

AWS 决策指南

# AWS Fargate 或者 AWS Lambda ?



# AWS Fargate 或者 AWS Lambda ? : AWS 决策指南

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

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

# Table of Contents

决策指南 .....	1
简介 .....	1
差异 .....	3
使用 .....	8
文档历史记录 .....	11
.....	xii

# AWS Fargate 或者 AWS Lambda ?

了解差异并选择适合您的差异

目的	探索是否 AWS Fargate 或 AWS Lambda 满足您对无服务器计算服务的需求。
上次更新	2024 年 11 月 15 日
承保服务	<ul style="list-style-type: none"><li><a href="#">AWS Fargate</a></li><li><a href="#">AWS Lambda</a></li></ul>

## 简介

在开始探索是选择 AWS Lambda 还是 AWS Fargate 作为无服务器计算服务之前，您可能已经考虑了更广泛的 AWS 计算服务（请参阅[选择 AWS 计算服务决策指南](#)），并将其范围缩小到这两种选择，因为它们提供：

- 减少运营开销：Lambda 和 Fargate 都省去了服务器管理，从而减少了对修补、维护和容量规划的需求。
- Pay-per-use 定价：您只需为实际使用的计算资源付费，从而有可能降低可变工作负载的成本。
- 更快的部署：与预置和配置 EC2 实例相比，部署时间通常更短。
- 内置高可用性：这两项服务都会自动处理基础设施冗余。
- 简化合规性：减少攻击面和内置安全功能可以简化合规工作。
- 专注于代码：开发人员可以更多地专注于编写应用程序代码，而不是管理基础架构。

虽然 Lambda 和 Fargate 都是无服务器选项，但它们之间有显著的区别：

AWS Fargate 是一款用于容器的无服务器计算引擎，主要用于 Amazon ECS。它会自动管理您的基础架构，使您能够专注于部署和扩展容器化应用程序。Fargate 非常适合长时间运行的应用程序、微服务或批处理，在这些应用中，您需要对资源分配（CPU、内存）进行精细控制，并希望避免管理底层服务器。

AWS Lambda 是一项无服务器计算服务，可自动运行您的代码以响应事件，并管理底层计算资源。它最适合事件驱动型应用程序，例如处理上传到 Amazon S3 的文件、响应 HTTP 请求或运行计划任

务。Lambda 还非常适合流处理和数据处理应用程序，因为它能够根据事件自动扩展并实时处理大量数据。Lambda 可以处理来自 Amazon Kinesis 或 Amazon DynamoDB 等来源的数据流，从而无需管理基础设施即可进行高效的无服务器数据转换、筛选和分析。Lambda 专为短期任务（最长 15 分钟）而设计，根据请求数量和执行时间计费，因此对于零星的工作负载，它具有成本效益。

如果您的项目涉及事件驱动的短期任务或不可预测的工作负载，则 AWS Lambda 可能更合适。如果您需要运行具有特定资源需求的容器化应用程序（或者您需要持久的进程），AWS Fargate 那就更合适了。

下表更详细地介绍了这些服务之间的一些区别，以帮助您入门。

功能	AWS Fargate	AWS Lambda
执行模型	基于容器的无服务器计算	事件驱动无服务器函数
支持的语言	任何可以在容器中运行的语言	支持的语言：Node.js、Python、Java、C#、Go、Ruby 和 PowerShell。您也可以 <a href="#">构建自定义运行时</a> ，以您选择的语言实现 AWS Lambda 函数。
使用案例	长时间运行的容器化应用程序	持续时间短、事件驱动的任务
扩展	根据所需的任务数自动缩放	根据请求自动缩放
冷启动	35 秒到 2 分钟	100 ms 到 2 秒
执行时间限制	没有硬性限制	最长 15 分钟
内存分配	高达 120 GiB	最大 10 GiB
CPU 分配	最多 16 个 vCPU	与内存成正比，最多 6 个 vCPU
Networking	在 VPC 中运行，可以使用 ENIs	可以在 AWS 托管 VPC 中运行，也可以使用 AWS Hyperplane 连接到客户管理的 VPC
状态管理	只要容器处于运行状态，Fargate 中的容器就可以	无状态设计（状态必须由外部管理，例如 Amazon

