

# AWS 云采用框架

运营论点

2016 年 2 月



© 2015 年，Amazon Web Services 有限公司或其附属公司版权所有。

## 通告

本文件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件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	4
简介	4
服务管理	8
注意事项	10
<b>SLA/OLA 策略</b>	<b>11</b>
注意事项	12
业务连续性规划	13
注意事项	14
事故与问题管理	15
注意事项	16
变更与配置管理	17
注意事项	19
性能与运营状态	20
注意事项	21
<b>CAF 分类与术语</b>	<b>22</b>
总结	23
备注	23

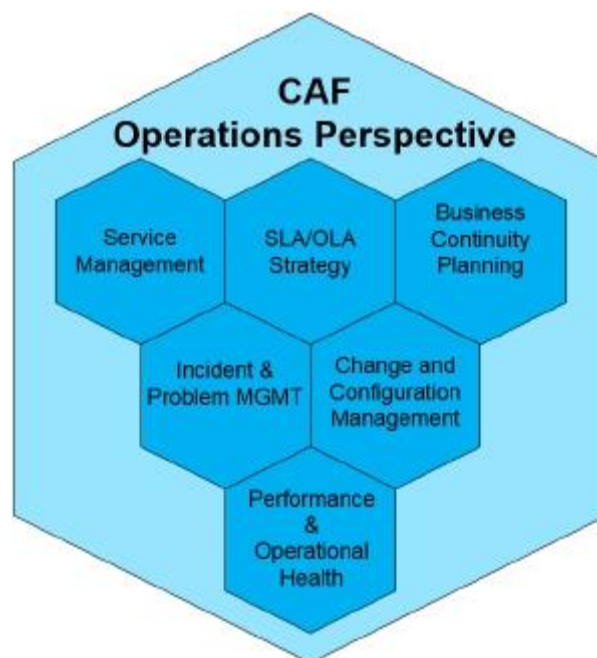
## 摘要

Amazon Web Services（简称 AWS）云采用框架（简称 CAF）<sup>1</sup> 提供各项最佳实践与规范性指导，帮助企业客户加快迈向云计算的转型步伐。本份 CAF 指南分为多个着眼领域，且全部与基于云的 IT 系统实现工作密切相关。这些重点领域被称为“论点”。每项论点由一份单独的白皮书负责论述。本份白皮书论述的“观点”方向为运营论点，其专注于如何更为高效地运营 AWS IT 环境。

## 简介

云采用业务目标通常包含众多关键词与短语，例如敏捷性、上市时间改进以及成本透明度，分别对应传统 IT 运营模式当中的低效率因素。在这些目标当中寻求平衡点并配合以适合的集中控制与管理，即可保证“业务”与 IT 运营之间的良好协同。在大多数企业当中，解决此项挑战要求运营团队采纳由商业合作伙伴提供的建议。相关建议通常包括创建 DevOps 模式甚至是 DevSecOps 模型，同时要求大家克服由组织变革、整体式结构乃至进展缓慢的传统融资模式带来的实施挑战。

