

AWS 白皮书

# AWS Outposts 高可用性设计和架构注意事项



# AWS Outposts 高可用性设计和架构注意事项: AWS 白皮书

Copyright © 2024 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

摘要和简介 .....	i
您使用 Well-Architected 了吗？ .....	1
简介 .....	1
将 AWS 基础设施和服务扩展到本地位置 .....	2
了解更新后的责任共担模型 .....	4
从故障模式的角度进行思考 .....	7
故障模式 1：网络 .....	7
故障模式 2：实例 .....	7
故障模式 3：计算 .....	8
故障模式 4：机架或数据中心 .....	8
故障模式 5：AWS 可用区或区域 .....	8
使用 AWS Outposts 机架构建 HA 应用程序和基础设施解决方案 .....	9
联网 .....	10
网络连接 .....	10
锚连接 .....	14
应用程序/工作负载路由 .....	16
计算 .....	19
容量规划 .....	20
容量管理 .....	23
实例置放 .....	24
存储 .....	27
数据保护 .....	27
更大规模的故障模式 .....	29
结论 .....	32
贡献者 .....	33
文档历史记录 .....	34
版权声明 .....	35
AWS 词汇表 .....	36
.....	xxxvii

# AWS Outposts 高可用性设计和架构注意事项

发布日期：2021 年 8 月 12 日 ( [文档历史记录](#) )

本白皮书讨论了 IT 经理和系统架构师在构建高度可用的本地应用程序环境时可以采用的架构注意事项和推荐实践。AWS Outposts

## 您的架构是否良好？

当您在云端构建系统时，[AWS Well-Architected Framework](#) 可助您了解所作决策的利弊。利用此框架的六个支柱，您可以了解到设计和运行可靠、安全、高效、经济有效且可持续的系统的架构最佳实践。您可以使用 [AWS Management Console](#) 免费提供的 [AWS Well-Architected Tool](#)，回答与每个支柱相关的一组问题，即可根据这些最佳实践检查自己的工作负载。

有关云架构的更多专家指导和最佳实践 ( 参考架构部署、图表和白皮书 )，请参阅 [AWS 架构中心](#)。

## 简介

本 paper 适用于希望使用 AWS 云平台部署、迁移和操作应用程序，并在本地使用机架 ( 42U [AWS Outposts 机架](#) 外形规格为 ) 运行这些应用程序的 IT 经理和系统架构师。[AWS Outposts](#)

它介绍了架构模式、反模式以及构建包括 AWS Outposts 机架在内的高可用性系统的推荐做法。您将学习如何管理 AWS Outposts 机架容量，以及如何使用网络和数据中心设施服务来设置高度可用的 AWS Outposts 机架基础设施解决方案。

AWS Outposts rack 是一项完全托管的服务，可提供云计算、存储和网络功能的逻辑池。借助 Outpost 机架，客户可以在本地环境中使用支持的 AWS 托管式服务，包括：[Amazon Elastic Compute Cloud](#) (Amazon EC2)、[Amazon Elastic Block Store](#) (Amazon EBS)、[Amazon S3 on Outposts](#)、[Amazon Elastic Kubernetes Service](#) (Amazon EKS)、[Amazon Elastic Container Service](#) (Amazon ECS)、[Amazon Relational Database Service](#) (Amazon RDS) 以及 Outpost 上的其他 [AWS 服务](#)。Outpost 上的服务将在 AWS 区域内的相同 [AWS Nitro 系统](#) 上提供。

通过利用 AWS Outposts 机架，您可以使用熟悉的 AWS 云服务和工具构建、管理和扩展高度可用的本地应用程序。AWS Outposts rack 非常适合需要低延迟访问本地系统、本地数据处理、数据驻留以及迁移具有本地系统相互依赖关系的应用程序的工作负载。

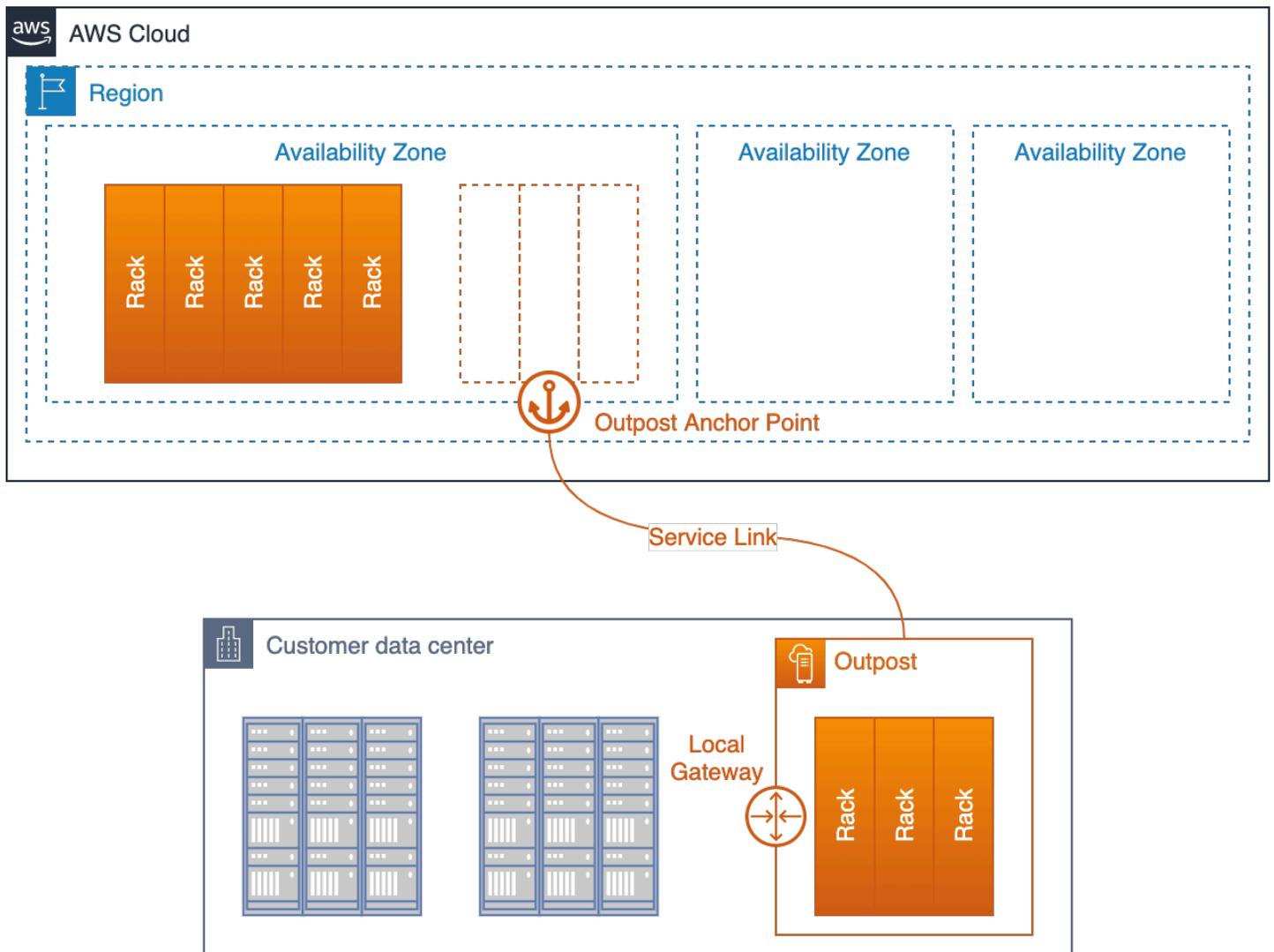
## 将 AWS 基础设施和服务扩展到本地位置

该 AWS Outposts 服务为 [50多个国家和地区的本地位置提供 AWS 基础设施和服务](#)，使客户能够将相同的 AWS 基础架构、AWS 服务、API和工具部署到几乎任何数据中心、托管空间或本地设施，以获得真正一致的混合体验。要了解如何使用 Outposts 进行设计，你应该了解构成云的不同等级。AWS

[AWS 区域](#)是指世界上的一个地理区域。每个 AWS 区域 数据中心都是按逻辑分组为 [可用区 \(AZ\)](#) 的数据中心的集合。AWS 区域 提供多个（至少两个）物理分离和隔离的可用区，这些可用区通过低延迟、高吞吐量和冗余网络连接相连。每个 AZ 包含一个或多个物理数据中心。

逻辑 [前哨](#)（以下简称 Outpost）是部署一个或多个物理连接的 AWS Outposts 机架，作为单个实体进行管理。Outpost 在您的一个站点上提供 AWS 计算和存储容量池，作为 AWS 区域中可用区的私有扩展。

也许最好的概念模型 AWS Outposts 是考虑从AZ中的数据中心拔掉一个或多个机架的电源。AWS 区域将这些机架从 AZ 数据中心部署到您的数据中心。然后，使用一根（特别）长的电缆将机架插入 AZ 数据中心中的锚点，以使机架能够作为 AWS 区域的一部分继续工作。您还可以将这些机架插入本地网络，以便在本地网络和这些机架上运行的工作负载之间提供低延迟连接。



部署在客户数据中心内并与其锚 AZ 和父级区域相连的 Outpost

前哨基地充当其停泊的 AZ 的延伸。AWS 作为其中的一部分运营、监控和管理 AWS Outposts 基础架构 AWS 区域。Outpost 通过一组名为服务链接的加密 VPN 隧道（而非一条特别长的物理电缆）连接回其父级区域。

服务链接的终点位于 Outpost 父级区域的可用区（AZ）中的一组锚点上。

您可以选择内容的存储位置。您可以将内容复制并备份到 AWS 区域 或其他位置。未经您的同意，您的内容不会被转移或复制到您所选择的位置以外，除非为了遵守法律或政府机构具有约束力的命令必须如此。有关更多信息，请参阅 [AWS 数据隐私 FAQ](#)。

您在这些机架上部署的工作负载在本地运行。而且，虽然这些机架中可用的计算和存储容量是有限的，无法容纳运行的云规模服务 AWS 区域，但部署在机架上的资源（您的实例及其本地存储）可以享受本地运行的好处，同时管理平面继续在中运行。AWS 区域

要在 Outpost 上部署工作负载，您需要向虚拟私有云 ( VPC ) 环境中添加子网，并指定 Outpost 作为子网的位置。然后，在通过 CLI、API AWS Management Console、CDK 或基础设施即代码 (IaC) 工具部署支持的 AWS 资源时，您可以选择所需的子网。Outpost 子网中的实例通过 VPC 联网与 Outpost 或区域中的其他实例通信。