	在请求之间保持状态，从而无需外部存储即可处理会话、缓存数据或保持内存状态。对于关键数据，建议使用外部存储。	S3、Amazon DynamoDB、Amazon EFS )
容器支持	支持容器	有限的容器支持 ( 通过容器镜像部署 )
编排	与 Amazon ECS 集成	无需编排
定价模式	每秒按使用的 vCPU 和内存计费	每次调用和持续时间 ( GB 秒 )
并发限制	基于集群容量	默认 1000 个并发执行 ( 可以增加 )
事件驱动调用	需要额外的设置	对各种 AWS 事件源的原生支持
缓解冷启动	使用 Seekable OCI 延迟加载图像可以加快 Fargate 任务的启动速度	预配置并发可用
Package 大小限制	没有具体限制 ( 容器大小受配置的临时存储空间的限制，最大 200 GiB )	已解压缩 250 MB ( 包括图层 ) ， 10GB 用于容器映像部署

## Fargate 和 Lambda 之间的区别

探索 Fargate 和 Lambda 在多个关键领域的区别。

### Languages supported

Fargate : AWS Fargate 是一项容器编排服务，这意味着它支持任何可以打包到 Docker 容器中的编程语言或运行时环境。这种灵活性允许开发人员使用几乎任何适合其应用程序需求的语言、框架或库。无论你使用的是 Python、Java、Node.js、Go、.NET、Ruby、PHP，还是自定义语言和环境，只要它们封装在容器中，Fargate 都可以运行它们。这种广泛的语言支持使 Fargate 非常适合运行各种应用程序，包括传统系统、多语言微服务和现代云原生应用程序。

Lambda : 与专为事件驱动函数设计的 Fargate 相比 , Lambda 为更有限的语言集 AWS Lambda 提供原生支持。到目前为止 , Lambda 正式支持以下语言 :

- Node.js
- Python
- Java
- Go
- Ruby
- C#
- PowerShell

Lambda 还支持自定义运行时 , 允许您使用自己的语言或运行时环境 , 但与使用本机支持的选项相比 , 这需要更多的设置和管理。如果您选择从容器映像部署 Lambda 函数 , 则可以使用 AWS 仅限操作系统的基础镜像并在映像中包含 Rust 运行时客户端 , 从而在 Rust 中编写函数。如果您使用的语言没有 AWS 提供的运行时接口客户端 , 则必须创建自己的运行时接口客户端。

## Event-driven invocation

Lambda 本质上是为事件驱动计算而设计的。Lambda 函数是为了响应各种事件而触发的 , 包括数据更改、用户操作或计划任务。它可以与许多系统无缝集成 AWS 服务 , 例如 Amazon S3 ( 例如 , 在上传文件时调用函数 )、DynamoDB ( 例如 , 在数据更新时触发 ) 和 API Gateway ( 例如 , 处理 HTTP 请求 )。Lambda 事件驱动架构非常适合需要立即响应事件而无需持久计算资源的应用程序。

Fargate 不是原生的事件驱动的 , 但是通过一些额外的样板逻辑 , 它可以与 Amazon SQS 和 Kinesis 等事件源集成。虽然 Lambda 会为您处理大部分集成逻辑 , 但您必须使用这些服务自己实现此集成。 APIs

## Runtime/use cases

Fargate 专为运行容器化应用程序而设计 , 提供了一个灵活的运行时环境 , 您可以在其中定义容器的 CPU、内存和网络设置。由于 Fargate 在基于容器的模型上运行 , 因此它支持长时间运行的进程、持久性服务和具有特定运行时要求的应用程序。Fargate 中的容器可以无限期运行 , 因为对执行时间没有硬性限制 , 因此非常适合需要持续启动和运行的应用程序。