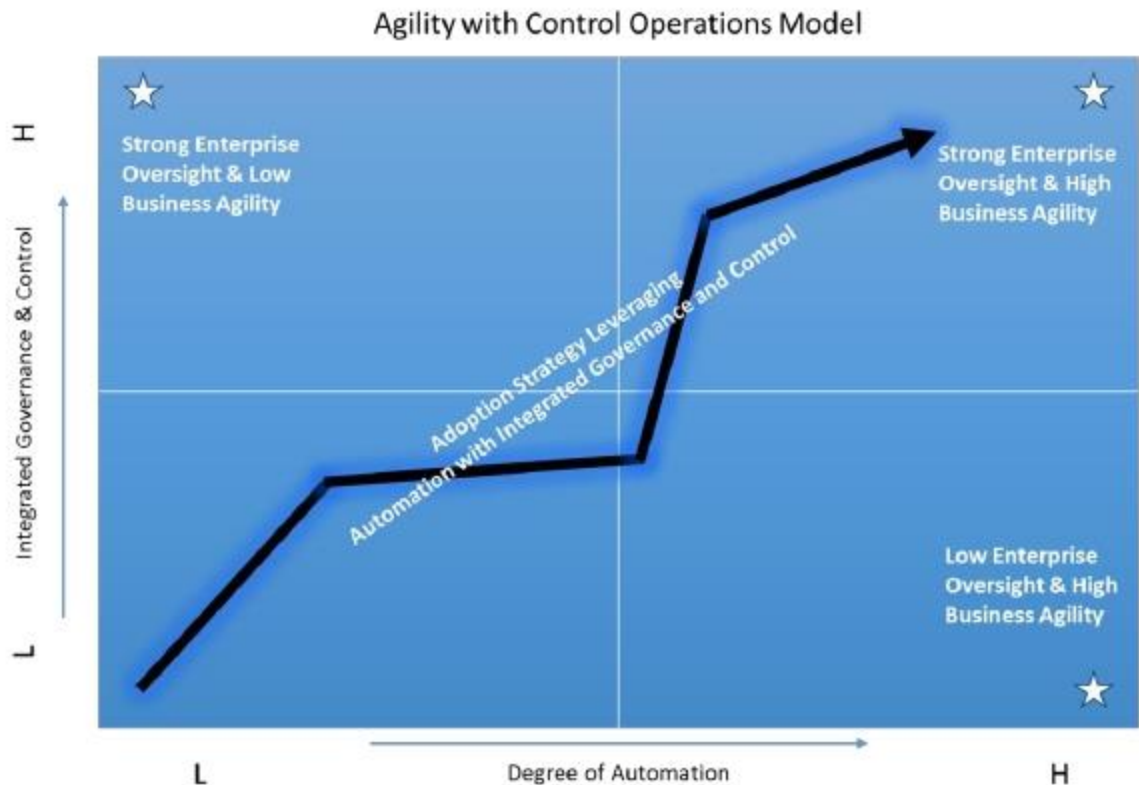
运营论点高度关注技术与解决方案架构师与其他 AWS 专业人员，需要由其观察并调整运营模式，从而由控制阻碍敏捷性转变为由控制实现敏捷性。敏捷性与控制手段的配合亦成为现代开发模式中目标与诉求的重要核心。



图一：运营论点各组成部分

当在服务目录当中采用云功能时，例如 AWS Service Catalog，您可以预先规划并提供运营指导意见，从而帮助开发团队立足于更为充实的发挥空间完成创新。

随着企业由内部环境逐步转向混合型内部/云环境，再到全面纯云 IT 环境，运营团队需要根据为支持模式提供增量式变更以响应各类实际需求。随着时间推移，转向云环境能够帮助 IT 环境运营与支持人员提升个人技能与经验，从而更为顺畅地完成云环境操作。面向 DevSecOps（即开发/安全/运维）等现代运营模式则能够跨越多个部门建立共同的目标与协作概念，从而在企业内促进并实现文化变革。最终，DevSecOps 模型将把运营业务功能扩展至开发团队当中，旨在建立起业务责任分担机制。



**Figure 2: Creating a Balanced Environment of Agility with Control**

无论您在云端需要运营的是传统、敏捷、DevOps 抑或是 DevSecOps 模式，最终都将通过一整套高度自动化的支持主干实现效率、控制与治理能力。

在各类模式当中，运营的目标始终基本保持一致——包括实现与业务需求相符的可用性、可靠性以及可维护性。而对这些目标的针对性调整，则是为了帮助产品开发团队与最终用户尽可能多地对当前环境加以控制。

- 可用性 - 内部 IT 环境当中往往存在多座数据中心，用于实现高可用性水平。利用 AWS 云，其中的多可用区架构与自动化机制的结合，能够以更低的资源成本实现同样的运营标准。
- 可靠性 - 在内部环境当中，您需要利用多套系统为 IT 体系内的每种组件提供冗余环境，旨在确保系统整体能够在发生故障后立足单点实现恢复。您也可以利用 AWS 云资源实现同样的冗余效果。然而，Amazon 的冗余机制可通过同等数量的 EC2 实例实现，其跨越多个分布式可用区并配合主动/主动设计，从而有效提高资源利用效率。
- 可维护性 - 在内部来表示当中，IT 运维团队负责为最终用户提供全方位服务。而在 AWS 云中，IT 运维团队将与 AWS 乃至业务部门自身共同分担环境维护职责。这意味着 IT 运维人员在保持原有环境维护能力的同时，减轻了维护责任压力，具体请参阅 AWS 责任分担页面<sup>2</sup>的阐述内容。
- 最终用户控制 - 在由内部环境负责交付自助服务功能时，环境对于最终用户的控制往往需要采用复杂的开发与工具架构。而在 AWS 云中，大家可以利用 AWS Service Catalog 等服务以集中化方式管理常见的 IT 服务，同时利用 AWS CloudFormation 创建并管理一整套相关 AWS 资源。再有，利用 AWS CodeDeploy 则允许业务部门及 IT 团队共同实现软件的自动化部署。

