AWS 白皮书

AWS Lambda 的安全性概览

AWS Lambda 的安全性概览: AWS 白皮书

Table of Contents

摘要 i
摘要 1
简介 2
关于 AWS Lambda 3
Lambda 的益处 3
无需管理服务器 3
连续扩缩 4
毫秒级计量 4
提高创新能力 4
实现应用程序现代化改造 4
丰富的生态系统 4
运行基于 Lambda 的应用程序的成本 4
责任共担模式 5
Lambda 函数 6
Lambda 调用模式 7
Lambda 执行 9
Lambda 执行环境 9
执行角色 10
Lambda MicroVM 和工件 10
Lambda 隔离技术 12
存储和状态 12
Lambda 中的运行时维护 14
监控和审计 Lambda 函数 15
Amazon CloudWatch 15
Amazon CloudTrail 15
AWS X-Ray 15
AWS Config 15
Lambda 函数架构和操作 16
Lambda 和合规性 17
Lambda 事件源 18
总结 19
贡献者 20
延伸阅读 21
文档修订 22

声明 23

AWS Lambda 的安全性概览

发布日期：2021 年 2 月 12 日（文档修订）

摘要

此白皮书从安全性角度深入探讨了 AWS Lambda 服务。它对该服务进行了充分诠释（这对新用户很有用），并让现有用户能够更深入地了解 Lambda。

本白皮书的目标读者是首席信息安全干事 (CISO)、信息安全工程师、企业架构师、合规性团队，以及任何有兴趣了解 AWS Lambda 基础知识的其他人员。

简介

如今，越来越多的工作负载使用 AWS Lambda 来实现可扩展性、性能和成本效益，而无需管理底层基础设施。这些工作负载可扩展到每秒数千个并发请求。Lambda 是 AWS 当今提供的众多重要服务之一。每月有成千上万的 Amazon Web Services (AWS) 客户使用 Lambda 来处理数万亿个请求。

Lambda 适用于许多行业的任务关键型应用程序。从媒体和娱乐到金融服务和其他受监管行业，各种各样的客户都在利用 Lambda。这些客户通过专注于自己最擅长的领域（即运营其业务），缩短了上市时间、优化了成本并提高了敏捷性。

托管式运行时环境模式使 Lambda 能够管理运行无服务器工作负载的大量实施细节。该模式进一步减少了攻击面，同时简化了云安全性。本白皮书为开发人员、安全分析师、安全与合规性团队以及其他利益攸关方介绍了该模式的基础以及最佳实践。

关于 AWS Lambda

AWS Lambda 是一项事件驱动的无服务器计算服务，它使用自定义逻辑扩展其他 AWS 服务，或创建其他具有规模、性能和安全性的后端服务。Lambda 可以自动运行代码来响应多个事件，例如，通过 Amazon API Gateway 发送的 HTTP 请求、Amazon S3 存储桶中的对象修改、Amazon DynamoDB 中的表更新以及 AWS Step Functions 中的状态转换。还可以直接从任何 Web 或移动应用程序运行代码。Lambda 在高可用性计算基础设施上运行代码，执行底层平台的所有管理工作，包括服务器和操作系统维护、容量预置和弹性伸缩、打补丁、代码监控和日志记录。

借助 Lambda，您只需上载代码并配置何时调用它；Lambda 负责处理以高可用性运行代码所需的所有其他事项。Lambda 可与许多其他 AWS 服务集成，并使您能够创建无服务器应用程序或后端服务，范围涵盖从定期触发的简单自动化任务到成熟的微服务应用程序。

还可以将 Lambda 配置为访问 Amazon Virtual Private Cloud 中的资源，并进一步访问本地部署资源。

可以使用 AWS Identity and Access Management (IAM) 以及本白皮书中讨论的其他技术，轻松地将 Lambda 与强大的安保状况相融合，以保持高水平的安全性和审计并满足合规性需求。

主题

- Lambda 的益处
- 运行基于 Lambda 的应用程序的成本

Lambda 的益处

希望释放其开发组织的创造力和速度（但前提是不影响 IT 团队提供可扩展、经济高效且易于管理的基础设施的能力）的客户发现，AWS Lambda 使他们能够在不损失规模或可靠性的情况下，以运营的复杂性来换取敏捷性和更优惠的定价。