Outpost 服务链接可同时传输 Outpost 管理流量和客户 VPC 流量 ( Outpost 上的子网与区域中的子网之间的 VPC 流量 )。

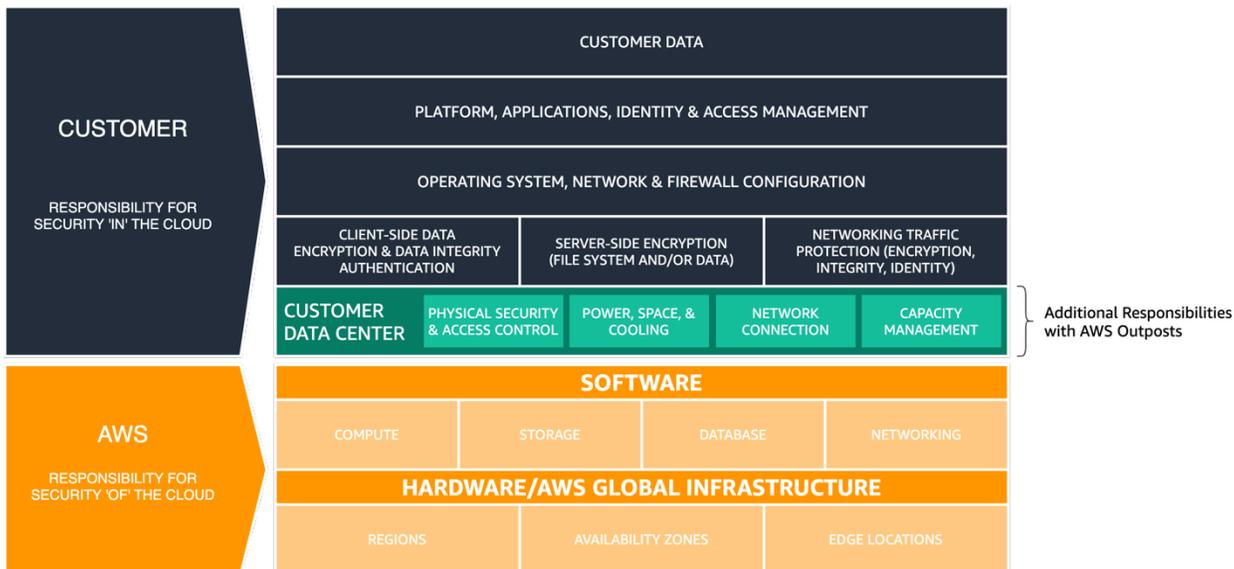
### 重要术语：

- AWS Outposts— 是一项完全托管的服务，可为几乎任何数据中心、托管空间或本地设施提供相同的 AWS 基础架构、AWS 服务、API 和工具，以实现真正一致的混合体验。
- Outpost — 是一个或多个物理连接的 AWS Outposts 机架的部署，这些机架作为单个逻辑实体进行管理，AWS 计算、存储和网络池部署在客户现场。
- 父区域 — 为 Outpost 部署提供管理、控制平面 AWS 服务和区域服务。AWS 区域
- 锚可用区 ( 锚 AZ )：父级区域中的可用区，用于托管 Outpost 锚点。Outpost 作为其锚可用区的扩展运行。
- 锚点：锚 AZ 中的端点，用于接收来自远程部署的 Outpost 的连接。
- 服务链接：一组加密的 VPN 隧道，用于将 Outpost 连接到其父级区域的锚可用区。
- 本地网关 ( LGW )：一种逻辑互连虚拟路由器，用于实现 Outpost 与本地网络之间的通信。

## 了解更新后的责任共担模型

当您将 AWS Outposts 基础设施部署到数据中心或主机托管设施时，您将在责任[AWS 共担模式中承担额外的责任](#)。例如，在区域中，AWS 提供多种电源、冗余核心网络和弹性广域网 ( WAN ) 连接，以确保在一个或多个组件出现故障时提供服务。

使用 Outpost 时，您负责为 Outpost 机架提供弹性电源和网络连接，以满足 Outpost 上运行的工作负载的可用性需求。



## AWS 已更新分担责任模型 AWS Outposts

使用 AWS Outposts，您负责数据中心环境的物理安全和访问控制。您必须提供足够的电源、空间和冷却能力以保持 Outpost 正常运行，并提供网络连接以使 Outpost 连接回区域。

由于 Outpost 的容量是有限的，由您站点 AWS 安装的机架大小和数量决定，因此您必须决定需要多少 EC2、EBS 和 S3 on Outposts 容量来运行初始工作负载，适应未来的增长，并提供额外的容量来缓解服务器故障和维护事件。

AWS 负责 Outposts 基础设施的可用性，包括机架内的电源、服务器和网络设备。AWS Outposts 还管理在 Outposts 上运行的虚拟化虚拟机管理程序、存储系统和 AWS 服务。

每个 Outpost 机架中的中央电源架可以将交流电转换为直流电，并通过汇流排架构为机架中的服务器供电。借助汇流排架构，即使机架中有一半的电源出现故障，所有服务器仍将继续不间断地运行。



图 3- AWS Outposts 交流-直流电源和母线电源分布

Outpost 机架内和机架之间的网络交换和布线也采用完全冗余的配置。光纤配线架提供 Outpost 机架和本地网络之间的连接，并充当客户管理的数据中心环境与托管环境之间的分界点。AWS Outposts

就像在该地区一样，负责Outpost AWS s上提供的云服务，并在您选择和部署更高级别的托管服务（例如Outposts上的Amazon RDS）时承担额外的责任。在考虑和选择要在 Outpost 上部署的服务时，应查看 [AWS 责任共担模型](#) 和针对不同服务的常见问题 (FAQ) 页面。这些资源提供了有关您与之间的责任划分的更多详细信息 AWS。

# 从故障模式的角度进行思考

在设计高度可用的应用程序或系统时，必须考虑到哪些组件可能会出现故障，组件故障会对系统产生哪些影响，以及可以实施哪些机制来缓解或消除组件故障的影响。应用程序是在单台服务器上、单个机架中还是单个数据中心中运行？当服务器、机架或数据中心出现临时或永久故障时会发生什么？当网络等关键子系统或应用程序本身出现故障时会发生什么？这些就是故障模式。

在规划 Outpost 和应用程序部署时，应考虑本部分所述的故障模式。以下各部分将介绍如何缓解这些故障模式，从而为应用程序环境提供更高级别的可用性。

## 故障模式 1：网络

Outpost 部署的管理和监控依赖于与其父级区域的弹性连接。网络中断可能是由各种故障引起的，例如操作员错误、设备故障和服务提供商的服务中断。Outpost 可能包含在站点连接在一起的一个或多个机架，如果其无法通过服务链接与区域通信，则被视为已断开连接。

冗余网络路径可以帮助降低发生断开连接事件的风险。您应该映射应用程序依赖关系和网络流量，以了解断开连接事件对工作负载操作的影响。规划足够的网络冗余以满足应用程序可用性需求。

在断开连接事件期间，在 Outpost 上运行的实例将继续运行，并且可通过 Outpost 本地网关 ( LGW ) 从本地网络进行访问。如果本地工作负载和服务依赖于区域的服务，则可能会受损或出现故障。当 Outpost 与区域断开连接时，变更请求 ( 例如启动或停止前哨基地上的实例 )、控制平面操作和服务遥测 ( 例如 CloudWatch 指标 ) 将失败。

## 故障模式 2：实例

如果支持 EC2 实例运行的服务器出现问题，或者实例遇到操作系统或应用程序故障，则这些实例可能会受损或失败。应用程序处理这些类型故障的方式取决于应用程序架构。单体应用程序通常使用应用程序或系统功能进行恢复，而以服务为导向的模块化架构或微服务架构通常会替换出现故障的组件以保持服务可用性。

您可以使用 EC2 自动扩缩组等自动机制将失败的实例替换为新实例。实例自动恢复可以重启因服务器故障而失败的实例，前提是其余服务器上有足够的可用容量。

## 故障模式 3：计算

服务器可能会出现故障或受损，并且可能由于各种原因（例如组件故障和定期维护操作）而需要（临时或永久）停止运行。Outpost 机架上的服务处理服务器故障和受损问题的方式各不相同，可能取决于客户如何配置高可用性选项。

您应该订购充足的计算容量以支持 N+M 可用性模型，其中 N 是所需的容量，M 是为应对服务器故障而预置的备用容量。

作为完全托管 AWS Outposts 机架服务的一部分，提供故障服务器的硬件更换。AWS 主动监控 Outpost 部署中所有服务器和网络设备的运行状况。如需进行物理维护，AWS 将安排时间前往您的站点以更换出现故障的组件。通过预置备用容量，您可以在出现故障的服务器停止使用和进行更换的同时保持工作负载运行。

## 故障模式 4：机架或数据中心

机架故障可能是由于机架完全断电或环境故障（例如冷却中断或数据中心因洪水或地震而受到物理损坏）所致。数据中心的配电架构存在缺陷或标准数据中心电源维护期间出现错误，都可能导致一个或多个机架甚至整个数据中心断电。

通过将基础设施部署到多个数据中心楼层或者同一园区或城区内相互独立的地点，可以缓解上述情况。

在 AWS Outposts 机架上采用这种方法需要仔细考虑应用程序的架构和分布方式，使其在多个独立的逻辑 Outposts 上运行，以保持应用程序的可用性。

## 故障模式 5：AWS 可用区或区域

每个 Outpost 都锚定到某个 AWS 区域内的特定可用区（AZ）。锚 AZ 或父级区域内的故障可能会导致 Outpost 失去管理和可变性，并可能中断 Outpost 与区域之间的网络通信。

与网络故障类似，AZ 或区域故障可能会导致 Outpost 与区域断开连接。如前所述，在 Outpost 上运行的实例将继续运行，并且可以通过 Outpost 本地网关（LGW）从本地网络进行访问，如果这些实例依赖于区域内的服务，则可能会受损或失败。

为了减轻 AWS 可用区和区域故障的影响，您可以部署多个 Outposts，每个 Outpost 都锚定到不同的可用区或区域。然后，您可以使用当今用于设计和部署的许多类似[机制和架构模式](#)来设计工作负载，使其在分布式多前哨部署模型中运行。AWS

# 使用 AWS Outposts 机架构建 HA 应用程序和基础设施解决方案

借 AWS Outposts 助 rack，您可以使用熟悉的 AWS 云服务和工具构建、管理和扩展高度可用的本地应用程序。云 HA 的架构和方法通常不同于当前可能在数据中心内运行的传统本地 HA，了解这一点非常重要。

