的大致风险、价值或成本成比例。主要审计领域包括组织面临的事件或中断风险，或对组织所做的贡献（以收入或品牌声誉进行衡量）、工作负载的成本（以资源的总成本和运营成本进行衡量）和工作负

载的使用量（以单位时间的组织产出量进行衡量）。如果这些领域在生命周期内发生变化，则需要对工作负载进行调整，例如全部停用或部分停用。

资源

相关文档：

- AWS Config
- AWS Systems Manager
- 针对工作职能的 AWS 托管策略
- AWS 多账户计费策略
- 使用 IAM 策略控制对 AWS 区域的访问

COST 3 如何监控使用情况和成本？

建立策略和程序以便监控并适当分配您的成本。这让您能够衡量和改进工作负载的成本效益。

最佳实践

- COST03-BP01 配置详细信息源
- COST03-BP02 确定成本归属类别
- COST03-BP03 建立组织指标
- COST03-BP04 配置账单和成本管理工具
- COST03-BP05 在成本和使用情况中添加组织信息
- COST03-BP06 根据工作负载指标分配成本

COST03-BP01 配置详细信息源

将 AWS 成本和使用情况报告以及 Cost Explorer 配置为以每小时为粒度，以便提供详细的成本和使用情况信息。配置工作负载，使交付的每个业务成果都有日志条目。

未建立这种最佳实践的情况下暴露的风险等级： 高

实施指导

在 AWS Cost Explorer 中启用每小时粒度，并创建 AWS 成本和使用情况报告（CUR）。这些数据源最确切地反映了整个组织中的成本和使用量。CUR 提供所有收费的 AWS 服务的每日或每小时使用粒度、费率、成本和使用属性。CUR 中的所有可能维度包括：标记、位置、资源属性和账户 ID。

使用以下自定义项配置 CUR：

- 包括资源 ID
- 自动刷新 CUR
- 每小时粒度
- 版本控制： 覆盖现有报告
- 数据集成： Amazon Athena（Parquet 格式和压缩）

使用 AWS Glue 准备分析数据、使用 Amazon Athena 执行数据分析、使用 SQL 查询数据。您也可以使用 Amazon QuickSight 构建复杂的自定义视图，并在整个组织内分发。

实施步骤

- 配置成本和使用情况报告： 使用账单控制台，至少配置一个成本和使用情况报告。配置以每小时为粒度的报告，以便包括所有标识符和资源 ID。还可以创建采用不同粒度的其他报告，以提供概括性摘要信息。
- 在 Cost Explorer 中配置每小时粒度： 使用账单控制台，启用每小时和资源级别数据。

> **Note**
>
> 启用此功能会产生相关成本。有关详细信息，请参阅定价。

- 配置应用程序日志记录： 确认应用程序记录所交付的每项业务成果，以便进行跟踪和衡量。确保该数据的粒度至少为每小时一次，以便与成本和使用情况数据匹配。有关日志记录和监控的更多详细信息，请参阅 《Well-Architected 卓越运营支柱》 。

资源

相关文档：

- AWS 账户设置
- AWS 成本和使用情况报告（CUR）
- AWS Glue
- Amazon QuickSight
- AWS 成本管理定价
- 标记 AWS 资源

- 使用 AWS Budgets 分析成本
- 使用 Cost Explorer 分析成本
- 管理 AWS 成本和使用情况报告
- Well-Architected 卓越运营支柱

相关示例：

- AWS 账户设置

COST03-BP02 确定成本归属类别

确定可以用于在组织内分摊成本的组织类别。

未建立此最佳实践暴露的风险等级： 高

实施指导

与您的财务团队和其他利益相关者合作，以了解必须在组织内部如何分摊成本的要求。必须将工作负载成本分摊至整个生命周期，包括开发、测试、生产和停用。了解组织如何对学习、员工培养和创意构思进行成本归类。这有助于将用于此目的的账户正确分配给培训和开发预算，而不是一般的 IT 成本预算。

实施步骤

- 定义组织类别： 与利益相关者召开会议，定义反映组织结构和要求的类别。这些将直接对应于现有财务类别的结构，例如业务单位、预算、成本中心或部门。了解云为您带来的业务成果（例如培训或教育），因为这些也是组织类别。可以将多个类别分配给一个资源，并且一个资源可以位于多个不同的类别中，因此可以根据需要定义任意多个类别。
- 定义功能类别： 与利益相关者召开会议，定义反映业务所含功能的类别。这可以是工作负载名称或应用程序名称以及环境类型（例如生产、测试或开发）。可以将多个类别分配给一个资源，并且一个资源可以位于多个不同的类别中，因此可以根据需要定义任意多个类别。

资源

相关文档：

- 标记 AWS 资源
- 使用 AWS Budgets 分析成本

- 使用 Cost Explorer 分析成本
- 管理 AWS 成本和使用情况报告

COST03-BP03 建立组织指标

建立此工作负载需要的组织指标。生成的客户报告或提供给客户的 Web 页面都属于工作负载指标。

未建立此最佳实践暴露的风险等级： 高

实施指导

了解如何根据业务成功来衡量工作负载的输出。每个工作负载通常有一组表示性能的主要输出。如果您的工作负载复杂且包含许多组件，则可以对列表进行优先级排序，或者为每个组件定义和跟踪指标。与团队合作，了解要使用哪些指标。此部分将用于了解工作负载的效率，或每项业务输出的成本。

实施步骤

- 定义工作负载结果： 与业务利益相关者召开会议，定义工作负载成果。这些主要用于衡量客户使用情况，因此必须是业务指标，而不是技术指标。每个工作负载应该有少量的概要指标（少于 5 个）。如果工作负载针对不同的使用案例产生多个结果，请将其分组为一个指标。
- 定义工作负载组件结果： 如果工作负载大而复杂，或者可以轻松地将工作负载分为输入和输出定义明确的多个组件（例如微服务），则可以选择为每个组件定义指标。这项工作应反映组件的价值和成本。按照从大到小的顺序，从最大的组件开始，逐步处理较小的组件。

资源

相关文档：

- 标记 AWS 资源
- 使用 AWS Budgets 分析成本
- 使用 Cost Explorer 分析成本
- 管理 AWS 成本和使用情况报告

COST03-BP04 配置账单和成本管理工具

配置符合组织策略的 AWS Cost Explorer 和 AWS Budgets。

未建立这种最佳实践的情况下暴露的风险等级： 高

实施指导

要修改使用量和调整成本，组织中的每个人都必须能够访问其成本和使用量信息。建议所有工作负载和团队在使用云时都配置以下工具：

- 报告： 汇总所有成本和使用情况信息
- 通知： 当成本或使用量超出定义的限值时触发通知。
- 当前状态： 配置显示当前成本和使用量水平的仪表板。仪表板应位于工作环境中的显眼位置（类似于操作仪表板）。
- 趋势分析： 能够以所需的粒度显示成本和使用量在指定时间段内的变化。
- 预测： 能够显示预计的未来成本。
- 跟踪： 对照配置的方向性目标或执行性目标显示当前的成本和使用量。
- 分析： 可让团队成员在所有可能的维度执行详尽至每小时粒度的自定义和深入分析。

您可以使用 AWS 原生工具（如 AWS Cost Explorer、 AWS Budgets和 Amazon Athena 以及 Amazon QuickSight ）来提供此功能。您还可以使用第三方工具，但是，必须确保为此工具花费的成本能够给组织带来价值。

实施步骤

- 创建成本优化组： 配置账户并创建可以访问所需成本和使用情况报告的组。该组必须包括负责或管理应用程序的所有团队的代表。这证明每个团队都可以访问他们的成本和使用情况信息。
- 配置 AWS Budgets： 在所有账户中为您的工作负载配置 AWS Budgets。通过使用标签设置账户总支出预算和工作负载预算。
- 配置 AWS Cost Explorer： 为工作负载和账户配置 AWS Cost Explorer。创建工作负载控制面板，以跟踪总支出和工作负载的关键使用指标。
- 配置高级工具： 可以选择为您的组织创建自定义工具，以便提供额外详细信息和粒度。可以使用 Amazon Athena实现高级分析功能，使用 Amazon QuickSight实现控制面板。

资源

相关文档：

- 标记 AWS 资源
- 使用 AWS Budgets 分析成本
- 使用 Cost Explorer 分析成本

- 管理 AWS 成本和使用情况报告

相关示例：

- Well-Architected 实验室：AWS 账户设置
- Well-Architected 实验室：账单可视化
- Well-Architected 实验室：成本和治理使用情况
- Well-Architected 实验室：成本和使用情况分析
- Well-Architected 实验室：成本和使用情况可视化

COST03-BP05 在成本和使用情况中添加组织信息

根据组织、工作负载属性和成本分摊类别来定义标记方案。在所有资源上应用标记。使用 Cost Categories，根据组织属性对成本和使用情况进行分组。

未建立此最佳实践暴露的风险等级： 低

实施指导

在 AWS 中实施标记， 以将组织信息添加到您的资源中，然后将其添加到成本和使用情况信息中。标签是键值对 — 键是定义的，必须在整个组织中唯一，值则对于一组资源唯一。键值对的一个示例是键为 Environment，值为 Production。生产环境中的所有资源都有这个键值对。借助标记，您可以使用有意义、相关的组织信息对成本进行分类和跟踪。您可以应用代表组织类别（例如成本中心、应用名称、项目或拥有者）的标签，标识工作负载和工作负载的特征（例如测试或生产），以在整个组织中分摊成本和使用量。

当您将标记应用于 AWS 资源（如 Amazon Elastic Compute Cloud 实例或 Amazon Simple Storage Service 存储桶）并激活标记后，AWS 会将此信息添加到成本和使用情况报告中。您可以在带标签和无标签的资源上运行报告并执行分析，以更好地遵守内部成本管理策略，并确保准确归属。

跨组织账户创建和实施 AWS 标记标准之后，您将能够一致且统一地管理和治理 AWS 环境。使用 标记策略（位于 AWS Organizations 中）定义有关如何在 AWS Organizations 账户的 AWS 资源上使用标签的规则。借助标记策略，您可以采用标准化方法轻松标记 AWS 资源

AWS 标签编辑器 可用于为多个资源添加、删除和管理标记。

AWS Cost Categories 可用于向成本分配组织含义，而无需在资源上添加标签。您可以将成本和使用量信息映射到唯一的内部组织结构。您可以定义类别规则，以使用账单维度（例如账户和标签）对成本进

行映射和分类。除了标记之外，这还提供了另外一个级别的管理能力。您还可以将特定账户和标记映射到多个项目。

实施步骤

- 定义标记方案： 召集整个业务的所有利益相关者来定义方案。这通常包括担任技术、财务和管理角色的人员。定义所有资源必须具有的标签列表，以及资源应该具有的标签列表。验证标签名称和值在整个组织中是否一致。
- 标记资源： 使用定义的成本归属类别，根据类别在工作负载中的所有资源上放置标签。使用 CLI、标签编辑器或 Systems Manager 等工具提高效率。
- 实施 Cost Categories： 您可以创建 Cost Categories，而无需采用标记方式。Cost Categories 使用现有的成本和使用量维度。根据方案创建类别规则，并在 Cost Categories 中加以实施。
- 自动标记： 要验证您是否在所有资源中保持高水平的标记，请自动标记，以便在创建资源时自动标记资源。使用 AWS CloudFormation 等服务中的功能确保在创建资源时进行标记。还可以创建自定义微服务，用于定期扫描工作负载并删除没有标记的任何资源，此方法非常适合测试和开发环境。
- 监控和报告标记： 要验证您是否在整个组织中保持高水平的标记，请报告和监控工作负载中的标记。可以使用 AWS Cost Explorer 查看标记资源和未标记资源的成本，也可以使用标签编辑器等服务。定期审核未标记资源的数量，并执行操作添加标记，直到达到所需的标记级别。

资源

相关文档：

- AWS CloudFormation 资源标签
- AWS Cost Categories
- 标记 AWS 资源
- Amazon EC2 和 Amazon EBS 增加对创建资源时加以标记的支持
- 使用 AWS Budgets 分析成本
- 使用 Cost Explorer 分析成本
- 管理 AWS 成本和使用情况报告

COST03-BP06 根据工作负载指标分配成本

根据指标或业务成果分配工作负载的成本，以便衡量工作负载的成本效益。实施一个流程，使用 Amazon Athena来分析 AWS 成本和使用情况报告，以便深入了解成本因素。