Lambda 提供了诸多益处，包括：

无需管理服务器

Lambda 在分布于单个区域中多个可用区 (AZ) 之间的高度可用、容错的基础设施上运行您的代码，同时无缝部署代码，并提供对基础设施的所有管理、维护和补丁。Lambda 还提供内置的日志记录和监控功能，包括与 Amazon CloudWatch、CloudWatch Logs 和 AWS CloudTrail 集成。

连续扩缩

Lambda 通过并行运行事件触发代码并单独处理每个事件，精确地管理函数（或应用程序）的扩缩。

毫秒级计量

使用 AWS Lambda，您需要按代码运行时间（以每毫秒为单位）和代码触发次数付费。您需要为一致的吞吐量或执行持续时间（而不是服务器单元）付费。

提高创新能力

Lambda 通过接管基础设施管理工作来释放您的编程资源，使他们能够更多地专注于业务逻辑的创新和开发。

实现应用程序现代化改造

借助 Lambda，您可以将函数与预训练的机器学习模型结合使用，轻松地将人工智能注入应用程序。单个应用程序编程接口 (API) 请求可以对图像进行分类、分析视频、将语音转换为文本、执行自然语言处理等。

丰富的生态系统

Lambda 通过以下服务为开发人员提供支持：AWS Serverless Application Repository，用于发现、部署和发布无服务器应用程序；AWS Serverless Application Model，用于构建无服务器应用程序以及与各种集成开发环境 (IDE) 集成，例如 AWS Cloud9、AWS Toolkit for Visual Studio、AWS Tools for Visual Studio Team Services 等等。Lambda 与其他 AWS 服务集成，可为您提供用于构建无服务器应用程序的丰富生态系统。

运行基于 Lambda 的应用程序的成本

Lambda 提供精细的随用随付定价模式。使用此模式，您需要根据函数调用的次数及其持续时间（代码运行所花的时间）付费。除了这种灵活的定价模式外，Lambda 还提供每月 100 万个永久免费请求，这使许多客户能够免费自动执行其流程。

责任共担模式

安全性和合规性是 AWS 与客户的共同责任。该责任共担模式可以帮助减轻您的运营负担，因为 AWS 运营、管理并控制各个方面的组件，从主机操作系统和虚拟化层直至运行服务的设施的物理安全，由其全权负责。

对于 AWS Lambda，AWS 管理底层基础设施和基础服务、操作系统及应用程序平台。您负责代码的安全性以及对 Lambda 服务和函数内部的身份和访问管理 (IAM)。