利用传统的本地 HA 应用程序部署方法，可将应用程序部署到虚拟机 ( VM ) 中。要保持这些虚拟机持续正常运行，则需要部署和维护复杂的 IT 系统和基础设施。VM 通常具有特定的身份，每台 VM 在整个应用程序架构中可能都扮演着一个关键角色。

架构角色与 VM 身份紧密耦合。系统架构师可以利用 IT 基础设施功能提供高度可用的 VM 运行时环境，使每台 VM 都能可靠地访问计算容量、存储卷和网络服务。如果 VM 出现故障，则会运行自动或手动恢复流程，以将出现故障的 VM 恢复到正常运行状态，恢复流程通常在其他基础设施上或完全在另一个数据中心内运行。

Cloud HA 架构采用不同的方法。AWS 云服务提供可靠的计算、存储和联网功能。应用程序组件可以部署到 EC2 实例、容器、无服务器函数或其他托管式服务。

实例是指某个应用程序组件 ( 可能是扮演该角色的众多组件之一 ) 的实例化。应用程序组件彼此之间以及这些组件与其在整个应用程序架构中所扮演的角色之间为松耦合关系。实例的单个身份通常并不重要。可以创建或销毁其他实例，以根据需求进行扩展或缩减。只需将失败的实例或运行不正常的实例替换为运行正常的新实例即可。

AWS Outposts rack 是一项完全托管的服务，可将 AWS 计算、存储、网络、数据库和其他云服务扩展到本地位置，以提供真正一致的混合体验。您不应将 Outpost 机架服务视为采用传统本地 HA 机制的 IT 基础设施系统的简易替代。尝试使用 AWS 服务和 Outposts 来支持传统的本地 HA 架构是一种反模式。

在 AWS Outposts 机架上运行的工作负载使用云高可用性机制，例如 [Amazon EC2 Auto Scaling \( 水平扩展以满足工作负载需求 \)](#)、[EC2 运行状况检查](#) ( 检测和删除不健康的实例 ) 和 [应用程序负载均衡器](#) ( 将传入的工作负载流量重新定向到扩展或替换的实例 )。将应用程序迁移到云端时，无论是迁移到云 AWS 区域 还是 AWS Outposts 机架，都应更新高可用性应用程序架构，以开始利用托管云服务和云高可用性机制。

以下各节介绍了架构模式、反模式以及在本地环境中部署 AWS Outposts 机架以运行具有高可用性要求的工作负载的推荐做法。这些部分介绍了多种模式和实操；但是，其中并未提供配置和实现细节。在

使环境为部署 Outpost 机架做好准备并使应用程序为迁移到 AWS 服务做好准备的过程中，您应阅读并熟悉 [AWS Outposts 机架常见问题](#) 和 [用户指南](#) 以及有关在 Outpost 机架上运行的服务的常见问题和服务文档。

## 主题

- [联网](#)
- [计算](#)
- [存储](#)
- [更大规模的故障模式](#)

## 联网

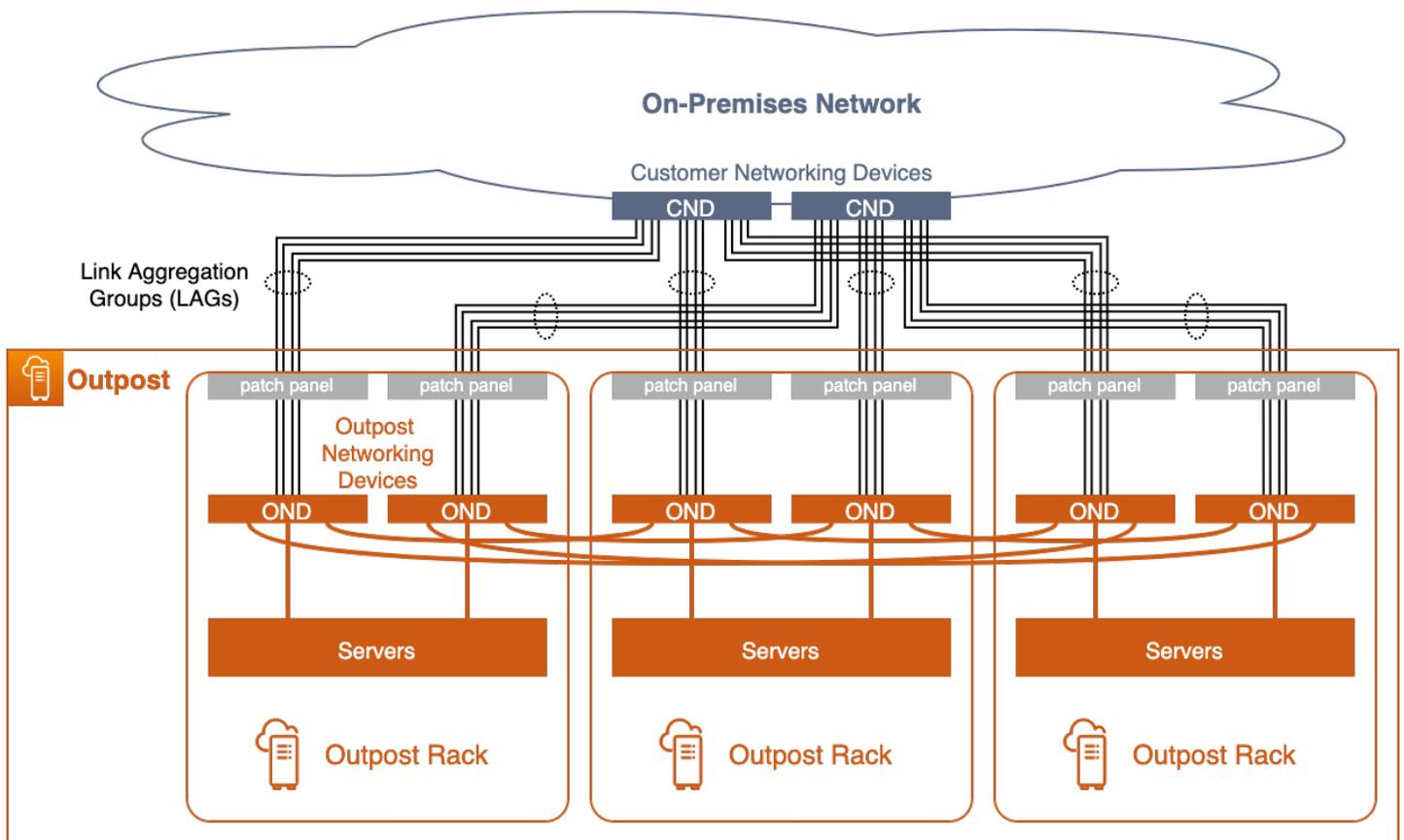
Outpost 部署依赖于与其锚 AZ 的弹性连接，从而使管理、监控和服务操作正常运行。您应该配置本地网络，为每个 Outpost 机架提供冗余的网络连接，以及与 AWS 云中锚点的可靠连接。此外，还要考虑在 Outpost 上运行的应用程序工作负载以及与之进行通信的其他本地系统和云系统之间的网络路径 – 您要如何在网络中路由这些流量呢？

## 主题

- [网络连接](#)
- [锚连接](#)
- [应用程序/工作负载路由](#)

## 网络连接

每个 AWS Outposts 机架都配置了名为 Outpost 网络设备 (OnD) 的冗余 top-of-rack 交换机。每个机架中的计算和存储服务器都连接到两个 OND。您应该将每个 OND 连接到数据中心内一台名为客户网络设备 (CND) 的独立交换机，以便为每个 Outpost 机架提供不同的物理和逻辑路径。借助光纤电缆和光学收发器，OND 通过一个或多个物理连接连接到 CND。[物理连接](#)是在逻辑[链路聚合组 \(LAG\) 链路](#)中配置的。



## 具有冗余网络连接的多机架 Outpost

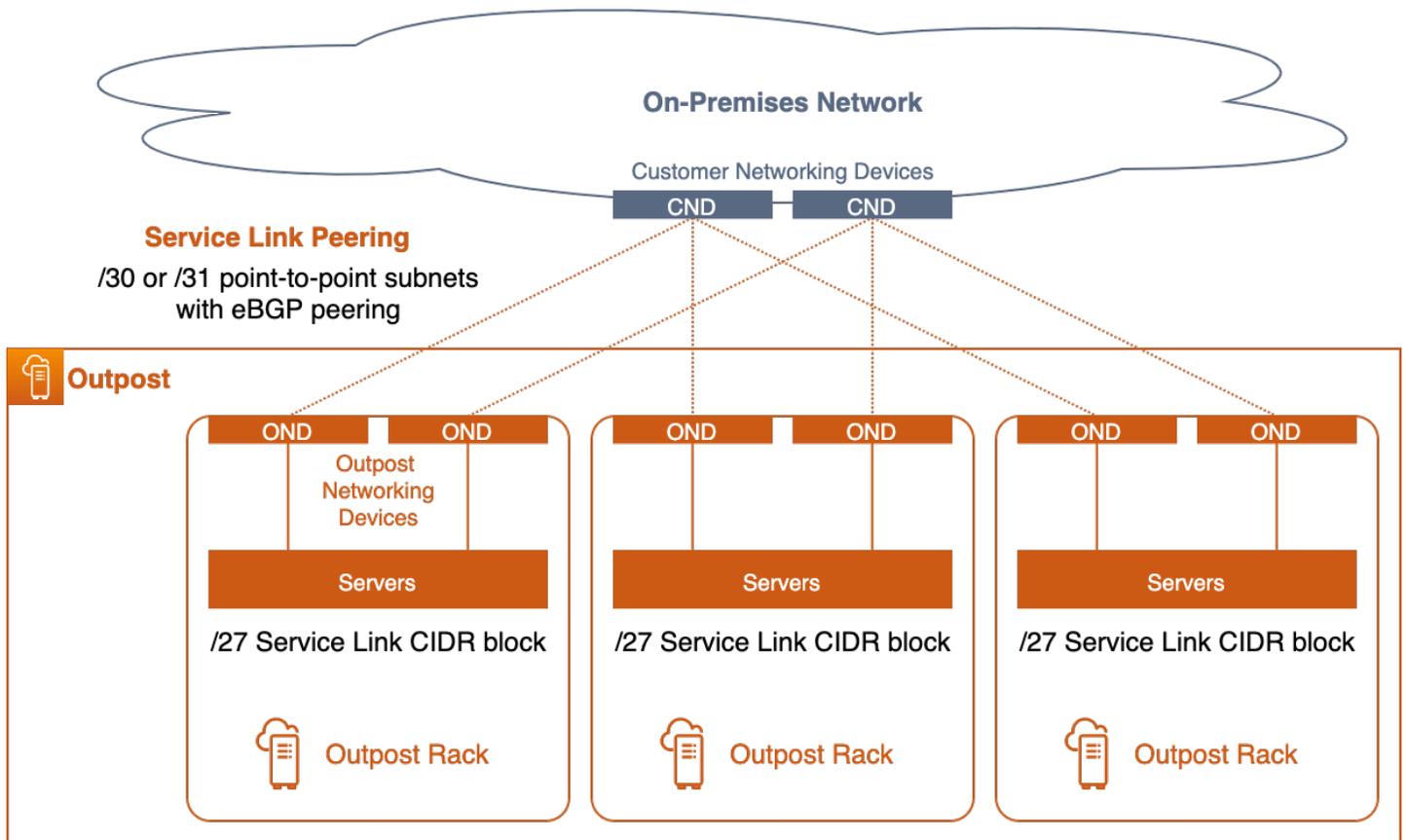
OND 到 CND 的链路始终在 LAG 中配置，即使物理连接是单根光纤电缆也是如此。若将链路配置为 LAG 组，则可以通过向逻辑组添加其他物理连接来增加链路带宽。将 LAG 链路配置为 IEEE 802.1q 以太网中继，以便实现 Outpost 和本地网络之间的网络隔离。