未建立这种最佳实践的情况下暴露的风险等级： 低

实施指导

成本优化旨在以最低的价格实现业务成果，这只能通过按工作负载指标分配工作负载成本（按工作负载效率衡量）来实现。通过日志文件或其他应用程序监控来监控定义的工作负载指标。将此数据与工作负载成本（可通过查看具有特定标签值或账户 ID 的成本获得）相结合。建议每小时进行一次分析。如果有一些静态成本要素（例如，全天候运行的后端数据库）且请求率不同（例如，使用量高峰在上午 9 点至下午 5 点，晚间的请求数量很少），则效率通常会变化。了解静态成本和可变成本之间的关系有助于您将精力集中在优化活动上。

实施步骤

- 将成本分配到工作负载指标： 使用定义的指标和配置的标记，创建结合工作负载输出和工作负载成本的指标。使用 Amazon Athena 和 Amazon QuickSight 等分析服务，为整个工作负载和任何组件创建效率控制面板。

资源

相关文档：

- 标记 AWS 资源
- 使用 AWS Budgets 分析成本
- 使用 Cost Explorer 分析成本
- 管理 AWS 成本和使用情况报告

COST 4 您如何停用资源？

在从项目开始到结束的过程中实施变更控制和资源管理。这可以确保您关闭或终止未使用的资源，以便减少浪费。

最佳实践

- COST04-BP01 在资源生命周期内跟踪资源
- COST04-BP02 实施停用流程
- COST04-BP03 停用资源
- COST04-BP04 自动停用资源

COST04-BP01 在资源生命周期内跟踪资源

制定和实施一种方法，在资源生命周期内跟踪资源及其与系统的关联。您可以使用标记来标识资源的工作负载或功能。

未建立此最佳实践暴露的风险等级： 高

实施指导

停用不再需要的工作负载资源。一个常见的示例是用于测试的资源，在测试完成后，可以将其删除。通过标签跟踪资源（并在这些标签上运行报告）可帮助您确定要停用的资产。使用标记是跟踪资源的一种有效方法，它通过标记资源的功能或资源的已知可停用日期来跟踪资源。然后，可以在这些标签上运行报告。功能标记的示例值是 `feature-X ###` 用于根据工作负载生命周期标识资源的用途。

实施步骤

- 实施标记方案： 实施标记方案，标识资源所属的工作负载，从而确认相应地标记工作负载中的所有资源。
- 实施工作负载吞吐量或输出监控： 实施工作负载吞吐量监控或警报，在输入请求或输出完成时触发。将其配置为在工作负载请求或输出下降到零时发出通知，指示不再使用工作负载资源。如果在正常情况下，工作负载周期性地下降到零，则加入时间因素。

资源

相关文档：

- AWS Auto Scaling
- AWS Trusted Advisor
- 标记 AWS 资源
- 发布自定义指标

COST04-BP02 实施停用流程

实施一个流程来确定和停用孤立的资源。

未建立这种最佳实践的情况下暴露的风险等级： 高

实施指导

在整个组织中实施标准化流程，以识别和删除未使用的资源。该流程应该定义执行搜索的频率以及删除资源的流程，以确保满足所有组织要求。

实施步骤

- 创建并实施停用流程： 与工作负载开发人员和负责人合作，为工作负载及其资源构建停用流程。该流程应涵盖一种方法，验证工作负载是否正在使用以及每个工作负载资源是否正在使用。该流程还应涵盖停用资源所需的步骤，将资源从服务中删除，同时确保符合任何法规要求。还应涵盖任何关联资源，例如许可证或附加存储。该流程应向工作负载负责人发送已执行停用流程的通知。

资源

相关文档：

- AWS Auto Scaling
- AWS Trusted Advisor

COST04-BP03 停用资源

停用由定期审计或使用情况发生变化等事件触发的资源。停用通常定期执行，可以手动停用，也可以自动停用。

未建立这种最佳实践的情况下暴露的风险等级： 中

实施指导

搜索未使用资源的频率和工作量应反映潜在的节省额，因此，与成本较高的账户相比，对成本较低的账户进行分析的频率应该更低。搜索和停用事件可由工作负载中的状态更改触发，比如产品生命周期结束或被更换。搜索和停用事件也可由外部事件触发，如市场条件发生变化或产品终止。

实施步骤

- 停用资源： 使用停用流程，停用确定为孤立的每个资源。

资源

相关文档：

- AWS Auto Scaling
- AWS Trusted Advisor

COST04-BP04 自动停用资源

设计您的工作负载，使其在您发现并停用非关键资源、不需要的资源或使用率低的资源时妥善处理资源的终止。

未建立这种最佳实践的情况下暴露的风险等级： 低

实施指导

使用自动化技术可以减少或消除停用流程中的相关成本。将工作负载设计为执行自动化停用将减少工作负载在其整个生命周期内的总成本。您可以使用 AWS Auto Scaling 执行停用流程。您还可以使用 API 或开发工具包 实施自定义代码以自动停用工作负载资源。

实施步骤

- 实施 AWS Auto Scaling： 对于受支持的资源，可使用 AWS Auto Scaling 配置它们。
- 配置 CloudWatch 以终止实例： 可以将实例配置为使用 CloudWatch 告警终止。使用停用流程的指标，实施包含 Amazon Elastic Compute Cloud（Amazon EC2）操作的告警。在推出之前，在非生产环境中验证操作。
- 在工作负载中实施代码： 您可以使用 AWS 开发工具包或 AWS CLI 停用工作负载资源。在与 AWS 集成的应用程序中实施代码，并终止或删除不再使用的资源。

资源

相关文档：

- AWS Auto Scaling
- AWS Trusted Advisor
- 创建停止、终止、重启或恢复实例的告警
- 开始使用 Amazon EC2 Auto Scaling

具有成本效益的资源

问题

- COST 5 您在选择服务时如何评估成本？
- COST 6 在选择资源类型、规模和数量时，如何实现成本目标？
- COST 7 您如何使用定价模式来降低成本？
- COST 8 您如何规划数据传输费用？

COST 5 您在选择服务时如何评估成本？

Amazon EC2、Amazon EBS 和 Amazon S3 属于构建块 AWS 服务。托管服务（如 Amazon RDS 和 Amazon DynamoDB）属于更高级别或应用程序级别的 AWS 服务。通过选择适当的基础服务和托管服务，您可以优化工作负载，从而降低成本。例如，使用托管服务，您可以节省或消除大部分管理和运营开销，从而使您有精力从事应用程序和业务相关活动。

最佳实践

- COST05-BP01 确定组织对成本的要求
- COST05-BP02 分析此工作负载的所有组件
- COST05-BP03 对每个组件进行彻底分析
- COST05-BP04 选择具有成本效益许可的软件
- COST05-BP05 选择此工作负载的组件，以便根据组织的优先事项优化成本
- COST05-BP06 对不同时间的不同使用情况执行成本分析

COST05-BP01 确定组织对成本的要求

与团队成员合作，为此工作负载确定成本优化与其他支柱（例如性能和可靠性）之间的平衡。

未建立这种最佳实践的情况下暴露的风险等级： 高

实施指导

在为工作负载选择服务时，了解组织的优先要务至关重要。确保在成本和其他 Well-Architected 支柱（例如性能和可靠性）之间取得平衡。完全成本优化的工作负载是最符合组织需求的解决方案，但不一定是成本最低的。与组织内的所有团队会面以收集信息，例如产品、业务、技术和财务。

实施步骤

- 确定组织对成本的要求： 与组织中的团队成员会面，这些成员包括产品管理、应用程序负责人、开发和运营团队、管理和财务角色。对此工作负载及其组件的 Well-Architected 支柱进行优先级排序，

输出是一个按顺序排列的支柱列表。您还可以为每个支柱添加一个权重，这可以指示一个支柱体现的额外关注程度，或者两个支柱之间的关注点的相似程度。

资源

相关文档：

- AWS 总拥有成本（TCO）计算器
- Amazon S3 存储类
- 云产品

COST05-BP02 分析此工作负载的所有组件

确认已分析工作负载的每个组件，无论当前大小或当前成本如何。审核工作应该体现出可能带来的好处，例如当前成本和预期成本。

在未建立这种最佳实践的情况下暴露的风险等级： 低

实施指导

对工作负载中的所有组件进行全面分析。确保在分析成本与工作负载在其生命周期内可能节省的成本之间取得平衡。必须确定组件的当前影响以及未来的潜在影响。例如，如果拟议资源的成本为每月 10 美元，在预测的负载下不会超过每月 15 美元，则花一天的时间将成本降低 50%（每月 5 美元）可能会超过系统使用寿命内的潜在收益。使用更快、更有效的基于数据的预估可为该组件带来最佳的总体结果。

工作负载可能会随时间变化，如果工作负载架构或使用量发生变化，原本合适的服务集可能不再是最优之选。为甄选服务进行分析时，必须考虑工作负载当前和未来的状态以及使用量水平。为将来的工作负载状态或使用量实施服务可以减少或消除未来进行更改所需的工作量，从而降低总体成本。

AWS Cost Explorer 和 AWS 成本和使用情况报告（CUR）可以分析概念验证（PoC, Proof of Concept）或运行环境的成本。您也可以使用 AWS Pricing Calculator 估算工作负载成本。

实施步骤

- 列出工作负载组件： 构建所有工作负载组件的列表，用于验证是否分析了每个组件。投入的工作量应体现出组织优先事项所规定的工作负载的关键性。如果有多个数据库，按功能（例如生产数据库存储）将资源分组可以提高效率。
- 对组件列表进行优先级排序： 获取组件列表，按工作顺序进行优先级排序。通常按照组件的成本从最昂贵到最便宜的顺序排列，或者按照组织优先事项规定的关键性排列。

- 执行分析： 对于列表中的每个组件，检查可用的选项和服务，然后选择最符合组织优先事项的选项。

资源

相关文档：

- AWS Pricing Calculator
- AWS Cost Explorer
- Amazon S3 存储类
- 云产品

COST05-BP03 对每个组件进行彻底分析

分析组织为每个组件付出的总体成本。通过考虑运营和管理成本（尤其是使用托管服务时）来分析总拥有成本。审核工作应该体现出可能带来的好处，例如用于分析的时间与组件成本成正比。

未建立这种最佳实践的情况下暴露的风险等级： 低

实施指导

利用节省下来的时间，您的团队将能够专注于解决技术债务、创新和增值功能。例如，您可能需要尽快将本地环境直接迁移到云端，然后再进行优化。值得探索的是，通过使用消除或减少许可证成本的托管服务，您可以节省多少成本。托管服务消除了维护服务的运营和管理负担，让您可以专注于创新。此外，由于托管服务在云级别运行，因此可以提供更低的单位事务或服务成本。

通常，可以设置托管服务的部分属性，以确保容量足够。您必须设置和监控这些属性，以便最大限度地减少多余容量，并最大限度地提高性能。您可以使用 AWS Management Console或 AWS API 和 SDK 修改 AWS Managed Services 的属性，以使资源需求匹配不断变化的要求。例如，您可以增加或减少 Amazon EMR 集群（或 Amazon Redshift 集群）上的节点数量，以扩展或缩减集群。

您还可以在 AWS 资源上打包多个实例，以实现更高密度的使用量。例如，您可以在单个 Amazon Relational Database Service（Amazon RDS）数据库实例上预置多个小型数据库。随着使用量的增长，您可以使用快照和还原过程将其中一个数据库迁移到专用 Amazon RDS 数据库实例。

在托管服务上预置工作负载时，您必须了解调整服务容量的要求。这些要求通常是时间、工作量和对正常工作负载运营的任何影响。预置的资源必须留出时间来进行任何更改，并预置必要的开销以允许这样做。通过使用与系统和监控工具（如 Amazon CloudWatch）集成的 API 和开发工具包，可以将修改服务所需的持续工作量减少至接近零。

Amazon RDS、Amazon Redshift和 Amazon ElastiCache 提供托管数据库服务。Amazon Athena、Amazon EMR和 Amazon OpenSearch Service 提供托管分析服务。

AMS 是代表企业客户和合作伙伴运营 AWS 基础设施的服务。它提供了一个安全且合规的环境，您可以将工作负载部署到其中。AMS 使用具有自动化功能的企业云运营模型，可以满足组织要求，更快地迁移到云中并降低持续的管理成本。