图 1 显示了责任共担模式，它适用于 AWS Lambda 的共同和不同组成部分。AWS 责任以橙色显示在虚线下方，客户责任以蓝色显示在虚线上方。

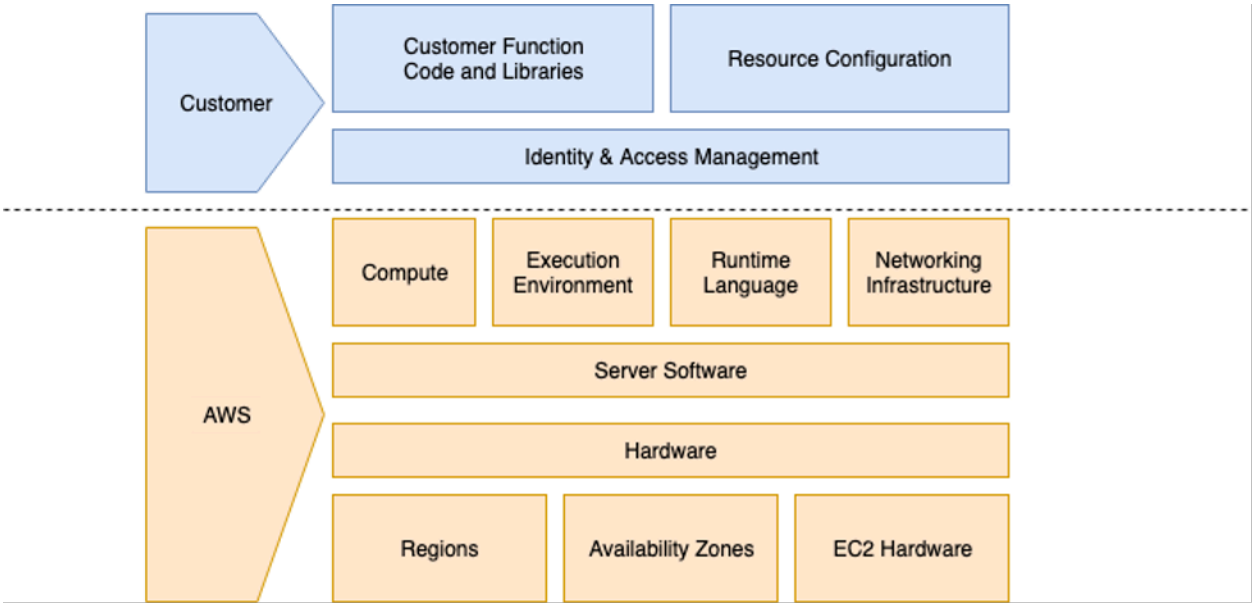

图 1 – AWS Lambda 的责任共担模式

Lambda 函数和层

借助 Lambda，您无需对底层基础设施进行管理即可通过虚拟方式运行代码。您仅负责您向 Lambda 提供的代码以及配置 Lambda 如何代表您运行该代码。目前，Lambda 支持两种类型的代码资源：函数 和层。

函数 是一种资源，可以调用它以在 Lambda 中运行代码。函数可以包括一个称为层 的公用或共享资源。层可用于在不同的函数或 AWS 账户之间共享公用代码或数据。您负责管理函数或层中包含的所有代码。当 Lambda 从客户那里收到函数或层代码时，Lambda 会使用 AWS Key Management Service (AWS KMS) 对函数或层代码进行静态加密，并在传输中使用 TLS 1.2+ 进行加密，从而保护对它的访问。

您可以通过 AWS Lambda 策略或基于资源的权限来管理对函数和层的访问。有关 IAM 上受支持的 IAM 功能的完整列表，请参阅使用 IAM 的 AWS 服务。

还可以通过 Lambda 的控制层面 API 控制函数和层的整个生命周期。例如，您可以选择通过调用 `DeleteFunction` 来删除您的函数，或通过调用 `RemovePermission` 撤消其他账户的权限。

Lambda 调用模式

调用 API 可以在两种模式下进行调用：事件 模式和请求-响应 模式。

- 事件 模式将有效负载排入队列以进行异步调用。
- 请求-响应 模式使用提供的有效负载同步调用函数并立即返回响应。

在这两种情况下，函数执行始终在 Lambda 执行环境中执行，但有效负载采用不同的路径。有关更多信息，请参阅本文档中的“Lambda 执行环境”。

您还可以使用代表您执行调用的其他 AWS 服务。使用哪种调用模式取决于您使用的 AWS 服务及其配置方式。有关其他 AWS 服务如何与 Lambda 集成的更多信息，请参阅将 AWS Lambda 与其他服务一起使用。

当 Lambda 收到请求-响应调用时，会直接将此调用传递给调用服务。如果调用服务不可用，调用方可能会暂时将有效负载客户端排入队列，以按设定的次数重试调用。如果调用服务收到有效负载，则该服务会尝试为请求确定可用的执行环境，并将有效负载传递给该执行环境以完成调用。如果不存在现有或适当的执行环境，则会根据请求动态创建一个执行环境。在传输过程中，发送到调用服务的调用有效负载受到 TLS 1.2+ 的保护。Lambda 服务内的流量（从负载均衡器向下）会通过隔离的内部 Virtual Private Cloud (VPC)，此 VPC 归 Lambda 服务拥有且位于请求发送到的 AWS 区域内。

