AWS 良好架构框架

2017 年 11 月

通告

本文档所提供的信息仅供参考，且仅代表截至本文件发布之日时 AWS 的当前产品与实践情况，若有变更恕不另行通知。客户有责任利用自身信息独立评估本文档中的内容以及任何对 AWS 产品或服务的使用方式，任何“原文”内容不作为任何形式的担保、声明、合同承诺、条件或者来自 AWS 及其附属公司或供应商的授权保证。AWS 面向客户所履行之责任或者保障遵循 AWS 协议内容，本文件与此类责任或保障无关，亦不影响 AWS 与客户之间签订的任何协议内容。

目录

摘要

本份白皮书介绍了 AWS 良好架构框架，旨在帮助客户审查并改进自身云架构，同时更好地理解其设计决策对于业务的实际影响。我们在良好架构框架当中定义出五大概念性领域——即五大支柱，并针对这些领域提供一般性设计原则以及最佳实践与指导方针。

内容简介

AWS 良好架构框架帮助您理解在 AWS 构建系统决策中的优势和不足。通过这个框架，您将了解到架构最佳实践，包括在云上设计并运行可靠、安全、有效和成本优化的系统。这为您提供了一致性的标准，以衡量架构符合了最佳实践，并指出可以改进之处。我们认为，良好架构系统的存在将能够极大提升您的企业实现商业成功的机率。

AWS 的解决方案架构师们多年以来一直在积累面向多种垂直业务与用例的架构解决方案经验，而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。立足于如此丰厚的经验储备，我们最终得以总结出云环境下架构系统的最佳实践与核心战略性结论。

AWS 良好架构框架系列白皮书囊括了一系列基础性问题，可帮助您了解某种特定架构是否与您所设想的云最佳实践相适应及能否符合其具体要求。该框架提供的一致性方法可帮助您根据自身对现代云系统的实际预期对系统进行评估，并以此类要求为基础采取必要举措。随着 AWS 的持续发展，我们将继续同客户协作并增进了解，同时将现实经验融入到良好架构定义的持续完善当中。

此份白皮书主要面向各类技术性角色，例如首席技术官（CTO）、架构师、开发人员以及运维团队成员等。在阅读本份白皮书后，您将了解到在进行云架构设计及运维时，可资运用的 AWS 最佳实践与各类策略。本份白皮书并不提供任何实现细节或架构模式，但是会包括为您提供此类信息引用的对应资源。了解更多信息，请访问 AWS 良好架构框架主页。

定义

每一天，AWS 的专家们都会协助客户进行系统架构设计，以期发挥云服务最佳实践的优势。我们与您携手并进，根据设计方案的演进对架构进行调整与权衡。当您将这些系统部署在运行环境当中后，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。

根据多年来积累的经验，我们创建起 AWS 良好架构框架，向客户和合作伙伴提供一系列最佳实践以评估架构，并包含一系列问题，您可用于评估自身架构与 AWS 最佳实践的具体契合程度。

AWS 良好架构框架基于五大核心支柱，分别为卓越操作、安全性、可靠性、性能效率与成本优化。

支柱名称	描述

卓越操作	运行并监控系统以交付商业价值，同时持续改进支持流程与规程的能力。
安全性	在交付商业价值的同时，通过风险评估与缓解策略保护信息、系统以及资产的能力。
可靠性	系统从基础设施或者服务故障当中实现恢复、以动态方式获取计算资源以满足需求，以及缓解配置错误或者暂时性网络问题等干扰因素的能力。
性能效率	高效利用计算资源以满足系统要求，同时在需求变更及技术演进时维持这种效率水平的能力。
成本优化	避免或者消除不必要成本或者欠佳资源的能力。

在构建解决方案时，您应根据实际业务情景在各核心支柱之间进行权衡，而此类业务决策亦直接反映着您的工程技术优先级判断。您可以在开发环境减少可靠性所需的成本，亦可在关键性系统解决方案当中提升成本预算以获得更理想的可靠性。在电子商务解决方案层面，性能往往决定着客户购买意愿乃至企业整体营收。另外，在实际权衡当中，安全性与卓越操作与其它支柱一般并不存在冲突。

关于架构

使用本地环境的客户，通常有一个集中的团队来设计技术架构，协调其他产品或者功能团队，确保遵循最佳实践。技术架构团队通常由一系列的角色来构成，包括技术架构师（基础设施）、方案架构师（软件）、数据分析师、网络架构师和安全架构师。这些团队通常使用 TOGAF 或者 Zachman 框架，作为企业架构能力的一部分。

对于亚马逊来说，我们倾向于把能力分散到不同的团队，而不是一个集中的团队。当您选择下放决定权时存在一定风险，例如，确保团队满足内部标准。我们通过两种方式减少风险。第一，我们的*实践*专注于每个团队有足够能力，我们的专家确保可以达到所需的标准条件。其次，我们有一些*机制*，可以进行自动检查，以满足标准。亚马逊的领导力准则支持这种分布式的方法，并为所有*以客户为始终*的角色建立一种文化。专注于客户的团队根据客户需求创造产品。

对于架构来说，这意味着每个团队有能力创建架构，并遵循最佳实践。为了帮助新团队获得这些能力，或者已有团队提升能力，我们开放了一个虚拟社区，让那些首席工程师可以审查他们的设计，帮助他们理解 AWS 最佳实践。首席工程师社区使最佳实践可见可用。一个方法是，在午餐时间讨论如何把最佳实践应用于实际场景。这些讨论被记录下来，成为团队新成员学习材料的一部分。

AWS 最佳实践来自于我们在互联网规模运转数千个系统的经验。我们愿意用数据来定义最佳实践，但是我们也邀请诸如首席工程师的特定领域专家来参与制定。当首席工程师发现新的最佳实践，他们会以社区形式确保遵循实践。同时，这些最佳实践会进入我们的内控流程，以及强制合规机制。

良好架构是我们内控流程面向客户的实施方式，使我们的首席工程师站在实施角色的角度去思考，例如方案架构师和内部工程团队。良好架构是可扩展的途径，使您在学习中占据先机。

通过遵循首席工程师社区里分散架构责任的方法，我们相信设计良好的企业架构由客户需求驱动并可以显现。技术领导者（例如 CTO 或者开发经理），在您的所有业务负载上进行良好架构审查，可以让您更好地理解您的技术栈的风险。以此方法，您可以确定不同团队间可使用的主题，通过机制、培训和午餐交流，您的首席工程师可以和多个团队分享他们在特定领域的想法。

一般性设计原则

良好架构框架定义了一系列一般性设计原则，以促成云上的良好设计：

- **不再猜测您的容量需求：**不再猜测您的基础设施容量需求。在以往系统实际部署前的容量决策环节当中，您通常会面临大量高成本闲置资源抑或是容量紧张而导致商业绩效受到影响这两类困扰。利用云计算，此类问题将得以解决。您可以首先尽可能降低容量水平，并根据后续需求自动对容量规模进行伸缩。

- **以生产规模进行系统测试：**在云环境当中，您可以根据需要创建起一套生产规模等级的测试环境、完成测试、而后清空这部分资源。由于您仅需要为实际运行的测试环境资源量付费，因此可以模拟出生产环境，而成本只相当于在自建机房测试的一小部分。

- **自动简化架构实验：**自动化机制能够帮助您以更低成本创建并复制系统方案，同时避免手动操作带来的种种投入。您可以追踪自动化机制中的各项变更、审计相关影响并在必要时恢复至原有参数。

- **允许实现架构演进：**在传统环境当中，架构决策通常是静态的一次性事件，一个系统的整个生命周期当中往往只有几个主要版本。而随着业务及其情景的持续变化，这些最初的决策结果可能无法适应不断变化的业务能力需求。在云环境当中，自动化与按需测试能力将大幅降低设计变更带来的影响及风险，允许系统随着时间推移而不断发展，最终帮助企业将创新作为标准实践。

- **使用数据驱动架构：**在云环境中，您可以收集相关数据，用以了解您的架构选择会给工作负载的具体行为带来哪些影响。以此为基础，您将能

够立足事实就工作负载改进作出决策。您的云基础设施以代码形式存在，因此您能够利用这部分数据指导您的架构选择并随时间推移加以改进。

- **通过竞赛日活动实现改进：**通过定期组织竞赛日活动模拟实际生产中的各类情况，并借此对您的架构及流程进行测试。这将帮助您了解潜在的改进空间并在处理此类情况时积累起丰富的实践经验。

良好架构框架中的五大支柱

软件系统的创建类似于建设建筑物。如果地基不够牢固，那么结构性问题很有可能破坏建筑物的完整性与功能。在架构技术解决方案方面，如果您忽略了安全性、可靠性、性能效率、成本优化与卓越操作五大基本支柱，那么系统恐怕也将无法满足您的期望与要求。如果能够将这些支柱纳入您的架构，您的系统将带来更加稳定且高效的实际表现。以此为基础，您将能够专注于设计工作的其它层面，例如满足受众功能需求等。

卓越操作支柱

卓越操作支柱包括运行和监控系统的能力，以交付商业价值，并且持续改进支持流程。

卓越操作支柱包括设计原则总览、最佳实践和问题。您可以从以下卓越操作白皮书中获得实践指导。

设计原则

在云环境当中，以下 6 个指导性原则可用于支持卓越操作：

- **利用代码执行操作**：在云中，可以将相同的程序代码应用于整个环境。您可以将整个工作负载（应用程序，基础架构等）定义为代码并使用代码进行更新。您可以为操作流程编写脚本，通过事件触发自动执行。以代码执行操作，可以减少人为错误，保持事件响应的一致。
- **为文档加上注释：**在自建机房环境中，文档是手工创建的，很难在变化时同步更新。在云中，您可以在每次部署后自动创建文档（或者自动注释手工文档）。注释文档可以人工或者系统完成。在操作代码中以注释作为输入。

- **进行定期、小型、增量式变更：**应在设计当中确保工作负载允许各组件进行定期更新。变更应以小型增量化方式进行，还应能够在不影响正常操作的前提下实现回滚。
- **经常改善运营流程：**使用运营流程时，发现改善的机会。当工作负载变化时，改善相应流程。定期举办比赛日，以确保所有流程有效，并且为团队所熟悉。
- **故障演习：**进行故障演习，指出潜在的问题，以便消除或者减轻问题。测试故障场景，检查对影响的理解程度。测试响应流程，确保行之有效，并且团队也熟悉流程。定期举办比赛日，测试工作负载和团队对于事件的响应。
- **从操作事件及故障中总结经验：**从操作事件及故障所学到的经验中改进流程。在团队和整个组织之间分享经验。

定义

云环境下的卓越操作最佳实践包括以下三点：

- 筹备
- 运营
- 演变

运营团队需要理解业务和客户需求，才能有效支持业务。运营创建并使用流程，以响应运维事件，并验证业务支持效率。运营收集用于衡量计划业务成果的指标。一切都在不断变化 —— 您的业务环境，业务优先级，客户需求等。设计流程以响应变化，并吸取经验教训是非常重要的。

最佳实践

筹备

需要有效的准备来推动卓越操作。业务成功是通过业务，开发和运营的共同目标和理解实现的。通用标准简化了工作负载的设计和管理，使运营成功。通过机制来设计工作负载，以监视并深入了解应用程序，平台和基础架构组件以及客户体验和行为。

创建机制来验证工作负载和修改，以确认可以转移到生产环境中，并可以运营。运营准备通过清单进行确认，以确保工作负载符合定义的标准，并且符合操作手册和指导里的流程。确认是否有足够的受过训练的人员来有效支持工作。在迁移之前，测试对运营事件和故障的响应。通过生成故障和比赛日事件，在支持的环境中进行演习。

AWS 在云上以代码来运营，并能够安全地进行实验，开发运营流程，进行故障演习。使用 AWS CloudFormation，您可以随着运营水平的不断提高，建立一致的、模板化的沙箱开发、测试和生产环境。AWS 通过各种日志收集和监控功能，使您能够查看所有层的工作负载。使用资源、应用程序编程接口（API）和网络流日志的数据，可以通过 Amazon CloudWatch，AWS CloudTrail 和 VPC Flow Logs 进行收集。您可以使用 CloudWatch Logs 代理或 collectd 插件将有关操作系统的信息汇总到 CloudWatch 中。

以下问题需要在卓越操作的准备阶段进行考虑。有关卓越操作问题、答案和最佳实践的列表，请参阅附录。）

问题 1：什么因素推动您的业务重点？

问题 2：如何设计工作负载以实现可操作性？

问题 3：如何知晓你已准备就绪支持一个工作负载？

工作负载采用最少数量的架构标准。在实施成本与工作利益，以及运营成本之间，找到平衡。减少支持的标准数量，以减少错误应用低于可接受的标准的可能性。运营人员通常访问资源要受到限制。