实施步骤

- 执行彻底分析： 使用组件列表，从最高优先级到最低优先级遍历每个组件。对于优先级较高且成本较高的组件，执行额外分析并评估所有可用选项及其长期影响。对于优先级较低的组件，评估使用情况的变化是否会更改组件的优先级，然后对适当的工作进行分析。

资源

相关文档：

- AWS 总拥有成本（TCO）计算器
- Amazon S3 存储类
- 云产品

COST05-BP04 选择具有成本效益许可的软件

开源软件无需软件许可成本，从而大大节省了工作负载的成本。如果需要许可软件，应避免使用绑定到任意属性（如 CPU）的许可证，而应使用绑定到输出或结果的许可证。这些许可证的成本与所提供的效益更为相当。

未建立这种最佳实践的情况下暴露的风险等级： 低

实施指导

使用开源软件可以消除软件许可成本。随着工作负载规模的扩展，这可能对工作负载成本产生重大影响。将许可软件能够带来的好处与总成本进行比较，确保拥有最优化的工作负载。对许可中的任何更改及其对工作负载成本的影响建模。如果供应商更改了数据库许可证的成本，请调查这会如何影响工作负载的整体效率。考虑供应商的历史定价公告，了解其产品中的许可更改趋势。许可成本也可以独立于吞吐量或使用量进行扩缩，例如按硬件扩缩的许可证（CPU 绑定许可证）。应避免使用这些许可证，因为成本会迅速增加，而且无法取得相应的结果。

实施步骤

- 分析许可证选项： 查看可用软件的许可条款。查看具有所需功能的开源版本，以及许可软件提供的效益是否大于成本。优惠条款可确保软件成本与所提供的效益相符。
- 分析软件提供商： 查看供应商的任何历史定价或许可变化。了解与成果不符的任何变化，例如在特定供应商硬件或平台上运行的惩罚性条款。此外，还要了解他们如何执行审计和处罚。

资源

相关文档：

- AWS 总拥有成本（TCO）计算器
- Amazon S3 存储类
- 云产品

COST05-BP05 选择此工作负载的组件，以便根据组织的优先事项优化成本

在选择所有组件时考虑成本因素。这包括使用 Amazon Relational Database Service（Amazon RDS）、 Amazon DynamoDB、Amazon Simple Notification Service（Amazon SNS）和 Amazon Simple Email Service（Amazon SES）等应用程序级别的托管服务降低组织的总体成本。使用无服务器服务和容器进行计算，例如 AWS Lambda、用于静态网站的 Amazon Simple Storage Service（Amazon S3）以及 Amazon Elastic Container Service（Amazon ECS）。使用开源软件或不收取许可证费用的软件，尽可能减少许可证成本：例如，对计算工作负载使用 Amazon Linux，或者将数据库迁移到 Amazon Aurora。

未建立这种最佳实践的情况下暴露的风险等级： 低

实施指导

您可以使用无服务器或应用程序级服务，如 AWS Lambda、Amazon Simple Queue Service (Amazon SQS)、 Amazon SNS和 Amazon SES。这些服务剔除了管理资源的需要，并提供代码执行、排队服务和消息传递功能。另一个好处是，它们可以根据使用量扩展性能和成本，从而实现有效的成本分配和归属。

有关无服务器的更多信息，请参阅《Well-Architected 无服务器应用程序剖析》白皮书。

实施步骤

- 选择每个服务以优化成本： 使用经过优先级排序的列表和分析，选择最符合组织优先事项的每个选项。

资源

相关文档：

- AWS 总拥有成本（TCO）计算器
- Amazon S3 存储类
- 云产品

COST05-BP06 对不同时间的不同使用情况执行成本分析

工作负载可能会随时间而变化。某些服务或功能在不同的使用水平下更具成本效益。通过随着时间的变化，根据每个组件的预期使用情况执行分析，工作负载可在其生命周期内保持成本效益。

未建立这种最佳实践的情况下暴露的风险等级： 低

实施指导

随着 AWS 发布新的服务和功能，适用于您的工作负载的最佳服务可能会发生变化。所需的工作量应反映出可能带来的好处。工作负载审核频率取决于您的组织要求。如果工作负载的成本很高，则尽早实施新服务可最大限度地节省成本，因此提高审核频率可能是有利的。审核的另一个触发因素是使用模式发生变化。使用量发生重大变化可能表明备用服务更加理想。例如，为获得更高的数据传输速率，直接连接服务可能比 VPN 便宜，并且会提供所需的连接。预测服务变更的潜在影响，以便您可以监控这些使用量水平触发器，并更快地实施最具成本效益的服务。

实施步骤

- 定义预计使用情况模式： 与组织中的相关人员（例如市场营销部门和产品负责人）合作，记录哪些预期和预计使用情况模式适用于工作负载。
- 根据预计使用情况进行成本分析： 使用定义的使用模式，在其中每个点执行分析。分析工作应该反映潜在的结果，例如，如果使用情况变化很大，应执行彻底分析，以验证任何成本和变化。

资源

相关文档：

- AWS 总拥有成本（TCO）计算器
- Amazon S3 存储类
- 云产品

COST 6 在选择资源类型、规模和数量时，如何实现成本目标？

确保选择适合当前任务的资源规模和资源数量。选择最经济实惠的资源类型、规模和数量可以尽可能减少浪费。

最佳实践

- COST06-BP01 执行成本建模
- COST06-BP02 根据数据选择资源类型、规模和数量
- COST06-BP03 根据指标自动选择资源类型、规模和数量

COST06-BP01 执行成本建模

确定组织要求，并对工作负载及其每个组件执行成本建模。对不同预计负载下的工作负载执行基准测试活动，并比较成本。建模工作应该反映出可能带来的好处，例如花费的时间与组件成本成正比。

未建立这种最佳实践的情况下暴露的风险等级： 高

实施指导

对工作负载及其每个组件执行成本建模，以了解资源之间的平衡，并在给定的具体性能水平下，确定工作负载中每个资源的正确规模。对不同预计负载下的工作负载执行基准测试活动，并比较成本。建模工作应该反映可能带来的好处，例如花费的时间与组件成本或预计可节省的成本成正比。有关最佳实践，请参阅 《性能效率支柱》白皮书 的 “审核”部分。

AWS Compute Optimizer 可协助对正在运行的工作负载进行成本建模。它根据历史使用量为计算资源提供合理调整大小的建议。这是计算资源的理想数据源，因为它是一项免费的服务，并且会利用机器学习根据风险等级提出多个建议。您还可以将 Amazon CloudWatch 和 Amazon CloudWatch Logs 与自定义日志一起用作数据源，用于其他服务和工作负载组件的合理调整大小操作。

以下是成本建模数据和指标的建议：

- 监控必须准确反映最终用户体验。为时间段选择正确的粒度，并仔细选择最大值或第 99 个百分位值而不是平均值。
- 为覆盖任何工作负载周期所需的分析时间段选择正确的粒度。例如，如果执行为期两周的分析，您可能会忽略高利用率的月度周期，这可能导致预置不足。

实施步骤

- 执行成本建模： 将工作负载或概念验证部署到具有特定资源类型和规模的单独账户，然后执行测试。使用测试数据运行工作负载，并记录输出结果以及运行测试时段的成本数据。然后，重新部署工作负载或更改资源类型和规模并重新运行测试。

资源

相关文档：

- AWS Auto Scaling
- Amazon CloudWatch 功能
- 成本优化：合理调整 Amazon EC2 的大小
- AWS Compute Optimizer

COST06-BP02 根据数据选择资源类型、规模和数量

根据工作负载和资源特征的相关数据选择资源规模或类型，例如计算、内存、吞吐量或写入密集型资源。通常使用工作负载的上一个版本（本地版本）、文档或关于工作负载的其他信息源进行选择。

未建立此最佳实践暴露的风险等级： 中

实施指导

根据工作负载和资源特征选择资源规模或类型，例如计算、内存、吞吐量或写入密集型资源。通常使用成本建模、工作负载的上一个版本（例如本地版本）、文档或关于工作负载的其他信息源（白皮书、发布的解决方案）进行选择。

实施步骤

- 根据数据选择资源： 使用成本建模数据，选择预期的工作负载使用情况水平，然后选择指定的资源类型和规模。

资源

相关文档：

- AWS Auto Scaling
- Amazon CloudWatch 功能
- 成本优化：合理调整 EC2 的大小

COST06-BP03 根据指标自动选择资源类型、规模和数量

使用当前运行的工作负载的指标选择正确的规模和类型，从而优化成本。针对 Amazon Elastic Compute Cloud（Amazon EC2）、Amazon DynamoDB、Amazon Elastic Block Store（Amazon EBS）（PIOPS）、Amazon Relational Database Service（Amazon RDS）、Amazon EMR 和联网等服务适当预置吞吐量、规模和存储。这可以通过自动扩展等反馈环路进行，也可以在工作负载中使用自定义代码来实现。

未建立这种最佳实践的情况下暴露的风险等级： 低

实施指导

在工作负载中创建一个反馈循环，此循环使用正在运行的工作负载中的活动指标来对该工作负载进行更改。您可以使用托管服务（如 AWS Auto Scaling），将其配置为代您执行合理调整大小的操作。AWS 还提供 API、开发工具包和功能，让您可以轻松修改资源。您可以对工作负载进行编程以停止和启动 Amazon Elastic Compute Cloud（Amazon EC2）实例，从而允许更改实例大小或实例类型。这带来双重好处：既合理调整了大小，又几乎消除了进行更改所需的所有运营成本。

某些 AWS 服务内置了自动类型或大小选项，如 Amazon Simple Storage Service（Amazon S3）Intelligent-Tiering。Amazon S3 Intelligent-Tiering 会根据您的使用模式，自动在两个访问层之间移动数据：频繁访问和非频繁访问。

实施步骤

- 配置工作负载指标： 确保捕获工作负载的关键指标。这些指标指明了客户体验（例如工作负载输出），并适应资源类型和规模之间的差异（例如 CPU 和内存使用情况）。
- 查看合理调整规模建议： 在 AWS Compute Optimizer 中使用合理调整规模建议来调整工作负载。
- 根据指标自动选择资源类型和规模： 使用工作负载指标，手动或自动选择工作负载资源。配置 AWS Auto Scaling 或在应用程序中实施代码可以减少频繁更改所需的工作量，而且实现更改的速度可能比手动操作更快。

资源

相关文档：

- AWS Auto Scaling
- AWS Compute Optimizer
- Amazon CloudWatch 功能
- CloudWatch 开始设置

- CloudWatch 发布自定义指标
- 成本优化：合理调整 Amazon EC2 的大小
- 开始使用 Amazon EC2 Auto Scaling
- Amazon S3 Intelligent-Tiering
- 使用 SDK 启动 EC2 实例

COST 7 您如何使用定价模式来降低成本？

使用最适合的资源定价模式可以尽可能减少支出。

最佳实践

- COST07-BP01 执行定价模式分析
- COST07-BP02 根据成本实施区域
- COST07-BP03 选择具有经济实惠的条款的第三方协议
- COST07-BP04 针对此工作负载的所有组件实施定价模式
- COST07-BP05 在主账户级别执行定价模式分析

COST07-BP01 执行定价模式分析

分析工作负载的每个组件。确定组件和资源是长时间运行（享受承诺折扣），还是短时间动态运行（采用竞价型实例或按需型实例）。利用 AWS Cost Explorer 中的建议功能对工作负载执行分析。

未建立这种最佳实践的情况下暴露的风险等级： 高

实施指导

AWS 有多种 定价模型 ，您可以符合组织需求、最具成本效益的方式支付资源费用。

实施步骤

- 执行承诺折扣分析： 在账户中使用 Cost Explorer 查看 Savings Plans 和预留实例建议。要验证您是否实施了具有所需折扣和风险的正确建议，请按照 Well-Architected 实验室操作。
- 分析工作负载弹性： 在 Cost Explorer 中使用每小时粒度或者使用自定义控制面板。分析工作负载弹性。确定正在运行的实例数量的规律性变化。短期实例是竞价型实例或竞价型实例集的候选实例。
  - Well-Architected 实验室：Cost Explorer
  - Well-Architected 实验室：成本可视化

资源

相关文档：

- 获取预留实例建议
- 实例购买选项

相关视频：

- 最高可节省 90% 并在竞价型实例上运行生产工作负载

相关示例：

- Well-Architected 实验室：Cost Explorer
- Well-Architected 实验室：成本可视化
- Well-Architected 实验室：定价模式