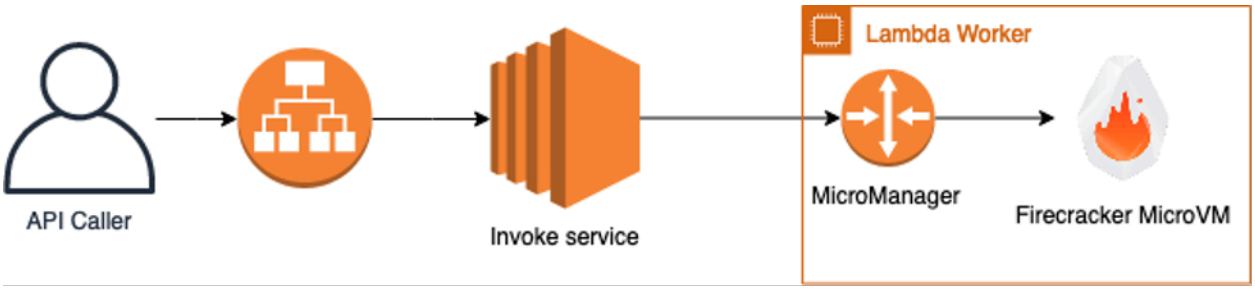

图 2 – AWS Lambda 请求-响应的调用模式

事件调用 模式有效负载在调用前始终排队等待处理。所有有效负载都在 Amazon Simple Queue Service (Amazon SQS) 队列中排队等待处理。排队的事件在传输过程中始终使用 TLS 1.2+ 进行保护，但它们目前尚未进行静态加密。Lambda 使用的 Amazon SQS 队列由 Lambda 服务管理，对客户不可见。排队的事件可以存储在共享队列中，但可能会根据客户无法直接控制的许多因素（例如，调用速率、事件大小等）迁移或分配到专用队列。

排队的事件由 Lambda 的轮询器机群 分批检索。轮询器机群是一组 EC2 实例，其目的是处理尚未处理的已排队事件调用。当轮询器机群检索到它需要处理的已排队事件时，它通过将其传递给调用服务来进行处理，就像客户使用请求-响应模式的调用一样。

如果无法执行调用，轮询器机群将在主机上的内存中临时存储事件，直到它能够成功完成执行或超过运行重试次数为止。没有有效负载数据会被写入轮询器机群本身的磁盘。轮询机群可以跨 AWS 客户执行任务，从而使调用时间最短。有关哪些服务可能采用事件调用模式的更多信息，请参阅将 AWS Lambda 与其他服务一起使用。

Lambda 执行

当 Lambda 代表您运行函数时，它将管理运行代码所需的底层系统的预置和配置工作。这使您的开发人员可以专注于业务逻辑和编写代码，而不是管理底层系统。

Lambda 服务分为控制层面 和数据层面。每个层面在服务中有不同的用途。控制层面提供管理 API（例如，`CreateFunction`、`UpdateFunctionCode`、`PublishLayerVersion` 等等），并管理与所有 AWS 服务的集成。与 Lambda 控制层面的通信在传输过程中受到 TLS 的保护。存储在 Lambda 控制层面中的所有客户数据都通过使用 AWS KMS 进行静态加密，旨在防止未经授权的泄露或篡改。

数据层面是 Lambda 的调用 API，用于触发 Lambda 函数的调用。调用 Lambda 函数时，数据层面会将 AWS Lambda 工件（或简称为工件，一种 Amazon EC2 实例类型）上的执行环境分配给该函数版本，或者选择已为该函数版本设置的现有执行环境，然后数据层面使用此环境来完成调用。有关更多信息，请参阅本文档的“AWS Lambda MicroVM 和工件”部分。

Lambda 执行环境

每个调用都由 Lambda 的调用服务路由到能够处理请求的工件上的一个执行环境。除了通过数据层面，客户和其他用户无法直接发起与执行环境的入站/进入网络通信。这有助于确保与执行环境的通信经过身份验证和授权。

执行环境是为特定函数版本预留的，不能在函数版本、函数或 AWS 账户之间重用。这意味着，单个可能具有两个不同版本的函数会导致至少两个独一无二的执行环境。

每个执行环境一次只能用于一个并发调用，并且出于性能考虑，它们可以在同一函数版本的多个调用之间重用。根据许多因素（例如，调用速率、函数配置等），给定函数版本可能存在一个或多个执行环境。通过这种方法，Lambda 能够为其客户提供函数版本级别隔离。

Lambda 目前不隔离在函数版本的执行环境中发生的调用。这意味着，一个调用可能会留下可能影响下一次调用的状态（例如，写入 /tmp 的文件或内存中数据）。如果您想确保一个调用不会影响另一个调用，Lambda 建议您创建其他不同的函数。例如，您可以为更容易出错的复杂解析操作创建不同的函数，并重用不执行安全敏感操作的函数。Lambda 目前对客户可以创建的函数数量没有限制。有关限制的更多信息，请参阅 Lambda 配额页面。