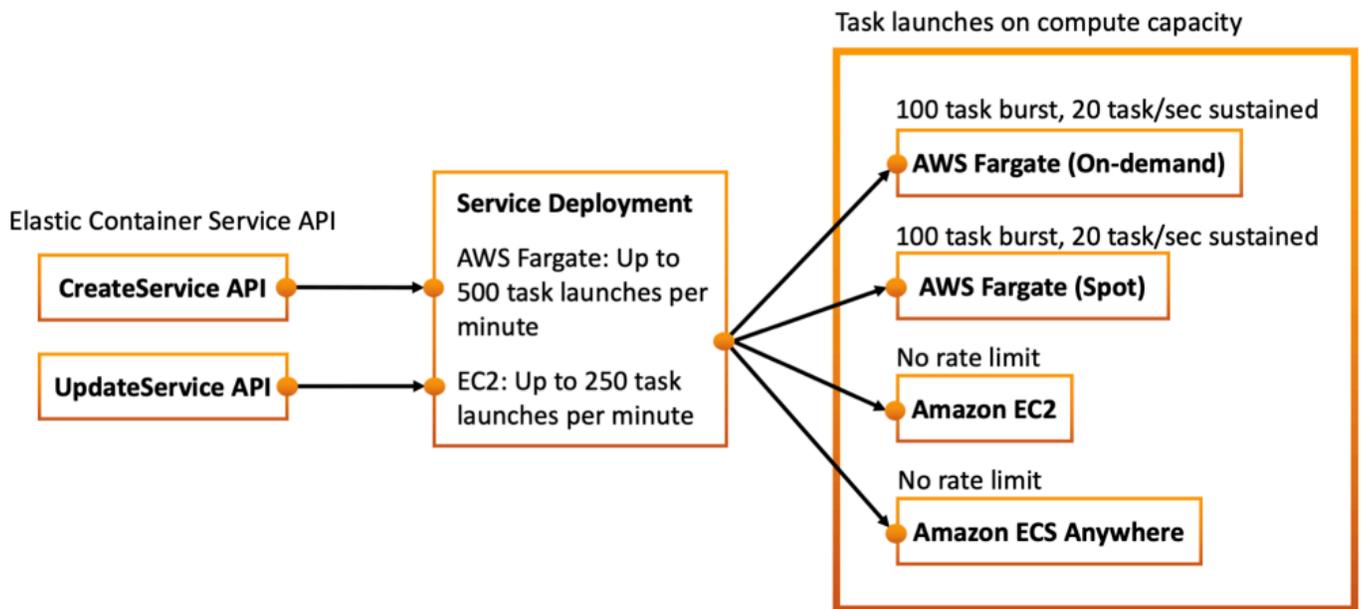
另一方面 , Lambda 针对短期、事件驱动的任务进行了优化。Lambda 函数在无状态环境中执行 , 其中最长执行时间上限为 15 分钟。这使得 Lambda 非常适合文件处理、实时数据流和 HTTP 请求处理等场景 , 在这些场景中 , 任务很简短 , 不需要长时间运行的进程。

在 Lambda 中，运行时环境更加抽象，对底层基础设施的控制也较少。Lambda 的无状态性质意味着每个函数调用都是独立的，需要在两次调用之间保留的任何状态或数据都必须由外部管理（例如，在数据库或存储服务中）。

