AWS 白皮书

# 托管于 AWS 云中的 Web 应用程序

# 托管于 AWS 云中的 Web 应用程序: AWS 白皮书

# Table of Contents

# 托管于 AWS 云中的 Web 应用程序

发布日期：2021 年 8 月 20 日 (文档修订)

## 摘要

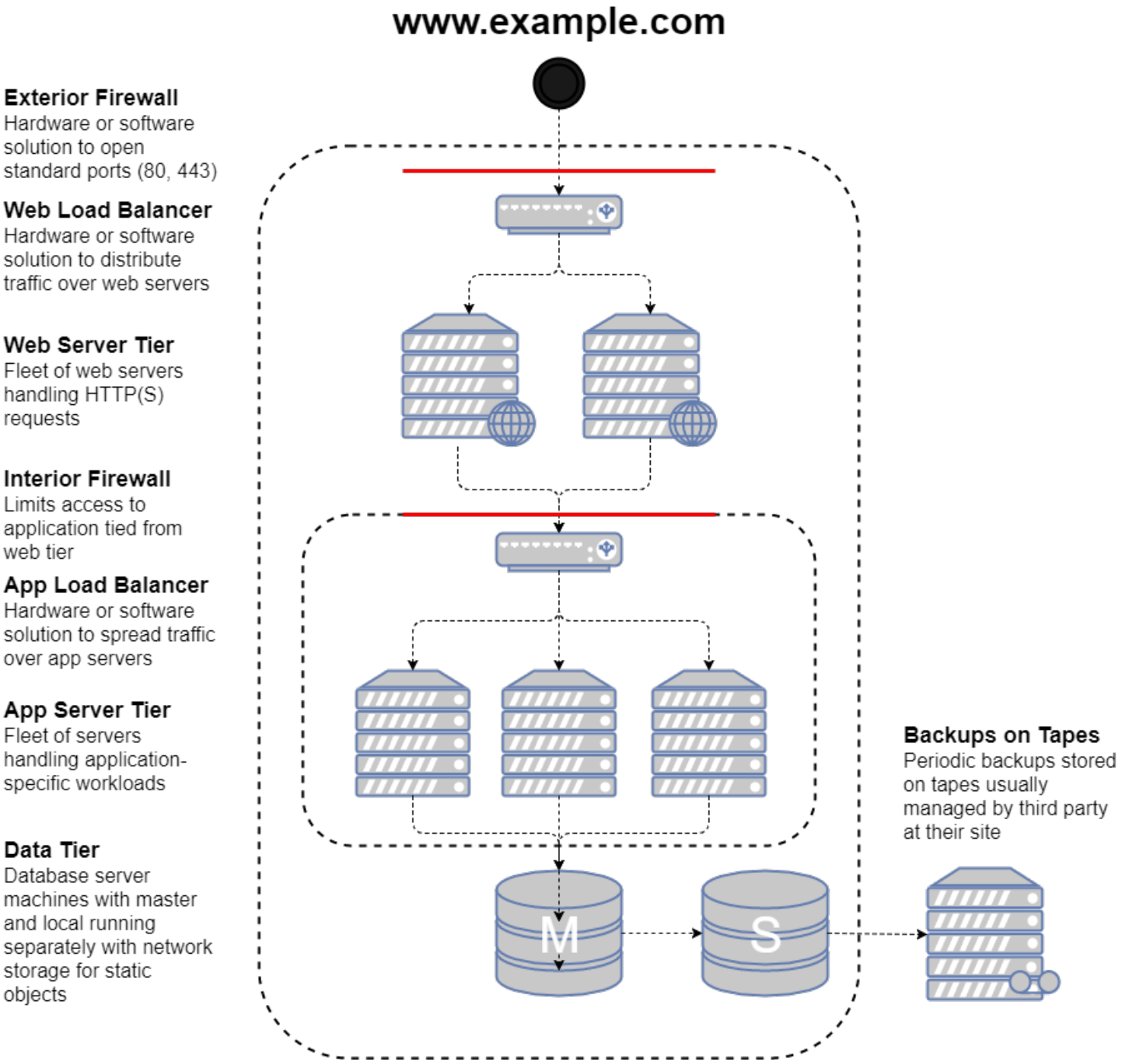
传统的本地部署 Web 架构需要复杂的解决方案和准确的预留容量预测，以确保可靠性。密集的流量高峰期和流量模式的急剧波动导致昂贵硬件的利用率较低。这样会导致维护闲置硬件的运营成本很高，而对于未充分利用的硬件，资金的使用效率也很低。

Amazon Web Services (AWS) 可为最严苛的 Web 应用程序提供可靠、可扩展、安全且高性能的基础设施。该基础设施可将 IT 成本与客户流量模式近乎实时地相匹配。

本白皮书面向希望了解如何在云中运行传统 Web 架构以实现弹性、可扩展性和可靠性的 IT 经理和系统架构师。

# 传统 Web 托管概述

可扩展的 Web 托管是一个众所周知的问题领域。下图描述了实施通用三层 Web 应用程序模型的传统 Web 托管架构。在此模型中，该架构分为表示层、应用程序层和持久性层。可扩展性是通过在这些层添加主机来实现的。另外该架构也包含内置的性能、故障转移和高可用性功能。传统的 Web 托管架构只需进行少量修改即可轻松移植到 AWS 云中。