使用脚本操作以最大限度提高操作人员的效率，最大限度地减少错误率，并实现自动化响应。利用云的弹性部署实践，来促进系统的预部署，以加快实施速度。

运营

成功的运营是通过业务和客户成果来衡量的。定义预期成果，确定成功标准，计算负载和运营指标来衡量是否成功。考虑运行状况健康，包括工作负载的健康状况，以及根据负载进行操作（例如部署和事件响应）的健康状况。建立标准，以判断运营是进步还是退步，收集并分析您的指标，验证您对成功运营的理解，以及如何随着时间而变化。收集指标来确定您是否满足客户和业务需求，并确定需要改进的地方。

运营事件的高效和有效管理是实现卓越操作所必需的。这适用于计划内和计划外的运营事件。使用运维手册处理定义事件，使用指导手册来解决其他事件。根据对业务和客户影响，对事件定义优先级。如果某个事件引发了警报，与指定人员一起启动相应流程。根据影响（持续时间，规模和范围），提前确定解决事件所需的人员，包括升级机制，以便在需要时让更多人员参与进来。确定并联系决策者，对于之前未定义的事件采取行动。

通过为目标受众定制的仪表板和通知（例如，客户，业务，开发人员，运营等），发布工作负载的运行状态，以便他们可以采取适当的行动，管理他们的期望，并在恢复时通知。

确定计划外事件的根本原因，以及计划事件的意外影响。这些信息将用于更新流程，以减少类似事件的发生。方便时在社区发布故障原因。

在 AWS 中，您可以生成仪表板视图，指标来自负载指标，以及 AWS 本地。您可以利用 CloudWatch 或第三方应用程序来汇总当前业务量、工作负载和运营水平视图。AWS 通过日志记录功能（包括 AWS X-Ray，CloudWatch，CloudTrail 和 VPC Flow Logs）提供指标数据，以分析负载问题，找到故障原因并修复。

以下问题需要在卓越操作时重点考虑。

问题 4：什么因素促进您对健康运营的理解？

问题 5：您如何管理运营事件？

常规操作和意外事件的响应应该是自动的。应避免手动部署、版本管理、更改和回滚。版本不应该进行不频繁的大批量更新。如果有大量更新，回滚更困难。如果没有回滚计划，或减轻故障影响的能力，将会影响运营的连续性。指标与业务需求相适应，以便能够有效维护业务连续性。手动的一次性的分散指标，会导致计划外事件中的操作受到更大影响。

演变

运营的演变是持续卓越操作所必需的。专注于工作周期以实现持续的渐进式改进。定期评估和优先处理改进机会（例如，功能请求，问题修复和合规性要求），包括负载和运营流程。对流程进行反馈，以快速确定需要改进的地方，并从运行中学习。

分享团队学到的经验教训，分享这些经验教训的好处。从经验教训中分析趋势，并对运营指标进行跨团队回顾分析，以确定改进的机会和方法。实施改进，并评估结果。

借助 AWS 开发工具，您可以持续进行构建，测试和部署活动，这些活动可与来自 AWS 和第三方的各种源代码，构建，测试和部署工具配合使用。部署活动的结果可用于分析部署和开发的改进机会。您可以对指标数据执行分析，将来自您的运营和部署活动的数据进行集成，以便分析这些活动对业务和客户结果的影响。这些数据可用于跨团队回顾性分析，以确定改进的机会和方法。

以下问题在卓越操作的演变时重点考虑。

问题 6：如何进行运营演变？

运营的成功演变源于：频繁的小改进；为实验、开发和测试改进提供安全的环境和时间；以及有环境鼓励从失败中学习。运营支持沙箱、开发、测试和生产环境，随着运营控制水平的提高，促进开发水平，增加生产环境部署成功的可能性。

关键性 AWS 服务

对卓越操作至关重要的 AWS 服务是 AWS CloudFormation，您可以使用它来基于最佳实践创建模板。这使您能够从开发到生产环境，以有序且一致的方式配置资源。以下服务和功能支持卓越操作的三个方面：

- **筹备：**AWS Config 和 AWS Config 规则可用于创建工作负载标准，并确定环境在投入生产之前是否符合这些标准。
- **运营：**Amazon CloudWatch 允许您监控工作负载的运行状况。
- **演变：**Amazon Elasticsearch Service（Amazon ES）允许您分析日志数据以快速安全地获取所需信息。

资源

请参阅以下资源，详细了解我们的卓越操作最佳实践。

文档

- DevOps and AWS

白皮书

- Operational Excellence Pillar

视频

- AWS re:Invent 2015 – DevOps at Amazon

安全性支柱

安全性支柱包含在交付商业价值的同时，通过风险评估与缓解策略保护信息、系统以及资产的能力。

安全性支柱提供了对于设计原则、最佳实践和问题的总览。您可以在安全性支柱白皮书中找到实施规范指导。

设计原则

在云环境对于安全性支柱有 6 个设计原则。

- **健壮的身份验证体系：**实现最少权限，对于 AWS 资源的所有访问都需要有正确的授权。权限集中管理，减少甚至取消长期证书。
- **实现可追踪性：**对于操作和更改进行实时的监控、报警和审计。集成日志和指标，自动响应并采取措施。
- **将安全引入一切层面：**相较仅在外层防护，引入包括其他安全控制的深度防护措施。应用于所有层面，包括边缘网络、VPC、子网、负载均衡器、每个实例、操作系统和应用程序。
- **自动化安全最佳实践：**基于软件的安全机制能够改善您以安全、快速且具备成本效益的方式实现规模扩展的能力。建立安全架构，包括如何实现以代码方式定义和管理版本控制。
- **保护动态和静态数据：**把数据按照敏感程度归类，使用安全机制，比如加密和使用令牌。减少甚至取消人工访问数据，以减少丢失或者篡改的风险。
- **为安全事件做预案：**根据组织要求，准备应急预案流程。进行故障仿真测试，利用自动化工具以快速检测、调查和恢复。

定义

云环境下的安全性最佳实践主要分为以下五点：

- 身份与访问管理
- 检测控制
- 基础设施保护
- 数据保护
- 事件响应

在构建任何系统之前，您需要切实采取以上实践以保障安全。您需要确切控制谁能够执行哪些操作。此外，您还需要识别各类安全事件、保护您的系统与服务，并通过数据保护机制维持数据的机密性与完整性。您应当建立起一套明确的可操作流程以应对安全事件。这些工具与技术非常重要，因为其是您实现各类安全目标的基础所在——包括避免财务损失或者遵循监管义务。

AWS 责任分担模式使得各组织机构能够利用云服务实现其安全性与合规性目标。由于 AWS 自身负责保护用于支持云服务的基础设施，因此 AWS 客户将能够专注于使用这些服务以完成自身业务目标。另外，AWS 云还提供更为出色的安全数据访问机制以及针对安全事件的自动化响应方法。

最佳实践

身份与访问管理

身份与访问管理属于信息安全规划当中的核心组成部分，负责确保仅经过验证及授权的用户方可以符合预期的方式访问您的资源。举例来说，您可定义各相关主体（包括用户、组、服务与角色对您帐户采取的具体操作）、制定与这些主体相统一的政策，同时实施强大的凭证管理。这些权限管理因素将共同构成业务系统内验证与授权的核心概念。

在 AWS 当中，权限管理主要由 AWS 身份与访问管理（简称 IAM）服务负责支持，其允许客户面向实际用户控制一切 AWS 服务与资源的访问行为。您可以应用细粒度政策，为各用户、组、角色或者资源分配具体权限。您亦能够要求采取强密码实践，例如复杂程度、避免重复使用以及使用多因素验证（简称 MFA）机制。您可以将其与其它现有目录服务相结合。对于那些需要系统接入 AWS 的工作负载，IAM 可通过实例配置、身份联动以及临时凭证等方式实现安全访问。

以下问题主要着眼于安全性层面的权限管理类注意事项（与安全性相关的详尽问题、回答及最佳实践清单，请参阅附录部分）。

问题 1：您如何保护 AWS root 帐户凭证的访问与使用活动？

问题 2：您如何为系统用户定义角色与责任，并借此控制指向 AWS 管理控制台与 API 的人为访问？

问题 3：您如何限制指向各 AWS 资源的自动访问？（例如应用程序、脚本以及/或者第三方工具或服务）

保护 root 帐户凭证至关重要。为了实现这一目标，AWS 建议您将 MFA 添加至 root 帐户当中，同时将 MFA 凭证锁在物理安全位置之内。IAM 服务允许您创建并管理其它非 root 用户权限，同时建立起指向各类资源的访问级别。

检测控制

您可以利用检测控制机制识别潜在的安全事件。此类控制手段属于治理框架当中的重要组成部分，可用于支持质量流程、法律或者合规性义务以及威胁识别与响应行动。检测控制分为多种不同类型，例如用于进行资产清单及其详尽属性整理以促进高效决策（与生命周期控制），进而帮助建立运营基准等等。或者，您也可以采取内部审计机制——即对与信息系统相关之控制方案加以检查，从而确保各项实践满足政策与要求，这部分明确定义的条件将成为进一步设置自动化警报通知机制的基础。这些控制手段皆属于极为重要的反应性因素，能够帮助企业有效识别并理解异常活动的影响范围。

在 AWS 当中，您可以通过处理日志、事件以及监控等方式实现审计、自动化分析与警报来实现检测控制。AWS CloudTrail 日志、AWS API 调用以及 Amazon CloudWatch 皆可提供面向特定指标的监控与警报功能，而 AWS Config 则可提供配置历史信息。AWS 同样提供服务层级日志记录，例如您可利用 Amazon 简单存储服务（即 Amazon Simple Storage Service ，简称 S3）记录访问请求。最后，Amazon Glacier 一项库锁功能，旨在利用专门用于支持可审计长期保留场景的合规控制机制保存关键性任务数据。

以下问题专注于检测控制手段中的安全性注意事项：

问题 4：您如何捕捉并分析日志内容？

日志管理对于一套良好架构设计方案而言非常重要，具体原因包括安全性/取证以及监管乃至法律要求。您必须有能力对日志进行分析与响应，并借此发现潜在的安全事件。AWS 为用户提供相关能力以轻松实现日志记录管理，这意味着用户将有能力定义数据保留生命周期，或者定义数据在何处进行保留、归档以及/或者最终删除。这一切都将令数据处理工作的可预测性与可靠性得到提升、实现流程得以简化、成本效益实现改善。

基础设施保护

基础设施保护工作当中包含多种控制方法，例如纵深防御以及多因素验证，这些已经成为实现最佳实践并满足行业或监管义务的必要前提。对于这些方法的运用亦属于立足云端或者本地环境实现成功持续运营的关键性条件。

在 AWS 当中，您可以实现有状态及无状态数据包检测，具体利用 AWS 原生技术或者 AWS Marketplace 提供的第三方合作伙伴产品及服务实现。您亦可利用 Amazon 虚拟私有云（Amazon Virtual Private Cloud，简称 VPC）创建起一套私有、安全且具备可扩展能力的环境。您可以在其中定义您的拓扑结构——包括网关、路由表以及公开与/或私有子网。

以下问题专注于基础设施保护层面的安全注意事项：

问题 5：您如何确保网络及主机层级边界受到保护？

问题 6：您如何运用 AWS 服务级安全功能？

问题 7：您如何在自己的 Amazon EC2 实例之上保护操作系统完整性？

在任何类型的环境当中，我们都建议您采用多层防御机制；而在基础设施保护层面，现已存在大量经过实践验证的跨云与本地环境安全保护模式概念与方法。强化边界保护、监控入站与出站点以及建立全局性日志记录、监控与警报机制都能够在实现信息安全保障方面起到至关重要的作用。

AWS 客户能够对单一 EC2 实例的配置进行定制或者强化，并将这套配置方案长期保存在 Amazon Machine Image（简称 AMI）镜像当中。在此之后，无论是通过 Auto Scaling 还是手动启动方式，全部新的虚拟服务器（实例）皆可利用此 AMI 进行启动并立即获得经过强化的配置方案。

数据保护

在为任何系统设计架构之前，我们首先应当确定可能影响安全性的各项基础实践。举例来说，数据分级提供的分类方法以敏感度为基础，旨在帮助企业立足不同加密级别对数据提供保护，确保其不会受到未经授权之访问活动的影响。这些工具与技术非常重要，能够有效支持预防经济损失或者遵循监管义务等目标。