## Scaling

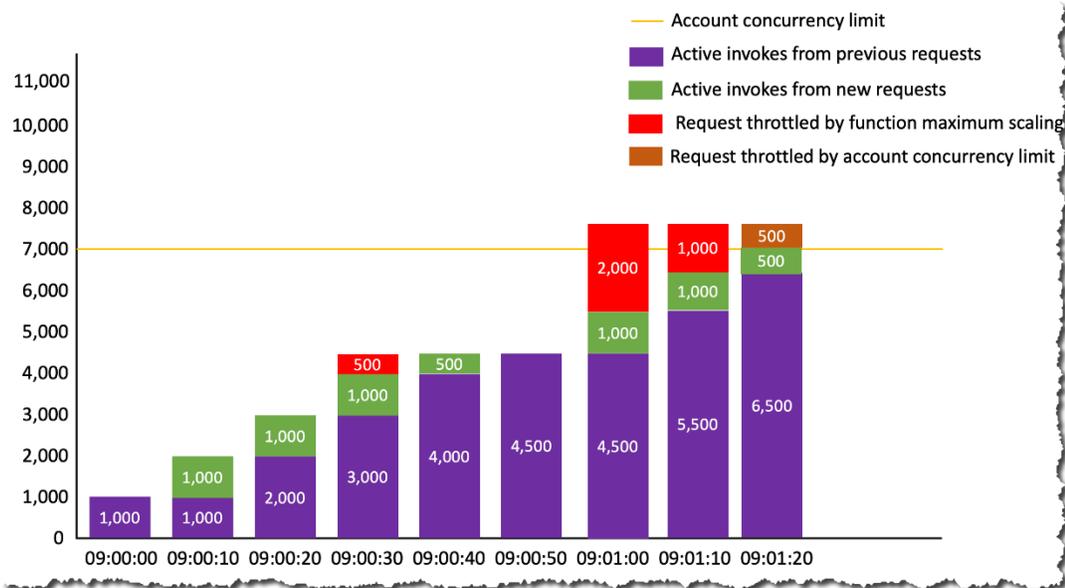
Fargate 通过根据容器编排服务 (Amazon ECS) 中定义的所需状态调整正在运行的容器的数量来进行扩展。这种扩展可以手动完成，也可以通过 Amazon A EC2 uto Scaling 自动完成。[这篇博文](#)提供了有关操作方法的更多详细信息。

在 Fargate 中，每个容器都在其隔离的环境中运行，扩展涉及启动其他容器或根据负载停止容器。对于网络和其他长期运行的服务，Amazon ECS 服务计划程序能够在不到一分钟的时间内为每项服务启动多达 500 个任务。



对于 Lambda，并发性是指您的 AWS Lambda 函数同时处理的正在处理的请求数。这与 Fargate 中的并发不同，在并发中，只要有可用的计算和网络资源，每个 Fargate 任务都可以处理并发请求。对于每个并发请求，Lambda 会预置单独的执行环境实例。当您的函数收到更多请求时，Lambda 会自动处理执行环境数量的扩展，直到您达到账户的并发限制。默认情况下，Lambda 为您的账户提供的所有函数并发执行 1,000 次的总并发限制 AWS 区域，如果需要，您可以请求增加配额。

对于区域中的每个 Lambda 函数，并发扩展速率为每 10 秒 1,000 个执行实例，不超过最大账户并发量。如[本博客中所述](#)，如果 10 秒钟内的请求数超过 1,000，则额外的请求将被限制。下图演示了假设账户并发度为 7000 的 Lambda 扩展是如何工作的。



## Cold start and cold-start mitigation

Lambda 可能会遇到冷启动，这种情况发生在函数空闲一段时间后被调用时。在冷启动期间，Lambda 服务需要初始化新的执行环境，包括加载运行时、依赖项和函数代码。此过程可能会带来延迟，特别是对于初始化时间较长的语言（例如 Java 或 C#）。冷启动可能会影响应用程序的性能，尤其是那些需要低延迟响应的应用程序。

为了缓解 Lambda 中的冷启动，可以采用多种策略：

- 最小化函数大小：减小函数包及其依赖关系的大小可以减少初始化所需的时间。
- 增加内存分配：较高的内存分配会增加 CPU 容量，从而有可能缩短初始化时间。
- 保持函数温度：定期调用 Lambda 函数（例如，CloudWatch 使用事件）可以使它们保持活动状态并降低冷启动的可能性。
- Lambda SnapStart：使用 SnapStart 适用于 Java 的 [Lambda](#) 函数来缩短启动时间。
- 预配置并发：此功能可使指定数量的函数实例保持温暖状态，随时可以处理请求，从而减少冷启动延迟。但是，这会增加成本，因为你要为预配置实例付费，即使它们没有积极处理请求。