## 传统 Web 托管架构

以下部分展示了为什么需要以及如何在 AWS 云中部署该架构。

# 使用 AWS 在云中托管 Web 应用程序

您应该问的第一个问题涉及到将经典 Web 应用程序托管解决方案迁移到 AWS 云中的价值。如果您确定云适合您，则您将需要一个合适的架构。本节将帮助您评估 AWS 云解决方案。它将在云中部署 Web 应用程序与在本地部署进行了比较、介绍了用于托管应用程序的 AWS 云架构并讨论了 AWS 云架构解决方案的关键组件。

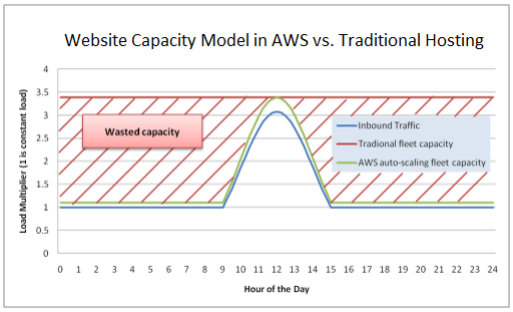
## AWS 如何解决常见 Web 应用程序托管问题

如果您负责运行 Web 应用程序，那么就必须面对各种基础设施和架构问题，而 AWS 可提供一套简单且经济实惠的无缝解决方案。以下列出了几项使用 AWS 的优势（与传统托管模式相比）。

### 用经济实惠的解决方案替换应对高峰流量所需要的庞大的机群

在传统托管模式中，您必须预置服务器来处理峰值容量。未使用的周期在峰值周期之外被浪费。AWS 托管的 Web 应用程序可以按需预置额外服务器，用户可以根据实际流量模式动态调整容量和成本。

例如，下图显示的 Web 应用程序的高峰期为每天上午 9 点到下午 3 点，而其他时间都处于使用不充分状态。自动扩展方法基于实际流量趋势，仅在需要时才预置资源，可使容量减少浪费，并将成本降低 50% 以上。




经典托管模式中的容量浪费示例

### 可处理意外流量高峰的可扩展解决方案

传统托管模式调配资源的速度慢，由此导致的一个严重后果是，无法及时对意外流量高峰作出响应。有很多这样的例子，由于网站在流行媒体中被提到而导致 Web 应用程序因为意外流量高峰而变得不可用。在 AWS 云中，帮助 Web 应用程序按需扩展以应对周期性流量高峰的这样一种按需分配的能力同样适用于意外负载。新主机可在几分钟内完成启动并投入使用，当流量恢复正常时，还可以迅速脱机。