在 AWS 当中，以下实践有助于保护您的业务数据：

- AWS 客户保有对其数据的全部控制权。
- AWS 帮助您更为轻松地实现数据加密与密钥管理，具体包括定期进行密钥轮换——AWS 提供的原生功能可轻松实现密钥自动轮换，用户亦可根据需求选择手动操作。
- 通过日志详尽记录各类重要内容，例如文件访问与变更等等。
- AWS 已经设计出多种具备卓越弹性水平的存储系统。举例来说，Amazon 简单存储服务（简称 S3）即可提供 11 个 9 持久性。（具体而言，如果您利用 Amazon S3 存储 1 万个对象，则平均每 1000 万年才会遭遇单一对象丢失。）
- 作为大规模数据生命周期管理流程当中的重要组成部分，版本控制能够保护您的数据免受意外覆盖、删除以及其它类似操作的影响。
- AWS 永远不会在各区域之间主动进行数据移动。保存在特定区域内的内容将始终存在于该区域，除非客户明确启用相关功能或者使用该功能以执行数据迁移。

以下问题侧重于数据安全性相关注意事项：

问题 8：您如何对数据进行分级？

问题 9：您如何对静态数据进行加密与保护？

问题 10：您如何管理密钥？

问题 11：您如何对传输中的数据进行加密与保护？

AWS 面向静态与传输中的数据提供多种加密实现方法。我们在产品及服务当中引入相关功能，旨在尽可能简化客户数据的加密流程。举例来说，我们在 Amazon S3 当中引入了服务器端加密（简称 SSE）机制，希望能够帮助客户以更为轻松的方式以加密格式进行数据存储。您亦可以采用弹性负载均衡（Elastic Load Balancing）服务全面采用 HTTPS 加密与解密流程（通常被称为 SSL 终止）。

事件响应

即使拥有非常成熟的预防与检测控制机制，企业仍然有必要建立相关流程以响应并缓解由安全事件引发的潜在影响。您工作负载的具体架构将极大影响到团队在事件发生期间采取行动，进而隔离或约束系统并将运行状态恢复至已知良好状态的能力。首先我们应当在安全事件发生前将工具部署到位，而后定期演练安全事件响应方案并及时更新架构，以确保在必要时其能够顺利完成调查取证与恢复任务。

在 AWS 当中，以下实践有助于提升安全事件响应效率：

- 采用详尽的日志记录方案，其中应包含文件访问与变更等重要内容。
- 事件可进行自动处理，而触发脚本则通过使用 AWS API 自动依照操作手册作出反应。
- 您可预先配置工具集并利用 AWS CloudFormation 建立“清洁屋”。如此一来，您将能够立足一套安全且经过隔离的环境进行取证。

以下问题侧重于**事件响应**层面的注意事项：

问题 12：您如何确保您拥有适当的事件响应机制？

确保您有能力快速为信息安全团队赋予访问权限，同时以自动化方式进行实例隔离，并捕捉各类数据与状态信息以作为事件取证。

关键性 AWS 服务

AWS 服务当中最为核心的安全性方案是 AWS 身份与访问管理（简称 IAM），其允许客户以安全方式控制用户访问各 AWS 服务与资源的。以下服务与功能则分别对安全保障层面的五个方面提供支持：

- **身份与访问管理:** IAM 允许您以安全方式控制一切指向 AWS 服务与资源的访问活动。多因素验证（简称 MFA）则在用户名与密码机制之外带来额外的保护层。

- **检测控制:** AWS CloudTrail 能够记录各项 AWS API 调用，AWS Config 负责提供一份关于您 AWS 资源与配置的详尽清单，而 Amazon CloudWatch 则是一项对各项 AWS 资源加以监控的服务。
- **基础设施保护：** Amazon 虚拟私有云（简称 VPC）允许您在 AWS 云当中配置出一套私有隔离化分区，您可在其中立足虚拟网络启动 AWS 资源。
- **数据保护:** 弹性负载均衡、Amazon 弹性块存储（Amazon Elastic Block Store，简称 EBS）、Amazon 简单存储服务（简称 S3）以及 Amazon 关系数据库服务（Amazon Relational Database Service，简称 RDS）等服务当中包含的加密功能可对您处于传输及静态数据提供保护。AWS 密钥管理服务（AWS Key Management Service，简称 KMS）则能够简化客户创建并控制各加密密钥的具体流程。
- **事件响应:** IAM 应用于为事件响应团队提供适当的验证机制。Amazon CloudFormation 服务则可用于创建一套受信环境，用以进行取证调查。

资源

您可以参考以下资源以了解更多与 AWS 云安全性最佳实践相关的信息。

文档

- AWS 安全中心
- AWS 合规性
- AWS 安全博客

白皮书

- AWS 安全性概述
- AWS 安全性最佳实践
- AWS 风险与合规性

视频

- AWS 云安全性
- 责任分摊模式概述

可靠性支柱

可靠性支柱包含系统从基础设施或者服务故障当中实现恢复、以动态方式获取计算资源以满足需求，以及缓解配置错误或者瞬态网络问题等干扰因素的能力。

可靠性支柱提供了对于设计原则、最佳实践和问题的总览。您可以在可靠性支柱白皮书中找到实施规范指导。

设计原则

在云环境当中，以下原则能够帮助您有效提升可靠性水平：

- **测试恢复规程：**在本地环境当中，测试工作通常用以证明系统在特定场景下的运作能力；这类测试一般不负责验证恢复策略。但在云环境内，您可以测试系统发生故障时的情景，并借此验证您的恢复规程是否切实有效。您可利用自动化方式模拟不同的故障状况，或者反复建立可能导致故障的场景。您能够借此发现故障的产生过程，同时在发生真实故障问题**之前**对潜在风险因素进行检查与纠正，并借此降低以往从未测试过的组件可能引发的潜在危害。
- **自动故障恢复：**通过对系统中的各关键性能指标（简称 KPI）进行监控，您可以在触发特定阈值时启动自动化响应机制。其允许自动进行故障通知与追踪，同时利用自动化恢复流程缓解或者修复具体故障。利用技术水平更高的自动化方案，我们甚至有可能在故障实际发生之前作出预测并加以解决。
- **横向扩展以提升总体系统可用性：**利用多种小型资源取代单一大型资源，借此降低单一故障对于整体系统产生的影响。跨越多种小型资源的分布式请求将能够确保其不致共同面对同一故障点。
- **不再依靠猜测确定容量需求：**内部系统当中的一大常见故障根源在于资源饱和，即系统请求的资源超过该系统的资源供给能力（拒绝服务攻击通常正是以此为目标）。在云环境当中，您可以监控系统资源的需求与利用率，并通过自动化方式添加或者移除部分资源以维持其最优水平，从而在无需过度或者过低配置的情况下满足实际需求。
- **自动管理变更：**基础设施内的变更应以自动化方式实现。需要加以管理的一切变更皆应被纳入自动化变更范畴之内。

定义

云环境下的可靠性保护包含以下三项最佳实践：

- 基础

- 变更管理
- 故障管理

为了实现这种可靠性，系统必须拥有经过良好规划的基础且将监控机制部署到位，同时有能力根据需求或者要求处理各项变更。该系统还应在设计当中引入故障检测与自动化自主修复能力。

最佳实践

基础

在构建任何系统之前，您应首先确定可能对其可靠性造成影响的基础性因素。举例来说，您必须拥有充足的网络数据中心传输带宽。此类要求有时会遭到忽略（因为其走出了单一项目的范畴），而这种忽略很可能对系统的可靠性水平造成重大影响。在本地环境当中，由于此类要求可能长期存在且面对极高的依赖度，因此必须在规划初期即被纳入考量。

在 AWS 当中，大部分此类基础性要求已经被纳入，亦可根据您的实际需要进行处理。云环境在设计层面拥有几乎无限的资源储备，因此 AWS 有责任满足一切对于网络与计算能力的需求——您可以随意调整资源规模，例如根据需求分配存储设备容量。

以下问题侧重于可靠性层面的注意事项：

问题 1：您如何管理您帐户当中的 AWS 服务限制？

问题 2：您如何规划您 AWS 之上的网络拓扑结构？

AWS 设置有服务限制（即您的团队所能够请求的每一种资源上限），旨在保护您免受资源过度配置问题的意外影响。您需要部署治理机制与相关流程以监控并变更这些限制，确保其始终与您的业务需求相匹配。在采用云服务的同时，您亦需要思考如何将其同您的现有本地资源加以整合（即混合型方案）。这类混合模式可随时间推移逐步转化为纯云解决方案，因此您显然有必要在网络拓扑设计当中充分考量 AWS 与内部资源间的交互方式。

变更管理

了解变更对于系统的实际影响能够确保您主动作出规划，而监控能力则可帮助您快速发现一切可能导致容量问题或者 SLA 违规的负面趋势。在传统环境当中，变更控制流程通常以手动方式进行，且必须配合认真协调与审计方可有效控制执行操作的具体个人及活动时间。

利用 AWS，您将能够监控系统活动并自动根据 KPI 作出响应——例如在系统迎来更多用户时自动添加服务器数量。您可以控制哪些用户拥有权限以作出系统变更，或者对系统变更记录进行审计。

以下问题侧重于变更操作与系统可靠性影响间关系的相关注意事项：

问题 3：您的系统如何根据需求适应各项变更？

问题 4：您如何监控各项 AWS 资源？

问题 5：您如何执行变更？

当您构建一套系统以自动根据变更需求添加及移除资源时，其不仅能够提升可靠性水平，同时亦可确保业务层面的成功不致给底层基础设施带来负担。利用监控方案，您的团队将能够以自动化方式在 KPI 偏离预期时发出提醒。自动变更记录则允许您对自有环境进行审计，同时快速发现可能对可靠性造成影响的具体活动。变更管理机制中的控制手段则可确保您强制执行规则，用以实现您所需要的可靠性水平。

故障管理

在任何具备一定复杂度水平的系统当中，故障或早或晚都会出现，因此我们自然有必要考虑如何发现这些故障、对其作出正确响应并避免同样的故障再度发生。

在 AWS 当中，我们可以充分发挥自动化优势以响应数据监控结果。举例来说，当特定指标超出阈值时，您可以触发自动化操作以解决问题。此外，相较于以往对生产环境下组成部分的特定资源进行诊断与修复，现在我们可以利用新资源加以替代并对被替换下来的旧有资源进行故障排查。由于云环境允许您以极低成本快速启动生产系统的临时性版本，因此您可以利用自动测试以验证整个故障恢复流程。

以下问题侧重于可靠性层面内的故障管理注意事项：

问题 6：您如何对自己的数据进行备份？

问题 7：您的系统拥有怎样的组件故障承受能力？

问题 8：您如何对弹性进行测试？

问题 9：您为灾难恢复制定了怎样的计划？

请定期备份您的数据并对备份文件进行测试，从而确保您有能力从逻辑及物理错误当中恢复过来。故障管理工作中的一大关键在于频度、系统故障以及恢复流程自动测试机制（在理想情况下，您应制定定期规划并在系统发生显著变更后触发此类测试）。主动追踪 KPI——例如恢复时间目标（简称 RTO）与恢复点目标（简称 RPO）——以评估系统弹性水平（特别是在故障测试场景之下）。追踪 KPI 将帮助您快速发现并解决单点故障因素。这方面的核心目标在于对系统的恢复流程进行全面检查，从而帮助您建立起充分的数据恢复与客户持续服务信心，甚至确保您有能力面对持续存在的挑战性难题。您的恢复流程应当如正常生产流程一样完备而可靠。

关键性 AWS 服务

在这方面，确保可靠性的关键性 AWS 服务为 Amazon CloudWatch——其负责监控各项运行时指标。其它可支持可靠性层面内三大领域的服务与功能还包括：

- **基础:** AWS 身份与访问管理（简称 IAM）允许您以安全方式控制访问 AWS 服务与资源。Amazon VPC 允许您立足 AWS 云配置一套私有且经过隔离的分区，您可在其中利用私有网络启动 AWS 资源。AWS Trusted Advisor 可以查看服务限制。AWS Shield 是防止 DDOS 的托管服务，可以保护运行在 AWS 的网站。
- **变更管理:** AWS CloudTrail 负责为您的帐户记录 AWS API 调用并为审计工作提供必要的日志记录文件。AWS Config 可提供一份包含 AWS 资源与配置信息的详尽清单，同时持续记录配置变更情况。Auto Scaling 服务可以对已经配置的负载进行自动容量管理。CloudWatch 提供了针对指标的报警，包括客户自定义指标。
- **故障管理:** AWS CloudFormation 提供多种模板以实现 AWS 资源创建，同时以按需且可预测方式对其加以配置。Amazon S3 提供了高持久性可用于备份。Amazon Glacier 提供高持久性归档。AWS KMS 提供了可靠的密钥管理服务，与许多 AWS 服务相集成。