执行环境由 Lambda 持续监控和管理，而创建或销毁执行环境的原因可能有很多，包括但不限于：

- 新的调用到达，但不存在合适的执行环境

- 发生内部运行时或工件软件部署
- 发布了新的预置并发配置
- 执行环境或工件上的租用时间接近或已超过最大生命周期
- 其他内部工作负载重新平衡流程

客户可以通过在其函数配置上配置预置并发来管理某个函数版本存在的预置执行环境的数量。配置为执行此操作后，Lambda 将创建、管理所配置数量的执行环境并确保它们始终存在。这确保了客户能够更好地控制其无服务器应用程序在任何规模下的启动性能。

除了通过预置并发配置外，客户无法确定地控制 Lambda 为响应调用而创建或管理的执行环境的数量。

执行角色

还必须为每个 Lambda 函数配置执行角色，执行角色是 Lambda 服务在执行与函数相关的控制层面和数据层面操作时代入的 IAM 角色。Lambda 服务代入此角色以提取临时安全凭证，这些凭证随后在函数调用期间可作为环境变量使用。出于性能方面的考虑，Lambda 服务将缓存这些凭证，并可能在使用相同执行角色的不同执行环境中重用它们。

为确保遵守最低权限原则，Lambda 建议每个函数拥有自己唯一的角色，并配置其所需的最低权限集。

Lambda 服务还可以代入执行角色来执行某些控制层面操作（例如，与为 VPC 函数创建和配置弹性网络接口 (ENI)、向 Amazon CloudWatch Application Insights 发送日志以及向 AWS X-Ray 发送跟踪相关的操作），或执行其他与调用无关的操作。客户始终可以通过在 AWS CloudTrail 中查看审计日志来查看和审计这些使用案例。

有关此主题的更多信息，请参阅 AWS Lambda 执行角色文档页面。

Lambda MicroVM 和工件

Lambda 将在名为 AWS Lambda 工件 的 Amazon EC2 实例的机群上创建其执行环境。工件是裸机 EC2 Nitro 实例，它们由 Lambda 在一个单独的隔离 AWS 账户（客户看不到此账户）中启动和管理。工件拥有一个或多个由 Firecracker 创建的硬件虚拟化微型虚拟机 (MVM)。Firecracker 是一款开源虚拟机监控器 (VMM)，它使用 Linux 的基于内核的虚拟机 (KVM) 来创建和管理 MVM。它专为创建和管理安全、多租户容器和基于函数的服务而构建，这些服务可提供无服务器操作模式。有关 Firecracker 的安全模式的更多信息，请访问 Firecracker 项目网站。

作为责任共担模式的一部分，Lambda 负责维护工件的安全配置、控制和补丁级别。Lambda 团队使用 Amazon Inspector 来发现已知的潜在安全问题以及其他自定义安全问题通知机制和预披露列表，这样客户就无需管理其执行环境的底层安保状况。

图 3 – AWS Lambda 工件的隔离模式

工件的最长租赁期限为 14 小时。当工件接近最长租赁时间时，不再向它发送进一步的调用，MVM 正常终止，底层工件实例终止。Lambda 持续监控其机群生命周期内的生命周期活动并发出有关此类活动的告警。

所有与工件的数据层面通信都通过使用伽罗瓦/计数器模式的高级加密标准 (AES-GCM) 进行加密。除了通过数据层面操作外，客户无法直接与工件进行交互，因为在 Lambda 服务账户中，工件托管在由 Lambda 管理的网络隔离 Amazon VPC 中。

当工件需要创建新的执行环境时，它会获得访问客户函数构件的限时授权。这些构件专门针对 Lambda 的执行环境和工件进行了优化。使用 ZIP 格式上载的函数代码经过一次优化，然后使用 AWS 托管式密钥和 AES-GCM 以加密格式存储。

使用容器镜像格式上载到 Lambda 的函数也进行了优化。容器镜像首先从其原始源下载，优化为不同的区块，然后使用经过身份验证的融合加密方法存储为加密块，该加密方法结合使用 AES-CTR、AES-GCM 和 SHA-256 MAC。融合加密方法允许 Lambda 安全地消除加密区块的重复数据。解密客户数据所需的所有密钥都使用客户托管式 AWS KMS 客户主密钥 (CMK) 进行保护。客户可以在 AWS CloudTrail 日志中使用 Lambda 服务所使用的 CMK，以便进行跟踪和审计。