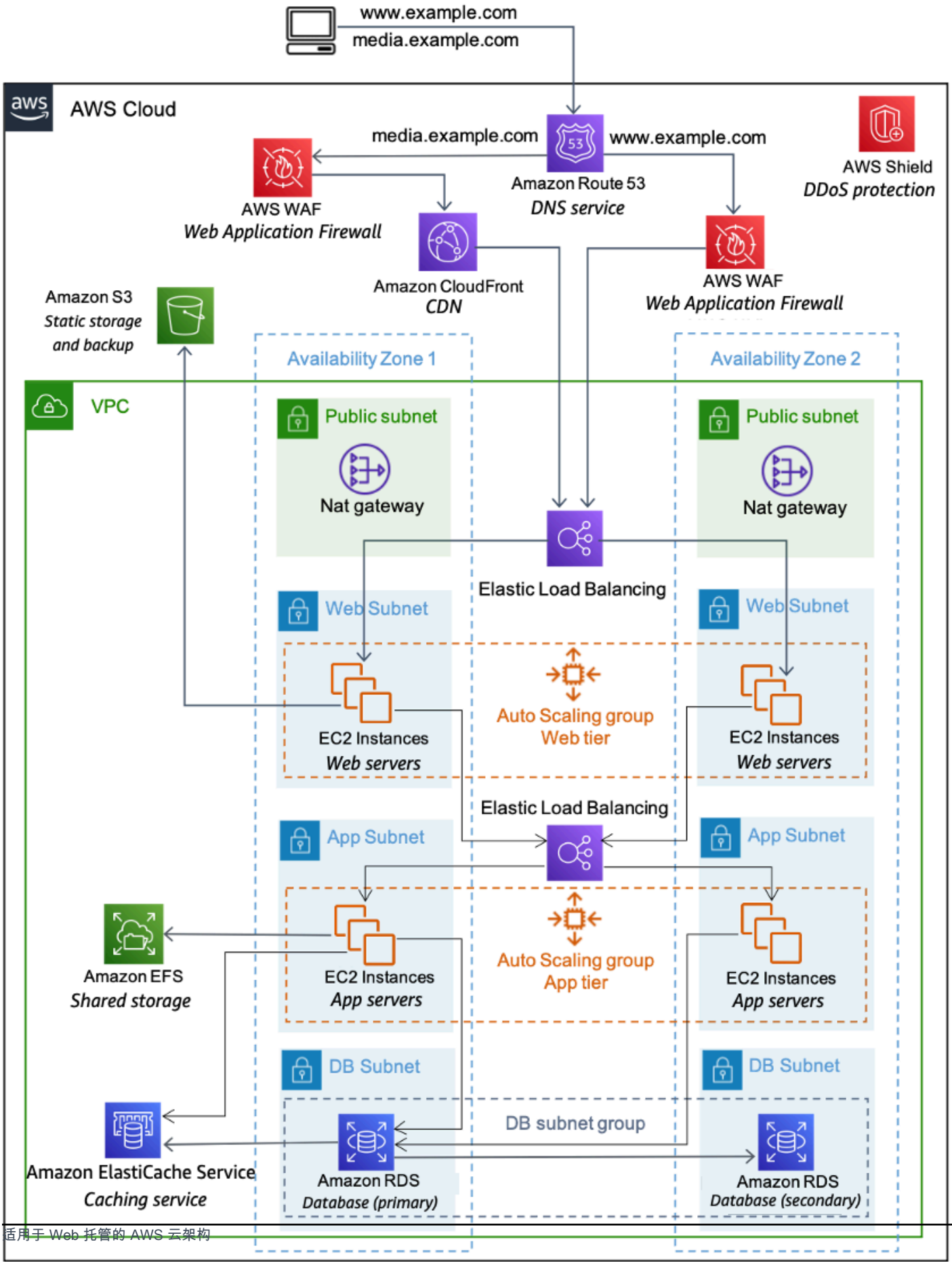
## 适用于测试、负载、测试版和生产前环境的按需解决方案

对于生产型 Web 应用程序而言，构建和维护传统托管环境的硬件成本不仅仅包括生产机群。通常，您需要创建预生产、测试版和测试机群，以确保 Web 应用程序在开发生命周期的每个阶段的质量。尽管可实施各种优化来尽可能提高此测试硬件的使用率，但这些并行机群的使用率并不会总是得到优化，许多昂贵的硬件长期处于闲置状态，得不到充分使用。

在 AWS 云中，可在需要时预置测试机群。这样不仅无需在实际使用前几天或几个月预置资源，而且还使您能够在不需要基础架构组件时灵活地将其删除。此外，您可以在负载测试期间模拟 AWS 云上的用户流量。您还可以将这些并行机群用作新生产版本的暂存环境。这样可以从当前的生产版本快速切换到新的应用程序版本，几乎没有服务中断。

# 适用于 Web 托管的 AWS 云架构

在下图中，让我们再看看经典 Web 应用程序架构，了解它如何能够利用 AWS 云计算基础设施。


www.example.com
media.example.com
AWS Cloud
media.example.com
www.example.com
AWS Shield
DDoS protection
AWS WAF
Web Application Firewall
Amazon Route 53
DNS service
Amazon CloudFront
CDN
AWS WAF
Web Application Firewall
Amazon S3
Static storage and backup
Availability Zone 1
Availability Zone 2
VPC
Public subnet
Nat gateway
Public subnet
Nat gateway
Elastic Load Balancing
Web Subnet
Web Subnet
EC2 Instances
Web servers
Auto Scaling group
Web tier
EC2 Instances
Web servers
Elastic Load Balancing
App Subnet
App Subnet
Amazon EFS
Shared storage
EC2 Instances
App servers
Auto Scaling group
App tier
EC2 Instances
App servers
DB Subnet
DB Subnet
DB subnet group
Amazon ElastiCache Service
Caching service
Amazon RDS
Database (primary)
Amazon RDS
Database (secondary)

AWS 上的 Web 托管架构示例

1. 使用 Amazon Route 53 的 DNS 服务 – 提供 DNS 服务以简化域管理。
2. 使用 Amazon CloudFront 进行边缘缓存 – 边缘缓存大量内容以减少对客户的延迟。
3. 使用 AWS WAF 为 Amazon CloudFront 提供边缘安全 – 通过客户定义的规则筛选恶意流量，包括跨站点脚本 (XSS) 和 SQL 注入。
4. 使用 Elastic Load Balancing (ELB) 实现负载均衡 – 使您能够将负载分散到多个可用区和 AWS Auto Scaling 组，以实现服务的冗余和解耦。
5. 使用 AWS Shield 实现 DDoS 防护 – 自动保护您的基础设施免受最常见的网络和传输层 DDoS 攻击。
6. 带安全组的防火墙 – 将安全性移至实例，以便为 Web 服务器和应用程序服务器提供有状态的主机级防火墙。
7. 使用 Amazon ElastiCache 进行缓存 – 通过 Redis 或 Memcached 提供缓存服务，以消除应用程序和数据库的负载，并降低频繁请求的延迟。
8. 使用 Amazon Relational Database Service (Amazon RDS) 托管数据库 – 创建具有六个可能的数据库引擎的高可用性、多可用区数据库架构。
9. 使用 Amazon Simple Storage Service (Amazon S3) 进行静态存储和备份 – 为备份和静态资产（如图像和视频）启用基于 HTTP 的简单对象存储。