本份白皮书将集中讨论如何实现敏捷性与控制相结合，同时演进企业文化以建立现代 IT 运营模式。我们关注 AWS CAF 运营论点中的下列组成部分，提供可资利用的举措（即规范性指导）以建立可迁移至云端并运营云解决方案的可行性规划：

- 服务管理
- SLA/OLA 策略
- 业务连续性规划
- 事故与问题管理
- 变更与配置管理
- 性能与运营状态

## 服务管理

作为AWS CAF运营论点的组成部分，服务管理主张在AWS环境当中将敏捷性与控制手段相结合。利用自动化机制以主动支持方式实现服务管理，从而区别于手动人为干预采取的被动性支持方式。其同时利用资源部署与自动化响应两种方式建立方案设计，从而解决潜在故障因素。AWS云平台提供全面的自动化功能，可在节约成本与时间投入之外，同时提升服务质量水平。在云方案采用过程中，我们应逐步减少手动任务执行元素，允许运维团队人员将精力集中在更具附加价值创造能力的工作身上。

要建立起敏捷性与控制手段相结合的环境，运营人员必须创建出一套资产清单，同时对所需要的新产品与新服务进行优先级排序。此份清单相当一套在服务与功能中内置控制手段的网格，且作为服务目录进行整理——例如将其纳入AWS Service Catalog。如果企业能够将此类目录以自助服务模式进行交付，则开发团队将能够获得更为理想的敏捷性水平。在此基础上，开发团队将能够根据需要配置资源，从而轻松完成功能实验并缩短新型业务组合功能的上市周期。开发团队随后可以自动化方式将治理与控制组件整合至服务当中，最终实现更高水平的运营效率。

组合管理章程中的部分内容用于定义构建服务目录所必需的能力。运营团队能够执行此类服务。另外，开发团队应当将AWS Service Catalog中的功能视为软件即服务（简称SaaS）方案。利用SaaS至上原则配合目录功能，即可最大程度缩减所需维护的代码规模，这亦应当成为团队在管理服务目录时的一项基本目标。在此基础上，维护现有代码的时间投入亦将得到有效控制，同时允许开发人员将更多精力投入到新功能的研究当中。

利用这种方式，开发人员将拥有更为可观的自由与开放空间，同时可利用自动化机制降低交付成本。在交付工作转化为自助服务工具之后，开发团队将能够快速向生产环境中投放解决方案，这将从根本上转变运营人员的职能定位——即由实现交付转化为管理交付。



另外，AWS Marketplace 亦提供商用现成（简称 COTS）方案——例如能够为第三方供应商所采用并维护的解决方案，这就消除了构建定制化解决方案的必要性。如此一来，我们将能够以最低投入显著提升服务目录的实际功能。这也意味着开发团队将拥有更高自主性，从而选择适合自己的合作伙伴以及用于实现解决方案交付的服务选项。

该服务目录的规模随后会呈指数级增长，同时消除以往功能交付过程中所必需的代码托管与支持要求。这意味着我们将能够立足于战略角度采购及运用最佳解决方案，而不必担心运营环境因此面临潜在风险。我们可以在采购流程中查询 AWS Marketplace 内可被纳入服务目录的供应商选项。通过仅批准纳入提供相关服务的供应商，我们将能够保持服务目录拥有极高的企业业务契合度并满足各类治理与控制预期。

在对可纳入服务目录的 AWS 服务进行审查时，大家应考虑在 AWS Marketplace 中物色合格的 AWS 技术合作伙伴。AWS 利用多种审查机制以确保 AWS Marketplace 中提供的服务拥有良好的质量水平。这套方案可被用于提升服务目录规模，同时有效控制我们用于搜索及评估供应方服务质量的时间投入。