每个 Outpost 都有至少两个逻辑上隔离的网络，这些网络需要与客户网络进行通信或跨客户网络进行通信：

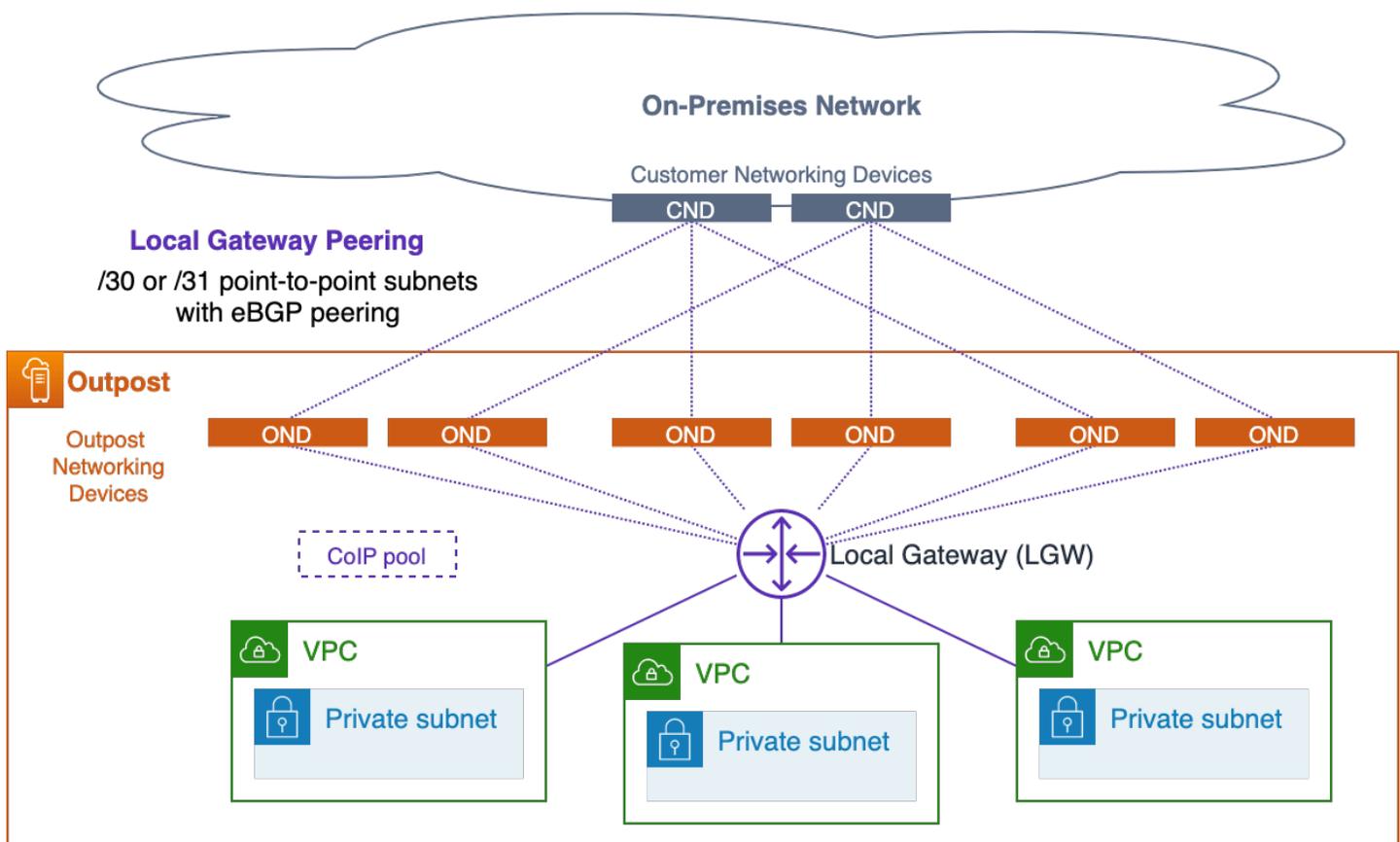
- 服务链接网络 — 将服务链路 IP 地址分配给 Outpost 服务器，并促进与本地网络的通信，从而允许服务器重新连接到该区域的 Outpost 锚点。
- 本地网关网络：通过 Outpost 本地网关 ( LGW ) 实现 Outpost 上的 VPC 子网与本地网络之间的通信。

这些隔离的网络通过一组 [point-to-point IP 连接](#) 通过 LAG 链路连接到本地网络。每个 OND 到 CND 延迟链路都配置了 VLAN ID、point-to-point ( /30 或 /31 ) IP 子网以及每个隔离网络 ( 服务链路和 LGW ) 的 eBGP 对等互连。您应该将 LAG 链路及其的 point-to-point VLAN 和子网视为第 2 层分段、

路由的第 3 层连接。路由的 IP 连接可提供冗余的逻辑路径，有利于 Outpost 上的隔离网络与本地网络之间的通信。



### 服务链接对等



## 本地网关对等

您应该终止直接连接的 CND 交换机上的第 2 层 LAG 链路（及其 VLAN），并在 CND 交换机上配置 IP 接口和 BGP 对等，而不应在数据中心交换机之间桥接 LAG VLAN。有关更多信息，请参阅《AWS Outposts 用户指南》中的[网络层连接](#)。

在逻辑多机架前哨基地内，OnD 以冗余方式互连，从而在机架和服务器上运行的工作负载之间提供高度可用的网络连接。AWS 负责前哨基地内的网络可用性。

## 针对高度可用的网络连接的建议实操

- 将 Outpost 机架中的每台 Outpost 网络设备（OND）连接到数据中心中的独立客户网络设备（CND）。
- 终止直接连接的客户网络设备（CND）交换机上的第 2 层链路、VLAN、第 3 层 IP 子网和 BGP 对等。切勿在不同的 CND 之间或跨本地网络将 OND 桥接到 CND VLAN。
- 向链路聚合组（LAG）添加链路，以增加 Outpost 和数据中心之间的可用带宽。不要依赖于通过两个 OND 的不同路径的聚合带宽。
- 使用通过冗余 OND 的不同路径，从而提供 Outpost 网络和本地网络之间的弹性连接。

- 为了实现最佳冗余并允许无中断的 OND 维护，我们建议客户按以下方式配置 BGP 通告和策略：
  - 客户网络设备应在不更改 BGP 属性的情况下接收 Outpost 发出的 BGP 通告，并启用 BGP 多路径/负载均衡以实现最佳的入站流量（从客户到 Outpost）。为 Outpost BGP 前缀使用 AS-Path 前置，以便在需要维护时，将流量从特定 OND/上行链路转移出去。客户网络应首选 Outpost 中 AS-Path 长度为 1 的路由，而不是 AS-Path 长度为 4 的路由，也就是说，应对 AS-Path 前置作出反应。
  - 客户网络应向 Outpost 中的所有 OND 传播具有相同属性的等价 BGP 前缀。默认情况下，Outpost 网络负载会均衡所有上行链路之间的出站流量（流向客户）。如果需要维护，可在 Outpost 端使用路由策略，以将流量从特定的 OND 转移出去。要执行此流量转移并以无中断的方式进行维护，需要在所有 OND 的客户端使用等价 BGP 前缀。当客户的网络需要维护时，我们建议使用 AS-Path 前置，以将来自特定上行链路或设备的流量暂时转移出去。

## 锚连接

[Outpost 服务链接](#) 连接到 Outpost 父区域中特定可用区 (AZ) 中的公共锚点或私有锚点（不能两者兼而有之）。前哨服务器启动从其服务链接 IP 地址到锚点 AZ 中锚点的出站服务链接 VPN 连接。这些连接使用 UDP 和 TCP 端口 443。AWS 负责区域内锚点的可用性。