Fargate 通常不会像 Lambda 那样受到冷启动的影响。启动 Fargate 任务所需的时间与从镜像注册表中[提取任务中定义的容器镜像](#)所花费的时间直接相关。Fargate 还支持延迟加载已使用 [Seekable OCI \(SOC\)](#) 索引的容器镜像。使用 SOC 延迟加载容器映像可以缩短在 Fargate 上启动 Amazon ECS 任务所需的时间。Fargate 运行的容器可以根据需要保持活动状态，这意味着它们随时准备处理请求。但是，如果您需要启动新容器以响应扩展事件，则容器初始化时可能会有一些延迟，但与 Lambda 冷启动相比，这通常不那么重要。

## Memory and CPU options

Fargate 可为您的容器化应用程序提供对内存和 CPU 资源的精细控制。在 Fargate 中启动任务时，可以根据应用程序的需求指定确切的 CPU 和内存要求。CPU 和内存分配是独立的，允许您选择最适合您的工作负载的组合。例如，根据您的配置，您可以选择介于 0.25 V CPUs 到 16 V 之间的 CPU 值 CPUs 以及每个容器 0.5 GB 到 120 GB 之间的内存。

这种灵活性非常适合运行需要特定性能特征的应用程序，例如内存密集型数据库或 CPU 密集型计算任务。Fargate 允许您优化资源分配，从而有效地平衡成本和性能。

在 Lambda 中，内存和 CPU 是相互关联的，CPU 会根据您选择的内存量按比例自动分配。您可以选择介于 128 MB 和 10 GB 之间的内存分配，以 1 MB 为增量。CPU 随内存而扩展，最多 6 个 vCPU，这意味着更高的内存设置会产生 CPU 功耗更高，但您无法直接控制 CPU 分配本身。

该模型专为简单性而设计，允许开发人员快速调整内存设置，而无需管理 CPU 配置。但是，对于需要在 CPU 和内存资源之间保持特定平衡的工作负载，它可能不那么灵活。Lambda 的模型适用于您希望根据内存需求进行直接扩展的任务，但对于具有复杂或高度特定资源需求的应用程序来说，它可能不是最佳选择。

## Networking

当您在 Fargate 中部署任务时，它们将在亚马逊 VPC ( 亚马逊虚拟私有云 ) 中运行，这样您就可以完全控制网络环境。这包括配置安全组、网络访问控制列表 (ACLs) 和路由表。每个 Fargate 任务都有自己的网络接口，带有专用的私有 IP 地址，如果需要，可以为其分配一个公有 IP 地址。

Fargate 支持高级网络功能，例如负载均衡 ( 使用 E AWS Elastic Load Balancing )、VPC 对等以及直接访问 VPC AWS 服务内的其他功能。您还可以使用 AWS PrivateLink 安全私密地连接到受支持者 AWS 服务，而无需穿越互联网。

默认情况下，Lambda 函数在托管网络环境中运行，不直接控制网络接口或 IP 地址。但是，您可以 AWS 使用 Hyperplane 将 Lambda 连接到客户管理的 VPC，这样您就可以控制对您的 VPC 内部资源的访问权限。

当 Lambda 函数附加到客户管理的 VPC 时，它们会继承该 VPC 的安全组和子网配置，从而允许它们与同一 VPC 中的 AWS 服务其他 ( 如 RDS 数据库 ) 安全地交互。

Lambda 服务使用网络功能虚拟化平台向客户提供 Lambda VPC 中的 NAT 功能。VPCs 这将在创建或更新 Lambda 函数时配置所需的弹性网络接口 (ENIs)。它还允许 ENIs 在多个执行环境之间共享您的账户，这使得 Lambda 能够在函数扩展时更有效地利用有限的网络资源。

由于资源 ENIs 会耗尽，而且 ENIs 每个区域的软限制为 250 个，因此，如果您要配置 Lambda 函数以访问 VPC，则应监控弹性网络接口的使用情况。同一可用区和同一个安全组中的 Lambda 函数