COST07-BP02 根据成本实施区域

资源定价在每个区域中可能各不相同。考虑区域成本有助于确保您为此工作负载支付最低的总体费用。

未建立此最佳实践暴露的风险等级： 中

实施指导

在架构解决方案时，最佳实践是设法将计算资源放在更接近用户的位置，以提供更低的延迟和强大的数据主权。对于全球用户，您应该使用多个位置来满足这些需求。您应该选择尽可能降低成本的地理位置。

AWS Cloud 基础设施围绕 区域和可用区构建。区域是指全球范围内的某个物理位置，每个区域由多个可用区组成。可用区由一个或多个分散的数据中心组成，每个都拥有独立的配套设施，其中包括冗余电源、联网和连接。

每个 AWS 区域都在当地市场条件下运营，每个区域的资源定价可能不同。选择特定区域来运行解决方案组件或整个解决方案，以便您可以在全球范围内以尽可能低的价格运行。您可以使用 AWS Pricing Calculator 估算各区域中工作负载的成本。

实施步骤

- 审核区域定价： 分析当前区域的工作负载成本。首先使用按服务和使用类型划分的最高成本，计算其他可用区域的成本。如果预测的节省超过移动组件或工作负载的成本，则迁移到新区域。

资源

相关文档：

- 获取预留实例建议
- Amazon EC2 定价
- 实例购买选项
- 区域表

相关视频：

- 节省高达 90% 并在竞价型实例上运行生产工作负载

COST07-BP03 选择具有经济实惠的条款的第三方协议

经济实惠的协议和条款可确保这些服务的成本与所提供的效益相称。选择与可为组织带来额外效益相称的协议和定价。

未建立此最佳实践暴露的风险等级： 中

实施指导

当您在云中使用第三方解决方案或服务时，确保定价结构与成本优化结果保持一致非常重要。定价应与其带来的结果和价值成比例。这方面的一个例子是可带来一定百分比节省额的软件，节省额（结果）越高，其价格也就越高。除非您能提供特定账单每一部分的结果，否则与账单成比例的协议通常不会与成本优化保持一致。例如，如果您使用的其他服务没有带来任何好处，提供 Amazon Elastic Compute Cloud（Amazon EC2）相关建议并收取整个账单一定比例费用的解决方案将会增加。另一个示例是根据所托管资源的成本按一定百分比收费的托管服务。实例越大并不一定意味着需要更多的管理工作，但会收取更多费用。确保这些服务定价安排包括成本优化计划或服务中的功能，以提高效率。

实施步骤

- 分析第三方协议和条款： 审核第三方协议中的定价。基于不同的使用情况水平执行建模，并考虑新成本，例如使用新服务，或当前服务由于工作负载增长而增加使用量。确定额外成本能否为业务提供所需效益。

资源

相关文档：

- 获取预留实例建议
- 实例购买选项

相关视频：

- 节省高达 90% 并在竞价型实例上运行生产工作负载

COST07-BP04 针对此工作负载的所有组件实施定价模式

永久运行的资源应利用预留容量，如 Savings Plans 或预留实例。短期容量配置为使用竞价型实例或竞价型实例集。按需型实例仅用于无法中断并且运行时间没有长到可以使用预留容量的短期工作负载，时间为使用时期的 25% 到 75%，具体取决于资源类型。

未建立这种最佳实践的情况下暴露的风险等级： 低

实施指导

考虑工作负载组件的要求并了解潜在的定价模型。定义组件的可用性要求。确定工作负载中是否存在执行功能的多个独立资源，以及工作负载随着时间推移的需求情况。使用默认的按需定价模型和其他适用模型比较资源成本。考虑资源或工作负载组件的任何潜在更改。

实施步骤

- 实施定价模式： 根据分析结果，购买 Savings Plans（SP）、预留实例或者实施竞价型实例。如果是首次购买 RI，请选择列表中的前 5 个或 10 个建议，然后在接下来的一两个月内监控并分析结果。购买少量的承诺折扣定期周期，例如每两周或每月。对可能会中断或者无状态的工作负载实施竞价型实例。
- 工作负载审核周期： 实施工作负载审核周期，用于专门分析定价模式覆盖范围。工作负载达到所需覆盖范围后，每两到四周再次购买承诺折扣，或者随着组织的使用情况变化进行购买。

资源

相关文档：

- 获取预留实例建议

- EC2 队列
- 如何购买预留实例
- 实例购买选项
- 竞价型实例

相关视频：

- 节省高达 90% 并在竞价型实例上运行生产工作负载

COST07-BP05 在主账户级别执行定价模式分析

使用 Cost Explorer Savings Plans 和预留实例建议，在管理账户级别执行承诺折扣定期分析。

未建立这种最佳实践的情况下暴露的风险等级： 低

实施指导

定期执行成本建模可确保能够跨多个工作负载进行优化。例如，如果总体上多个工作负载使用按需型实例，则变更的风险较低，并且实施基于承诺的折扣可降低总体成本。建议每两周到一个月定期执行一次分析。这样您可以进行少量调整性采购，因此定价模型的覆盖范围会随着工作负载及其组件的变化而不断变化。

使用 AWS Cost Explorer 建议工具寻找享受承诺折扣的机会。

要为竞价型工作负载寻找机会，请查看总体使用量的小时视图，并确定使用量或弹性的定期变化周期。

实施步骤

- 执行承诺折扣分析： 在账户中使用 Cost Explorer 查看 Savings Plans 和预留实例建议。要验证您是否实施了具有所需折扣和风险的正确建议，请按照 Well-Architected 实验室操作。

资源

相关文档：

- 获取预留实例建议
- 实例购买选项

相关视频：

- 最高可节省 90% 并在竞价型实例上运行生产工作负载

相关示例：

- Well-Architected 实验室：定价模式

COST 8 您如何规划数据传输费用？

务必要监控和规划您的数据传输费用，以便制定架构决策，尽可能降低成本。持续以小步迭代的方式进行架构优化可以实现运营成本的大幅降低。

最佳实践

- COST08-BP01 执行数据传输建模
- COST08-BP02 选择组件以便优化数据传输成本
- COST08-BP03 实施服务以便降低数据传输成本

COST08-BP01 执行数据传输建模

收集组织要求，并对工作负载及其每个组件执行数据传输建模。这样可以确定满足当前数据传输要求的最低成本点。

未建立这种最佳实践的情况下暴露的风险等级： 高

实施指导

了解数据传输在您的工作负载中发生的位置、传输的成本以及相关的收益。因此，您可以做出明智的决定来修改或接受架构决策。例如，您可能有一个多可用区配置，可以在可用区之间复制数据。您可以对结构成本建模，并确定这是可接受的成本（类似于在两个可用区中支付计算和存储费用），以实现所需的可靠性和弹性。

对不同使用级别的成本进行建模。工作负载的使用量可能随时间而变化，不同的服务可能在不同的级别上更具有成本效益。

使用 AWS Cost Explorer 或 AWS 成本和使用情况报告（CUR）来了解数据传输成本并对其建模。配置概念证明（PoC）或测试您的工作负载，并在实际的模拟负载下运行测试。您可以根据不同的工作负载需求对成本进行建模。

实施步骤

- 计算数据传输成本： 使用 AWS 定价页面 ，计算工作负载的数据传输成本。计算不同使用情况水平的数据传输成本，包括工作负载使用量的增加和减少这两种情况。如果工作负载架构有多个选项，则计算每个选项的成本以便进行比较。
- 将成本与成果相关联： 对于产生的每项数据传输成本，指定其实现的工作负载成果。如果在组件之间传输，则可能是为了实现解耦，如果在可用区之间传输，则可能是为了实现冗余。

资源

相关文档：

- AWS 缓存解决方案
- AWS 定价
- Amazon EC2 定价
- Amazon VPC 定价
- 使用 Amazon CloudFront 更快地交付内容

COST08-BP02 选择组件以便优化数据传输成本

选择所有组件然后设计架构，以便降低数据传输成本。其中包括使用广域网（WAN）优化和多可用区（AZ）配置等组件

未建立这种最佳实践的情况下暴露的风险等级： 低

实施指导

针对数据传输进行架构，可确保您最大限度地降低数据传输成本。这可能涉及使用内容分发网络来定位更靠近用户的数据，或者使用从您的本地设施到 AWS 的专用网络链接。您还可以使用 WAN 优化和应用程序优化来减少组件之间传输的数据量。

实施步骤

- 选择用于数据传输的组件： 使用数据传输建模，关注产生最多数据传输成本之处，或者工作负载使用情况发生变化时产生最多数据传输成本之处。查找替代架构或其他组件，以消除或减少数据传输的需要，或降低其成本。

资源

相关文档：

- AWS 缓存解决方案
- 使用 Amazon CloudFront 更快地交付内容

COST08-BP03 实施服务以便降低数据传输成本

实施服务以减少数据传输。例如，使用 Amazon CloudFront 等内容分发网络（CDN）向最终用户传输内容、使用 Amazon ElastiCache 建立缓存层，或者使用 AWS Direct Connect 而不是 VPN 来连接 AWS。

未建立这种最佳实践的情况下暴露的风险等级： 低

实施指导

Amazon CloudFront 是一个全球内容分发网络，可提供低延迟、高传输速度的数据。它在世界各地的边缘站点缓存数据，从而减少资源负担。通过使用 CloudFront，您可以减少向全球大量用户分发内容的管理工作，同时将延迟降到最低。

AWS Direct Connect 允许您建立到 AWS 的专用网络连接。与基于互联网的连接相比，这可以降低网络成本、增加带宽并提供更一致的网络体验。

AWS VPN 可让您在专用网络和 AWS 全局网络之间建立安全的专用连接。它是小型办公室或业务合作伙伴的理想之选，因其提供快速简便的连接，并且是完全托管的弹性服务。

VPC 终端节点 允许通过专用网络在 AWS 服务之间建立连接，可用于减少公共数据传输并且 NAT 网关 成本。网关 VPC 终端节点 不按小时收费，支持 Amazon Simple Storage Service（Amazon S3）和 Amazon DynamoDB。接口 VPC 终端节点 由 AWS PrivateLink 提供，有小时费和每 GB 使用成本。

实施步骤

- 实施服务： 使用数据传输建模，了解产生最大成本和最多数据流的地方。查看 AWS 服务并评估是否存在一种服务，可以减少或消除传输，特别是联网和内容交付。在需要重复访问数据或存在大量数据时查找缓存服务。

资源

相关文档：

- AWS Direct Connect
- AWS 探索我们的产品

- AWS 缓存解决方案
- Amazon CloudFront
- 使用 Amazon CloudFront 更快地交付内容

管理需求和供应资源

问题

- COST 9 如何管理需求和供应资源？

COST 9 如何管理需求和供应资源？

为了工作负载的性能与支出实现平衡，请确保您支付过费用的所有资源都得到利用，并避免出现资源利用率过低的情况。无论是从运维成本（由于过度使用导致性能下降）还是从浪费 AWS 支出（由于超额配置）的角度衡量，利用率指标过高或过低都会对您的组织产生负面影响。

最佳实践

- COST09-BP01 对工作负载需求执行分析
- COST09-BP02 实施缓冲区或节流来管理需求
- COST09-BP03 动态供应资源

COST09-BP01 对工作负载需求执行分析

分析工作负载需求随时间的变化。确认分析涵盖季节性趋势，并准确反映整个工作负载生命周期内的运行条件。分析工作应该体现出可能带来的好处，例如花费的时间与工作负载成本成正比。

未建立这种最佳实践的情况下暴露的风险等级： 高

实施指导

了解工作负载的要求。组织需求应指出工作负载对于请求的响应时间。响应时间可用于确定是否管理了需求，或者资源的供应是否会改变以满足需求。

分析应包括需求的可预测性和可重复性、需求的变化速率以及需求的变化量。确保在足够长的时间内执行分析，以纳入任何季节性变化，例如月末处理或假期高峰。

确保分析工作反映实施扩展的潜在好处。查看组件的预期总成本，以及在工作负载生命周期内增加和减少的使用量和成本。

您可以将 AWS Cost Explorer 或者 Amazon QuickSight 与 AWS 成本和使用情况报告（CUR）或应用程序日志一起使用，以便对工作负载需求进行可视化分析。