# AWS Web 托管架构的关键组件

以下各节将概述 AWS Web 托管架构的一些关键组件以及它们与传统 Web 托管组件的区别。

## 网络管理

在 AWS 云中，能够将您的网络与其他客户的网络分割开来，从而实现更加安全、更加可扩展的架构。安全组可提供主机级安全性（请参阅主机安全部分），而 Amazon Virtual Private Cloud (Amazon VPC) 则使您能够在您定义的逻辑隔离的虚拟网络中启动资源。

Amazon VPC 是一项服务，可让您完全控制在 AWS 中设置的网络的详细信息。这种控制的示例包括为 Web 服务器创建面向公众的子网，以及为数据库创建没有 Internet 访问权限的私有子网。此外，借助 Amazon VPC，您还可以使用硬件虚拟专用网络 (VPN) 创建混合架构，并将 AWS 云用作您自己的数据中心的扩展。

除了为您的网络提供传统的 IPv4 支持之外，Amazon VPC 还包括 IPv6 支持。

## 内容分发

如果您的 Web 流量是地理上分散的，那么在全球复制您的整个基础设施并不总是可行，当然也不具有成本效益。借助 内容分发网络 (CDN)，您能够利用其边缘站点的全球网络向客户分发视频、网页、图像等 Web 内容的缓存副本。CDN 使用距离客户或原始请求位置最近的边缘站点，以便缩短响应时间。从缓存中提供 Web 资产可以大幅提高吞吐量。对于动态数据，许多 CDN 都可以配置为从源服务器中检索数据。

您可以使用 CloudFront，借助全球边缘站点网络交付您的网站，包括动态、静态和流媒体内容。CloudFront 会自动将您的内容请求路由到最近的边缘站点，从而以尽可能最佳的性能交付内容。CloudFront 经过优化，可与其他 AWS 服务配合使用，例如 Amazon S3 和 Amazon Elastic Compute Cloud (Amazon EC2)。CloudFront 也可与任何非 AWS 源服务器的源服务器无缝配合，这些服务器存储您的最新版本原始文件。

与其他 AWS 服务相同，使用 CloudFront 无须签订合同或购买月度额，您只需依照使用该服务交付的实际内容流量多少支付费用。

此外，Web 应用程序基础设施中的任何现有边缘缓存解决方案都应在 AWS 云中运行良好。

## 管理公有 DNS

将 Web 应用程序迁移到 AWS 云需要进行一些域名系统 (DNS) 更改。为了帮助您管理 DNS 路由，AWS 提供了 Amazon Route 53，它是一种高度可用且可扩展的云 DNS Web 服务。Route 53 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，把名称（如“www.example.com”）转换为计算机用于互相连接的数字 IP 地址（如 192.0.2.1），从而将最终用户路由到 Internet 应用程序。Route 53 也与 IPv6 完全兼容。

## 主机安全

除了在边缘进行入站网络流量筛选外，AWS 还建议 Web 应用程序在主机级别应用网络流量筛选。Amazon EC2 提供了一个名为安全组的功能。安全组类似于入站网络防火墙，让您能够指定允许访问 EC2 实例的网络协议、端口和以及源 IP 范围。

您可以为每个 EC2 实例分配一个或多个安全组。每个安全组均允许适当的流量进入每个实例。可以配置安全组，以便只有特定的子网、IP 地址和资源才能访问 EC2 实例。或者，它们可以引用其他安全组来限制对特定组中 EC2 实例的访问。

在图 3 中的 AWS Web 托管架构中，Web 服务器集群的安全组可能只允许从 Web 层负载均衡器进行访问，并且仅允许通过端口 80 和 443（HTTP 和 HTTPS）上的 TCP 进行访问。另一方面，应用程序服务器安全组可能只允许从应用程序层负载均衡器进行访问。在此模型中，您的支持工程师还需要访问

EC2 实例，这一点可以通过 AWS Systems Manager Session Manager 来实现。有关安全性的更深入讨论，请参阅 AWS 云安全性，其中包含安全公告、认证信息和解释 AWS 安全功能的安全白皮书。

## 跨集群的负载均衡

硬件负载均衡器是传统 Web 应用程序架构中常见的网络设备。AWS 通过 Elastic Load Balancing (ELB) 服务提供此功能。ELB 自动将传入的应用程序流量分配到多个目标，例如 EC2 实例、容器、IP 地址、AWS Lambda 函数和虚拟设备。它可以在单个可用区内处理不断变化的应用程序流量负载，也可以跨多个可用区处理此类负载。Elastic Load Balancing 提供四种负载均衡器，它们均能实现高可用性、自动扩展和可靠的安全性，因此能让您的应用程序获得容错能力。

## 查找其他主机和服务

在传统的 Web 托管架构中，大多数主机均具有静态 IP 地址。在 AWS 云中，您的大多数主机均具有动态 IP 地址。尽管每个 EC2 实例都拥有公有和私有 DNS 条目并可通过 Internet 进行寻址，但是 DNS 条目和 IP 地址是在实例启动时动态分配的。它们不能手动分配。静态 IP 地址（在 AWS 术语中指弹性 IP 地址）可在实例启动后分配给运行中的实例。所以对于需要一致的终端节点的服务和实例（例如主数据库、中央文件服务器和 EC2 托管的负载均衡器等），都应该分配弹性 IP 地址。