服务条款与条件（简称 T&C）可由 IT 人员负责初步设置及治理，从而提升目录稳定性。在此之后，新型服务经由治理流程接受评判并最终得以引入。这种保障性流程能够在提供服务价值的同时，帮助我们摆脱采购活动带来的沉重业务成本。当采购流程初步批准目标供应商后，该供应商的产品即成为大家服务目录中的组成部分。而在设定检查与权衡标准（以及发布必要工作负载以进行服务目录运营）时，可考虑将新服务批准流程从目录维护工作当中拆分出来。

## 注意事项

- **应当**建立一整套开发/安全/运营文化，旨在帮助全部团队建立起解决方案设计、交付与运营工作层面的自主权与主动性。
- **应当**针对故障状况做出设计。
- **应当**利用 Amazon Machine Image（简称 AMI）中的探针实现配置管理，从而确保治理与控制能力。
- **应当**创建一套基础设施指导性服务目录，用于快速设计基础设施环境并实现其自动化运营。
- **应当**管理各类引导实例，包括额外的治理与控制功能。
- **不应**继续以人工配置方式管理基础设施环境。尽早考虑基础设施部署工作的自动化实现手段，从而消除人为错误并改善基础设施的可扩展能力。

## SLA/OLA 策略

作为 AWS CAF 运营论点中的组成部分，SLA/OLA 策略鼓励大家由以往以 IT 为核心的 IT 服务交付视角转化为以用心为核心的新型视角。服务水平协议（简称 SLA）与运营水平协议（简称 OLA）标准并不算是 IT 服务交付领域的新鲜事物。从传统角度讲，这些标准通过最终用户或者投资组合拥有者同运营团队间的协商所最终建立。其中常用的重要指标包括可用性、可靠性以及响应时间等。这些指标应当与实现相关要求所必需的投资水平进行权衡，确保解决方案的实施成本不至于超过解决方案本身的实际价值。

AWS 采用各类具备良好架构的解决方案，包括多可用区设计，旨在简化实现 SLA/OLA 标准时带来的运营负担。考虑到功能变更带来的巨大影响，大多数企业会选择在应用或者服务迁移项目部署之初审查预期 SLA 与 OLA。

当按照最佳实践指导推进云采用工作时，您会发现随着时间的推移，为正常运行时间或者可用性等传统指标设定 SLA 及 OLA 将失去实际意义。相反，我们开始由以往以 IT 为核心的服务与运营水平评估视角，逐步转移至以最终用户体验为核心的全新视角。AWS 鼓励大家在服务及运营水平定义当中，转向用户为核心或者以用户体验为核心的新思路。

您可以有效削减由采购、配置以及部署高可用性（简称 HA）解决方案所带来的时间与资金投入，从而改善开发与业务部门的敏捷性水平。通过遵循各项最佳实践指导并复用高效云架构模式，开发团队将能够轻松实现高可用性解决方案。

在云环境下，当我们执行自动化故障检测与恢复时，大家的系统会采取持续故障与自动恢复处理机制，意味着 IT 环境的用户不会直接感受到故障的出现。另外，开发人员能够操作代码以显示当前 SLA 或者 OLA，从而确保开发出的代码满足最终用户所认同的体验水平。

## 注意事项

- 应当 复查 SLA/OLA 标准水平。
- 应当 建立并遵循最佳实践，同时立足于各支持层为应用程序制定良好架构设计标准。
- 应当 利用自动化手段支持多可用区架构，从而实现应用程序的高可用性（简称 HA）。
- 应当 转而从最终用户的角度出发审视 SLA 与 OLA 预期。
- 不应在未采用先进云架构的前提下急于“上线并转向”顶层架构。

## 业务连续性规划

作为 AWS CAF 运营论点中的组成部分，业务连续性鼓励企业以制定流程及规程的方式在发生灾难情况时维护业务功能。从基础设施与技术功能角度来看，AWS 提供多种解决方案以消除灾难发生时实现业务连续性的成本与复杂性。能够在多个不同跨地理位置服务区内添加额外云功能，企业客户能够轻松应对各类潜在灾难事故。

IT 部门应当确保自身能够在受灾害问题影响时，继续保持正常运营。灾难恢复规划与流程必须提前部署到位，同时接受业务部门的定期测试，这意味着 IT 组织将能够从灾难情况下快速恢复并继续支持业务运营。云环境下的运营管理工作须着眼于主动性、端到端自动化层面。Netflix 公司亦全面采用了此类方案。Netflix 的 Simian Army 工具——其中包括 Chaos Monkey<sup>3</sup>——即是以主动方式模拟故障条件并实现测试的典型示例。信息技术基础设施库（简称 ITIL）亦针对各类不利条件提供诸多与 IT 服务管理（简称 ITSM）息息相关的有价值指导意见。这份指南同样适用于云环境。