资源

欲了解更多与可靠性最佳实践相关的细节信息，请参阅以下资源。

文档

- 服务限制 Limits
- 服务限制报告博客
- AWS Shield
- Amazon CloudWatch

- Amazon S3
- AWS KMS

白皮书

- 可靠性支柱
- 利用 AWS 实现备份归档与恢复
- 以规模化方式管理您的 AWS 基础设施
- AWS 灾难恢复
- AWS Amazon VPC 连接选项

视频

- 如何管理 AWS 服务限制？
- 拥抱故障：错误注入与服务可靠性

报告

- 云环境下的可用性与可靠性基准

AWS 技术支持

- AWS Premium 技术支持服务
- Trusted Advisor

性能效率支柱

性能效率支柱专注于高效利用计算资源以满足系统要求，同时根据需求变更及技术演进维持这种效率水平的能力。

性能效率支柱提供了对于设计原则、最佳实践和问题的总览。您可以在性能效率白皮书中找到实施规范指导。

设计原则

在云环境当中，以下设计原则能够帮助您实现性能效率目标：

- **普及先进技术：**原本难以实现的技术方案，如今通过将相关知识与复杂性因素转由云服务供应商承担而更容易实现。相较于要求 IT 团队学习如何托管并运行新型技术方案，现在 IT 团队能够更轻松以服务方式实

现服务使用。举例来说，NoSQL 数据库、媒体转码以及机器学习皆属于要求相当程度专业知识，甚至很难在技术业界之内获取。但在云环境当中，这些技术方案转化为可供团队直接消费的服务项目，意味着客户能够专注于产品开发——而非资源配置与管理。

- **数分钟内实现全球化：**只需要数次点击，即可轻松将您的系统部署至全球范围内的多个区域当中。这意味着您将可以投入极低成本为客户提供更低延迟水平与更佳使用体验。
- **使用无服务器架构：**在云环境当中，无服务器架构能够帮助您在实施传统计算活动的同时，摆脱运行并维护服务器的日常负担。举例来说，存储服务可充当静态网站，从而摆脱对 Web 服务器的依赖；事件服务则可为您实现代码托管。无服务器架构不仅消除了服务器管理带来的运营负担，同时亦能够凭借着云规模下的托管服务运营机制降低事务处理成本。
- **提升实验频率：**凭借着虚拟与可自动化资源，您将能够快速利用多种不同实例、存储或者配置类型完成各类综合性测试。
- **制度化选择：**切实使用能够与您欲达成之目标相匹配的技术方案。例如在选择数据库或者存储方案时，考虑具体数据访问模式。

定义

云环境下的性能效率分为四项最佳实践：

- 选择
- 审查
- 监控
- 权衡

采取一套数据驱动型方案以选择高性能架构选项。面向架构中的各个方面收集数据，从高级设计到资源类型的选择及配置皆在其中。通过定期审查您的选择，您将能够确保自身充分发挥 AWS 平台持续演进所带来的优势。监控将确保您可随时发现意料之外的性能问题并采取针对性措施。最后，您的架构亦应作出权衡以进一步改善性能水平，例如采用压缩、缓存或者放松一致性要求等举措。

最佳实践

选择

特定系统当中的最佳解决方案将根据您工作负载类型的不同而有所区别，且一般需要将多种方案加以结合。良好架构系统采用多种解决方案，且可实现多种功能以改善性能水平。

在 AWS 当中，资源以虚拟化形式存在，且通过多种不同类型及配置方案进行交付。这意味着客户能够更轻松地找到契合自身需求的方案，亦可建立起更多在本地环境下往往很难实现的选项。举例来说，Amazon DynamoDB 等托管服务能够提供一套全托管的 NoSQL 数据库，在任意运行规模之下都能保证毫秒级延迟。

以下问题侧重于选择层面的注意事项：

问题 1：您如何选择最佳性能的架构？

当您为自己的架构选定模式及实现方式，应采用数据驱动型方案以达成最优效。AWS 解决方案架构师、AWS 参考架构以及 AWS 合作伙伴皆可根据我们多年来积累起的丰富经验帮助您作出明智的架构选择，同时通过基准测试或者负载测试提取数据信息以优化这套架构。

您的架构将可能引入其它多种不同架构方案（例如事件驱动型、ETL 或者管道）。您的架构实现方案将使用各项专门针对架构性能作出优化的 AWS 服务。在以下章节当中，我们将着眼于您应当认真考量的四大主要资源类型（即计算、存储、数据库以及网络）。

计算

针对特定系统的优化计算解决方案往往取决于应用程序设计、使用模式以及配置设置等几大因素。架构可能利用多种不同计算解决方案以支持其中的不同组件，同时启用多种功能以实现性能提升。如果在架构当中选择错误的计算方案，则有可能导致性能效率低下。

在 AWS 当中，计算资源以三种方式进行交付：实例、容器与函数：

- **实例** 属于虚拟化服务器，因此您可以通过一键式方式或者 API 调用对其容量作出调整。由于云资源决策不再长期固定，因此您可以以实验性方式测试不同服务器类型。在 AWS 当中，这些虚拟服务器**实例**拥有不同的家族及大小，且可提供相当广泛的不同容量选项，例如固态驱动器（简称 SSD）以及图形处理单元（简称 GPU）。
- **容器** 属于一种操作系统虚拟化方法，其允许您立足资源隔离型流程运行应用程序及其依赖关系。
- **函数** 从您希望执行的代码当中抽象出执行环境。举例来说，AWS Lambda 允许您在无需运行实例的前提下实现代码执行。

以下问题侧重于计算方面的注意事项：

问题 2：您如何选择自己的计算解决方案？

在选择架构中的计算解决方案时，您应当充分发挥弹性机制优势，以确保您有能力根据业务需求不断调整与之相匹配的性能水平。

存储

针对特定系统的最优存储解决方案往往取决于访问方式类型（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。良好架构系统采用多种存储解决方案并启用多种功能以提升性能表现。

在 AWS 当中，存储资源经过虚拟化且可通过多种不同形式加以交付。这意味着您将能够更为轻松地将存储方法与实际需求结合起来，同时实现本地基础设施当中往往难以实现的存储选项。举例来说，Amazon S3 被设计为 11 个 9 持久性。您亦可选择将磁性硬盘驱动器（简称 HDD）替换为固态驱动器（简称 SSD），或者在数秒钟之内轻松完成不同实例间的虚拟驱动器迁移。

以下问题侧重于性能效率层面的存储注意事项：

问题 3：您如何选择自己的存储解决方案？

当您选择存储解决方案时，请确保其与您所需要的访问模式切实匹配——这一点对于达成性能预期非常关键。

数据库

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。多数系统采用多种不同数据库解决方案以适应其中各子系统的实际需要，同时启用各类不同功能以进一步提升性能水平。为系统选择错误的数据库解决方案及功能可能导致性能效率低下。

在 AWS 当中，Amazon 关系数据库服务（简称 RDS）提供一套全托管的关系数据库。利用 Amazon RDS，您能够对数据库的计算与存储资源进行任意规模伸缩，且通常不会造成服务中断。Amazon DynamoDB 是一套全托管 NoSQL 数据库，其可在任意规模之下提供个位数毫秒延迟水平。Amazon Redshift 是一套 PB 级别数据仓库，允许您根据性能或者容量需求的变化调整其节点数据或者类型。

以下问题侧重于性能效率层面的数据库注意事项：

问题 4：您如何选择自己的数据库解决方案？

尽管某一工作负载的配套数据库方案（例如 RDBMS、NoSQL 等等）会对具体性能效率产生巨大影响，但企业往往倾向于采用默认方案选项——而非使用数据驱动型方案。在存储方面，您务必根据自身工作负载的访问模式进行考量，同时思考其它非数据库类解决方案是否能够更为高效地解决问题（例如使用搜索引擎或者数据仓库）。

网络

针对特定系统的最优网络解决方案往往取决于延迟、数据吞吐量要求等因素。用户或者本地资源等物理限制条件往往决定着位置的选项，而您可利用边缘技术或者资源安置方案解决此类问题。

在 AWS 当中，网络资源以虚拟化形式存在，且可采用多种不同类型及配置选项。这意味着您能够更轻松地确保网络方案与自身需求相匹配。AWS 提供多种产品功能（例如高速网络实例类型，Amazon EBS 优化实例、Amazon S3 传输加速以及动态 Amazon CloudFront 等）以优化网络流量。AWS 亦提供多种网络功能（例如 Amazon Route53 基于延迟路由、Amazon VPC 端点以及 AWS Direct Connect）以减少网络距离或者卡顿现象。

以下问题侧重于性能效率层面的存储注意事项：

问题 5：您如何选择自己的网络解决方案？

当您选择自己的网络解决方案时，需要认真考虑位置这一关键性因素。利用 AWS，您可以选择将资源安置在距离您最近的位置，从而尽可能减少使用端与资源供给端之间的距离。通过发挥区域、安置组以及边缘位置机制的固有优势，您将能够显著提升性能表现。

审查

在构建解决方案时，您可以立足于一组有限选项作为起步模板。然而随着时间推移，新型技术与方案将持续出现，您可能需要将其引入系统以进一步提升架构的性能水平。

利用 AWS，您能够充分发挥我们持续创新工作带来的优势，且我们的创新努力始终以客户需求为核心。我们会定期推出新的区域、边缘位置、服务以及功能。这一切都将给您架构的性能效率带来积极改进。

以下问题侧重于性能效率层面的审查工作：

问题 6：面对持续推出的资源类型与功能，您如何确保能够始终采用与自身需求最为匹配的资源类型？

了解您架构当中的性能限制所在，将帮助您密切关注能够解决此类局限的资源类型或新型功能。

监控

一旦完成了架构构建，您将需要对其性能进行监控，以便抢在客户之前纠正一切可能影响使用体验的问题。您应采用指标监控机制，确保任一指标达到阈值时立即发送警报。该警报亦可自动触发对应操作，用以处理任何无法正常工作的系统组件。

利用 AWS，Amazon CloudWatch 允许您监控并发送警报通知，您亦可利用自动化方案通过 Amazon Kinesis、Amazon 简单队列服务（Amazon Simple Queue Service ，简称 SQS）以及 AWS Lambda 等协同应对性能问题。

以下问题侧重于性能效率的监控需求：

问题 7：您如何在启动之后对自身资源加以监控，从而确保其运行符合预期？

确保控制误报与被忽视数据比例是提升监控解决方案实际成效的关键所在。自动触发器能够避免人为错误，同时减少修复问题所需要的时间。另外，您可规划“竞赛日”活动，立足生产环境模拟并测试您的解决方案，借此确保其能够正确识别各类问题。

权衡

当您设计架构解决方案时，请认真考量权衡方面的工作，确保您能够从中选出最优方案。根据实际需求，您可以在一致性、持久性、空间、时间或者延迟等因素之间作出取舍，最终得到更理想的性能表现。

利用 AWS，您可以在数分钟之内完成全球化拓展，包括在世界范围内选定距您用户最近的多个区域以进行资源部署。您亦能够向信息存储体系（例如数据库）内动态添加只读副本，借以降低主数据库所承受的负载压力。AWS 还提供 Amazon ElastiCache 等缓存解决方案，其可提供一套内存内数据存储或缓存体系; Amazon CloudFront 则可将您的静态内容副本缓存在与最终用户更近的位置。

以下问题侧重于性能效率层面的空间与时间权衡考量：

问题 8：您如何通过权衡改善性能表现？

权衡可能提升您的架构复杂度，同时亦要求进行负载测试以确保获得可量化的切实收益。

关键性 AWS 服务

性能效率层面的关键性 AWS 服务为 Amazon CloudWatch，能够对您的资源与系统进行监控，同时面向整体性能与运营健康程度提供可见性能力。以下服务亦在性能效率领域发挥着重要作用：

选择：

计算： Auto Scaling 是确保您拥有充足实例以满足需求并维持响应能力的关键所在。

存储： Amazon EBS 提供广泛的存储方案选项（例如 SSD 和预配置每秒输入/输出操作，简称 PIOPS），允许您针对自身用例实现优化。Amazon S3 提供无服务器内容交付能力，而 Amazon S3 传输加速（Transfer Acceleration）能够实现长距离条件下快速、简便且安全的文件传输效果。

数据库： Amazon RDS 提供一系列数据库功能（例如配置 IOPS 与只读副本），允许您针对自身用例进行优化。Amazon DynamoDB 则可立足任意规模提供个位数毫秒延迟表现。