## 在 Web 应用程序内进行缓存

内存中的应用程序缓存可以通过缓存常用信息来降低服务负载并提高数据库层的性能和扩展性。Amazon ElastiCache 是一种让用户能够在云中轻松部署、操作和扩展内存缓存的 Web 服务。您可以将创建的内存中的缓存配置为随负载自动扩展并自动替换故障节点。ElastiCache 兼容 Memcached 和 Redis 协议，从而简化了当前本地解决方案的迁移。

## 数据库配置、备份和故障转移

许多 Web 应用程序均包含某种形式的持久性，通常表现为关系型或者非关系型数据库。AWS 同时提供关系型和非关系型数据库服务。或者，您可以在 EC2 实例上部署您自己的数据库软件。下表汇总了这些选项，将在本节对它们进行更详细的讨论。

表 1 — 关系型和非关系型数据库解决方案

| | 关系数据库解决方案 | NoSQL 解决方案 |
|---|---|---|
| 托管数据库服务 | Amazon RDS for MySQL、Oracle、SQL | Amazon DynamoDB、Amazon Keyspaces、Amazon |

| | 关系数据库解决方案 | NoSQL 解决方案 |
|---|---|---|
| | Server、MariaDB、PostgreSQL、Amazon Aurora | Neptune、Amazon QLDB、Amazon Timestream |
| 自行管理 | 在 Amazon EC2 实例上托管关系数据库管理系统 (DBMS) | 在 EC2 实例上托管非关系数据库解决方案 |

## Amazon RDS

Amazon Relational Database Service (Amazon RDS) 使您可以访问熟悉的 MySQL、PostgreSQL、Oracle 和 Microsoft SQL Server 数据库引擎的功能。您已经使用的代码、应用程序和工具可以在 Amazon RDS 上继续使用。Amazon RDS 可自动为数据库软件打补丁，自动备份您的数据库，并且仅保存用户定义的保存期内的备份数据。它还支持时间点恢复。只需进行一次 API 调用，即可灵活地扩展与关系数据库实例相关联的计算资源或存储容量，让您受益。

Amazon RDS 多可用区部署提高了数据库的可用性，确保数据库不会意外中断。Amazon RDS 读取副本提供了数据库的只读副本，这样对于频繁读取的数据库工作负载，您可以将容量扩展到一个数据库以上。与所有 AWS 服务一样，不需要前期投资，您只需为所使用的资源付费。

## 在 Amazon EC2 实例上托管关系数据库管理系统 (RDBMS)

除了托管的 Amazon RDS 产品之外，您还可以在 EC2 实例上选择安装 RDBMS（如 MySQL、Oracle、SQL Server 或 DB2）并自行管理。在 Amazon EC2 上托管数据库的 AWS 客户可成功使用各种主/备用和复制模型，包括只读副本镜像和使用日志传送实现的始终可用的被动从数据库实例。

在 Amazon EC2 上直接管理您自己的数据库软件时，您还应该考虑容错和持久性存储的可用性。为此，我们建议在 Amazon EC2 上运行的数据库采用 Amazon Elastic Block Store (Amazon EBS) 卷，它类似于网络挂载存储。

对于运行数据库的 EC2 实例，您应将所有数据库数据和日志放在 EBS 卷上。即使数据库主机出现故障，它们仍会保持可用。这样的配置可以实现简单的故障转移方案，在主机出现故障时，可以启动一台新的 EC2 实例，将现有 EBS 卷挂载到新的实例。数据库随后可从中断位置继续运行。

EBS 卷自动在可用区内提供冗余。如果一个 EBS 卷的性能无法满足数据库的需要，可以在几个 EBS 卷间进行捆绑以提高数据库的每秒输入/输出操作数 (IOPS) 性能。

对于过重的工作负载，您还可以使用 EBS 预置 IOPS，只需要在使用时指定所需的 IOPS。如果您使用 Amazon RDS，该服务会自动管理存储，您可以集中精力管理您的数据。

### 非关系数据库

除了支持关系数据库外，AWS 还提供了许多托管的非关系数据库：

- Amazon DynamoDB 是一种完全托管的 NoSQL 数据库服务，提供快速且可预测的性能，能够实现无缝扩展。使用 AWS Management Console 或 DynamoDB API，您可以扩大或缩小容量，而不会导致停机或性能下降。由于 DynamoDB 将运维和扩展分布式数据库的管理负担交给 AWS 处理，因此您无需担心硬件预置、安装和设置、数据复制、软件打补丁或集群扩展等问题。
- Amazon DocumentDB（兼容 MongoDB）是一种数据库服务，专为大规模 JSON 数据管理而打造，由 AWS 全面托管，并在 AWS 上运行，具有适合大型企业的高持久性。
- Amazon Keyspaces (for Apache Cassandra) 是一种可扩展、高度可用、托管的 Apache Cassandra 兼容数据库服务。借助 Amazon Keyspaces，您可以继续使用当前的相同 Cassandra 应用程序代码和开发工具，在 AWS 上运行 Cassandra 工作负载。
- Amazon Neptune 是一项快速、可靠且完全托管的图形数据库服务，可帮助您轻松构建和运行使用高度互连数据集的应用程序。Amazon Neptune 的核心是专门构建的高性能图形数据库引擎，它进行了优化以存储数十亿个关系并将图形查询延迟降低到毫秒级。
- Amazon Quantum Ledger Database (Amazon QLDB) (QLDB) 是一个完全托管的分类帐数据库，可提供由中央权威机构持有的透明、不可变且可加密验证的事务日志。QLDB 可用于跟踪每次的应用程序数据更改，并不断维护完整且可验证的更改历史记录。
- Amazon Timestream 是一种快速、可扩展的无服务器时间序列数据库服务，适用于物联网和运营应用程序，使用该服务每天可以轻松存储和分析数万亿个事件，速度提高了 1000 倍，而成本仅为关系数据库的十分之一。