根据业务连续性的具体实现要求，单一 AWS 服务区提供的可用性一般来讲已经能够满足灾难恢复与业务连续性需求。不过对于更为严格的要求，AWS 服务亦允许客户将其数据在不同服务区间进行复制。

AWS 建议客户定期横跨不同地理位置利用不同账户进行备份快照传输。一旦发生灾难性事件，大家将能够借此启动灾难恢复流程。

您也可以利用多个账户以降低主账户中登录凭证受到影响的可能性。设置多个账户以管理 AWS 服务，能够帮助您利用最低所需权限完成操作，同时为业务连续性维护提供额外保障。

当大家将应用程序设置为主动/主动而非主动/被动，并将其在多个可用区内启动时，您即建立起了一套具备高可用性保障且能抵御灾难侵袭的解决方案。您无需像以往那样通过将工作负载转移到第二站点或者服务区以实现业务持续性。然而，这种新型实现手段也会给需要支持大规模整体式遗留应用的技术团队带来一系列挑战。将应用程序架构由整体式转化为松散耦合式所带来的精力与资源投入往往相当可观。不过，企业完全能够在生产力前景与降低未来投资需求层面实现投资回报，从而鼓励其将上述转型工作纳入一级投资组合。欲了解更多细节信息，请参阅 James Hamilton 在 AWS re:Invent 2014 大会上带来的 AWS 规模化创新 4 演讲。

通过遵循以上模式，企业将能够以 AWS 服务为基础运用良好架构解决方案，从而显著改善恢复时间目标（简称 RTO）与恢复点目标（简称 RPO）。

## 注意事项

- 应当立足于各支持层为应用程序创建最佳实践与良好架构设计标准。
- 应当利用多可用区与多服务区架构为业务持续性规划带来安全的设施距离间隔设计。
- 应当考虑利用 AWS 云资源在低成本前提下扩展现有内部设施的业务持续性规划。
- 应当利用多账户方法制定妥善规划，从而应对单一账户遭到滥用时产生的安全问题。
- 应当利用常规业务连续性规划中的 RPO 与 RTO 要求，对现有设计、架构与实现方案进行测试及验证。
- 不应在纯云环境下继续使用传统的内部业务持续性规划实践。

## 事故与问题管理

作为 AWS CAF 运营论点中的组成部分，事故与问题管理鼓励大家设计一套自动化解决方案，旨在从故障状态下实现恢复。事故与问题管理方案的目标在于缩短事故与问题解决以及恢复正常服务运营所需要的时间，从而最大程度降低此类状况对最终用户造成的影响。

在云解决方案的帮助下，IT 团队能够针对故障状况提供主动式弹性解决方案，从而确保将注意力集中在系统问题管理而非故障点控制身上。在考虑到故障状况且与经过验证（通过主动模拟故障场景）的自我修复解决方案当中，设置集中化验收标准以确保其能够在面对事故与问题时提供理想的恢复能力。

这种方法层面的转化使得事故与问题响应成为自动化解决方案中的组成部分，而不再像以往那样要求技术团队以手动方式进行事故与问题的故障排查、修复与记录。

由于云环境拥有天然存在的自我修复特性，因此大多数事故与问题都会快速得到恢复，而无需人为介入整个恢复流程。自动化问题与事故提交记录机制应当成为企业内解决方案设计与开发文化的一部分。故障恢复的自动化特性也将帮助事故管理团队将更多精力集中在关注发展趋势身上，而非用于应对故障点。

## 注意事项

- 应当 利用容错性架构设计故障应对机制，从而实现自动化事故与问题响应能力。
- 应当在现有问题管理平台上实现自动化申报与记录流程。
- 应当 建立良好架构模式，从而改善 AWS 环境内的一致性与故障承受能力。
- 应当 建立一套基础设施模板库，并利用其实现基础设施环境的快速设计与自动化运营。
- 应当 测量并监控事故与问题的响应时间，从而确保这一指标能够随时间推移而不断得到改善。
- 应当 为经常发生的事故与问题找到系统性解决办法。
- 不应 以人工方式处理反复发生的事故与问题。