实施步骤

- 分析现有工作负载数据： 分析现有工作负载中的数据、以前工作负载版本中的数据或预测使用模式中的数据。使用日志文件和监控数据，了解客户如何使用工作负载。典型的指标有实际需求（以每秒请求数为单位）、需求率变化的时间或处于不同级别时的时间，以及需求变化速率。务必分析整个工作负载周期，从而确保收集任何季节性变化数据，如月末或年末活动。分析中反映的工作应该体现出工作负载特征。应将最多的精力放在需求变化最大的高价值工作负载上。应将最少的精力放在需求变化最小的低价值工作负载上。衡量价值的常用指标有风险、品牌知名度、收入或工作负载成本。
- 预测外部影响： 与组织中会影响或更改工作负载需求需求的团队成员会面。通常涉及的团队包括销售、营销或业务拓展团队。与他们合作，了解其运作周期，以及是否有改变工作负载需求的任何活动。使用这些数据预测工作负载需求。

资源

相关文档：

- AWS Auto Scaling
- AWS Instance Scheduler
- 开始使用 Amazon SQS
- AWS Cost Explorer
- Amazon QuickSight

COST09-BP02 实施缓冲区或节流来管理需求

缓冲和限流可修改工作负载需求，从而避免出现任何峰值情形。在客户端执行重试时实施限流。实施缓冲以存储请求并将处理任务往后推迟一段时间。确认设计节流和缓冲区时客户端能够在所需的时间内收到响应。

未建立这种最佳实践的情况下暴露的风险等级： 低

实施指导

节流： 如果需求源具有重试功能，可以实施限流。限流会告诉需求源，如果当前无法处理请求，则应稍后再试。需求源将等待一段时间，然后重新尝试请求。实施限流的优势是可限制最大资源量和工作负

载成本。在 AWS 中，您可以使用 Amazon API Gateway 实施限流。请参阅 《Well-Architected 可靠性支柱》白皮书 以了解有关实施限流的更多详细信息。

基于缓冲区： 与限流类似，缓冲区会延迟请求处理，从而允许以不同速率运行的应用程序有效通信。基于缓冲区的方法使用队列来接受来自产生方的消息（工作单元）。然后消息将由使用方读取并处理，这样消息就能够以满足使用方业务需求的速率运行。无需担心产生方必须处理数据持久性和反向压力等限流问题（因为使用方运行缓慢，导致产生方运行缓慢）。

在 AWS 中，您可以从多个服务中进行选择，以便实施缓冲方法。Amazon Simple Queue Service（Amazon SQS）是一项托管服务，提供允许单个使用方读取单个消息的队列。Amazon Kinesis 提供允许众多使用方读取相同消息的流。

使用基于缓冲区的方法进行架构时，请确保架构工作负载以在所需时间内处理请求，并且您能够处理重复的工作请求。

实施步骤

- 分析客户端需求： 分析客户端请求，确定它们是否能够执行重试。对于无法执行重试的客户端，需要实施缓冲区。分析总体需求、变化率和所需的响应时间，以确定所需的限流或缓冲区大小。
- 实施缓冲区或节流： 在工作负载中实施缓冲区或限流。Amazon Simple Queue Service（Amazon SQS）之类的队列可以为工作负载组件提供缓冲区。Amazon API Gateway 可以为工作负载组件提供节流。

资源

相关文档：

- AWS Auto Scaling
- AWS Instance Scheduler
- Amazon API Gateway
- Amazon Simple Queue Service
- 开始使用 Amazon SQS
- Amazon Kinesis

COST09-BP03 动态供应资源

资源按计划预置。这种预置可以基于需求（例如通过自动扩展来实现），也可以基于时间（需求可以预测，基于时间提供资源）。这些方法可以尽可能减少超额预置或预置不足的情况。

未建立这种最佳实践的情况下暴露的风险等级： 低

实施指导

您可以使用 AWS Auto Scaling，或者通过 AWS API 或 SDK 在代码中加入扩展。这样省去了手动更改环境的操作成本，因而工作负载的总体成本得以降低，而且执行速度变得更快。这将确保工作负载资源在任何时候都最符合需求。

基于需求的供应： 利用云的弹性来供应资源以满足不断变化的需求。利用 API 或服务功能，以编程方式动态改变架构中云资源的数量。这使您能够在架构中扩展组件，并在需求高峰期间自动增加资源数量以保持性能，也可以在需求量降低时减少容量以降低成本。

AWS Auto Scaling 可帮助您调整容量以维持稳定、可预测的性能，并确保成本最低。它是一项完全托管的免费服务，与 Amazon Elastic Compute Cloud（Amazon EC2）实例和竞价型实例集、Amazon Elastic Container Service（Amazon ECS）、Amazon DynamoDB 和 Amazon Aurora 集成。

Auto Scaling 提供自动资源发现功能，以帮助您在工作负载中找到可以配置的资源，它具有内置的扩展策略来优化性能、成本或者在两者之间取得平衡，并提供预测性扩展来协助应对定期出现的峰值。

Auto Scaling 可以实施手动、计划或基于需求的扩展。您还可以使用来自 Amazon CloudWatch 的指标和警报触发工作负载的扩展事件。典型的指标可以是标准 Amazon EC2 指标，例如 CPU 利用率、网络吞吐量和 Elastic Load Balancing（ELB）观察到的请求或响应延迟。如果可能，应该使用指示客户体验的指标，通常是可能来自工作负载中的应用程序代码的自定义指标。

当构建基于需求的方法时，请注意两个重要事项。首先，了解您必须以多快的速度预置新资源。其次，了解供应和需求之间的差额将发生变化。您必须准备好应对需求变化的速度，并准备好应对资源故障。

ELB 通过在多种资源之间分配需求来帮助您扩展规模。随着实施的资源越来越多，您可以将它们添加到负载均衡器中以满足需求。Elastic Load Balancing 支持 Amazon EC2 实例、容器、IP 地址和 AWS Lambda 函数。

基于时间的供应： 基于时间的方法可以协调资源容量以满足可预测或时间明确定义的需求。此方法通常不依赖资源的利用水平。基于时间的方法可以确保资源在需要的特定时间可用，并且提供时不会因启动流程和系统或一致性检查而发生延迟。使用基于时间的方法，您可以在繁忙时段提供额外的资源或增加容量。

您可以使用计划的 Auto Scaling 来实施基于时间的方法。工作负载可以在定义的时间按计划扩展或缩减（例如办公时间开始时），从而确保用户就位或需求出现时资源可用。

您还可以利用 AWS API 和 SDK 以及 AWS CloudFormation 在需要时自动预置和停用整个环境。此方法非常适合仅在定义的办公时间或时间段运行的开发或测试环境。

您可以使用 API 来扩展环境中的资源大小（纵向扩展）。例如，可以通过更改实例大小或分类纵向扩展生产工作负载。这可以通过停止和启动实例，以及选择不同的实例大小或分类来实现。这种技巧也可以应用于其他资源，如 Amazon Elastic Block Store（Amazon EBS）弹性卷，您可以在使用时对其进行修改以增加大小、调整性能（IOPS）或更改卷类型。

当构建基于时间的方法时，请注意两个重要事项。首先，使用模式的一致性如何？ 其次，如果模式发生更改会产生什么影响？ 您可以通过两种方式提高预测的准确性：监控工作负载和使用商业智能。如果您发现使用模式发生重大更改，可以调整时间，以确保提供覆盖范围。

实施步骤

- 配置基于时间的调度： 对于可预测的需求变化，基于时间的扩展可以及时提供正确的资源量。如果资源创建和配置的速度不够快，无法响应需求变化，也可使用这种方法。根据工作负载分析，使用 AWS Auto Scaling 配置计划扩缩。
- 配置 Auto Scaling： 要根据活动工作负载指标配置扩展，请使用 Amazon Auto Scaling。使用分析并配置 Auto Scaling 以在正确的资源级别上触发，并确保工作负载在所需的时间内扩展。

资源

相关文档：

- AWS Auto Scaling
- AWS Instance Scheduler
- 开始使用 Amazon EC2 Auto Scaling
- 开始使用 Amazon SQS
- Amazon EC2 Auto Scaling 的计划扩缩

随着时间的推移不断优化

问题

- COST 10 如何评估新服务?

COST 10 如何评估新服务?

AWS 不断发布新服务和功能，因此您最好不断审视现有架构决策，以便确保其始终最具成本效益。

最佳实践

- COST10-BP01 制定工作负载审核流程
- COST10-BP02 定期审核和分析此工作负载

COST10-BP01 制定工作负载审核流程

制定一个流程，定义工作负载的审核标准和流程。审核工作应该体现出潜在优势。例如，核心工作负载或费用占比超过 10% 的工作负载每季度审核一次，而费用占比低于 10% 的工作负载每年审核一次。

未建立此最佳实践暴露的风险等级： 高

实施指导

为确保工作负载始终最具成本效益，您必须定期对其进行审核，以了解是否有机会实施新的服务、功能和组件。为确保整体成本尽可能低，此过程必须与潜在的节省额成比例。例如，与占总支出 5% 的工作负载相比，应更经常、更彻底地审核占总支出 50% 的工作负载。考虑任何外部因素或波动。如果工作负载服务于特定的地理位置或市场领域，并且您已预测出该领域会出现的变化，则提高审核频率可能会节省成本。审核时要考虑的另一个因素是实施更改的工作量。如果测试和验证变更的成本很高，则审核的频率应该降低。

考虑维护过时和旧式组件及资源的长期成本，以及无法在其中实施新功能的事实。当前的测试和验证成本可能会超过预计的收益。但是，随着时间的推移，工作负载和当前技术之间的差距会增大，进行更改的成本可能会大幅增加，导致成本升高。例如，迁移到新的编程语言的成本当前可能不具成本效益。然而，五年之后，熟练使用该语言的人员的成本可能会增加，并且由于工作负载的扩展，迁移到新语言的系统规模更大，这其中涉及的工作量甚至高于以前。

将工作负载分解成多个组件，分配组件的成本（估算即可），然后在每个组件旁边列出因素（例如工作量和外部市场）。使用这些指示信息来确定每个工作负载的审核频率。例如，您可能觉得 Web 服务器的成本高、变更的工作量小、外部因素多，因而审核频率很高。而中央数据库的成本可能中等、变更的工作量很大、外部因素较少，因而审核频率也为中等。

实施步骤

- 定义审核频率： 定义工作负载及其组件的审核频率。应考虑多种因素，且这些因素可能会因组织中的工作负载而异，也可能因工作负载中的组件而异。常见的因素包括：从收入或品牌角度来讲对组织的重要性、运行工作负载的总成本（包括运营和资源成本）、工作负载的复杂性、实施变革的难易程度、任何软件许可协议以及变革是否会因惩罚性的许可而显著增加许可成本。可以在功能上或技术上定义组件，例如 Web 服务器和数据库，或者计算和存储资源。相应地权衡这些因素，并为工作负载及其组件制定一个周期。您可能决定每 18 个月审核一次完整的工作负载，每 6 个月审核一次 Web 服务器，每 12 个月审核一次数据库，每 6 个月审核一次计算资源和短期存储，每 12 个月审核一次长期存储。

- 定义审核的彻底性： 定义在工作负载或工作负载组件的审核上投入的工作量。与审核频率类似，这也需要权衡多种因素。您可能决定投入一周的时间对数据库组件进行分析，投入四小时的时间进行存储审核。

资源

相关文档：

- AWS 新闻博客
- 云计算类型
- AWS 的新增功能

COST10-BP02 定期审核和分析此工作负载

根据每个定义的流程定期审核现有工作负载。

未建立这种最佳实践的情况下暴露的风险等级： 低

实施指导

为实现新 AWS 服务和功能的优势，必须对工作负载执行审核流程，并根据需要实施新服务和功能。例如，您可以审核工作负载，并使用 Amazon Simple Email Service（Amazon SES）替换消息收发组件。这省去了运行和维护实例队列的成本，同时以更低的成本提供所有功能。

实施步骤

- 定期审核工作负载： 使用您定义的流程，按照指定的频率执行审核。确认在每个组件上投入正确的工作量。此流程类似于您选择服务进行成本优化的初始设计流程。分析服务及其带来的优势，这一次需考虑实施更改所产生的成本，而不仅仅是长期优势。
- 实施新服务： 如果分析结果表明可以实施更改，请先执行工作负载基线，以了解每项产出的当前成本。实施更改，然后执行分析以确认每项产出的新成本。