此外，您可以使用 Amazon EC2 托管您可能正在使用的其他非关系数据库技术。

## 数据和资产的存储与备份

AWS 云中有许多选项可用于存储、访问和备份 Web 应用程序数据和资产。Amazon S3 提供了高度可用且冗余的对象存储。对于静态或变化缓慢的对象，如图像、视频和其他静态媒体，Amazon S3 是非常棒的存储解决方案。Amazon S3 还通过与 CloudFront 交互，支持对这些资产执行边缘缓存和流式传输。

对于类似于附加文件系统的存储，EC2 实例可以附加 EBS 卷。它们的作用类似于用于运行 EC2 实例的可装载磁盘。对于需要作为块存储进行访问且要求持久性超过运行实例生命周期的数据，如数据库分区和应用程序日志，Amazon EBS 是很好的选择。

除了拥有独立于 EC2 实例的生命周期外，您还可以拍摄 EBS 卷的快照并将其存储在 Amazon S3 中。由于 EBS 快照仅备份自上次快照以来的更改，因此更频繁的快照会缩短快照拍摄时间。您还可以将 EBS 快照用作跨多个 EBS 卷复制数据的基线并将这些卷挂载到其他运行实例。

EBS 卷最大可达 16TB，可将多个 EBS 卷捆绑在一起来获得更大的卷或提升输入/输出 (I/O) 性能。为了最大程度发挥 I/O 密集型应用程序的性能，您可以使用预置 IOPS 卷。预置 IOPS 卷为满足 I/O 密集型工作负载而设计，尤其是数据库工作负载，此类工作负载对随机存取 I/O 吞吐量的存储性能和一致性较敏感。

您可以在创建卷时指定 IOPS 速率，之后 Amazon EBS 会在使用该卷使用过程中预置该速率。Amazon EBS 目前支持每卷 IOPS 数，范围从最大 16,000（适用于所有实例类型）到 64,000（适用于构建在 Nitro 系统上的实例）。您可将多个卷条带化，共同为您的应用程序的每个实例提供数千 IOPS。除此之外，对于吞吐量更高和需要亚毫秒级延迟的任务关键型工作负载，您可以使用 io2 Block Express 卷类型，该卷类型可支持高达 256,000 IOPS，最大存储容量为 64TB。

## 自动扩展机群

AWS 云架构与传统托管模型之间的一个重要区别在于，AWS 能够按需自动扩展 Web 应用程序机群来应对流量变化。在传统托管模式中，一般使用流量预测模式，根据预测流量提前调配主机。在 AWS 中，可以根据一组触发条件来动态预置实例，以增加或减少机群。

Auto Scaling 服务可用于创建按需伸缩的服务器容量组。也可以直接将 Auto Scaling 与 CloudWatch 配合使用以用于监控指标数据，以及与 Elastic Load Balancing 配合使用以添加和删除用于负载分发的主机。例如，如果 Web 服务器报告在一段时间里 CPU 的使用率超过 80%，系统会迅速再部署一个 Web 服务器，然后自动将其添加到负载均衡器中，以便立即将其纳入负载均衡循环。

如在 AWS Web 托管架构模型中所示，可针对架构的不同层级创建多个Auto Scaling 组，以支持各个层级独立进行扩展。例如，Web 服务器 Auto Scaling 组可以根据网络 I/O 量的波动触发扩展和收缩，而应用程序服务器 Auto Scaling 组可能会根据 CPU 使用率进行扩展和收缩。您可以设置下限和上限，以帮助确保全天候可用性以及最高用量。

可以设置 Auto Scaling 触发条件来增加和缩减某层级的机群总量，从而使资源使用率与实际流量需要相匹配。除了 Auto Scaling 服务外，还可以直接通过 Amazon EC2 API 扩展 Amazon EC2 机群，Amazon EC2 API 可用于启动、终止或检查实例。

## 其他安全功能

分布式拒绝服务 (DDoS) 攻击的数量和复杂程度都在上升。从传统上讲，这些攻击很难抵御。它们通常会导致在缓解时间和能耗方面付出高昂的代价，以及在攻击期间失去对您网站进行访问的机会成本。

有许多 AWS 因素和服务可以帮助您抵御此类攻击。其中之一是 AWS 网络的规模。AWS 基础设施非常庞大，使您能够利用我们的规模来优化防御。一些服务（包括 Elastic Load Balancing、Amazon CloudFront 和 Amazon Route 53）可以有效地扩展您的 Web 应用程序，以应对流量的大幅增加。