网络： Amazon Route 53 提供基于延迟的路由服务。Amazon VPC 端点与 Direct Connect 则可减少网络距离或者卡顿现象。

审查： AWS 官方网站上的 AWS 博客与最新消息频道专门为您提供最新功能与服务资讯。

监控： Amazon CloudWatch 提供指标、警报与通知机制，您可将其与现有监控解决方案加以结合。另外，您亦可配合 AWS Lambda 以根据指标触发对应操作。

权衡： Amazon ElastiCache、Amazon CloudFront 以及 AWS Snowball 等服务允许您进一步实现性能提升。Amazon RDS 当中的只读副本则允许您实现高强度工作负载的规模化扩展。

资源

请参阅以下资源，以了解与我们性能效率最佳实践相关的细节信息。

文档

- Amazon S3 性能优化
- Amazon EBS 分卷性能

白皮书

- 性能效率支柱

视频

- AWS re:Invent 2016: Scaling Up to Your First 10 Million Users (ARC201)
- Performance AWS re:Invent 2016: Deep Dive on Amazon EC2 Instances, Featuring Performance Optimization (CMP301)

成本优化支柱

成本优化支柱包括避免或者消除不必要的成本或者次优资源。

成本优化支柱提供了设计原则、最佳实践和问题的概要。您可以在成本优化支柱白皮书中参考更多实践指导。

设计原则

在云环境当中，有以下 5 项指导原则以实现成本优化：

- **采用消费模式：**仅根据您的业务消费要求（而非通过详细的预测）申请资源量，同时随时添加或减少资源需求，最终为您的实际消费量付费。举例来说，开发及测试环境在每周工作日内往往每天仅运行八小时。您可以在此类环境处于静止状态时清空其资源分配量，从而实现高达 75% 的潜在成本节约效果（将原本的 168 小时运行时长降低至 40 小时）。
- **衡量整体效率：**衡量业务产出与成本。以此更加理解增加产出并且减少成本。
- **不再承担数据中心运营支出：**AWS 为您承担起服务器机架安装、部署以及电力供应等任务，因此您可以将精力集中在自身客户以及业务项目身上，而不再需要为 IT 基础设施分神。
- **支出分析与归因：**云环境能够帮助您更轻松且更准确地发现系统使用情况与成本，这同时意味着我们能够更为透明地对 IT 成本进行归因。这不仅有助于实现投资回报率（简称 ROI）量化，同时亦可为系统拥有者提供良好的资源优化与成本削减机会。
- **使用托管服务以降低拥有成本：**在云环境当中，托管服务能够消除邮件发送或者数据库管理等任务给服务器维护工作带来的负担。另外，由于托管服务以云规模方式运行，因此亦能够带来更低的每事务或每服务成本开销。

定义

云环境下的成本优化领域共包含四项最佳实践：

- 成本效率资源
- 供需匹配
- 支出认知
- 随时间推移优化

与其它支柱一样，成本优化工作同样存在多种因素需要权衡。举例来说，您更强调上市速度还是成本控制？在某些情况下，速度优化最为重要——即更快将产品推向市场、更快发布新功能或者单纯满足进度表规划，而非单纯立足经济成本进行优化。有时候，设计决策可能较为匆忙而导致未遵照经验数据进行核查，这意味着设计者只能利用过度预估用量来“以防万一”，而无法真正根据时间基准测试制定出最具成本效益的部署方案。在这样的背景之下，大规模过度配置与未经优化的部署方案几乎必都会存在。在下面的内容当中，我们将探讨能够切实用于部署成本优化的各类技术成果与战略指导方针。

最佳实践

成本效益资源

运用适当的实例与资源装备您的系统是实现成本节约的核心所在。举例来说，在小型服务器上一项报告流程可能需要耗时五个小时，但一台大型服务器尽管使用成本翻倍，但却能够在一小时内完成任务。这两种方案能够为您带来一样的结果，而小型服务器由于时间投入更多，因此将带来更高实际成本。

一套拥有良好架构的系统应使用最具成本效益的资源，且借此实现显著而积极的经济影响。您亦有机会利用托管服务实现成本削减。举例来说，相较于维护自有服务器进行邮件服务交付，您可以使用云服务供应商提供的按条消息计费服务。

AWS 提供一系列灵活且具备成本效益的服务价格选项，旨在确保 Amazon EC2 实例能够真正满足您的具体需求。**按需实例**允许您以小时为单位支付计算容量成本，且不需要任何最低承诺使用量。**预留实例**（简称 RI）允许您为自身业务保留必要的资源量，且相较于按需实例可带来最高 75%的价格折扣。**竞价实例**则允许您以极为可观的折扣使用闲置 Amazon EC2 容量。竞价实例适用于那些

能够接受集群内各服务器节点以动态方式添加/移除的系统体系——例如高性能计算与大数据场景。

以下问题侧重于成本优化层面中的资源选择需求：

问题 1：您是否会在为解决方案选择 AWS 服务时考虑成本因素？

问题 2：您是否会调整资源规模以满足成本目标要求？

问题 3：您是否已经选择了适当的计费模式以满足您的成本要求？

通过使用 AWS Trusted Advisor 等工具对 AWS 资源使用量进行定期审查，您能够主动监控容量利用率并据此作出部署方案调整。

匹配供需

最优供需匹配应能够为特定系统提供最低成本，同时亦需要具备充足的额外供应以控制供应时间并应对个别资源故障。需求可为固定或是可变形式，且应利用指标及自动化机制确保管理工作不致产生过高成本。

良好架构的系统使用最优成本的资源，能够看到显著并且积极的经济影响。您可以使用托管服务以减少成本。例如，不使用邮件服务器，而是使用按需计费的邮件服务。

在 AWS 当中，您能够自动进行资源配置以匹配实际需求。Auto Scaling 以及多种其它基于需求、缓冲与时间的方案允许您根据需要随时添加及移除资源配额。如果您能够预测资源的需求变化情况，则可进一步节约资金并确保资源供应与系统需求相契合。

以下问题侧重于成本优化层面的供需匹配注意事项：

问题 4：您如何确保您的容量在匹配需求的同时，不致大幅超出具体需求量？

在设计供需匹配架构时，您需要以主动方式考量资源使用模式与新资源的配置时耗。

支出认知

云环境凭借着强大的灵活性与敏捷性，能够鼓励并显著提升您以更快节奏实现创新型开发与部署工作。其能够消除与基础设施相关的各类手动操作流程与本地部署配置带来的时间投入，具体包括硬件规格判断、报价谈判、采购订单管

理、出货计划以及资源部署等等。然而，这种使用极为便捷且几乎毫无限制的按需容量供应亦要求我们建立起新的支出认知方式与之相匹配。

大多数企业都由多支团队以及其下的多套系统共同构成。将资源成本归因至个别业务或者产品拥有者能够有效驱动资源使用活动，同时有助于降低资源浪费。精准的成本归因亦可帮助您理解产品的真实利润空间，同时以更为明智的方式作出预算划分决策。

以下问题侧重于成本优化层面的支出认知注意事项：

问题 5：在进行架构设计时，您是否考虑过数据传输费用？

问题 6：您如何监控资源使用量与支出情况？

问题 7：您是否会清理不再需要的资源，或者中止暂时不需要的资源？

问题 8：您采取了怎样的访问控制与规程以实现 AWS 资源使用治理？

您可以利用成本分配标签对您的 AWS 使用成本进行分类与追踪。在将标签应用于您的 AWS 资源时（例如 Amazon EC2 实例或者 Amazon S3 存储桶），AWS 会根据您的使用情况生成成本分摊报告，并给出各标签所对应的总体成本水平。您可以利用这些标签代表各项业务类别（例如成本中心、系统名称或者拥有者等），从而组织起您自己的跨服务成本核算体系。

将标记资源与各实体生命周期（包括员工、项目等）相结合即可发现那些无法继续为企业创造价值且应当被清退的孤立资源或项目。您可以设置结算通知以提醒您超出预算的支出，并利用 AWS 简单月度计费器（AWS Simple Monthly Calculator）计算您的数据传输成本。

随时间推移进行优化

随着 AWS 不断推出新型服务与功能，另一最佳实践在于审查您的现有架构决策以确保其仍然符合最佳成本效益。如果您的要求发生变化，则应主动清退那些您不再需要的资源、整体服务或者系统。

AWS 提供的托管服务通常能够显著实现解决方案优化，因此在上线之后，您应积极对其进行了解。举例来说，运行 Amazon RDS 数据库在成本方面要远优于立足 Amazon EC2 运行您自己的数据库。

以下问题侧重于成本优化层面的成本估算注意事项：

问题 9：您如何管理及/或考量新服务的采用？

通过定期审查您的部署方案，您通常能够利用新型 AWS 服务降低实际运行成本。另外，您亦有必要评估新型服务以帮助您节约资金开销。举例来说，AWS RDS for Aurora 能够帮助您降低关系数据库的实现成本。

关键性 AWS 服务

成本优化支持层面的关键性 AWS 功能为成本分配标签，其能够帮助您了解系统的具体实现成本。以下服务与功能在成本优化的四大领域当中亦起到重要作用：

- **成本效益资源：**您可以使用预留实例以及预付费容量降低业务运行成本。AWS Trusted Advisor 可用于检查您的 AWS 环境并找到潜在的成本节约空间。
- **供需匹配：**Auto Scaling 允许您添加或移除资源以匹配实际需求，同时保证不存在超支现象。
- **支出认知：**Amazon CloudWatch 警报与 Amazon 简单通知服务（简称 SNS）能够在发生超支或者预测可能发生超支时向您发出提醒。
- **随时间推移进行优化：**AWS 官方网站上的 AWS 博客与**最新消息**频道为您提供与各新型功能与服务相关的资讯。AWS Trusted Advisor 可检查您的 AWS 环境并发现一切可通过清除未使用或闲置资源、或采用预留实例容量实现的成本节约机会。

资源

参阅以下资源以了解更多与 AWS 成本优化最佳实践相关的细节信息。

文档

- 利用成本浏览分析成本
- AWS 云经济中心
- AWS 详细账单报告

白皮书

- 成本优化支柱

视频

- AWS 成本优化

工具

- AWS 总体拥有成本（简称 TCO）计算器
- AWS 简单月度计费器

审查流程

体系结构的审查需要以一致的方式完成，采用不追究的方法鼓励钻研。这应该是一个很快的过程（几小时而不是几天），这是对话而并非审计。审查架构的目的是找出关键问题，或者可以改进之处。审查之后采取一些措施，以改善客户体验。

正如“架构体系”部分所讨论的那样，需要每个团队成员对架构的质量负责。我们建议负责架构的团队，使用良好架构框架持续检查架构，而不是举行正式的审查会议。持续的方法使您的团队成员在架构演变时继续更新，并在您交付功能时改进架构。

AWS 良好架构与 AWS 在内部审查系统和服务的方式保持一致。它的前提是一套影响架构方法的设计原则，以及确保不会忽视经常产生问题的部分。只要内部系统、AWS 服务或客户存在重大问题，我们就会检查根本原因，看看是否可以改进审核流程。

应该从设计阶段开始，在工作负载的生命周期多次审查，以避免难以挽回的修改，并在上线之前进行评估。系统上线后仍然要随着新功能和技术变化而更新。架构也会随时间而变化。您需要遵循良好的实践，避免架构随着发展而退化。当架构有重大变更时，应该遵循包括良好架构审查在内的健康流程。

如果您想将审核当做一次性快照或独立审计，您需要确保与合适的人进行对话。通常我们会发现，审查才使团队第一次真正了解他们所做的事情。当审查另一个团队的工作负载时，一个好方法是对架构进行一些非正式交流，您可以收集大多数问题的答案。然后，您可以进行一两次会议，以获得准确答案，或者深入探讨未知领域。

以下是对于会议的一些建议：

- 带白板的会议室
- 打印任何图表或设计说明
- 需要继续研究答案的问题行动列表（例如，我们是否启用了加密？）

在您完成审核之后，您应该有一个问题清单，您可以根据您的业务设置优先级。您还需要考虑这些问题对您团队的日常工作的影响。如果您尽早解决这些问题，您有时间开展创造性的工作，而不是解决重复发生的问题。在解决问题时，您可以更新您的审查，以了解架构如何改进。