资源

相关文档：

- AWS 新闻博客
- 云计算类型
- AWS 最新内容

可持续性

主题

- 区域选择
- 用户行为模式
- 软件和架构模式
- 数据模式
- 硬件模式
- 开发和部署流程

区域选择

问题

- SUS 1 如何选择区域来支持您的可持续发展目标？

SUS 1 如何选择区域来支持您的可持续发展目标？

根据您的业务需求和可持续发展目标，选择您将在其中实施工作负载的区域。

最佳实践：

SUS01-BP01 选择 Amazon 可再生能源项目附近的区域和其电网公布的碳强度低于其他位置（或区域）的区域

选择亚马逊可再生能源项目附近的区域和其电网公布的碳强度低于其他位置（或区域）的区域。

未建立这种最佳实践的情况下暴露的风险等级： 中

实施指导

选择亚马逊可再生能源项目附近的区域和其电网公布的碳强度低于其他位置（或区域）的区域。

资源

相关文档：

- Amazon 遍布全球

- 可再生能源方法
- 为工作负载选择区域时应考虑的事项

用户行为模式

问题

- SUS 2 您如何利用用户行为模式来支持您的可持续发展目标？

SUS 2 您如何利用用户行为模式来支持您的可持续发展目标？

用户使用您的工作负载和其他资源的方式可以帮助您确定改进措施，以实现可持续性目标。扩展基础设施以持续匹配用户负载，并确保仅部署支持用户所需的最少资源。使服务水平与客户需求保持一致。定位资源以限制用户使用它们所需的网络。移除现有的未使用资产。识别已创建但未使用的资产并停止生成它们。为您的团队成员提供满足其需求的设备，同时最大限度地减少对可持续性的影响。

最佳实践：

SUS02-BP01 扩缩基础设施以匹配用户负载

确定利用率低或无利用率的时段，缩减资源以消除过剩容量并提高效率。

常见反模式：

- 您没有扩缩基础设施以匹配用户负载。
- 您一直在手动扩缩基础设施。
- 在扩展事件之后，您将保留增加的容量，而不是缩减容量。

建立此最佳实践的好处：配置和测试工作负载弹性将有助于减小工作负载环境影响，节省资金，并维护性能基准。您可以利用云中的弹性，在用户负载峰值期间和之后自动扩缩容量，以确保只使用满足客户需求所需的确切数量的资源。

未建立这种最佳实践的情况下暴露的风险等级：中

实施指导

- 弹性可根据对您拥有的资源的需求来提供这些资源。实例、容器和函数都能够与自动扩展功能相结合或作为此服务的一项功能来提供可实现弹性的机制。在架构中使用弹性，以确保在用户负载较低的时期，可以快速轻松地缩减工作负载：

    - 使用 Amazon EC2 Auto Scaling 验证您拥有适量的 Amazon EC2 实例，可处理您应用程序的用户负载。
    - 使用 Application Auto Scaling 自动扩缩 Amazon EC2 以外的各项 AWS 服务的资源，比如 Lambda 函数或 Amazon Elastic Container Service (Amazon ECS) 服务。
    - 使用 Kubernetes Cluster Autoscaler 自动扩缩 AWS 上的 Kubernetes 集群。
- 验证衡量扩展或缩减的指标已根据所部署的工作负载类型进行了验证。如果您正在部署一个视频转码应用程序，CPU 利用率预计为 100%，并且不应将此作为您的主要指标。如果需要，您可以为您的扩缩策略使用一个 自定义指标 （如内存利用率）。要选择正确的指标，请考虑以下关于 Amazon EC2 的指导：
    - 该指标应该是有效的利用率指标，并描述实例的繁忙程度。
    - 该指标值必须随 Auto Scaling 组中的实例数量成比例地增加或减少。
- 使用 动态扩展 而不是 手动扩展 （对于 Auto Scaling 组）。我们还建议您在动态扩展中使用 目标跟踪扩缩策略 。
- 验证工作负载部署可以处理扩展事件和缩减事件。创建缩减事件的测试方案，以确保工作负载按预期方式运行。您可以使用 活动历史记录 来测试和验证 Auto Scaling 组的扩缩活动。
- 评估您的工作负载以获得可预测的模式，并在您预期需求会发生预测和计划的变化时主动扩缩。使用 Amazon EC2 Auto Scaling 预测式扩缩 来消除过度增加容量的需求。

资源

相关文档：

- 开始使用 Amazon EC2 Auto Scaling
- 由机器学习提供支持的 EC2 预测式扩缩
- 使用 Amazon OpenSearch Service、Amazon Data Firehose 和 Kibana 分析用户行为
- 什么是 Amazon CloudWatch？
- 什么是 AWS X-Ray？
- VPC 流日志
- 在 Amazon RDS 上使用 Performance Insights 监控数据库负载
- 介绍对 Amazon EC2 Auto Scaling 预测式扩缩的原生支持
- 如何基于内存利用率指标创建 Amazon EC2 Auto Scaling 策略（Linux）
- 介绍 Karpenter - 高性能开源 Kubernetes Cluster Autoscaler

相关视频：

- 更好、更快、更便宜的计算：Amazon EC2 成本优化（CMP202-R1）

相关示例：

- 实验室：Amazon EC2 Auto Scaling 组示例
- 实验室：使用 Karpenter 实施自动扩展

SUS02-BP02 使 SLA 与可持续性目标保持一致

定义和更新服务等级协议（SLA，Service Level Agreements），例如可用性或数据留存期，以最大限度地减少支持工作负载所需的资源数量，同时继续满足业务需求。

未建立此最佳实践暴露的风险等级： 低

实施指导

- 定义 SLA，在支持可持续性目标的同时满足您的业务需求。
- 重新定义 SLA 以满足业务需求，而不是超越它们。
- 做出权衡，显著降低可持续性影响，以换取可接受的服务等级降低幅度。
- 使用优先考虑业务关键功能的设计模式，并允许非关键功能具有较低的服务等级（例如响应时间或恢复时间目标）。

资源

相关文档：

- AWS 服务等级协议（SLA）
- Importance of Service Level Agreement for SaaS Providers

相关视频：

- AWS 上的可持续构建

SUS02-BP03 停止创建和维护未使用的资产

分析应用程序资产（例如预编制的报告、数据集和静态图像）和资产访问模式，以识别冗余、利用率低下的情况和潜在的淘汰目标。整合具有冗余内容的生成资产（例如，具有重叠或共用数据集和输出的月度报告），以消除重复输出时消耗的资源。淘汰未使用的资产（例如，已停售产品的图片）以释放消耗的资源，并减少用于支持工作负载的资源数量。

未建立这种最佳实践的情况下暴露的风险等级： 低

实施指导

- 管理静态资产并移除不再需要的资产。
- 管理生成的资产并停止生成和删除不再需要的资产。
- 整合生成的重叠资产以消除冗余处理。
- 指示第三方停止生成和存储代您管理但不再需要的资产。
- 指示第三方整合代表您生成的多余资产。

资源

相关文档：

- 优化您的 AWS 基础设施以实现可持续性，第 II 部分：存储

相关视频：

- AWS 上的可持续构建

SUS02-BP04 针对用户位置优化工作负载的地理位置

分析网络访问模式以识别您的客户建立连接的地理位置。选择可减少网络流量必须传输的距离的区域和服务，以减少支持您的工作负载所需的总网络资源。

常见反模式：

- 您根据自己所在的位置选择工作负载的区域。

建立此最佳实践的好处： 将工作负载放在接近客户的地方可以提供极低的延迟，同时减少网络中的数据移动并减小对环境的影响。

未建立这种最佳实践的情况下暴露的风险等级： 中

实施指导

- 请根据以下关键元素，为您的工作负载部署选择区域：
  - 您的可持续发展目标： 如 区域选择中所述。
  - 数据所在位置： 对于数据密集型应用程序（如大数据和机器学习），应用程序代码的执行应尽量接近数据。
  - 用户所在位置： 对于面向用户的应用程序，选择一个接近您工作负载的客户群的区域。
  - 其他制约： 考虑安全性和合规性等制约，如 为工作负载选择区域时应考虑的事项中所述。
- 使用 AWS Local Zones 运行视频渲染和图形密集型虚拟桌面应用程序等工作负载。Local Zones 使计算和存储资源更接近终端用户，从而使您受益。
- 对常用资源使用本地缓存或 AWS 缓存解决方案 ，以提高性能，减少数据移动并减小对环境的影响。
  - 使用 Amazon CloudFront 缓存静态内容（如图像、脚本和视频）以及动态内容（如 API 响应或 Web 应用程序）。
  - 使用 Amazon ElastiCache 缓存 Web 应用程序的内容。
  - 使用 DynamoDB Accelerator 将内存中加速添加到您的 DynamoDB 表。
- 使用可帮助您在更接近工作负载用户的位置运行代码的服务：
  - 使用 Lambda@Edge 执行计算密集型操作，当对象不在缓存中时执行这些操作。
  - 使用 Amazon CloudFront Functions 处理简单使用场景，如 HTTP(s) 请求或响应操作，这些操作可由短期运行的函数执行。
  - 使用 AWS IoT Greengrass 为互联设备运行本地计算、消息收发和数据缓存。
- 使用连接池来实现连接重用并减少所需资源。
- 使用不依赖于持久连接和同步更新的分布式数据存储来保持一致性，从而为区域人口提供服务。
- 用共享的动态容量代替预先配置的静态网络容量，并与其他用户共享网络容量的可持续性影响。

资源

相关文档：

- 优化您的 AWS 基础设施以实现可持续性，第 III 部分：联网
- Amazon ElastiCache 文档
- 什么是 Amazon CloudFront？

- Amazon CloudFront 主要功能
- Lambda@Edge
- CloudFront Functions
- AWS IoT Greengrass

相关视频：

- AWS 上的可持续构建

相关示例：

- AWS 联网研讨会

SUS02-BP05 针对执行的活动优化团队成员资源

优化提供给团队成员的资源，在支持其需求的同时最大程度地降低对可持续性的影响。例如，在利用率高的共享云桌面上，而不是在利用率不高的强力单用户系统上，执行渲染和编译等复杂的操作。

未建立这种最佳实践的情况下暴露的风险等级： 低

实施指导

- 按照工作站和其他设备的使用方式对它们进行预置。
- 使用虚拟桌面和应用程序串流来限制升级和设备要求。
- 将处理器或内存密集型任务移至云端。
- 评估流程和系统对您的设备生命周期的影响，并选择在满足业务需求的同时最大限度减少设备更换需求的解决方案。
- 对设备实施远程管理以减少所需的商务旅行。

资源

相关文档：

- 什么是 Amazon WorkSpaces？
- Amazon AppStream 2.0 文档
- NICE DCV

- AWS Systems Manager Fleet Manager

相关视频：

- AWS 上的可持续构建

软件和架构模式

问题

- SUS 3 您如何利用软件和架构模式来支持您的可持续发展目标？

SUS 3 您如何利用软件和架构模式来支持您的可持续发展目标？

实施用于执行负载平滑和保持已部署资源始终如一的高利用率的模式，以最大限度地减少资源消耗。由于用户行为会随着时间的推移而发生变化，因此组件可能会因缺乏使用而变得空闲。修改模式和架构以整合未充分利用的组件，从而提高整体利用率。停用不再需要的组件。了解工作负载组件的性能，并优化消耗资源最多的组件。注意客户用来访问您服务的设备，并实施相应的模式以最大限度地减少设备升级需要。

最佳实践：

SUS03-BP01 针对异步和计划作业优化软件和架构

使用高效的软件设计和架构来尽可能减少每个工作单元所需的平均资源。实施可促成均匀的组件利用率的机制，以减少任务之间的空闲资源并最大限度地减少负载峰值的影响。

未建立这种最佳实践的情况下暴露的风险等级： 低

实施指导

- 不需要立即处理的队列请求。
- 提高序列化程度以扁平化整个管道的利用率。
- 修改单个组件的容量，防止空闲资源等待输入。
- 创建缓冲区并建立速率限制，以使外部服务的使用更顺畅。
- 使用最高效的可用硬件进行软件优化。
- 使用队列驱动的架构、管道管理和按需型实例工件，最大限度地提高批处理的利用率。
- 安排任务以避免同时执行导致的负载峰值和资源争用。

- 将作业安排在一天中碳强度最低的时段中处理。

资源

相关文档：

- 什么是 Amazon Simple Queue Service？
- 什么是 Amazon MQ？
- 基于 Amazon SQS 进行扩展
- 什么是 AWS Step Functions？
- 什么是 AWS Lambda？
- 将 AWS Lambda 与 Amazon SQS 配合使用
- 什么是 Amazon EventBridge？