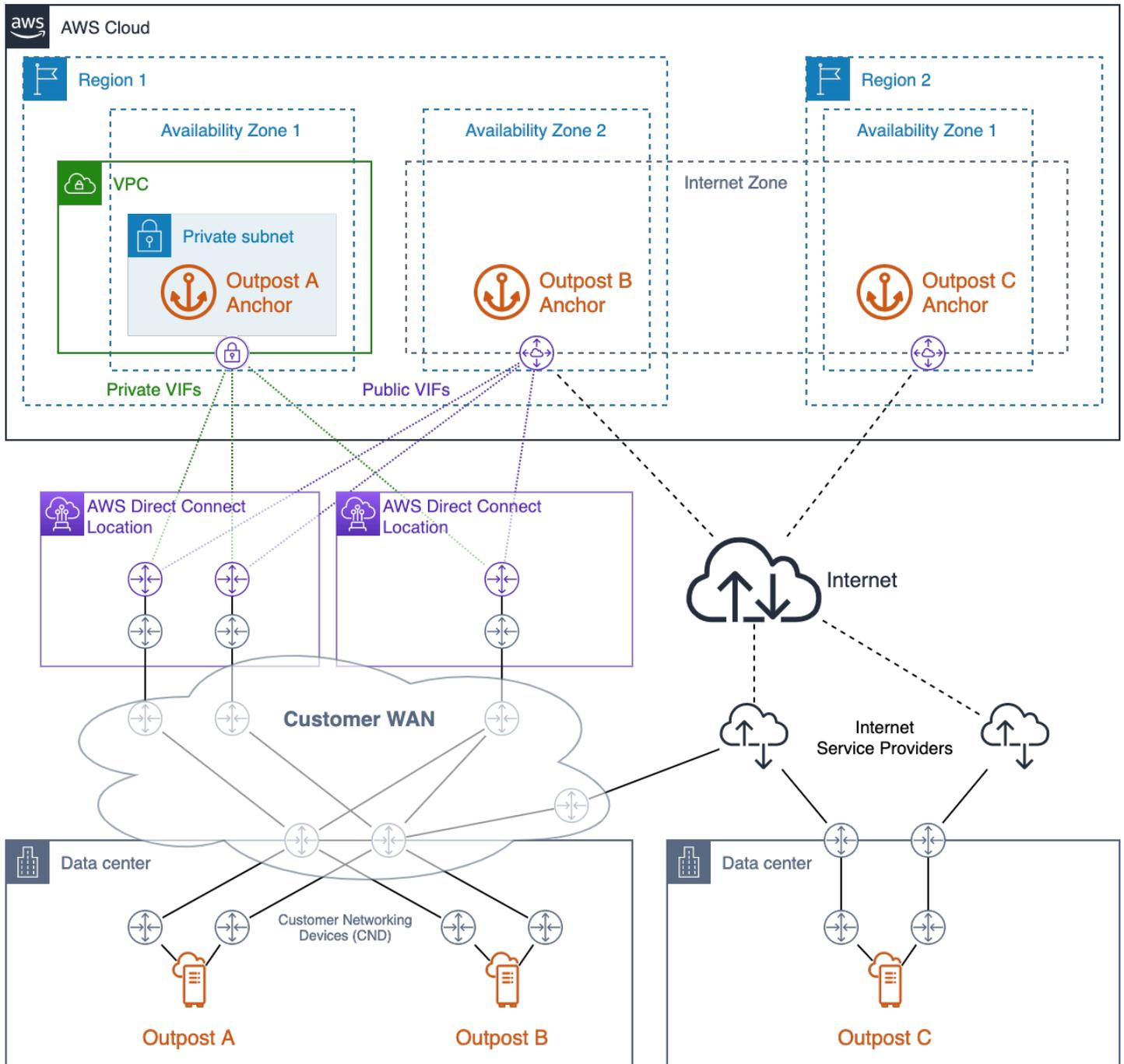
您必须确保 Outpost 服务链接 IP 地址可以通过您的网络连接到锚点 AZ 中的锚点。服务链接 IP 地址无需与本地网络上的其他主机通信。

公有锚点位于区域的 [公有 IP 范围](#)（EC2 服务 CIDR 块中）内，可以通过互联网或 [AWS Direct Connect](#)（DX）公有虚拟接口（VIF）进行访问。使用公共锚点可以更灵活地选择路径，因为服务链路流量可以通过任何可以成功到达公共互联网上的锚点的可用路径进行路由。

私有锚点支持您使用自己的 IP 地址范围进行锚连接。私有锚点是使用客户分配的 IP 地址在 [专用 VPC 内的私有子网](#) 中创建的。VPC 是在拥有 Outpost 资源的中创建的，您负责确保 VPC 可用且配置正确（不要将其删除！）。AWS 账户私有锚点必须使用 [Direct Connect 私有 VIF](#) 进行访问。

您应该预置 Outpost 与区域内锚点之间的冗余网络路径，同时在多个位置的独立设备上终止连接。动态路由的配置应可实现以下功能：在连接或网络设备出现故障时，自动将流量重新路由到备用路径。您应该预置充足的网络容量，以确保一条 WAN 路径的故障不会使剩余路径不堪重负。

下图显示了三个 Outposts，这些 Outposts 具有通往其锚可用区的冗余网络路径 AWS Direct Connect 以及公共互联网连接。Outpost A 和 Outpost B 锚定到同一区域的不同可用区。Outpost A 连接到区域 1 的 AZ 1 中的私有锚点。Outpost B 连接到区域 1 的 AZ 2 中的公有锚点。Outpost C 连接到区域 2 的 AZ 1 的公有锚。



## 高度可用的锚点连接 AWS Direct Connect 和公共互联网接入

Outpost A 有三条冗余网络路径可以访问其私有锚点。其中两条路径通过单个 Direct Connect 位置的冗余 Direct Connect 线路实现。第三条路径通过位于另一个 Direct Connect 位置的 Direct Connect 线路实现。这种设计将 Outpost A 的服务链路流量保持在专用网络上，并提供路径冗余，允许任何一个 Direct Connect 电路出现故障或整个 Direct Connect 位置出现故障。

Outpost B 有四条冗余网络路径可以访问其公有锚点。其中三条路径通过在 Direct Connect 线路上预置的公有 VIF 和 Outpost A 使用的位置实现。第四条路径通过客户 WAN 和公有互联网实现。Outpost B 的服务链接流量可以通过任何可以成功到达公共互联网锚点的可用路径进行路由。使用 Direct Connect 路径可以提供更稳定的延迟和更高的带宽可用性，而公有互联网路径可用于灾难恢复 (DR) 或带宽增强方案。

Outpost C 有两条冗余的网络路径可以访问其公有锚点。Outpost C 部署在与 Outpost A 和 B 不同的数据中心中。Outpost C 的数据中心没有连接到客户 WAN 的专用线路。相反，数据中心拥有冗余互联网连接，分别由两个不同的互联网服务提供商 (ISP) 提供。Outpost C 的服务链路流量可以通过任一 ISP 网络进行路由，以到达公共互联网上的锚点。这种设计允许灵活地通过任何可用的公共互联网连接路由服务链路流量。但是，该 end-to-end 路径依赖于带宽可用性和网络延迟波动的公共第三方网络。

前哨基地与其服务链路锚点之间的网络路径必须符合以下带宽规范：

- 每个 Outpost 机架的可用带宽为 500 Mbps – 1 Gbps (例如，3 个机架：1.5 – 3 Gbps 可用带宽)

针对高度可用的锚连接的建议实操：

- 预置每个 Outpost 与其在区域内的锚点之间的冗余网络路径。
- 使用 Direct Connect (DX) 路径来控制延迟和带宽可用性。
- 确保从 Outpost 服务链接 CIDR 块到父级区域内的 [EC2 IP 地址范围](#) 的 TCP 和 UDP 端口 443 处于打开 (出站) 状态。确保所有网络路径上的端口均处于打开状态。
- 确保每条路径都满足带宽可用性和延迟需求。
- 使用动态路由，从而自动重定向流量以绕过网络故障。
- 测试在每条规划的网络路径上路由服务链路流量，以确保路径按预期运行。

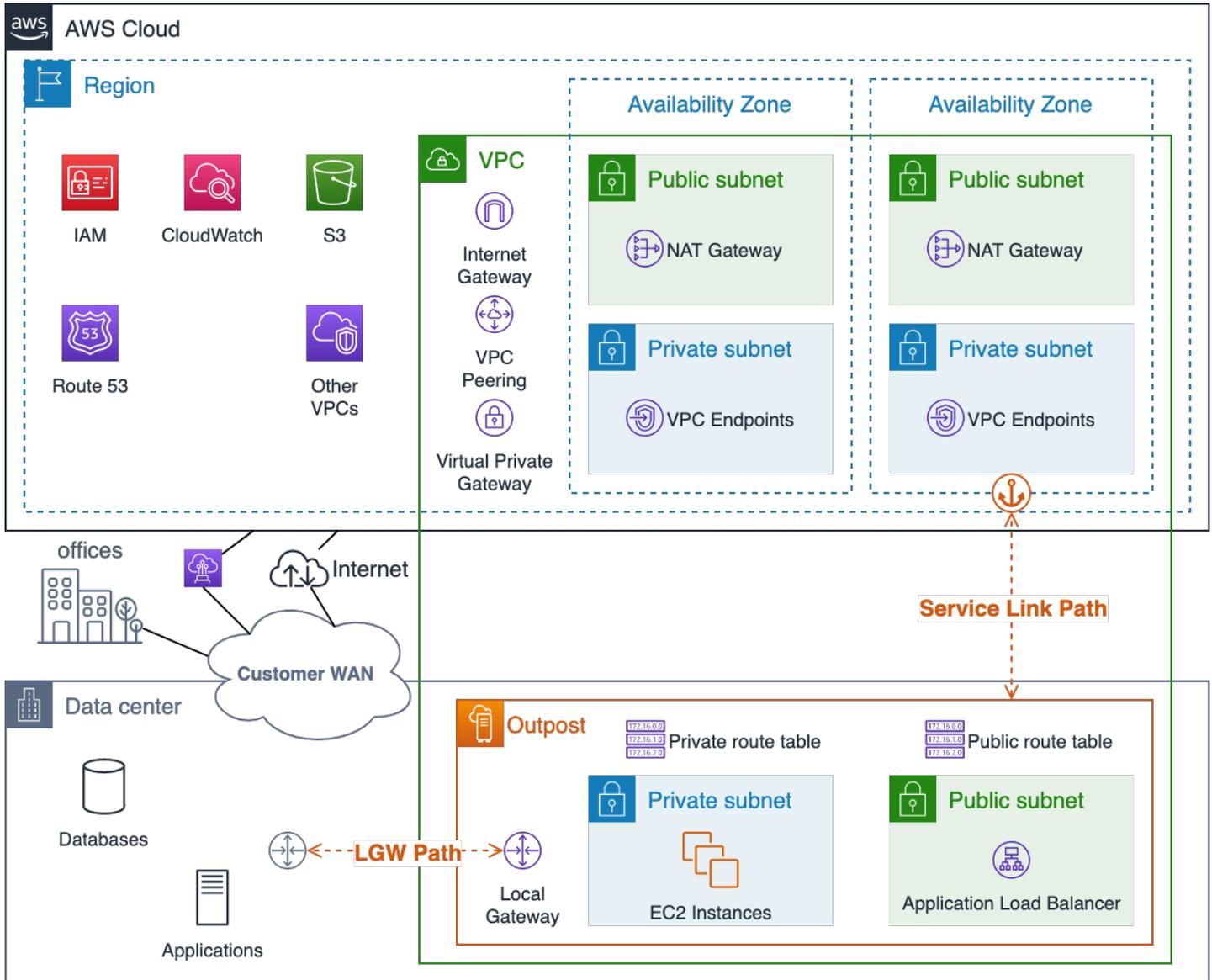
## 应用程序/工作负载路由

应用程序工作负载有两条从 Outpost 传出的路径：

- 服务链接路径
- 本地网关 (LGW) 路径

您可以配置 Outpost 子网路由表，以控制通过哪条路径访问目标网络。指向 LGW 的路由会使流量从本地网关传出并将其定向到本地网络。指向该区域服务和资源的路由 (例如 Internet Gateway、NAT Gateway、Virtual Private Gateway 和 TGW) 将使用 [服务链接](#) 来到达这些目标。如果您与同一