Lambda 隔离技术

Lambda 使用各种开源和专有隔离技术来保护工件和执行环境。每个执行环境都包含以下各项的专用副本：

- 特定函数版本的代码
- 为您的函数版本选择的任何 AWS Lambda 层
- 所选函数运行时（例如 Java 11、NodeJS 12、Python 3.8 等）或函数的自定义运行时
- 可写入的 /tmp 目录
- 基于 Amazon Linux 2 的最小 Linux 用户空间

执行环境使用 Linux 内核中内置的几种类似容器的技术以及 AWS 专有的隔离技术相互隔离。这些技术包括：

- cgroups – 用于限制函数对 CPU 和内存的访问。
- namespaces – 每个执行环境都在专用的命名空间中运行。我们通过拥有唯一的组进程 ID、用户 ID、网络接口以及其他由 Linux 内核管理的资源来实现这一点。
- seccomp-bpf – 限制可以在执行环境中使用的系统调用 (syscall)。
- iptables 和 路由表 – 阻止入口网络通信并隔离 MVM 之间的网络连接。
- chroot – 提供针对底层文件系统的限定访问。
- Firecracker 配置 – 用于限制块储存设备和网络设备吞吐量的速率。
- Firecracker 安全功能 – 有关 Firecracker 的当前安全设计的更多信息，请参阅 Firecracker 的最新设计文档。

这些机制与 AWS 专有的隔离技术一起，在执行环境之间提供了强大的隔离。

存储和状态

执行环境始终不会在不同的函数版本或客户之间重用，但可以在同一函数版本的多个调用之间重用单个环境。这意味着数据和状态可以在调用之间保留。作为正常执行环境生命周期管理的一部分，数据和/或状态可能会持续存在数小时，然后才会被销毁。出于性能方面的考虑，函数可以利用此行为，通过在调用之间保留和重用本地缓存或长寿命连接来提高效率。在执行环境中，这些多次调用由单个进程处理，因此，如果调用发生在重用的执行环境中，则任何进程范围的状态（例如 Java 中的静态状态）都可以供将来的调用重用。

每个 Lambda 执行环境还包括一个可写入的文件系统（位于 /tmp）。此存储无法跨执行环境访问或共享。与进程状态一样，写入 /tmp 的文件将在执行环境的生命周期内保留。这允许在多个调用间分摊昂贵的传输操作，例如下载机器学习 (ML) 模型。不希望在两次调用之间保留数据的函数不应写入 /tmp，或者应在调用之间从 /tmp 中删除其文件。/tmp 目录由 Amazon EC2 实例存储提供支持，并进行静态加密。

希望将数据保留到执行环境之外的文件系统的客户应考虑使用 Lambda 与 Amazon Elastic File System (Amazon EFS) 的集成。有关更多信息，请参阅将 Amazon EFS 与 AWS Lambda 配合使用。

如果客户不希望在调用之间保留数据或状态，Lambda 建议他们不要使用执行上下文或执行环境来存储数据或状态。如果客户希望主动防止在调用之间泄漏数据或状态，Lambda 建议他们为每个状态创建不同的函数。Lambda 不建议客户在执行环境中使用或存储安全敏感状态，因为该状态可能会在调用之间发生变化。我们建议改为在每次调用时重新计算状态。

Lambda 中的运行时维护

Lambda 通过持续扫描和部署兼容的更新和安全补丁以及执行其他运行时维护活动来为这些运行时提供支持。这使客户能够仅专注于其功能和层中包含的任何代码的维护和安全性。Lambda 团队使用 Amazon Inspector 来发现已知的安全问题以及其他自定义安全问题通知机制和预披露列表，以确保我们的运行时语言和执行环境保持已修补状态。如果确定了任何新的补丁或更新，Lambda 将在没有客户参与的情况下测试和部署运行时更新。有关 Lambda 合规性计划的更多信息，请参阅本文档的“Lambda 和合规性”部分。

通常，为受支持的 Lambda 运行时选取最新补丁不需要执行任何操作，但有时可能需要在部署补丁之前先对补丁进行测试（例如，已知的不兼容的运行时补丁）。如果客户需要执行任何操作，Lambda 将通过 Personal Health Dashboard、AWS 账户的电子邮件或其他方式与他们联系，告知他们需要执行的具体操作。