相关视频：

- AWS 上的可持续构建
- 迁移到事件驱动型架构

SUS03-BP02 删除或重构很少或没有使用的工作负载组件

监控工作负载活动以识别各个组件的利用率随时间的变化。移除未使用且不再需要的组件，并重构利用率低的组件，以限制资源浪费。

未建立这种最佳实践的情况下暴露的风险等级： 低

实施指导

- 分析功能正常的组件上的负载（使用事务流和 API 调用等指标），以识别未使用和未充分利用的组件。
- 停用不再需要的组件。
- 重构未充分利用的组件。
- 将未充分利用的组件与其他资源整合以提高利用效率。

资源

相关文档：

- 什么是 AWS X-Ray？
- 什么是 Amazon CloudWatch？
- 使用 ServiceLens 监控应用程序的运行状况
- 自动清理 Amazon ECR 中未使用的镜像

相关视频：

- AWS 上的可持续构建

SUS03-BP03 优化消耗最多时间或资源的代码区域

监控工作负载活动以识别消耗最多资源的应用程序组件。优化在这些组件中运行的代码，以最大限度地减少资源使用和提高性能。

未建立这种最佳实践的情况下暴露的风险等级： 低

实施指导

- 根据资源使用情况监控性能，以将单个工作单元的资源需求高的组件确定为优化目标。
- 使用代码分析器确定使用时间最长或使用资源最多的代码区域作为优化目标。
- 将算法替换为产生相同结果的更高效版本。
- 使用硬件加速来提高执行时间长的代码块的效率。
- 对工作负载使用最高效的操作系统和编程语言。
- 删除不必要的排序和格式。
- 使用数据传输模式，根据数据更改的频率和使用方式，最大限度地减少使用的资源。例如，将状态更改信息推送到客户端，而不是让它消耗资源来轮询和接收无价值的“无更改”消息。

资源

相关文档：

- 什么是 Amazon CloudWatch？

- 什么是 Amazon CodeGuru Profiler？
- FPGA 实例
- 在 AWS 上进行构建所需工具的 AWS 开发工具包

相关视频：

- AWS 上的可持续构建

SUS03-BP04 优化对客户设备的影响

了解客户用来使用您服务的设备、它们的预期生命周期，以及更换这些组件对财务和可持续性的影响。实施软件模式和架构，以最大限度地减少客户更换和升级设备的需求。例如，使用与旧硬件和操作系统版本向后兼容的代码实现新功能，或管理有效负载的大小，使其不超过目标设备的存储容量。

未建立这种最佳实践的情况下暴露的风险等级： 低

实施指导

- 盘点客户使用的设备。
- 使用具有代表性硬件集的托管式设备场进行测试，以了解更改的影响，并迭代开发以最大限度增加支持的设备数。
- 在构建有效负载时考虑网络带宽和延迟，并实施有助于您的应用程序在低带宽、高延迟链路上良好运行的功能。
- 预处理数据有效负载，以减少本地处理要求并限制数据传输要求。
- 在服务器端执行计算密集型活动（例如图像渲染），或使用应用程序串流来改善旧设备上的用户体验。
- 对输出进行分段和分页，尤其是对于交互式会话，以管理有效负载并限制本地存储要求。

资源

相关文档：

- 什么是 AWS Device Farm？
- Amazon AppStream 2.0 文档
- NICE DCV

- Amazon Elastic Transcoder 文档

相关视频：

- AWS 上的可持续构建

SUS03-BP05 使用最能支持数据访问和存储模式的软件模式和架构

了解数据在工作负载中的使用方式、用户使用数据的方式，以及数据的传输和存储方式。选择相应的技术以最大限度地减少数据处理和存储要求。

未建立这种最佳实践的情况下暴露的风险等级： 低

实施指导

- 分析您的数据访问和存储模式。
- 以高效的文件格式（如 Parquet）存储数据文件，防止进行不必要的处理（例如在运行分析时）并减少预置的总存储。
- 使用可以原生处理压缩数据的技术。
- 使用最能支持您的主导查询模式的数据库引擎。
- 管理您的数据库索引以确保索引设计支持高效的查询执行。
- 选择可减少消耗的网络容量的网络协议。

资源

相关文档：

- Athena 压缩支持文件格式
- 使用 Amazon Redshift 从列数据格式复制
- 在 Firehose 中转换您的输入记录格式
- AWS Glue 中 ETL 输入和输出的格式选项
- 通过转换为列格式提高 Amazon Athena 上的查询性能
- 使用 Amazon Redshift 从 Amazon S3 加载压缩数据文件
- 在 Amazon Aurora 上使用 Performance Insights 监控数据库负载
- 在 Amazon RDS 上使用 Performance Insights 监控数据库负载

- AWS IoT FleetWise

相关视频：

- AWS 上的可持续构建

数据模式

问题

- SUS 4 您如何利用数据访问模式和使用模式来支持您的可持续发展目标？

SUS 4 您如何利用数据访问模式和使用模式来支持您的可持续发展目标？

实施数据管理实践以减少支持工作负载所需的预置存储，以及使用存储所需的资源。了解您的数据，并使用最能支持数据的商业价值及其使用方式的存储技术和配置。当需求减少时，将数据移到更高效、性能更低的存储中，并删除不再需要的数据。

最佳实践：

SUS04-BP01 实施数据分类策略

对数据进行分类以了解其对业务成果的重要性。使用此信息来确定何时可以将数据移动到更节能的存储，或者何时可以安全删除数据。

未建立此最佳实践暴露的风险等级： 低

实施指导

- 确定数据的分发、保留和删除要求。
- 对卷和对象使用标记来记录用于确定其管理方式的元数据，包括数据分类。
- 针对未标记和未分类的数据定期审核您的环境，并对数据进行适当的分类和标记。

资源

相关文档：

- 数据分类过程
- 利用 AWS Cloud 支持数据分类

- AWS Organizations 中的标记策略

SUS04-BP02 使用支持数据访问和存储模式的技术

使用最能支持您的数据访问和存储方式的存储，以在支持您的工作负载的同时最大限度地减少预置资源。例如，固态硬盘（SSD，Solid State Device）比磁性驱动器更耗能，应该仅用于活跃的数据使用场景。对不常访问的数据使用节能的存档级存储。

未建立此最佳实践暴露的风险等级： 中

实施指导

- 监控您的数据访问模式。
- 根据访问模式将数据迁移到适当的技术。
- 将存档数据迁移到为此目的设计的存储中。

资源

相关文档：

- Amazon EBS 卷类型
- Amazon EC2 实例存储
- Amazon S3 Intelligent-Tiering
- 使用 Amazon S3 存储类
- 什么是 Amazon CloudWatch？
- 什么是 Amazon S3 Glacier？

相关视频：

- AWS 上数据湖的架构模式

SUS04-BP03 使用生命周期策略删除不必要的数据

管理所有数据的生命周期并自动执行删除时间表，以最大限度地减少工作负载的总存储需求。

未建立这种最佳实践的情况下暴露的风险等级： 低

实施指导

- 为所有数据分类类型定义生命周期策略。
- 设置自动化生命周期策略以强制实施生命周期规则。
- 删除未使用的卷和快照。
- 在适当情况下根据生命周期规则汇总数据。

资源

相关文档：

- Amazon ECR 生命周期策略
- Amazon EFS 生命周期管理
- Amazon S3 Intelligent-Tiering
- 使用 AWS Config 规则 评估资源
- 在 Amazon S3 上管理存储生命周期
- AWS Elemental MediaStore 中的对象生命周期策略

相关视频：

- Amazon S3 生命周期

SUS04-BP04 最大限度地减少数据块存储中的过度预置

要尽可能减少总预置存储，请创建大小分配适合工作负载的数据块存储。随着数据的增长，使用弹性卷扩展存储，而无需调整附加到计算资源的存储大小。定期检查弹性卷并缩小过度配置的卷，以适应当前数据大小。

未建立这种最佳实践的情况下暴露的风险等级： 低

实施指导

- 监控数据卷的利用率。
- 使用弹性卷和托管式数据块数据服务，随着持久性数据的增长自动分配额外的存储。
- 为您的数据卷设置目标利用率水平，并调整超出预期范围的卷大小。

- 调整只读卷的大小以适应数据。
- 将数据迁移到对象存储，以避免使用数据块存储上的固定卷大小预配多余容量。

资源

相关文档：

- Amazon EBS 弹性卷
- Amazon FSx 文档
- 什么是 Amazon CloudWatch？
- 什么是 Amazon Elastic File System？

SUS04-BP05 删除不需要或多余的数据

仅在必要时复制数据，以最大程度地减少消耗的总存储空间。使用备份技术在文件和数据块级别进行重复数据删除。限制使用独立驱动器冗余阵列（RAID，Redundant Array of Independent Drives）配置，除非需要满足服务等级协议（SLA，Service Level Agreements）。

未建立这种最佳实践的情况下暴露的风险等级： 低

实施指导

- 使用可以在数据块和对象级别删除重复数据的机制。
- 使用可以进行增量备份的备份技术，并在数据块、文件和对象级别删除重复数据。
- 仅在需要满足您的 SLA 时才使用 RAID。
- 集中日志和跟踪数据，对相同的日志条目进行重复数据删除，并在需要时建立调整详细程度的机制。
- 仅在合理的情况下预填充缓存。
- 建立缓存监控和自动化以相应地调整缓存大小。
- 推送新版本的工作负载时，从对象存储和边缘缓存中删除过时的部署和资产。

资源

相关文档：

- Amazon EBS 快照

- 更改 CloudWatch Logs 中的日志数据留存
- Amazon FSx for Windows File Server 上的重复数据删除
- Amazon FSx for ONTAP 的功能，包括重复数据删除
- 使 Amazon CloudFront 上的文件失效
- 使用 AWS Backup 备份和还原 Amazon EFS 文件系统
- 什么是 Amazon CloudWatch Logs？
- 在 Amazon RDS 上使用备份

相关示例：

- 实验：使用 Amazon Redshift 数据共享优化数据模式

SUS04-BP06 使用共享文件系统或对象存储来访问通用数据

采用共享存储和单一事实来源，以避免重复数据删除并降低工作负载的总存储需求。仅在需要时从共享存储中获取数据。分离未使用的卷以使更多资源可用。

未建立此最佳实践暴露的风险等级： 低

实施指导

- 当数据具有多个使用者时，将数据迁移到共享存储。
- 仅在需要时从共享存储中获取数据。
- 根据您的使用模式删除数据，并实施生存时间（TTL，time-to-live）功能来管理缓存的数据。
- 将卷与未积极使用它们的客户端分离。

资源

相关文档：

- Amazon FSx
- 缓存策略
- 什么是 Amazon Elastic File System？
- 什么是 Amazon S3？

SUS04-BP07 最大限度地减少跨网络的数据移动

使用共享存储和访问区域数据存储中的数据，以最大限度地减少支持工作负载数据移动所需的总网络资源。

未建立这种最佳实践的情况下暴露的风险等级： 低

实施指导

- 将数据存储在尽可能靠近使用者的位置。
- 按区域对使用的服务进行分区，以便将其特定于区域的数据存储在使用它的区域内。
- 跨网络复制更改时，使用数据块级重复数据删除，而不是文件或对象级重复数据删除。
- 在通过网络移动数据之前，先对其进行压缩。

资源

相关文档：

- 优化您的 AWS 基础设施以实现可持续性，第 III 部分：联网
- AWS 全球基础设施
- Amazon CloudFront 主要功能，包括 CloudFront 全球边缘网络
- 在 Amazon OpenSearch Service 中压缩 HTTP 请求
- 使用 Amazon EMR 进行中间数据压缩
- 将压缩数据文件从 Amazon S3 加载到 Amazon Redshift
- 通过 Amazon CloudFront 提供压缩文件

SUS04-BP08 仅在难以重新创建时备份数据

为了最大限度地减少存储消耗，仅备份具有商业价值或满足合规性要求所必需的数据。检查备份策略并在恢复方案中排除没有价值的临时存储。

未建立这种最佳实践的情况下暴露的风险等级： 低

实施指导

- 使用数据分类来确定需要备份的数据。
- 排除您可以轻松重新创建的数据。
- 从备份中排除临时数据。

- 排除数据的本地副本，除非从公共位置恢复该数据所需的时间会超过您的服务等级协议（SLA，service level agreements）。

资源

相关文档：

- 使用 AWS Backup 备份和还原 Amazon EFS 文件系统
- Amazon EBS 快照
- 在 Amazon Relational Database Service 上使用备份