Outpost 上的多个 VPC 建立了 VPC 对等连接，则这些 VPC 之间的流量仍保留在前哨站上，并且不会使用返回该地区的服务链接。有关 VPC 对等互连的信息，请参阅 Amazon [VPC 用户指南中的使用 VPC 对等连接连接 VPC](#)。



### Outpost 服务链接和 LGW 网络路径的可视化

在规划应用程序路由时，应保持谨慎，考虑到正常操作和网络故障期间的有限路由和服务可用性。如果 Outpost 与区域断开连接，则服务链接路径不可用。

您应该在 Outpost LGW 和关键本地应用程序、系统和用户之间预置不同的路径并配置动态路由。冗余网络路径允许网络绕过故障路由流量，并确保在部分网络出现故障期间，本地资源能够与 Outpost 上运行的工作负载进行通信。

Outpost VPC 路由配置是静态的。您可以通过 CLI、AWS Management Console、API 和其他基础设施即代码 (IaC) 工具配置子网路由表；但是，在断开连接事件期间，您将无法修改子网路由表。您必须重新建立 Outpost 和区域之间的连接才能更新路由表。正常操作时使用的路由与计划在断开连接事件期间使用的路由相同。

前哨基地上的资源可以通过服务链接和该地区的 Internet Gateway (IGW) 或通过本地网关 (LGW) 路径访问互联网。通过 LGW 路径和本地网络路由互联网流量允许您使用现有的本地互联网入口/出口点，与使用通往该地区的 IGW 的服务链路路径相比，可以提供更低的延迟、更高的 MTU 和更低 AWS 的数据出口费用。

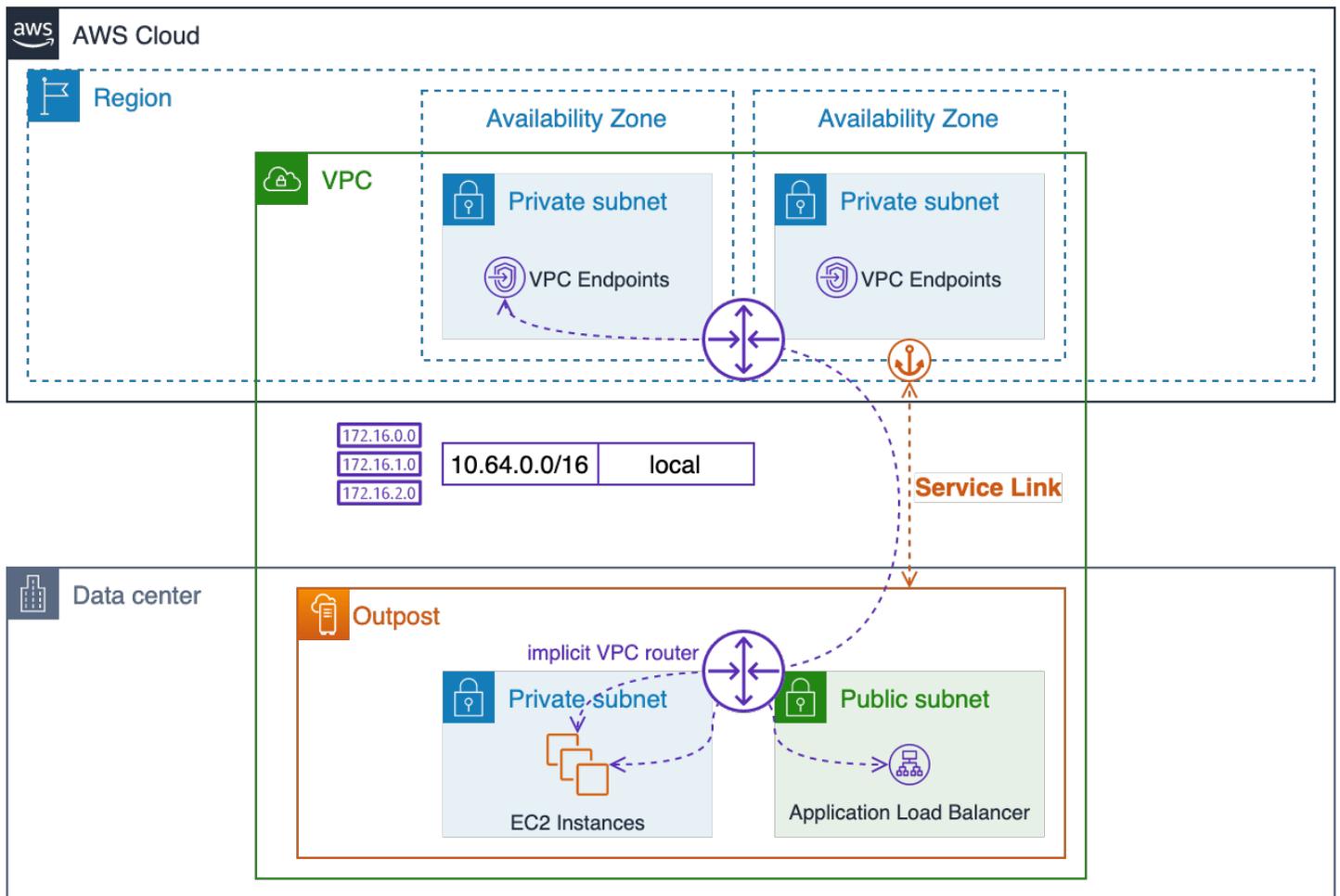
如果应用程序必须在本地运行且需要支持通过公有互联网进行访问，则应通过本地互联网连接将应用程序流量路由到 LGW，以访问 Outpost 上的资源。

虽然您可以用类似于配置区域内的公有子网的方式在 Outpost 上配置子网，但对于大多数使用场景来说，这可能是一种不良实操。入站互联网流量将通过进入，AWS 区域并通过服务链接路由到前哨基地上运行的资源。

反过来，响应流量将通过服务链接路由，然后通过互联网连接返回。AWS 区域在流量离开区域流向 Outpost 的过程中，以及在返回流量通过区域返回并传出到互联网的过程中，这种流量模式可能会增加延迟，并且会产生数据输出费用。如果应用程序可以在区域内运行，则区域是其最佳运行位置。

不同 VPC 资源 (同一 VPC 内) 之间的流量将始终由本地 VPC CIDR 路由，并由隐式 VPC 路由器在子网之间路由。

例如，在 Outpost 上运行的 EC2 实例与该地区的 VPC 终端节点之间的流量将始终通过服务链接进行路由。



通过隐式路由器进行的本地 VPC 路由

针对应用程序/工作负载路由的建议实操：

- 尽可能使用本地网关 (LGW) 路径而不是服务链接路径。
- 通过 LGW 路径路由互联网流量。
- 使用一组标准路由配置 Outpost 子网路由表，这些路由将用于正常操作和断开连接事件。
- 预置 Outpost LGW 和关键本地应用程序资源之间的冗余网络路径。使用动态路由，从而自动重定向流量以绕过本地网络故障。

## 计算

虽然 Amazon EC2 的容量看似 AWS 区域 是无限的，但 Outposts 的容量是有限的。您负责规划和管理 Outpost 部署的计算容量。

## 主题

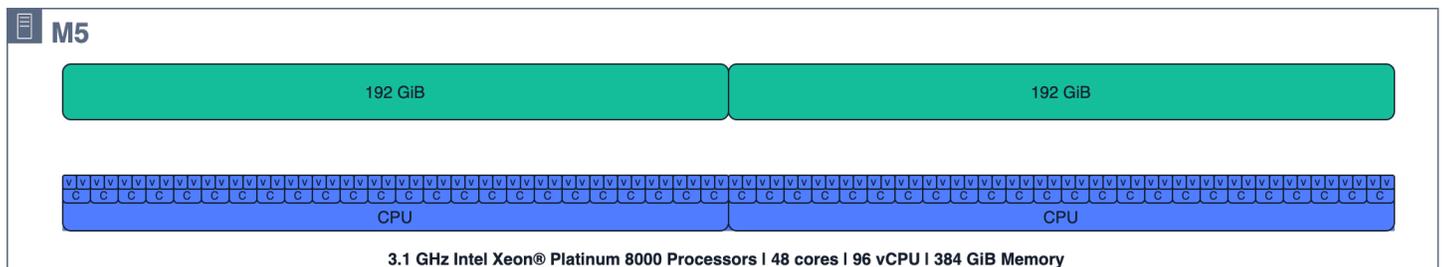
- [容量规划](#)
- [容量管理](#)
- [实例置放](#)

## 容量规划

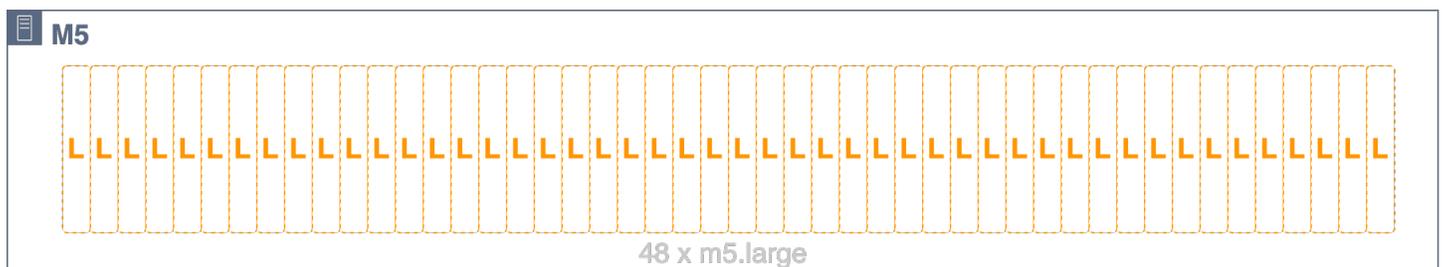
虽然 Amazon EC2 的容量看似 AWS 区域 是有限的，但 Outposts 的容量是有限的，受订购的计算容量总量的限制。您负责规划和管理 Outpost 部署的计算容量。您应该订购充足的计算容量以支持 N+M 可用性模型，其中 N 是所需的服务器数量，M 是为应对服务器故障而预置的备用服务器数量。N+1 和 N+2 是最常见的可用性级别。