通过实施自定义运行时，客户可以在 Lambda 中使用其他编程语言。对于自定义运行时，运行时的维护成为客户的责任，包括确保自定义运行时包含最新的安全补丁。有关更多信息，请参阅《AWS Lambda 开发人员指南》中的自定义 AWS Lambda 运行时。

当上游运行时语言维护者将其语言标记为生命周期结束 (EOL) 时，Lambda 将不再支持该运行时语言版本。当运行时版本在 Lambda 中标记为已弃用时，Lambda 将停止支持创建新函数，并且不再支持对在弃用的运行时中编写的现有函数创建更新。为了提示客户即将发生的运行时弃用，Lambda 会向客户发送通知，告知即将弃用的日期以及预期情况。Lambda 不会为弃用的运行时提供安全更新、技术支持或修补程序，并保留权利随时禁止调用配置为在弃用的运行时上运行的函数。如果客户希望继续运行已弃用或不受支持的运行时版本，他们可以创建自己的自定义 AWS Lambda 运行时。有关何时弃用运行时的详细信息，请参阅 AWS Lambda 运行时支持策略。

监控和审计 Lambda 函数

您可以使用许多 AWS 服务和方法（包括以下服务）监控和审计 Lambda 函数。

Amazon CloudWatch

AWS Lambda 自动代表您监控 Lambda 函数。它通过 Amazon CloudWatch 报告指标，例如请求数、每个请求的执行持续时间以及导致错误的请求数。这些指标在函数级别公开，然后您可以利用这些指标来设置 CloudWatch 告警。有关 Lambda 公开的指标的列表，请参阅 AWS Lambda 指标。

Amazon CloudTrail

使用 AWS CloudTrail，您可以对整个 AWS 账户（包括 Lambda）实施监管、合规性、运营审计和风险审计。CloudTrail 使您能够记录、持续监控和保留与整个 AWS 基础设施中的操作相关的账户活动，同时提供通过 AWS Management Console、AWS 软件开发工具包、命令行工具和其他 AWS 服务执行的操作的完整事件历史记录。使用 CloudTrail，您可以选择使用 AWS KMS 加密日志文件，还可以利用 CloudTrail 日志文件完整性验证来实施肯定断言。

AWS X-Ray

使用 AWS X-Ray，您可以分析和调试基于 Lambda 的生产和分布式应用程序，这样您就可以了解应用程序及其底层服务的性能，从而可以最终确定性能问题和错误的根本原因并排除故障。当请求在应用程序中传输时，X-Ray 的端到端请求视图会显示应用程序底层组件的地图，因此您可以在开发和生产过程中分析应用程序。

AWS Config

借助 AWS Config，您可以跟踪对 Lambda 函数（包括已删除的函数）、运行时环境、标签、处理程序名称、代码大小、内存分配、超时设置、并发设置以及 Lambda IAM 执行角色、子网和安全组关联的配置更改。这可让您全面了解 Lambda 函数的生命周期，并使您能够显示该数据以满足潜在的审计和合规性要求。

Lambda 函数架构和操作

本节讨论 Lambda 架构和操作。有关无服务器应用程序的标准最佳实践的信息，请参阅《无服务器应用程序剖析》白皮书，该白皮书定义并探讨了无服务器环境中 AWS Well Architected Framework 的各个支柱。

- 卓越运营支柱 – 运行和监控系统以创造商业价值并不断改进支持流程和程序的能力。
- 安全性支柱 – 在通过风险评估和缓解策略创造商业价值的同时保护信息、系统和资产的能力。
- 可靠性支柱 – 系统从基础设施或服务中断中恢复、动态获取计算资源以满足需求以及减少诸如配置错误或暂时性网络问题等中断的能力。
- 性能效率支柱 – 高效利用计算资源来满足需求，以及在需求发生变化和技术不断演进的情况下保持这种效率。
- 成本优化支柱 – 不断完善和改进的过程，以确保实现业务成果，同时随着需求的变化和技术的发展，最大限度地降低成本。