尽管在完成一项评估后，评估的价值显而易见，但您可能会发现新团队一开始可能会遇到阻力。以下是一些反对意见，可以通过对团队进行利益评估来处理：

- “我们太忙了！”（通常在团队准备重大发布时）
    - 如果准备好进行重大发布，您一定希望顺利进行。审查将让您了解您可能错过的任何问题。
    - 我们建议您在设计生命周期的早期进行审查，以发现风险，并制定与功能交付路线图一致的优化计划。
- “我们没有时间做任何有结果的事情！”（说这话的时候，通常对于他们来说，有些时间不可更改，比如超级杯）。
- 这些事件无法更改。你真的想在不知道架构中的风险的情况下开始吗？即使你没有解决所有这些问题，如果问题真的发生，你仍然可以有流程来处理它们。
- “我们不希望别人知道我们方案实施的秘密！”
- 如果您让团队看看架构框架白皮书附录中的问题，他们将会知道，没有任何显示任何商业或技术细节。

在您与组织团队进行多项评审时，您可能会讨论某个主题。例如，您可能会发现团队在特定支柱或主题中存在一系列问题。您将希望以全面的方式查看您的所有审查，并确定可帮助解决这些问题的任何机制、培训或工程师交流。

总结

AWS 良好架构框架涵盖五大支柱提供架构最佳实践，旨在立足云端设计并操作可靠、安全、高效且具备成本效益的系统体系。该框架提供一系列参考问题，允许您借此审查您的现有或者预期架构，同时亦针对各项支柱提供多项 AWS 最佳实践。在架构当中运用这套框架将帮助您建立起稳定且高效的系统，保证您能够将主要精力集中在功能需求身上。

贡献者

以下个人及组织对本文件的编撰作出贡献：

- Philip Fitzsimons: Sr. Manager Well-Architected, Amazon Web Services
- Rodney Lester: Reliability Lead Well-Architected, Amazon Web Services
- Brian Carlson: Operations Lead Well-Architected, Amazon Web Services
- Jon Steele: Sr. Technical Account Manager, Amazon Web Services
- Ryan King: Technical Program Manager, Amazon Web Services
- Ben Potter: Security Lead Well-Architected, Amazon Web Services
- Erin Rifkin: Senior Product Manager, Amazon Web Services
- Max Ramsay: Principal Security Solutions Architect, Amazon Web Services
- Scott Paddock: Security Solutions Architect, Amazon Web Services
- Callum Hughes: Solutions Architect, Amazon Web Services

扩展阅读

查看以下文档以获得更多信息:

- Operational Excellence Pillar
- Security Pillar
- Reliability Pillar
- Performance Efficiency
- Cost Optimization Pillar

文档历史

2017 年 11 月，卓越操作部分移至前面并进行重写，更新其他框架，更新其他支柱以反映 AWS 的演变。

2016 年 11 月，更新框架以加入卓越操作支柱，同时修订并更新其它支柱以删减重复内容并纳入来自数千家客户的学习与评估意见。

2015 年 11 月，在附录中更新当前 Amazon CloudWatch 日志信息。

2015 年 10 月，原始版本。

附录：良好架构相关问题、答案与最佳实践

卓越操作支柱

筹备

问题 1：什么因素推动您的业务重点？

企业的存在是为了满足客户的需求。运营要满足业务需求。业务优先级由业务和客户需求决定。诸如法规遵从要求或行业最佳实践等外部因素也可能影响运营优先级。在制定运营优先级时，基于利益和风险，作出决策。

最佳实践：

- **业务需求：**让业务和开发团队参与确定运营优先级。
- **合规要求：**监管或行业标准等外部因素可能会使您的业务承担特定要求，例如 SOX 法规遵从要求与 PCI 行业最佳实践。
- **风险管理：**平衡决策与潜在利益的风险。

问题 2：您如何设计工作负载以实现可操作性？

考虑运营需求是系统设计的一部分。设计工作负载，提供对其运行状态、客户行为和客户体验的洞察力，从而对问题作出响应以及确定需要改进的领域。实施低风险、零停机时间的部署方法，以便轻松识别故障和快速恢复。

最佳实践：

- **共享设计标准：**在团队之间分享现有最佳实践、指导和管理要求，并纳入系统设计。流程可以对设计标准进行更改、添加和异常处理。
- **针对云操作的设计：**包含云启用的功能，可在系统设计中提供优于物理资源的优势。
- **提供有关工作负载行为的深入见解：**在您的系统设计中设计组件，以便了解系统内正在发生的事情，并衡量各个组件的性能。

- **深入了解客户行为：**在您的系统设计中设计组件，以便了解客户如何使用系统以及他们的体验质量。
- **减少缺陷，简化补救措施并改善流程：**采用包括快速反馈质量的方法，实现重构和缺陷修复。
- **降低部署风险：**使用频繁、小型、可回溯、自动部署、测试、灰度部署、蓝绿部署等方法。

问题 3：如何确定准备好支持工作负载？

采用流程来验证运营就绪情况以支持工作量。足够和适当的技能人员已经到位，并且准备好支持工作负载。公开运营流程，经常审查，并根据情况进行更新。运营流程以代码形式出现，并在适当的地方自动执行程序。

最佳实践：

- **持续改进文化：**控制您的运营方式，并认识到变化是一直存在的。您需要继续尝试和发展，寻找机会并采取行动。
- **对企业价值的共识：**让团队了解企业工作负载的价值，并制定流程吸引更多团队寻求支持。
- **人员能力：**确保您拥有适当数量的经过培训的人员来支持您的工作负载需求。定期检查工作量需求，并根据需要培训或调整人员能力。
- **把指导和操作文档公开：**确保标准易于获取，易于理解，并且可以衡量合规性。建立标准流程，以及特殊流程。
- **使用清单：**使用清单来评估是否准备好运行工作负载。 这些包括操作准备和安全检查表。
- **运营手册：**具有充分理解的事件和程序的运行手册。
- **剧本：**为故障场景制作剧本。
- **恢复演习：**识别潜在的故障情况并测试您的响应（例如，竞赛日，模拟失败）。

运营

问题 4：什么因素加深您对健康运营的理解？

以业务成就定义成功，确定成功标准，确定工作负载和运营指标，以满足这些衡量标准，并创建成功运营的业务级视图。建立基准并进行主动评估以确定趋势并回应。使用跨职能团队验证工作负载和运营状况，并酌情调整回应。

最佳实践：

- **定义预期的业务和客户结果：**从业务和客户的角度对成功的工作负载进行定义。
- **确定成功指标：**定义根据业务和客户期望来衡量工作负载行为的指标。
- **识别工作负载指标：**定义根据成功标准来衡量工作负载及其组件状态的指标。
- **识别操作指标：**定义衡量运营活动（操作手册和剧本）执行情况的指标。
- **建立基线：**建立基准作为衡量标准，提供预期值作为比较的基础。
- **收集并分析您的指标：**定期进行主动评估，以确定趋势并确定回应。
- **验证洞察力：**与跨职能团队和企业所有者一起检查分析结果和回应。酌情调整回复。
- **业务级运营视图：**确定您是否满足客户需求，并可确定需要改进的地方以实现业务目标。

问题 5：你如何管理运营事件？

通过业务影响来指定运营优先级。为任何警报的事件定义流程，并进行响应。在适当的地方自动执行这些动作。对于运营状态的变化进行沟通，以便受影响的各方可以根据需要做出响应。找出计划外事件的根本原因，以及计划事件的意外影响，以便可以使用这些信息来减轻未来事件的发生可能。方便时与受影响方共享此信息。

最佳实践：

- **根据业务影响确定运营事件的优先级：**当多个事件需要干预时，基于业务影响制定优先级。
- **事件、意外和问题管理流程：**流程适用于处理观察到的事件，需要干预的事件（意外），以及需要干预并重现和/或目前无法解决的事件（问题）。基于警报的通知机制：在特定指标超出安全范围时接收来自监控系统的自动警报。

- **以警报进行处理：**对于任何警报事件，应该明确定义责任人（例如个人，团队或角色）的明确响应（操作手册或剧本）。
- **定义升级路径：**操作手册和剧本应该定义，事件升级，升级过程以及具体确定每个步骤的责任人。升级可能会包括第三方（例如供应商，AWS 支持）。
- **确定决策者：**当行为可能对业务有潜在影响时，确定决策者有权以组织的名义作出决策。
- **通过仪表板发布状态：**仪表板存在业务当前运行状态的信息，针对目标受众（例如，内部技术团队，领导力和客户）量身定制。示例包括 CloudWatch 仪表板，个人健康仪表板（PHD）和服务健康仪表板（SHD）。
- **推送通知：**当用户所使用的服务受到影响，并恢复时与用户及时沟通（例如，通过电子邮件或短信）。
- **根本原因分析流程：**确定并记录事件根本原因的流程。
- **对于问题原因进行沟通：**适当时，对于问题原因和影响，与特定客户进行沟通。

演变

问题 6：如何对管理进行演变？

专注于工作周期以实现持续的渐进式改进。评估从反馈、经验教训、分析和跨团队评论中获得的改进机会。验证机会并确定优先顺序。在团队中分享经验和益处。

最佳实践：

- **持续改进流程：**运营流程包括特定的工作周期，以实现持续的变化改进。对机会进行评估并优先采取行动。
- **改进动力：**考虑所需的特性、功能和改进、不可接受的问题、错误和漏洞；以及在评估改进机会时所需的更新，以保持遵守供应商提供的政策或支持。
- **反馈：**流程包括反馈以确定需要改进的地方。
- **经验教训：**有流程来获取并记录运营中经验教训，以便其他团队可以利用这些经验教训。
- **分享学习：**有适当的流程来分享团队中学到的经验教训。

- **分析经验教训：**有适当的流程来学习趋势，并找出需要调查以寻求改进机会的领域。
- **运营指标评估：**对跨业务参与者的运营指标进行回顾性分析，以确定改进的机会和方法。
- **进行更改：**实施修改以改进并评估结果，确保成功。

安全性支柱

身份与访问管理

问题 1：您如何保护与 AWS root 帐户凭证相关的访问与使用活动？

AWS root 帐户凭证类似于其它操作系统当中的 root 或者本地管理员机制，因此应尽量少使用。目前的最佳实践在于创建 AWS 身份与访问管理（简称 IAM）用户，将其关联至管理员组，并利用 IAM 用户管理该帐户。该 AWS root 帐户不应具有 API 密钥，应拥有一条高强度密码且应关联硬件多因素验证（简称 MFA）设备。这意味着使用者必须利用惟一 root 身份通过 AWS 管理控制台进行访问，且该 root 帐户无法通过应用程序编程接口（简称 API）进行调用。需要注意的是，部分经销商或者区域并未发布或者支持 AWS root 帐户凭证。

最佳实践：

- **MFA 与最低 Root 使用原则：** AWS root 帐户凭证应仅用于最低必要活动。
- **不使用 Root**

问题 2： 您如何为系统用户定义角色与责任，并借此控制 AWS 管理控制台与 API 的人为访问？

目前的最佳实践在于创建用户组，从而面向客户对各系统用户的角色与责任进行拆分及定义。用户组可通过多种不同技术方案进行定义，具体包括身份与访问管理（简称 IAM）组、跨帐户访问 IAM 角色、网络身份、通过安全断言标记语言（简称 SAML）集成（例如在 Active Directory 当中进行定义）或者采用通常经由 SAML 或者 AWS 安全令牌服务（AWS Security Token Service ，简称 STS）实现的第三方解决方案（例如 Okta、Ping Identity 或者其它定制化技术方案）。我们明确反对直接使用共享帐户。

最佳实践：

- **受控员工生命周期：** 经过严格定义与执行的员工生命周期政策。
- **最低权限原则：**用户、组以及角色经过明确定义且仅被分配完成业务要求所必需的最低权限。

问题 3：您如何限制指向 AWS 资源的自动访问？（例如应用程序、脚本以及/或者第三方工具或服务）

应当以类似的方式定义系统访问行为，因为用户组手工创建。对于 Amazon EC2 实例，此类组被称为 IAM EC2 角色，且目前的最佳实践在于使用 EC2 专用 IAM 角色以及 AWS SDK 或 CLI，内置支持对 EC2 对应 IAM 角色凭证进行检索。立足于传统作法，用户凭证往往会被注入至 EC2 实例当中；但我们强烈建议您不要将凭证以硬编码方式注入脚本及源代码当中。

最佳实践：

- **用于自动化访问的静态凭证：**以安全方式加以存储。
- **用于自动化访问的动态凭证：**利用实例配置文件或者 Amazon STS 加以管理。

检测控制

问题 4：您如何对日志进行获取与分析？

获取日志记录对于从性能到安全事件在内的各类调查工作而言至关重要。目前的日志记录最佳实践在于定期将相关信息由来源直接移动至日志处理系统当中（例如 CloudWatch Logs、Splunk 以及 Papertrail 等等）或者将其存储于 Amazon S3 存储桶内以待后续根据需求加以处理。常见的日志来源包括 AWS API 以及用户相关日志（例如 AWS CloudTrail）、AWS 特定服务日志（例如 Amazon S3 及 Amazon CloudFront 等）、操作系统生成日志以及第三方特定应用日志。您可以利用 Amazon CloudWatch Logs 对来自 Amazon EC2 实例、AWS CloudTrail 或者其它来源的日志文件进行监控、存储与访问。