可以共享。ENIs通常，如果在 Lambda 中提升了并发限制，则您应评估是否需要增加弹性网络接口。如果达到限制，则这会导致启用 VPC 的 Lambda 函数的调用受到限制。

## Pricing model

Fargate 的定价基于分配给容器的资源，特别是您为每项任务选择的 vCPU 和内存。对于容器使用的 CPU 和内存，按秒计费，最低收费为一分钟。成本与您的应用程序消耗的资源直接相关，这意味着无论应用程序是否在积极处理请求，您都需要为预配置的资源付费。Fargate 非常适合需要特定资源配置的可预测工作负载，并且可以通过调整分配的资源来优化成本。此外，数据传输、存储和联网等相关服务（例如 VPC、Elastic Load Balancing）可能会收取额外费用。

Lambda 具有不同的定价结构，该结构是由事件驱动的，. pay-per-execution 根据您的函数收到的请求数量和每次执行的持续时间（以毫秒为单位）向您收费。Lambda 还会考虑您分配给函数的内存量，成本会根据使用的内存和执行时间进行调整。定价模式包括免费套餐，每月提供 100 万个免费请求和 400,000 GB 秒的计算时间，这使得 Lambda 对于小容量、零星的工作负载特别具有成本效益。

Lambda 定价模式非常适合流量模式不可预测或突发的应用程序，因为您只需为实际的函数调用和执行时间付费，无需为空闲容量预置或付费。

## 使用

现在，您已经阅读了在 AWS Fargate 和之间进行选择的标准 AWS Lambda，您可以选择满足您需求的服务，并使用以下信息来帮助您开始使用每种服务。

### AWS Fargate

- [学习如何为 Fargate 启动类型创建 Amazon ECS Linux 任务](#)

使用 Fargate 启动类型 AWS Fargate 来完成你的 Linux 任务，开始开启 Amazon ECS。

#### [浏览指南](#)

- [了解如何为 Fargate 启动类型创建 Amazon ECS Windows 任务](#)

使用 Fargate 启动类型执行你 AWS Fargate 的 Windows 任务，开始开启 Amazon ECS。

#### [浏览指南](#)

- [Fargate 和 Amazon EKS 入门](#)

本指南介绍如何开始 AWS Fargate 使用您的 Amazon EKS 集群运行您的 pod。

### [浏览指南](#)

- AWS Fargate 定价

使用本指南了解 vCPU、内存、存储和操作系统配置如何影响 AWS Fargate 定价。

### [浏览指南](#)

- AWS Fargate 经常问的问题

获取有关 AWS Fargate 功能的常见问题的答案以及实施的最佳实践。

### [浏览指南](#)

## AWS Lambda

- 创建无服务器文件处理应用程序

设置和使用 Amazon SNS 的 step-by-step 演练。它涵盖了创建主题、为终端节点订阅主题、发布消息和配置访问权限等主题。

### [浏览指南](#)

- 无服务器开发人员指南

本指南可帮助您更好地从概念上理解无服务器应用程序开发，以及如何将各种应用程序 AWS 服务组合在一起以创建构成云应用程序核心的应用程序模式。

### [浏览指南](#)

- 无服务器土地

该网站汇集了 AWS Serverless 的最新信息、博客、视频、代码和学习资源。学习使用和构建可在低成本、完全托管的无服务器架构上自动扩展的应用程序。

### [浏览该网站](#)

- AWS Lambda 定价

使用本指南估算费用并根据功能使用和配置优化成本。它包括一个定价计算器，用于在单个估算中计算您的成本 AWS Lambda 和架构成本。

### [浏览指南](#)

- AWS Lambda 经常问的问题

获取有关 AWS Lambda 功能的常见问题的答案以及实施的最佳实践。

[浏览指南](#)

# AWS Fargate 或 AWS Lambda ? 的文档历史记录

下表描述了本决策指南的重要更改。要获取有关本指南更新的通知，您可以订阅 RSS feed。

变更	说明	日期
<a href="#">初始版本</a>	决策指南的首次发布。	2024 年 11 月 15 日

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