《无服务器应用程序剖析》白皮书包括日志记录指标和告警、限制和局限性、向 Lambda 函数分配权限以及使敏感数据可供 Lambda 函数使用等主题。

Lambda 和合规性

正如“责任共担模式”部分所述，您有责任确定哪种合规性制度适用于您的数据。确定合规性制度需求后，您可以使用各种 Lambda 功能来匹配这些控制措施。您可以联系 AWS 专家（如解决方案构架师、领域专家、技术客户经理和其他人力资源）寻求帮助。然而，AWS 无法针对合规性制度是否（或哪些合规性制度）适用于特定使用案例向客户提供建议。

截至 2020 年 11 月，Lambda 已纳入 SOC 1、SOC 2 和 SOC 3 报告的范围，此类报告是独立的第三方检查报告，阐明 AWS 如何达成关键合规性控制和目标。有关合规性信息的最新列表，请参阅合规性计划范围内的 AWS 服务页。

由于某些合规性报告的敏感性质，它们无法公开共享。要访问这些报告，您可以登录 AWS Management Console并使用 AWS Artifact（免费的自助式门户网站）按需访问 AWS 合规性报告。

Lambda 事件源

Lambda 通过直接集成方式与 140 多项 AWS 服务集成，并与 Amazon EventBridge 事件总线集成。常用的 Lambda 事件源有：

- Amazon API Gateway
- Amazon CloudWatch Events
- Amazon CloudWatch Logs
- Amazon DynamoDB Streams
- Amazon EventBridge
- Amazon Kinesis Data Streams
- Amazon S3
- Amazon SNS
- Amazon SQS
- AWS Step Functions

使用这些事件源，您可以：

- 使用 AWS Identity and Access Management 安全地管理对服务和资源的访问。
- 加密静态数据。*所有服务都会加密传输中的数据。
- 使用 VPC 终端节点从您的 Amazon Virtual Private Cloud 进行访问（由 AWS PrivateLink 提供支持）
- 使用 Amazon CloudWatch Application Insights 收集、报告指标并对指标发出告警。
- 使用 AWS CloudTrail 记录、持续监控和保留与整个 AWS 基础设施中的操作相关的账户活动，提供通过 AWS Management Console、AWS 软件开发工具包、命令行工具和其他 AWS 服务执行的操作的完整事件历史记录。

*在发布之时，静态数据加密功能尚不适用于 Amazon EventBridge。继续监控服务主页以获取有关这些功能的更新。

总结

AWS Lambda 提供了强大的工具包，用于构建安全且可扩展的应用程序。Lambda 中的许多安全性和合规性最佳实践与所有 AWS 服务中的最佳实践相同，但有些是 Lambda 特有的。本白皮书介绍 Lambda 的益处、它对应用程序的适用性以及 Lambda 托管式运行时环境。它还包括有关监控和审计以及安全性和合规性最佳实践的信息。在考虑下一个实施时，请考虑一下您学到了有关 AWS Lambda 的什么内容，以及它将如何改进下一个工作负载解决方案。

贡献者

本文档的贡献者包括：

- Mayank Thakkar，Global Life Sciences 解决方案构架师
- Marc Brooker，高级首席工程师（无服务器）
- Osman Surkatty，高级安全工程师（无服务器）

延伸阅读

如需更多信息，请参阅：

- 责任共担模式，它解释了 AWS 对安全性的总体看法。
- AWS 安全最佳实践，其中涵盖了针对 AWS Identity and Access Management (IAM) 服务的建议。
- 无服务器应用程序剖析 涵盖 AWS Well-Architected Framework，并指明了确保工作负载架构设计符合最佳实践的关键元素。
- AWS 安全性简介 对 AWS 中的安全思想进行了广泛的介绍。
- AWS 风险与合规性 概述了 AWS 的合规性。

文档修订

要获得有关此白皮书的更新通知，请订阅 RSS 源。

更新-历史记录-更改	更新-历史记录-描述	更新-历史记录-日期
已更新	重大更新	2021 年 2 月 15 日
初次发布	白皮书首次发布	2019 年 1 月 3 日

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) 代表当前提供的 AWS 产品和实践，如有更改，恕不另行通知；并且 (c) AWS 及其附属机构、供应商或许可方不做任何承诺或保证。AWS 产品或服务“按原样”提供，不提供任何形式的保证、陈述或条件，无论是明示还是暗示。AWS 对其客户的责任和义务由 AWS 协议决定，本文档与 AWS 和客户之间签订的任何协议无关，亦不影响任何此类协议。