基础设施保护服务尤其有助于您制定防御策略：

- AWS Shield 是一项托管 DDoS 防护服务，有助于抵御各种形式的 DDoS 攻击途径。AWS Shield 的标准产品是免费的，可在整个账户中自动激活。此标准产品有助于抵御最常见的网络和传输层攻击。除此级别外，高级产品还可让您近乎实时地了解正在进行的攻击，并在更高级别与前面提到的服务进行集成，从而针对您的 Web 应用程序提供更高级别的保护。此外，您还可以联系 AWS DDoS 响应团队 (DRT)，以帮助缓解针对您资源的大规模复杂攻击。
- AWS WAF（Web 应用程序防火墙）旨在保护您的 Web 应用程序免受可能危及可用性或安全性或以其他方式消耗过多资源的攻击。AWS WAF 与 CloudFront 或 Application Load Balancer 以及您的自定义规则保持一致，以抵御跨站点脚本、SQL 注入和 DDoS 等攻击。与大多数 AWS 服务一样，AWS WAF 附带了功能齐全的 API，可帮助您在安全需求发生变化时自动创建和编辑 AWS WAF 实例的规则。
- AWS Firewall Manager 是一项安全管理服务，可让您跨 AWS Organizations 中的账户和应用程序集中配置和管理防火墙规则。在创建新应用程序时，您可以借助 AWS Firewall Manager 实施一套通用的安全规则，轻松地让新应用程序和资源从一开始就达到合规要求。

## 通过 AWS 实现故障转移

与传统 Web 托管相比，使用 AWS 的另一个关键优势是，可用区可让您方便地访问冗余部署位置。可用区设计为可与其他可用区的故障完全隔离的物理上的不同位置。它们向同一 AWS 区域中的其他可用区提供低成本、低延迟的网络连接。正如 AWS Web 托管架构图所示，AWS 建议您跨多个可用区部署 EC2 主机，以保证 Web 应用程序具有更大的容错能力。

要确保在发生故障时可以跨可用区迁移单一访问点，这非常重要。例如，您应该在第二可用区中设置备用数据库，以确保即使在意外故障时，数据的持久性是仍然是一致且高度可用的。只需单击一个按钮，即可在 Amazon EC2 或 Amazon RDS 上执行此操作。

尽管将现有 Web 应用程序迁移到 AWS 云时，通常必须对架构进行一些修改，但是迁移后应用程序的可扩展性、可靠性和成本效益都将得到大幅提升，因此，使用 AWS 云完全值得这些代价。下一节将讨论这些改进。

# 将 AWS 用于 Web 托管时要考虑的关键因素

AWS 云和传统 Web 应用程序托管模型相比有一些重要区别。以上部分突出强调了将 Web 应用程序部署到云中时需要考虑的众多因素。本部分将指出将应用程序迁移到云中时，需要对架构执行的一些重要调整。

## 不再需要实体网络设备

您不能在 AWS 中部署物理网络设备。例如，AWS 应用程序的防火墙、路由器和负载均衡器不再驻留在物理设备上，而是必须以软件解决方案取代。无论是负载均衡还是建立 VPN 连接，都有各种各样的企业级软件解决方案。这不是限制可以在 AWS 云上运行的内容，而是如果您现在就使用这些设备，应用程序的架构会发生改变。

## 防火墙无处不在

以前在传统托管模型中，您拥有一个简单的非管制区域 (DMZ)，然后开启主机之间的通信，现在，AWS 强制实施了更加安全的模式，在这种模式中每个主机均被锁定。在计划 AWS 部署时，其中一个步骤是分析主机之间的流量。该分析将会确定究竟需要打开哪些端口。您可以为架构中每种类型的主机创建安全组。您还可以创建各种简单的分层安全模型，以便在架构中的主机间提供最低访问权。在 Amazon VPC 中使用网络访问控制列表有助于在子网级别锁定您的网络。

## 考虑多个数据中心的可用性

将 AWS 区域内的可用区视为多个数据中心。不同可用区的 EC2 实例在逻辑上和物理上是分离的，从而为跨数据中心部署应用程序提供了简单易用的模型，以帮助应用获取高可用性和可靠性。Amazon VPC 作为区域服务使您能够利用可用区，同时将所有资源保存在同一个逻辑网络中。

## 将主机当作临时、动态的存在

在构建 AWS 应用程序架构时，最重要的一个转变就是 EC2 主机应该是临时的、动态的。任何面向 AWS 云构建的应用程序都不应假定主机始终可用，而应在设计时充分考虑到，如果 EC2 实例发生故障，EC2 即时存储中的任何数据都将丢失。

启动新主机时，不应假设主机可用区内的 IP 地址或位置。所以配置模型必须灵活，引导启动主机的方法也必须充分考虑云的动态特性。这些技术对于构建和运行高度可扩展、容错能力强的应用程序至关重要。