## 变更与配置管理

作为 AWS CAF 运营论战中的绯闻，变更与配置管理鼓励大家以自动化方式实现基础设施与软件部署，具体包括问题识别与缓解、以及显著提升变更速度等。作为 AWS 采用战略的一部分，其需要着眼于敏捷性与可控性相结合的变更管理机制，同时负责对变更管理结构进行调整。当所有团队皆参与到解决方案的设计、创建、交付与运营当中时，也即意味着他们亦共同分担着变更实现、问题识别与缓解等责任。通过这种方式，企业将能够由以往的每年数次成果发布升级至如今的每天数次成果发布。

为了实现这种分布式模式，同时改善企业内部的整体敏捷性水平，作为应用程序交付体系的各个团队都必须利用自动化手段实现控制能力改进。最重要的是建立起合理的检查与权衡机制，其中包括在做出可能影响生产环境的决策之前进行人为介入。随着企业云采用标准的逐步成熟，这些检查与权衡工作亦将被纳入自动化体系当中，从而降低人为介入的影响力水平。

AWS 服务将相关责任进行分散，从而实现更为轻松的环境管理与监控能力。大家可以获得资源清单、配置历史以及配置变更通知等便携功能。另外，大家也可以监控多种指标、收集日志文件并设定警报等等。您可立足于系统整体直接查看资源使用情况、应用程序性能以及运营健康状况。所有服务交付管理流程都应当以自动化方式实现，因为这将显著提高效率与精度，同时降低执行成本。

原本需要由系统及网络工程师手动进行的硬件配置任务，如今可以通过基础设施即代码与统一变更与配置方案加以自动化处理。运营与维护工作将由关注物理硬件转变为关注所需维护目录内的各项服务。在 AWS 环境下，在需要做出变更时，工程师们将不再需要真正进入计算机房并以物理方式配置硬件，而是利用模板直接配置实际需要的服务器数量、网络以及其它云资源。

基础设施即代码堪称 IT 基础设施部署自动化的实现基础，且应作为行业标准性手段被纳入各类配置管理工具当中。在这里，企业能够实现持续集成/持续部署（简称 **CI/CD**），从而将基础设施与软件的自动化部署结合起来。

变更管理应当专注于利用可重复流程由一种已知良好状态转移至另一种已知良好状态。使用传统变更管理方案会拖慢这种变更速度，而自动化机制则能够大大加快变更速度。为了支持云环境下的变更实现速度水平，变更管理团队必须转换自身实践模式。他们需要摆脱以往每周或者每月召开变更研讨会议的作法，转而立足于 **CI/CD** 技术手段制定软件开发最佳实践。这种方案允许我们在新功能与用户预期不相适应的情况下，快速回滚至原有已知良好状态。

要实现云环境的全部潜在价值，变更工作可以亦应当通过由团队开发及交付的解决方案或者发布产品实现。变更速度可由原本的每周或者每月数次提升到每天数次。**CI/CD** 技术通常属于 **DevOps** 或者 **DevSecOps** 文化的最终产物，而这些技术本身亦作为敏捷性与可控性相结合的最佳水平的直接体现。

## 注意事项

- 应当努力建立起开发/安全/运营文化。
- 应当将变更管理责任进一步分配至各组织与开发团队当中。
- 应当通过利用各 AWS 服务及第三方监控、日志记录与控制工具实现控制与监督。
- 应当建立一套基础设施模板库，用于快速设计并自动化运营基础设施环境。
- 应当利用 CI/CD 技术作为早期实践以推进分布式控制模式；包括采取基础设施即代码机制中的源代码管理手段。
- 不应强迫拥有现代开发规程的开发团队以月或者半月为单位进行变更控制。

## 性能与运营状态

作为 AWS CAF 运营论战中的组成部分,性能与运营状态鼓励企业利用 AWS 服务以提供监控云资产运作状态所必需的各类工具,同时确保其达到理想的性能水平。AWS 还提供多种工具以帮助我们对实例进行取证与分析,包括各类正在或者不再运行及可访问的实例。

在 AWS 云环境下,我们能够轻松配置必要的信息集合,从而监控性能与运营状态。在此之后,这部分信息可经由 AWS 服务实现进一步分析,并被导入至其它用于监控性能与运行状态的服务当中。AWS 环境可通过配置在满足阈值设定时实现自动化规模伸缩,同时自动解决运行状态问题。