硬件模式

问题

- SUS 5 您的硬件管理和使用实践如何支持您的可持续发展目标？

SUS 5 您的硬件管理和使用实践如何支持您的可持续发展目标？

寻找机会，通过更改硬件管理实践来降低工作负载可持续性影响。最大限度地减少预置和部署所需的硬件数量，并为您的各项工作负载选择最高效的硬件。

最佳实践：

SUS05-BP01 使用最少的硬件来满足您的需求

通过使用云的功能，您可以对工作负载实施进行频繁更改。在需求变化时更新已部署的组件。

未建立这种最佳实践的情况下暴露的风险等级： 中

实施指导

- 启用横向扩缩，并使用自动化在负载增加时扩展，在负载减少时缩减。
- 通过小增量扩缩来适应可变的工作负载。
- 在负载随时间（天、周、月或年）而变化时，根据周期性利用模式（例如，具有两周一次的密集处理活动的工资系统）进行扩缩。
- 协商服务等级协议（SLA，service level agreements），允许暂时减少容量，同时利用自动化功能部署替换资源。

资源

相关文档：

- AWS Compute Optimizer 文档
- 运行 Lambda：性能优化
- 弹性伸缩文档

SUS05-BP02 使用影响最小的实例类型

持续监控新实例类型的发布并利用能效改进，包括那些旨在支持特定工作负载（例如机器学习训练、推理以及视频转码）的实例类型。

常见反模式：

- 您只使用一个系列的实例。
- 您只使用 x86 实例。
- 您在 Amazon EC2 Auto Scaling 配置中指定一种实例类型。
- 您使用 AWS 实例的方式与其预期用途不匹配（例如，您将计算优化的实例用于内存密集型工作负载）。
- 您没有定期评估新的实例类型。
- 您不查看 AWS 合理调整大小工具（如 AWS Compute Optimizer）的建议。

建立此最佳实践的好处： 通过使用节能且大小合适的实例，您可以大大减小工作负载对环境的影响并降低其成本。

未建立这种最佳实践的情况下暴露的风险等级： 低

实施指导

- 学习和探索可以减小工作负载对环境影响的实例类型。
  - 订阅 AWS 新增功能 及时了解新增的 AWS 技术和实例。
  - 了解不同的 AWS 实例类型。
  - 通过观看如下视频，了解基于 AWS Graviton 的实例（这些实例在 Amazon EC2 中每瓦能耗方面提供出色性能）： re:Invent 2020 - 深入了解 AWS Graviton2 处理器提供支持的 Amazon EC2 实例 和 深入了解 AWS Graviton3 和 Amazon EC2 C7g 实例。
- 规划工作负载并将其转换为影响极小的实例类型。

- 定义一个流程来评估工作负载的新功能或实例。利用云中的敏捷性，快速测试新的实例类型如何改善工作负载的环境可持续性。使用代理指标来衡量完成一个单元的工作需要多少资源。
- 如有可能，修改工作负载以使用不同数量的 vCPU 和不同数量的内存，以最大限度地增加您的实例类型选项。
- 考虑将工作负载转换为基于 Graviton 的实例，以提高工作负载的性能效率（请参阅 AWS Graviton Fast Start 和 适用于 ISV 的 AWS Graviton2）。请记住这些 注意事项，以便将工作负载转换为基于 AWS Graviton 的 Amazon Elastic Compute Cloud 实例。
- 考虑选择 AWS Graviton 选项（在使用 AWS 托管服务时）。
- 将工作负载迁移到提供对可持续性影响极小的实例且仍满足您的业务要求的区域。
- 对于机器学习工作负载，请使用基于定制 Amazon Machine Learning 芯片的 Amazon EC2 实例，例如 AWS Trainium、 AWS Inferentia和 Amazon EC2 DL1。
- 使用 Amazon SageMaker Inference Recommender 来调整 ML 推理端点的大小。
- 对于具有实时视频转码的工作负载，请使用 Amazon EC2 VT1 实例。
- 对于突增工作负载（不经常需要额外容量的工作负载），请使用 可突增性能实例。
- 对于无状态和容错工作负载，请使用 Amazon EC2 竞价型实例 提高云的整体利用率并减少未使用资源对可持续性的影响。

- 运营和优化您的工作负载实例。
  - 对于临时工作负载，请评估 实例 Amazon CloudWatch 指标 （例如 `CPUUtilization` ），以确定实例是空闲还是未充分利用。
  - 对于稳定的工作负载，请定期检查 AWS 合理调整大小工具（如 AWS Compute Optimizer ），以确定优化和合理调整实例大小的机会。

资源

相关文档：

- 优化您的 AWS 基础设施以实现可持续性，第 I 部分：计算
- AWS Graviton 处理器
- AWS Inferentia
- AWS Trainium
- Amazon EC2 DL1
- Amazon EC2 可突增性能实例
- Amazon EC2 容量预留实例集

- Amazon EC2 竞价型实例集
- Amazon EC2 竞价型实例
- Amazon EC2 VT1 实例
- Amazon EC2 实例类型
- AWS Compute Optimizer
- 函数：Lambda 函数配置

相关视频：

- 深入了解 AWS Graviton2 处理器提供支持的 Amazon EC2 实例
- 深入了解 AWS Graviton3 和 Amazon EC2 C7g 实例

相关示例：

- 实验室：合理调整大小建议
- 实验室：使用 Compute Optimizer 合理调整大小
- 实验室：优化硬件模式并遵守可持续性 KPI

SUS05-BP03 使用托管服务

托管服务将维持已部署硬件的高平均利用率和可持续性优化的责任转移给 AWS。使用托管服务将服务的可持续性影响分散到服务的所有租户，从而减少您的个人份额。

未建立这种最佳实践的情况下暴露的风险等级： 低

实施指导

- 将自托管服务迁移到托管服务。例如，使用托管式 Amazon Relational Database Service（Amazon RDS）实例而不是维护自己的 Amazon RDS 实例（在 Amazon Elastic Compute Cloud（Amazon EC2）上），或者使用托管式容器服务（例如 AWS Fargate），而不是实施您自己的容器基础设施。

资源

相关文档：

- AWS Fargate

- Amazon DocumentDB
- Amazon Elastic Kubernetes Service（EKS）
- Amazon Managed Streaming for Apache Kafka（Amazon MSK）
- Amazon Redshift
- Amazon Relational Database Service (RDS)

SUS05-BP04 优化您对 GPU 的使用

图形处理单元（GPU，Graphics Processing Units）可能是高功耗的来源，许多 GPU 工作负载是高度可变的，例如渲染、转码以及机器学习训练和建模。仅在需要时运行 GPU 实例，并在不需要时自动停用它们，以最大限度地减少资源消耗。

未建立此最佳实践暴露的风险等级： 低

实施指导

- 仅将 GPU 用于比基于 CPU 的替代方案更高效的任务。
- 使用自动化功能在不使用 GPU 实例时将其释放。
- 使用灵活的图形加速而不是专用的 GPU 实例。
- 利用特定于您的工作负载的定制用途硬件。

资源

相关文档：

- 加速计算型
- AWS Inferentia
- AWS Trainium
- EC2 实例的加速计算
- Amazon EC2 VT1 实例
- Amazon Elastic Graphics

开发和部署流程

问题

- SUS 6 您的开发和部署流程如何支持您的可持续发展目标？

SUS 6 您的开发和部署流程如何支持您的可持续发展目标？

寻找机会，通过对开发、测试和部署实践进行更改来降低可持续性影响。

最佳实践：

SUS06-BP01 采用可以快速引入可持续性改进的方法

在将潜在改进部署到生产环境之前对其进行测试和验证。在计算改进的潜在未来收益时，考虑测试成本。开发低成本的测试方法，以实现细微的改进。

未建立此最佳实践暴露的风险等级： 中

实施指导

- 在您的开发过程中添加可持续性要求。
- 允许资源并行工作以开发、测试和部署可持续性改进。
- 在将潜在可持续性影响改进部署到生产环境之前，对其进行测试和验证。
- 使用最小可行代表性组件测试潜在改进。
- 在经过测试的可持续性改进可用时将其部署到生产环境中。

资源

相关文档：

- AWS 支持可持续性解决方案

相关示例：

- 实验：将 成本和使用情况报告转化为效率报告

SUS06-BP02 让您的工作负载保持最新状态

最新的操作系统、库和应用程序可以提高工作负载效率，并简化更高效技术的采用。最新的软件可能还包括更准确地衡量工作负载对可持续性的影响的功能，因为供应商提供的功能是为了满足其自身的可持续性目标。

常见反模式：

- 您认为当前的架构将为静态并且不会随着时间的推移而更新。
- 您没有任何系统（也不会定期）评估更新的软件和软件包是否与您的工作负载兼容。
- 您可以随着时间的推移对架构进行更改，而无需提供理由。

建立此最佳实践的好处： 通过建立一个及时更新工作负载的流程，您将能够采用新的特性和功能，解决问题，并提高工作负载效率。

未建立这种最佳实践的情况下暴露的风险等级： 低

实施指导

- 定义一个流程和计划来评估工作负载的新功能或实例。利用云中的敏捷性，快速测试新功能如何改善工作负载以：
  - 减小对可持续性的影响。
  - 提升性能效率。
  - 为计划改进消除障碍。
  - 提高衡量和管理可持续性影响的能力。
- 盘点工作负载软件和架构，并确定需要更新的组件。您可以使用 AWS Systems Manager 清单 从 Amazon EC2 实例中收集操作系统（OS）、应用程序和实例元数据，并快速了解哪些实例正在运行您的软件策略所需的软件和配置，以及哪些实例需要更新。
- 了解如何更新工作负载的组件。
  - 管理适用于 Linux 或 Windows 服务器映像的 Amazon Machine Images (AMI) 的更新（使用 EC2 Image Builder）。
  - 您应该将 Amazon Elastic Container Registry (Amazon ECR) 与现有管道配合使用以 管理 Amazon Elastic Container Service (Amazon ECS) 映像 和 管理 Amazon Elastic Kubernetes Service 映像。
  - AWS Lambda 包括 版本管理功能。
- 采用自动化更新流程，以减少部署新功能的工作量，并减少手动过程引起的错误。使用 AWS Systems Manager Patch Manager 等工具自动执行系统更新流程，并使用 AWS Systems Manager 维护时段安排活动。

资源

相关文档：

- AWS Architecture Center
- AWS 新增功能
- AWS 开发人员工具
- AWS Systems Manager Patch Manager

相关示例：

- Well-Architected 实验室：清单和补丁管理
- 实验室：AWS Systems Manager

SUS06-BP03 提高构建环境的利用率

使用自动化和基础设施即代码功能，在需要时启动预生产环境，并在不使用时将其关闭。一种常见模式是安排与开发团队成员的工作时间相吻合的可用时段。休眠是一个有用的工具，它可以保存状态，并且只在需要时才快速将实例上线。使用具有爆增容量的实例类型、竞价型实例、弹性数据库服务、容器和其他技术，使开发和测试能力与使用相一致。

未建立此最佳实践暴露的风险等级： 低

实施指导

- 使用自动化功能来最大程度地利用开发和测试环境。
- 使用自动化功能来管理开发和测试环境的生命周期。
- 使用最小可行代表性环境来开发和测试潜在的改进。
- 使用按需型实例来补充您的开发人员设备。
- 使用自动化可以最大化构建资源的效率。
- 使用具有爆增容量的实例类型、竞价型实例和其他技术，使构建容量与使用保持一致。
- 采用原生云服务来实现安全的实例 Shell 访问，而不是部署堡垒主机群。

资源

相关文档：

- AWS Systems Manager Session Manager
- Amazon EC2 突发性能实例
- 什么是 AWS CloudFormation？

SUS06-BP04 使用托管式 Device Farm 进行测试

托管式设备场将硬件制造和资源使用的可持续性影响分散到多个租户。托管式设备场提供多种设备类型，使您能够支持不太受欢迎的较旧硬件，并避免不必要的设备升级对客户可持续性的影响。

未建立此最佳实践暴露的风险等级： 低

实施指导

使用具有代表性硬件集的托管式设备场进行测试，以了解更改的影响，并迭代开发以最大限度增加支持的设备数。

资源

相关文档：

- 什么是 AWS Device Farm？

声明

客户负责对本文档中的信息进行独立评估判断。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 协议规定，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。