最佳实践：

- **适合的主动监控：**Amazon CloudWatch 日志事件、VPC 流日志、ELB 日志以及 S3 存储桶日志等等。
- **启用 AWS Cloud Trail**。
- **监控操作系统或者应用程序日志**。

基础设施保护

问题 5：您如何确保对网络及主机级边界进行保护？

在本地数据中心当中，DMZ 方案能够利用防火墙将系统拆分为多个受信与非受信区。在 AWS 当中，客户可使用有状态与无状态两类防火墙。有状态防火墙被称为安全组，而无状态防火墙则被称为网络访问控制列表（简称 ACL），负责保护 Amazon 虚拟私有云（简称 VPC）之内的各子网。目前的最佳实践在于立足 VPC 内运行一套系统，并在安全组内定义基于角色的安全机制（例如网络层、应用层等），而后在网络 ACL 之内定义基于位置的安全机制（例如在每个可用区内的一套子网中定义弹性负载均衡层、并在各可用区内的另一子网中定义网络层等）。

最佳实践：

- **VPC 中的受控网络流量：** 例如使用防火墙、安全组、NACLS 以及堡垒主机等。
- **边界位置上的受控网络流量：** 例如使用 AWS WAF、基于主机的防火墙、安全组以及 NACLS 等等。

问题 6：您如何运用 AWS 服务级安全功能？

AWS 服务提供多种额外安全功能（例如 Amazon S3 存储桶政策、Amazon SQS、Amazon DynamoDB 以及 KMS 密钥政策等等）。

最佳实践：

- 采用适合的其它各项功能。

问题 7：您如何立足 Amazon EC2 实例保护操作系统完整性？

另一项传统控制要求在于保护操作系统完整性。您能够利用基于主机的传统技术方案（例如 OSSEC、Tripwire 以及 Trend Micro Deep Security 等等）在 Amazon EC2 实例当中轻松实现这一点。

最佳实践：

- **文件完整性：**在 EC2 实例当中使用文件完整性控制机制。
- **EC2 入侵检测：**在 EC2 实例当中使用基于主机的入侵检测控制机制。

- **AWS Marketplace 或合作伙伴解决方案：**采用来自 AWS Marketplace 或者 APN 合作伙伴的解决方案。
- **配置管理工具：**采用定制化 Amazon Machine Image（简称 AMI）或者默认受到安全保护的配置管理工具（例如 Puppet 或 Chef）。

数据保护

问题 8：您如何对数据进行分级？

数据分级机制能够根据敏感程度对数据进行分类组织，其中包括可用数据类型、数据所在位置、访问级别以及数据保护机制（例如通过加密或者访问控制实现）。

最佳实践：

- **采用数据分级模式。**
- **全部数据皆被视为敏感类型。**

问题 9：您如何对处于静态的数据进行加密与保护？

传统的安全控制机制强调对静态数据进行加密。AWS 利用客户端（例如 SDK 支持的、受支持操作系统、Windows Bitlocker、dm-crypt 以及 Trend Micro SafeNet 等）与服务器端（例如 Amazon S3）双管齐下的方式支持这种控制能力。您亦可以采用服务器端加密（简称 SSE）与 Amaozn 弹性块存储服务中的加密卷。

最佳实践：

- **不要求：**不要求对静态数据进行加密。
- **对静态数据进行加密。**

问题 10：您如何管理密钥？

密钥属于应得到严格保护的机密信息，同时亦应配合适当的轮换政策以实现定义与使用。最佳实践在于绝对不可以硬编码方式将这些机密信息添加至管理脚本及应用程序当中——但此类作法确实时有发生。

最佳实践：

- **AWS CloudHSM**：采用 AWS CloudHSM。
- **采用 AWS 服务控制**：静态数据可利用 AWS 特定服务控制（例如 Amazon S3 SSE、Amazon EBS 加密分卷、Amazon 关系数据库服务（简称 RDS）以及传输数据加密（简称 TDE）等）实现加密。
- **采用客户端加密**：静态数据可利用客户端技术方案进行加密。
- **AWS Marketplace 或合作伙伴解决方案**：使用来自 AWS Marketplace 或者 APN 合作伙伴的解决方案。（例如 Trend Micro SafeNet 等）。

问题 11：您如何对传输的数据进行加密与保护？

最佳实践在于利用加密机制对传输的数据进行保护。AWS 支持面向各服务 API 的加密端点方案。另外，客户亦可在其 Amazon EC2 实例之内使用多种技术方案。

最佳实践：

- **不要求**：不要求对传输的数据进行加密。
- **加密通信**：利用 TLS 或者其它同类方案对通信内容加以恰当保护。

问题 12：您如何确保采用适当的事件响应方案？

在安全事件发生之前，确保将相关工具部署到位，而后定期进行事件响应演练以确保架构始终保持更新并有能力及时实现调查与恢复。

最佳实践：

- **预配置访问**：信息安全人员应有权或者有能力快速实现访问。这种能力应以预配置方式实现，从而保证可在发生事件时作出适当响应。
- **预部署工具**：信息安全人员应在 AWS 内预部署正确工具，用以确保在事件发生时可进行适当响应。
- **非生产竞赛日活动**：应定期在非生产环境开展事件响应模拟活动，同时将由此获取的经验与教训纳入架构及操作体系当中。
- **生产竞赛日活动**：应定期在生产环境开展事件响应模拟活动，同时将由此获取的经验与教训纳入架构及操作体系当中。

可靠性支柱

基础

问题 1：您如何为自己的帐户管理 AWS 服务限制？

AWS 帐户默认配置有服务上限，旨在防止新用户遭遇意料之外的资源配置超量。AWS 客户应评估自身对于 AWS 服务的实际需求，并提交申请以适当变更各区域内的可用资源上限。

最佳实践：

- **限制监控与管理**：评估您对于 AWS 资源的潜在使用量，合理上调所在区域资源上限，同时允许根据实际使用情况进行计划内增长。
- **设置自动化监控机制**：利用 SDK 等工具在触及指标阈值时发出警报。
- **了解固定服务限制**： 了解不可变更的服务限制，并在架构设计中考虑到相关因素。
- **确保您的服务限制与最大资源使用量之间始终存在充足空间，借以适应故障转移需求**。
- **应跨越全部相关帐户与区域考量服务限制因素**。

问题 2：您如何在 AWS 上规划自己的网络拓扑结构？

应用程序可存在于一套或者多套环境当中：EC2 Classic、VPC 或者默认 VPC 等等。网络考量则包括系统连接性、弹性 IP/公共 IP 地址管理、VPC/私有地址管理以及作为云环境下资源利用根本性前提的名称解析等。经过良好规划及记录的部署方案能够切实降低重叠和争用等风险。

最佳实践：

- **不需要返回数据中心的连接**。
- **AWS 与本地环境间的高可用性连接（如果必要）**： 可根据需求使用多 DX 线缆、多 VPN 隧道以及 AWS Marketplace 方案。
- **面向工作负载用户的高可用性网络连接**： 高可用性负载均衡与/或代理、基于 DNS 的解决方案、AWS Marketplace 方案等。
- **非重叠私有 IP 地址范围**： 虚拟私有云内 IP 地址范围与子网的使用不应存在彼此重叠、与其它云环境重叠或者与本地环境相重叠。
- **IP 子网分配**：各 Amazon VPC IP 地址范围应足以适应应用程序的实际需要，包括未来的扩展规模以及跨可用区间各子网的 IP 地址分配需求。

变更管理

问题 3：您的系统如何适应需求变化？

一套可扩展系统能够提供必要弹性以自动添加及移除各类资源，从而确切满足任意给定时间点之内的当前需求。

最佳实践：

- **自动化规模伸缩**：使用各类自动化规模伸缩服务，例如 Amazon S3、Amazon CloudFront、Auto Scaling、Amazon DynamoDB 以及 AWS Elastic Beanstalk 等等。
- **负载测试**：利用负载测试方法衡量规模伸缩活动是否满足应用程序的实际需要。

问题 4：您如何监控各类 AWS 资源？

日志与指标是了解您应用程序运行状态的强大工具。您可以通过系统配置监控日志与指标，并在触及阈值或者发生重大事件时发出通知。

在理想条件下，一旦低性能阈值被触及或者发生故障，系统应拥有必要架构设计以自动实现自我修复或规模伸缩响应。

最佳实践：

- **监控**：利用 Amazon CloudWatch 或者其它第三方工具对您的应用程序进行监控。
- **通知** ：在计划中考虑到发生重大事件时接收通知的需求。
- **自动响应**：利用自动化方案在检测到故障时执行对应操作，例如替换故障组件。

问题 5：您如何执行变更？

您环境当中的非受控变更将使变更影响变得难以预测。要确保各应用程序与操作环境始终运行已知软件并以可预测方式安装补丁或进行组件替换，我们必须保证利用受控变更进行 AWS 资源与应用程序配置。

最佳实践：

- **自动化：**以自动化方式执行部署与补丁安装。

故障管理

问题 6：您如何对数据进行备份？

对您的数据、应用程序以及操作环境（所谓操作环境，是指纳入应用程序且经过配置的操作系统）进行备份，从而满足平均恢复时间（简称 MTTR）与恢复点目标（简称 RPO）的要求。

最佳实践：

- **自动备份：**使用 AWS 功能、AWS Marketplace 解决方案或者其它第三方软件以实现自动化备份。
- **定期恢复测试：**通过恢复测试验证备份流程是否符合 RTO 与 RPO 要求。

问题 7：您的系统如何承受组件故障？

您的应用程序是否存在（隐性或明确的）可用性及平均恢复时间（简称 MTTR）要求？如果是，您应建立起应用弹性机制并通过资源分配确保其能够承受中断问题。为了进一步提升可用性水平，资源分配机制应跨越多个不同物理位置。在架构当中纳入多个独立层（例如 Web 服务器、数据库等）以实现弹性，其中包含监控、自我修复、重大事件通知以及故障应对等因素。

最佳实践：

- **多可用区/区域：**跨越多个可用区/区域分发应用程序负载（例如 DNS、ELB、应用程序负载均衡器以及 API 网关等）。
- **松耦合的依赖关系：** 例如使用队列系统、流式系统、工作流以及负载均衡器等。
- **优雅降级机制：**当组件间的依赖关系存在问题，并不代表组件本身的运行状态不佳。应确保组件能够在降级模式下继续响应请求。
- **自动修复：**利用自动化方式检测故障并执行修复操作。持续监控系统的运行状态并计划接收一切关于重大事件的通知信息。

问题 8：您如何对弹性进行测试？

当您对弹性进行测试时，可能会发现某些潜在 bug 只存在于生产环境中。通过竞赛日活动定期进行模拟，从而帮助企业顺利贯彻并执行既定规程。

最佳实践：

- **响应剧本**：面向故障场景建立响应剧本。
- **故障注入**：定期进行故障测试（例如使用 Chaos Monkey）以确保覆盖故障情况。
- **规划竞赛日活动**。
- **根本原因分析(简称 RCA)**：根据重大事件执行系统故障审查并进行架构评估。

问题 9：您如何规划灾难恢复机制？

灾难恢复（简称 DR）机制在立足备份方案实现数据恢复这类任务当中发挥着重要作用。您应当对执行工作的目标、资源、位置以及功能进行定义与执行，且确保相关举措同数据的 RTO 及 RPO 目标相吻合。

最佳实践：

- **定义目标**：定义 RTO 与 RPO。
- **灾难恢复**：建立一套灾难恢复策略。
- **配置草案**：确保灾难恢复站点/区域内的 Amazon Machine Image（简称 AMI）与系统配置状态处于最新。
- **灾难恢复测试与验证**：定期测试灾难恢复策略的故障转移能力，确保其能够满足 RTO 与 RPO 目标。
- **实施自动化恢复**：利用 AWS 与/或第三方工具以实现自动系统恢复。

性能效率支柱

选择

问题 1：您如何选择最佳性能架构？

特定系统的最佳解决方案根据工作负载的具体类型而有所区别，且通常需要将多种方案加以结合。良好架构系统采用多种解决方案并通过不同功能共同提升性能水平。

最佳实践：

- **基准测试：** 立足 AWS 对已知工作负载进行测试，并借此确定最佳选项。
- **负载测试：**利用多种不同资源类型与规模在 AWS 之上部署您系统的最新版本，同时利用监控机制收集性能指标，而后根据性价比计算结果得出最终选择。