每台服务器 ( C5M5、R5、等 ) 都支持单个 EC2 实例系列。在 EC2 计算服务器上启动实例之前，必须提供时段布局，指定您希望每台服务器提供的 [EC2 实例大小](#)。AWS 使用请求的时段布局配置每台服务器。

如果所有插槽的实例大小相同 ( 例如，48 个插槽 )，则服务器可以采用同质 m5.large 插槽，也可以使用混合实例类型 ( 例如 4、4 m5.large、3 m5.2xlarge、1 和 1m5.8xlarge ) 进行异质插槽 m5.xlarge，有关这些插槽配置的可视化效果 m5.4xlarge，请参阅接下来的三个图。



### m5.24xlarge 服务器计算资源



m5.24xlarge 服务器均匀地插入 48 个插槽 m5.large



*m5.24xlarge* 服务器异构地分为 4 *m5.large*、4、3 *m5.xlarge* *m5.2xlarge* *m5.4xlarge*、1 和 1 个插槽 *m5.8xlarge*

服务器的容量并非必须全部都分配给槽。槽可以添加到具有可用未分配容量的服务器上。您可以通过打开支持工单来修改槽布局。如果新的槽布局因某些槽被当前运行的实例占用而无法应用，则 Enterprise Support 可能会要求您关闭或重启某些实例以完成槽重新布局的请求。

所有服务器的预置槽都可以提供给 Outpost 上的 EC2 容量池，并且具有指定实例类型和大小的所有槽都可以作为单个 EC2 容量池进行管理。例如，之前带有、*m5.large**m5.xlarge**m5.2xlarge**m5.4xlarge*、和 *m5.8xlarge* 插槽的异构插槽服务器会将这些插槽贡献给五个 EC2 容量池，每种实例类型和大小各占一个池。

在规划备用容量以实现 N+M 服务器可用性时，必须考虑服务器时段和 EC2 容量池。AWS 检测服务器何时出现故障或性能降级，并安排现场访问以更换出现故障的服务器。在设计 EC2 容量池时，应确保其能够容忍 Outpost 中每个实例系列 (N+1) 中至少有一台服务器出现故障。凭借该最低级别的服务器可用性，当服务器出现故障或需要停止使用时，您可以在同系列的其余服务器的备用槽上重启失败或降级的实例。

如果您拥有多台采用同构槽配置的服务器或一组采用异构槽配置且具有相同槽布局的服务器，则 N+M 可用性的规划十分简单。您只需计算运行所有工作负载所需的服务器数量 (N)，然后再增加一定数量 (M) 的额外服务器，即可满足故障和维护事件期间的服务器可用性需求。

由于 NUMA 边界，以下插槽配置不可用：

- 3 *m5.8xlarge*
- 1 *m5.16xlarge* 和 1 *m5.8xlarge*

请咨询您的 AWS 账户 团队，以验证您计划的 AWS Outposts 机架插槽配置。

在下图中，四 *m5.24xlarge* 台服务器的插槽是异质的，时隙布局相同。这四台服务器创建了五个 EC2 容量池。每个池都以最大使用率 (75%) 运行，以保持在这四台服务器上运行的实例的 N+1 可用性。如果任何服务器出现故障，则有充足的空间在其余服务器上重启失败的实例。



### EC2 服务器槽、正在运行的实例和槽池的可视化

对于更复杂的槽布局，如果服务器的槽配置不相同，则需要计算每个 EC2 容量池的 N+M 可用性。您可以使用以下公式来计算有多少台服务器（为指定的 EC2 容量池提供槽）可以出现故障而其余服务器仍能支撑正在运行的实例：

$$M = \left\lceil \frac{poolSlots_{available}}{serverSlots_{max}} \right\rceil$$

其中：

- $poolSlots_{available}$  是指定 EC2 容量池中可用槽的数量（池中的槽总数减去正在运行的实例数）
- $serverSlots_{max}$  是任何服务器提供给指定 EC2 容量池的最大槽数
- M 是可以出现故障而其余服务器仍能支撑正在运行的实例的服务器数量

示例：前哨基地有三台服务器为 m5.2xlarge 容量池提供插槽。第一台服务器提供 4 个槽，第二台提供 3 个槽，第三台提供 2 个槽。Outpost 上的 m5.2xlarge 实例池总容量为 9 个插槽（4 + 3 + 2）。前哨基地有 4 个正在运行的 m5.2xlarge 实例。有多少台服务器可以出现故障而其余服务器仍能支撑正在运行的实例？

$$poolSlots_{available} = total\ capacity - running\ instances = 9 - 4 = 5$$

$$serverSlots_{max} = \max([4, 3, 2]) = 4$$

$$M = \left\lceil \frac{poolSlots_{available}}{serverSlots_{max}} \right\rceil = \left\lceil \frac{5}{4} \right\rceil = [1.25] = 1$$

答案：在丢失其中任何一台服务器的情况下，其余服务器仍可支撑正在运行的实例。

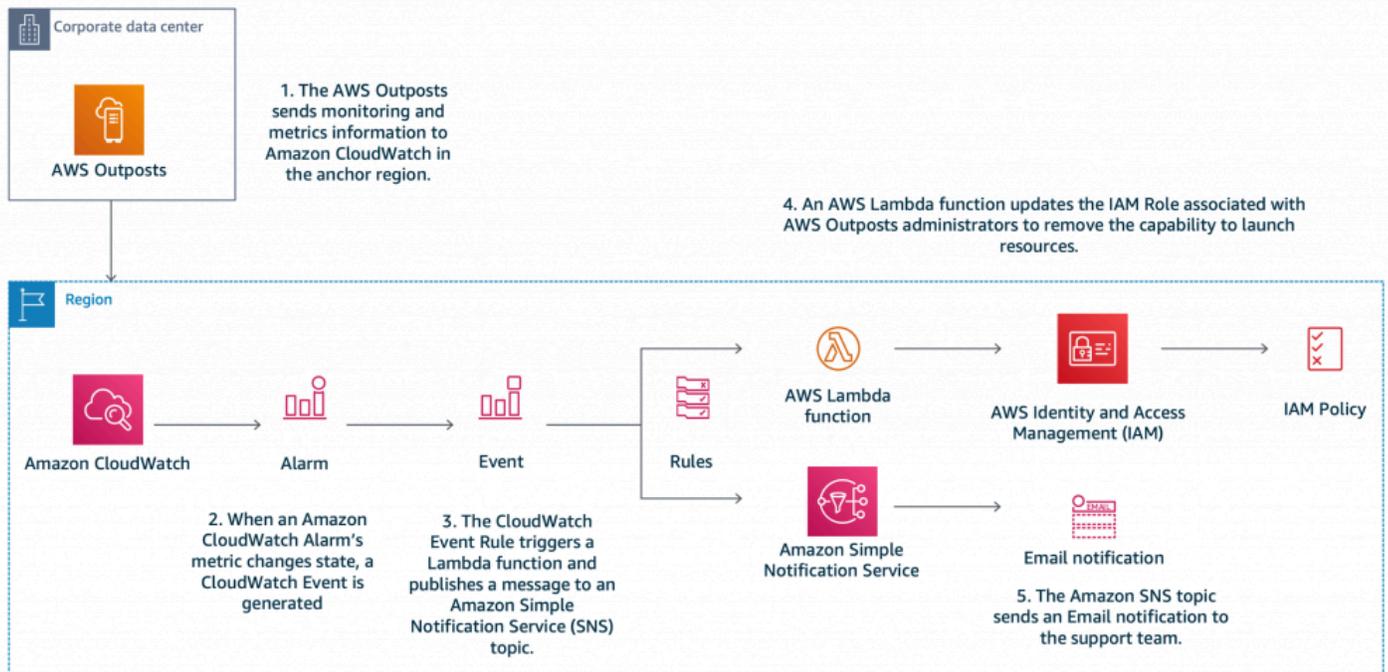
针对计算容量规划的建议实操：

- 调整计算容量，从而为 Outpost 上的每个 EC2 容量池提供 N+M 冗余。
  - 为采用同构槽配置或具有相同布局的异构槽配置的服务器部署 N+M 服务器。
  - 计算每个 EC2 容量池的 N+M 可用性，并确保每个容量池都满足您的可用性需求。

## 容量管理

您可以通过亚马逊指标 AWS Management Console 和通过亚马逊 CloudWatch 指标监控 Outpost EC2 实例池的利用率。请联系 Enterprise Support 以检索或更改 Outpost 的槽布局。

您可以使用相同的[实例自动恢复](#)和[EC2 自动扩缩](#)机制来恢复或替换受服务器故障和维护事件影响的实例。您必须对 Outpost 容量进行监控和管理，以确保始终有充足的备用容量来应对服务器故障。[使用 Amazon 管理 AWS Outposts 容量 CloudWatch 和 AWS Lambda](#) 博客文章提供了一个动手教程，向您展示了如何合并 AWS CloudWatch 和 AWS Lambda 管理 Outpost 容量以保持实例可用性。



使用 Amazon 管理 AWS Outposts 容量 CloudWatch 和 AWS Lambda

针对计算容量管理的建议实操：

- 在自动扩缩组中配置 EC2 实例，或者使用实例自动恢复机制来重启失败的实例。
- 自动监控 Outpost 部署的容量，并配置容量警报的通知和（可选的）自动响应。