大家可以利用 AWS 服务从实例当中收集并合并事件日志。相较于面向各单一实例进行记录,并分别审查记录以查找事件,云环境下的日志记录被整理至统一源并供用户访问。在 AWS 云中,此类“机下”型解决方案可轻松为大家服务。

事件信息可被合并至控制台或者仪表板当中,从而提供近实时型性能与运行状态统一数据视图。(如果需要,大家亦可手动审查日志。)这种作法能够将实例的直接访问数量控制在最低水平,从而极大改善企业的安全态势。利用这种机下方案,员工能够以集中化方式访问到必需的日志信息,但却无法接触到实际实例。

云环境下的日志存储成本亦得到有效控制。企业能够控制所记录数据的总体数量,从而实现更理想的敏捷性水平。如此一来,应用开发团队将能够记录更多应用使用数据,而后根据这部分数据做出针对性改进,最终实现理想的最终用户体验。企业亦可利用商务智能(简称 BI)工具简化且加快日志分析流程。结合预测分析,我们将能够将性能与运营状态的处理方式由被动转为主动。另外,利用商务智能工具还可降低分析应用程序或者服务运行状态问题时所需要的时间投入。

## 注意事项

- **应当** 利用 AWS 服务将运营与状态监控机制扩展至云环境当中。
- **应当** 将 AWS 监控服务与现有监控解决方案加以结合。
- **应当** 利用机下方案对不同日志记录与监控信息资源加以聚合。
- **应当** 评估高层应用日志记录的实际功效，利用其提升应用智能化水平以简化并缩短部署周期。
- **应当** 利用现代商务智能工具，包括预测分析，更为高效且有效地实现信息聚合。
- **不应** 继续利用内部方法及技术所提供的日志记录与监控手段。
- **不应** 在拥有其它选择的情况下继续使用机上日志记录方案。

## CAF 分类与术语

AWS 云采用框架（简称 CAF）是一套由 AWS 创建的框架，用于从以往客户的参与活动中收集指导依据与最佳实践。一项 CAF “论点” 代表的是与云 IT 系统在企业内实现所需要高度关注的某一相关领域。举例而言，当客户计划实施一项云解决方案时，运营论点负责为企业向 AWS 环境迁移时提供可资利用的现代化敏捷运营模式。

每项 CAF 论点都由多项组成部分及举措构成。其中，每项组成部分代表的是论点的一种子领域，指代企业需要关注的某个特定方向。本份白皮书探讨了成熟度论点范畴之内的各组成部分。另外，举措代表的则是建立可行性规划的指导性意见，企业可利用其完成云环境迁移并操作各类云解决方案。

举例来说，服务管理即为运营论点中的组成部分，而面向故障的主动设计则属于该组成部分内的一项举措。

而通过进一步合并，云采用框架（简称 CAF）与云采用方法（简称 CAM）二者可作为指导方针，帮助大家更顺利地完成 AWS 云过渡之旅。

## 总结

采用 AWS 服务能够帮助客户得以更为明确地审视自身 IT 成本，同时改善 IT 功能的交付速度。在生产环境下加快功能交付速度要求我们在速度、敏捷性以及敏捷性相关的运营控制及稳定性之间找到平衡点。在确认平衡点后，配合适当的控制、治理与可支持能力即可建立起一套涵盖“业务”与 IT 运营工作且健康的相互牵制、相互配合关系。在大多数企业当中，这方面挑战需要由运营团队根据商业合作伙伴给出的建议加以解决。

运营论点侧重于 AWS 所强调的技术与方案层面，同时为由敏捷性受可控性影响向敏捷性与可控性相结合的转变做出规定。运营论点提供的指导意见直指问题核心，并可作为建立现代开发模式的一种榜样性参考。

## 备注

- [https://do.awsstatic.com/whitepapers/aws\\_cloud\\_adoption\\_framework.pdf](https://do.awsstatic.com/whitepapers/aws_cloud_adoption_framework.pdf)
- <https://aws.amazon.com/compliance/shared-responsibility-model/>
- <http://techblog.netflix.com/2011/07/netflix-simian-army.html>
- [https://www.youtube.com/watch?v=JIQETrFC\\_SQ](https://www.youtube.com/watch?v=JIQETrFC_SQ)