# 考虑容器和无服务器

本白皮书主要关注更传统的 Web 架构。但是，请考虑通过迁移到容器和无服务器技术来实现 Web 应用程序的现代化，利用 AWS Fargate 和 AWS Lambda 等服务使您能够抽象使用虚拟机来执行计算任务。由于无服务器计算以及容量预置和修补等基础设施管理任务可由 AWS 处理，您就可以构建更敏捷的应用程序，从而更快地进行创新和响应变化。

# 考虑自动部署

- Amazon Lightsail 是一款易于使用的虚拟专用服务器 (VPS)，它为您提供构建应用程序或网站所需的一切，并提供经济高效的每月计划。Lightsail 是简化工作负载、加快部署和开始体验 AWS 的理想选择。它旨在帮助您从小规模开始，然后随着您的发展而扩展。
- AWS Elastic Beanstalk 是一项易于使用的服务，用于在熟悉的服务器（例如 Apache、NGINX、Passenger 和 IIS）上部署和扩展使用 Java、.NET、PHP、Node.js、Python、Ruby、GO 和 Docker 开发的 Web 应用程序和服务。您只需上传代码，Elastic Beanstalk 即可自动处理部署、容量预置、负载均衡、自动扩展和应用程序运行状况监控工作。同时，您能够完全控制为应用程序提供支持的 AWS 资源，并可以随时访问底层资源。
- AWS App Runner 是一项完全托管式服务，使开发人员能够轻松、快速、大规模地部署容器化 Web 应用程序和 API，而无需事先具备基础设施方面的经验。从源代码或容器映像开始。App Runner 自动构建和部署 Web 应用程序，并通过加密实现流量的负载均衡。App Runner 还能自动扩大或缩小规模以满足您的流量需求。
- AWS Amplify 是一组既可组合使用也可单独使用的工具和服务，能够帮助前端 Web 和移动开发人员构建由 AWS 提供支持的可扩展全栈式应用程序。借助 Amplify，您可以在几分钟内配置应用程序后端并连接应用程序，单击几下即可部署静态 Web 应用程序，并在 AWS Management Console 外轻松管理应用程序内容。

# 总结与贡献者

## 总结

在考虑将 Web 应用程序迁移到 AWS 云时，有许多架构和概念上的考虑。拥有一个经济高效、高度可扩展、容错能力强且可随业务一起增长的基础设施所带来的优势远远超过迁移到 AWS 云所付出的努力。

## 贡献者

以下是对本文做出贡献的个人和组织：

- AMIR Khairalomoum，AWS 高级解决方案架构师
- Dinesh Subramani，AWS 高级解决方案架构师
- Jack Hemion，AWS 高级解决方案架构师
- Jatin Joshi，AWS 云支持工程师
- Jorge Fonseca，AWS 高级解决方案架构师
- Shinduri K S，AWS 解决方案架构师

# 延伸阅读

- 将基于 Django 的应用程序部署到 Amazon LightSail
- 将高可用性 Drupal 网站部署到 Elastic Beanstalk
- 将高可用性 PHP 应用程序部署到 Elastic Beanstalk
- 将带 DynamoDB 的 Node.js 应用程序部署到 Elastic Beanstalk
- 在 AWS 云中开始使用 Linux Web 应用程序
- 托管静态网站
- 使用 Amazon S3 托管静态网站
- 教程：使用 Elastic Beanstalk 部署 ASP.NET Core 应用程序
- 教程：如何使用 Elastic Beanstalk 部署 .NET 示例应用程序

# 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 |
| --- | --- | --- |
| 已更新白皮书 | 使用新服务、功能和已更新服务限制更新了多个部分和图表。 | 2021 年 8 月 20 日 |
| 已更新白皮书 | 已更新图 3 中“使用 ElastiCache 缓存”的图标标签。 | 2019 年 9 月 29 日 |
| 已更新白皮书 | 为新服务添加和更新了多个部分。已更新图表以增加清晰度和服务水平。在“网络管理”中添加了 VPC 作为 AWS 中的标准联网方法。 在“附加安全功能”中添加了有关 DDoS 防护和缓解的部分。 添加了有关用于 Web 托管的无服务器架构的一小部分。 | 2017 年 7 月 1 日 |
| 已更新白皮书 | 已更新多个部分来增加清晰度。已更新图表以使用 AWS 图标。添加了“管理公有 DNS”部分来详细说明 Amazon Route 53。为清晰起见，已更新“查找其他主机和服务”部分。为了清晰性和 DynamoDB，已更新“数据库配置、备份和故障转移”部分。已扩展“数据和资产的存储与备份”部分以涵盖 EBS 预置 IOPS 卷。 | 2012 年 9 月 1 日 |
| 初次发布 | 已发布白皮书。 | 2010 年 5 月 1 日 |

# 声明

本文档仅用于参考。本文档代表截至其发行之日的 AWS 的最新产品服务和实践，如有变更，恕不另行通知。客户负责对此文件的信息以及对 AWS 的产品或服务的任何使用进行自我独立的评估，每项产品或服务均按“原样”提供，无任何类型的保证，不管是明示还是暗示。本文档不形成 AWS、其附属公司、供应商或许可方的任何保证、表示、合同承诺、条件或担保。AWS 对其客户承担的责任和义务受 AWS 协议制约，本文档不是 AWS 与客户直接的协议的一部分，也不构成对该协议的修改。