问题 2：您如何选择计算解决方案？

特定系统的最优计算解决方案取决于应用程序设计、使用量模式以及配置设置等多种因素。各类架构方案可能采用多种不同计算解决方案以满足各组件的实际要求，同时启用各项功能以提升性能水平。在架构当中选用错误的计算解决方案可能导致性能效率低下。

最佳实践：

- **选项考量：**考虑各实例、容器以及函数所使用的具体选项以实现最佳性能表现。
- **实例配置选项：**如果您使用实例，请考虑家族、实例大小以及特性（GPU、I/O、可突发）等配置选项。
- **容器配置选项：**如果您使用容器，请考虑内存、CPU 以及租户配置等容器配置选项。
- **函数配置选项：**如果您使用函数，则应考虑内存、运行时及状态等配置选项。
- **弹性：**利用弹性（例如 Auto Scaling、Amazon EC2 容器服务（简称 ECS）以及 AWS Lambda 等）功能以满足变更需求。

问题 3：您如何选择自己的存储解决方案？

特定系统的最优存储解决方案取决于访问方法（块、文件或者对象）、访问模式（随机或者连续）、数据吞吐量要求、访问频率（在线、离线及归档）、更新频率（WORM 与动态）以及可用性与持久性限制等因素。良好架构系统利用多种存储解决方案并启用不同功能以提升性能表现。

最佳实践：

- **特性考量**：考虑您所需要的不同特性（例如共享性、文件大小、缓存大小、访问模式、延迟、数据吞吐量、数据持久性等）以选择您需要使用的具体服务（包括 Amazon S3、Amazon EBS、Amazon 弹性文件系统（简称 EFS）以及 EC2 实例存储等）。
- **考虑配置选项**：考虑 PIOPS、SSD、磁盘以及 Amazon S3 传输加速等配置选项。
- **考虑访问模式**： 根据访问模式优化您的存储系统使用方式（例如条带化、密钥分发与分区等）。

问题 4：您如何选择自己的数据库解决方案？

面向特定系统的最优数据库解决方案取决于可用性、一致性、分区容限、延迟、持久性、可扩展性以及查询容量等因素。多数系统采用不同的数据库解决方案以适合各子系统的需求，同时启用多种功能以提升性能水平。为系统选择错误的数据库解决方案与功能可能导致性能效率低下。

最佳实践：

- **考虑特性**：考虑各项不同特性（例如可用性、一致性、分区容限、延迟、持久性、可扩展性以及查询容量等）以选择最符合实际需求的数据库方案（关系、NoSQL、仓库、内存内）。
- **考虑配置选项**：考虑各类配置选项，包括存储优化、数据库级设置、内存以及缓存等。
- **考虑访问模式**：根据实际访问模式优化您的数据库系统使用方式（例如索引、密钥分发、分区以及横向扩展等）。
- **考虑其它方案**：考虑利用搜索索引、数据仓库以及大数据等其它方案实现数据可查询性。

问题 5：您如何配置自己的网络解决方案？

面向特定系统的最优网络解决方案取决于延迟、数据吞吐量要求等因素。用户或者本地资源等物理限制将决定位置的选择，且可利用边缘技术或资源安置等方式解决。

最佳实践：

- **位置考量：**为降低网络延迟考虑位置选项（例如区域、可用区、安置组、边缘位置）。
- **考虑产品功能：**考虑利用各产品功能（例如 EC2 实例网络容量、多种网络实例类型、Amazon EBS 优化实例、Amazon S3 传输加速、动态 Amazon CloudFront 等）优化您的网络流量。
- **考虑网络功能：**考虑利用各项网络功能（例如 Amazon Route 53 基于延迟路由、Amazon VPC 端点、AWS Direct Connect）减少网络距离或者卡顿现象。
- **适当的 NACLS：**使用 NACLS 最小设置以维持良好的网络数据吞吐能力。
- **考虑加密卸载：**考虑利用负载均衡机制卸载加密终端（TLS）。
- **考虑协议：**考虑优化网络性能您需要哪些协议。

审查

问题 6：面对持续推出的资源类型与功能，您如何确保能够始终采用与自身需求最为匹配的资源类型？

在设计架构解决方案时，您面对着相对有限的实际方案选项。然而随着时间推移，将出现更多能够提升您架构性能的新型技术与方法。

最佳实践：

- **审查：** 制定流程以审查新的资源类型与规模。反复运行性能测试以评估性能效率层面的一切改进机会。

监控

问题 7：您如何在布发后监控各项资源以确保其表现符合预期？

系统性能可能随着时间推移受到内部及/或外部因素的影响而有所降低。监控系统性能将帮助您发现这类降低问题并修复相关的内部或外部因素（例如操作系统或应用程序负载）。

最佳实践：

- **监控：** 利用 Amazon CloudWatch、第三方或者定制化监控工具监控性能表现。

- **基于警报的通知机制**：如果指标超出安全范围，则从您的监控系统处接收到自动警报。
- **基于触发条件的操作**：设置由警报触发的自动操作，用以恢复或者升级问题。

权衡

问题 8：您如何利用权衡机制提升性能表现？

在设计架构解决方案时，主动考虑权衡机制将帮助您找到最优方案。您通常可以在一致性、持久性以及空间/时间与延迟之间进行权衡，从而实现更佳性能表现。

最佳实践：

- **服务考量**：使用能够提升性能的服务，例如 Amazon ElastiCache、Amazon CloudFront 以及 AWS Snowball 等。
- **模式考量**：使用能够提供性能表现的各类模式，例如缓存、只读副本、分片、压缩以及缓冲等。

成本优化支柱

成本效益资源

问题 1：您在为解决方案选择 AWS 服务时，是否考虑到了成本因素？

Amazon EC2、Amazon EBS 以及 Amazon S3 等属于“基础构建块”型 AWS 服务。Amazon RDS 与 Amazon DyanmoDB 等托管服务则属于“高级”AWS 服务。通过选择基础构建块与托管服务，您可以对自己的架构进行成本优化。举例来说，利用托管服务，您可以降低或者彻底摆脱管理及操作类日常开销，从而将精力投入在应用程序及业务相关工作上。

最佳实践：

- **选择服务以降低成本**：分析服务以了解您能够通过哪些服务实现成本节约。
- **许可成本优化**。

- **利用无服务器与基于容器类方案实现成本优化：**利用 AWS Lambda、Amazon S3 网站、Amazon DynamoDB 以及 Amazon ECS 等服务降低成本。
- **利用适当存储解决方案实现成本优化：** 根据使用模式使用最具成本效益的存储解决方案（例如 Amazon EBS 冷存储、Amazon S3 标准-低频访问、Amazon Glacier 等等）。
- **利用适当数据库实现成本优化：**根据需求使用 Amazon 关系数据库服务（简称 RDS）（Postgres、MySQL、SQL Server 以及 Oracle Server）或者 Amazon DynamoDB（或者其它键-值存储、NoSQL 替代方案）。
- **利用其它应用程序级服务实现成本优化：** 根据实际需求使用 Amazon 简单队列服务（简称 SQS）、Amazon 简单通知服务（简称 SNS）以及 Amazon 简单邮件服务（简称 SES）。

问题 2：您是否对资源进行规模调整以契合成本目标？

确保您为当前任务选择适当的 AWS 资源规模。AWS 鼓励您使用基准评估机制以确保您选择的资源类型符合工作负载的最优需求。

最佳实践：

- **指标驱动型资源规模：** 利用性能指标选择正确的资源规模/类型以实现成本优化。合理配置数据吞吐量、规模以及具体存储服务，例如 Amazon EC2、Amazon DynamoDB、Amazon EBS（预配置 IOPS）、Amazon RDS、Amazon EMR 以及网络等。

问题 3：您是否选择了适当的计费模式以契合成本目标？

使用最适合您工作负载的计费模式以降低使用成本。最优部署方案可能为全按需实例、按需与预留实例相结合，或者在适用时选择竞价实例。

最佳实践：

- **预留容量与提交申请：** 定期分析使用情况并据此购买预留实例（例如 Amazon EC2、Amazon DynamoDB、Amazon S3 以及 Amazon ClouFront 等等）。
- **竞价实例：**为选定工作负载（例如批处理、EMR 等）使用竞价实例（例如竞价时段、竞价队列）。
- **区域成本考量：**在区域选择中纳入成本考量。

供需匹配

问题 4：您如何确保自己的容量匹配但又不会过度超出需求？

在对架构进行支出与性能平衡的过程中，确保您付费的一切资源皆得到使用，从而避免实例未得到充分利用的问题。任何不够准确的利用率量化指标都会对您的业务运营成本产生影响（过度使用造成性能下降）或浪费 AWS 支出（供过于求）。

最佳实践：

- **基于需求的方案**：使用 Auto Scaling 以响应需求变化。
- **基于缓冲区的方案**：利用缓冲区（例如使用 Amazon Kinesis 或者 Amazon 简单队列服务（简称 SQS））推迟工作，直到您腾出充足的处理容量。
- **基于时间的方案**：基于时间类方案包括工作时间外任务分配、在周末期间关闭开发与测试实例以及遵循季度或者年度规划（例如黑色星期五）等。

支出认知

问题 5：您是否在架构设计当中考虑到数据传输成本？

确保对数据传输成本进行监控，以便在架构决策时尽可能降低其中部分成本。举例来说，如果您身为内容供应商，以往直接 Amazon S3 存储桶向最终用户交付内容，您可以将您的内容推送至 Amazon CloudFront 的内容交付网络（简称 CDN），从而显著降低使用成本。请记住，一项小型但有效的架构变更能够显著降低运营支出。

最佳实践：

- **优化**： 在架构当中对数据传输进行优化（应用程序设计、WAN 加速、多可用区以及区域选择等）。
- **CDN**： 在适当情况下使用 CDN。
- **AWS Direct Connect**：分析实际情况并在适当时使用 AWS Direct Connect。

问题 6：您如何监控使用情况与支出？

建立监控与控制政策与规程，同时合理分配成本预算。利用 AWS 提供的可视化工具了解谁在使用哪些资源以及与之对应的成本水平。这将帮助您更为深入地理解您的业务需求以及团队运营情况。

最佳实践：

- **标记一切资源**：标记一切可标记的资源，借以将其同您帐单中的变更进行关联，并最终反映基础设施与资源使用量中的变化。
- **利用计费与成本管理工具**：建立一套标准化流程以加载并解释详尽的结算报告或成本管理器。利用 Amazon CloudWatch 或者第三方工具（例如 Cloudability、CloudCheckr 以及 CloudHealth 等）定期监控资源使用量与支出情况。
- **通知**：让团队当中的关键成员了解您的支出是否已经超出了明确定义的范畴。
- **财务驱动型退费/显示退费方法**： 借此向成本中心分配实例与资源（例如使用标记机制）。

问题 7：您是否会清理不再需要的资源，或者中止暂时不需要的资源？

从项目启动到生命周期结束，始终坚持采用变更控制与资源管理方案，从而确保您有能力发现必要的流程变更或者功能增强空间。您亦可通过 AWS 技术支持服务获取针对工作负载实现项目优化的详尽建议，例如何时使用 Auto Scaling、AWS OpsWorks、AWS Data Pipeline 或其它不同 Amazon EC2 配置方案，或者参阅 Trusted Advisor 给出的成本优化建议。

最佳实践：

- **自动化**：精心设计您的系统，确保其能够在您发现并停用非关键性或者不再需要的低利用率资源时实现符合预期的处理效果。
- **定义流程**：制定一套流程以发现并停用不再需要的资源。

问题 8：您采用哪些访问控制与规程以治理 AWS 资源使用？

制定相关政策与机制，确保您能够在达成目标的同时拥有合理的成本水平。通过将检查与平衡方案贯彻至标记与 IAM 控制体系，您能够在不受过度支出影响的前提下顺利实现创新。

最佳实践：

- **建立组与角色：** （例如：开发/测试/生产）利用治理机制控制各分组内谁能够启动实例及资源。（可利用 AWS 服务或者第三方解决方案实现。）
- **追踪项目生命周期：**追踪、衡量并审计项目、团队及环境的生命周期，借以避免使用并支付不必要的资源。

随时间推移进行优化

问题 9：您如何管理及/或考量对新服务的采用？

随着 AWS 不断发布各类新型服务与功能，最佳实践要求您对现有架构决策进行审查以确保其仍然拥有最佳成本效益。

最佳实践：

- 建立成本优化功能。
- **审查：** 制定流程以审查各项新服务、资源类型以及具体规模。反复进行性能测试以评估潜在成本节约机会。