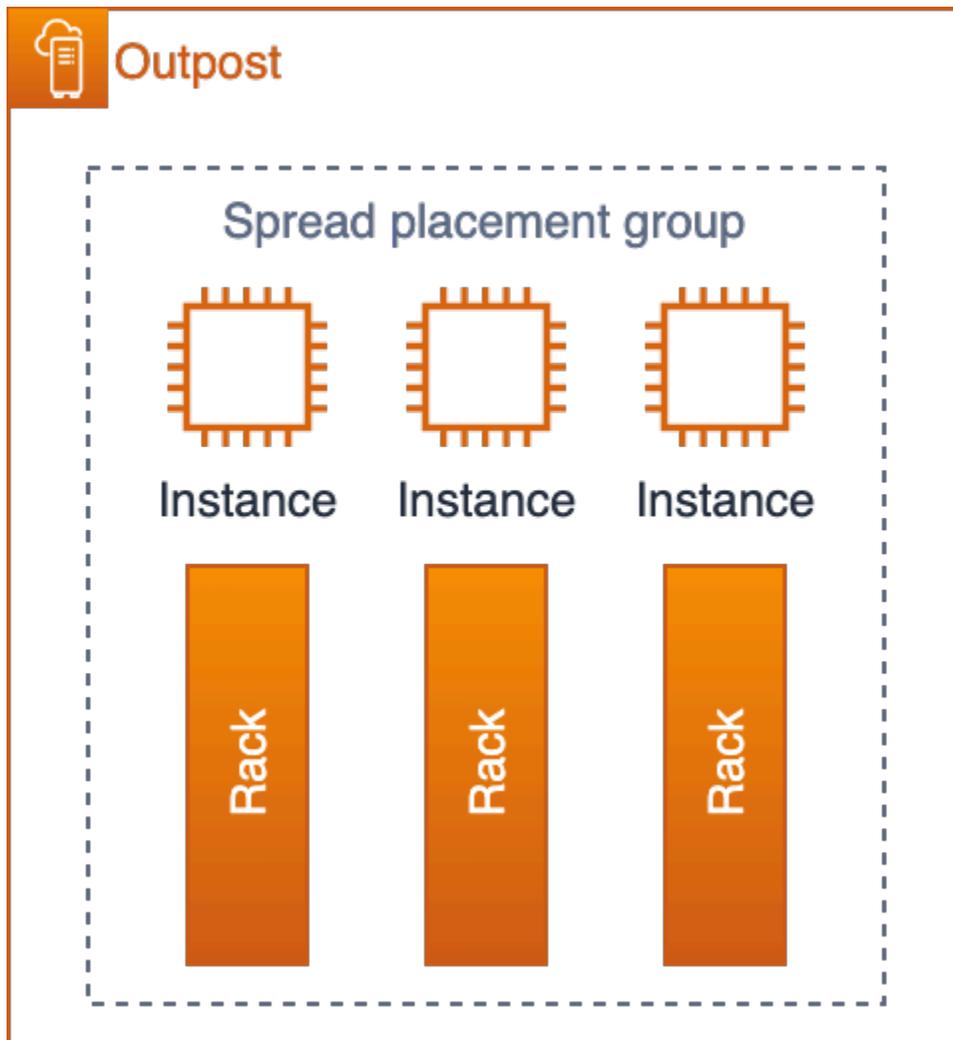
## 实例置放

Outpost 的计算服务器数量有限。如果应用程序在 Outpost 上部署了多个相关实例，在没有额外配置的情况下，这些实例可能会部署在同一台服务器上或同一机架中的多台服务器上。如今，您可以使用三种机制来分配实例，以降低在同一基础设施上运行相关实例的风险：

**多 Outpost 部署：**与区域内的多 AZ 策略类似，您可以将 Outpost 部署到单独的数据中心，并将应用程序资源部署到特定的 Outpost。这使您能够在所需的 Outpost（一组逻辑机架）上运行实例。可以采用多 Outpost 策略来提供针对机架和数据中心故障模式的保护，如果 Outpost 锚定到独立的 AZ 或区域，也可以提供针对 AZ 或区域故障模式的保护。有关多 Outpost 架构的更多信息，请参阅[更大规模的故障模式](#)。

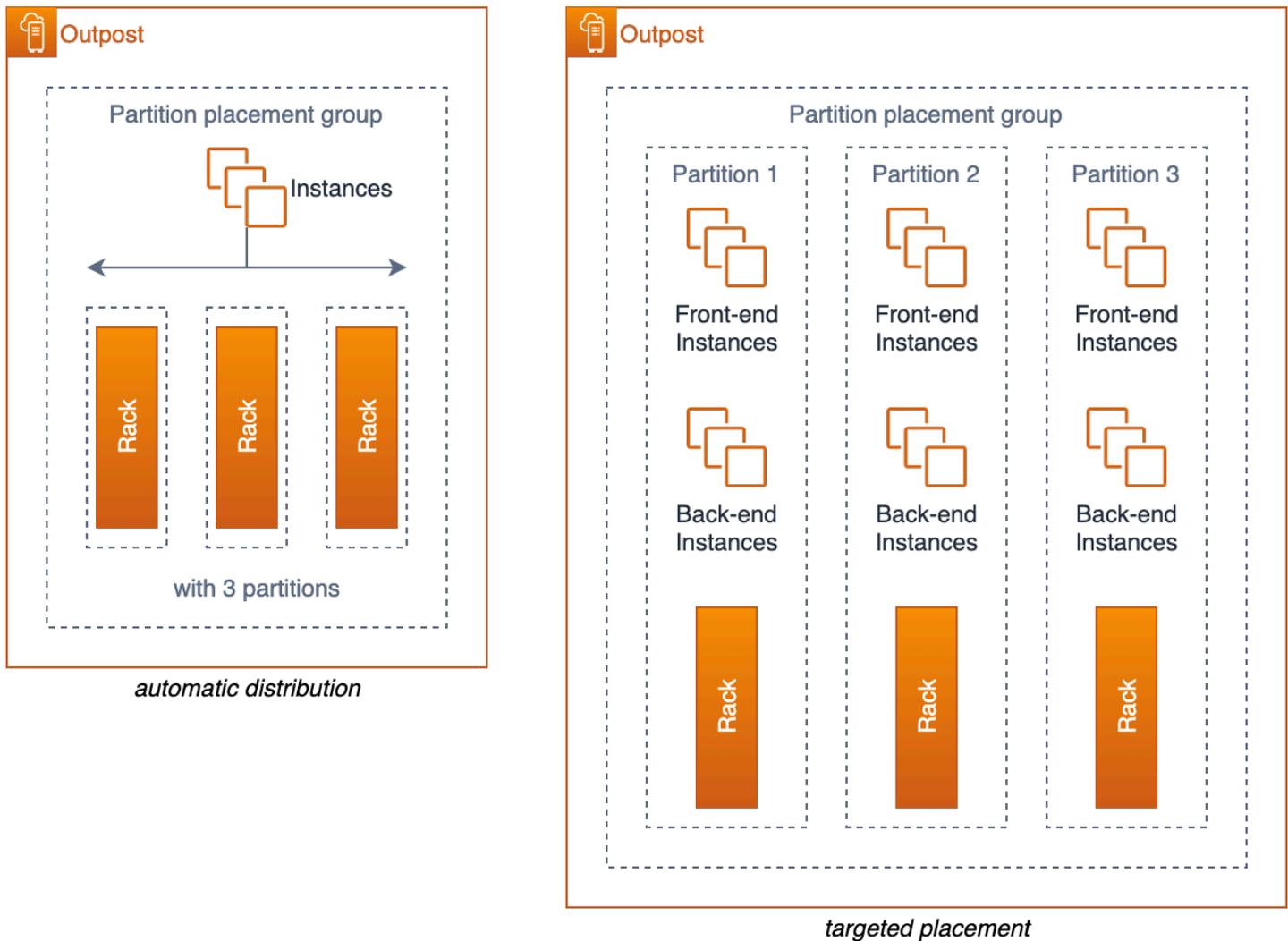
**Outpost 上的 Amazon EC2 置放群组（单 Outpost 多机架实例置放）：**允许您使用[集群](#)、[分布](#)和[分区](#)策略来影响置放。借助分布和分区置放策略，您可以在多机架 Outpost 中跨机架分配实例。

分布置放群组提供了一种跨机架分配单个实例的简单方法，可降低出现相关故障的可能性。在群组中部署的实例数量只能与 Outpost 中的机架数量相同。



### 具有三个机架的 Outpost 上的 EC2 分布置放群组

您还可以使用分区置放群组跨多个机架分配实例。使用自动分配跨群组中的分区分布实例或将实例部署到选定的目标分区。通过将实例部署到目标分区，您可以将选定的资源部署到同一个机架，同时跨机架分配其他资源。例如，如果您有一个具有三个机架的逻辑 Outpost，则通过创建一个包含三个分区的分区置放群组，您可以跨机架分配资源。



### 具有三个机架的 Outpost 上的 EC2 分区置放群组

富有创造性的服务器槽配置：如果您使用的是单机架 Outpost，或者您在 Outpost 上使用的服务不支持置放群组，则可以使用富有创造性的槽配置来确保实例不会部署在同一台物理服务器上。如果相关实例的 EC2 实例大小相同，则可以对服务器进行槽配置，以限制每台服务器上配置的该大小的槽数量，从而跨服务器分布槽。服务器槽配置将限制可在单台服务器上运行的（该大小）实例的数量。

以前文中图 13 所示的槽布局为例。如果您的应用程序需要在配置了此时段布局的 Outpost 上部署三个 m5.4xlarge 实例，EC2 会将每个实例放在单独的服务器上，并且这些实例不可能在同一台服务器上运行，前提是这些实例不可能在同一台服务器上运行，前提是时段配置不更改为在服务器上打开更多 m5.4xlarge 插槽。

### 针对计算实例置放的建议实操：

- 在 Outpost 上使用 Amazon EC2 置放群组来控制单个 Outpost 中的跨机架实例置放。

- 与其订购具有单个中型或大型 Outpost 机架的 Outpost，不如考虑将容量分成两个小型或中型机架，这样您便可以利用 EC2 置放群组功能来跨机架分配实例。

## 存储

机 AWS Outposts 架服务提供三种存储类型：

- 支持的 EC2 实例类型上的[实例存储](#)
- 用于持久块存储的 [Amazon Elastic Block Store \( EBS \) gp2 卷](#)
- 用于本地对象存储的 [Amazon Simple Storage Service on Outposts \( S3 on Outposts \)](#)

实例存储在支持的服务器 ( C5d、M5d、R5d、G4dn 和 I3en ) 上提供。就像在区域内一样，实例存储中的数据仅在[实例的 \( 运行 \) 生命周期](#)内保留。

Outposts EBS 卷和 S3 on Outposts 对象存储作为 AWS Outposts 机架托管式服务的一部分提供。客户负责管理 Outpost 存储池的容量。客户在订购 Outpost 时指定他们对 EBS 和 S3 存储的存储要求。AWS 使用提供请求的存储容量所需的存储服务器数量来配置 Outpost。AWS 负责 Outposts 上的 EBS 和 S3 存储服务的可用性。预置足够数量的存储服务器，从而为 Outpost 提供高度可用的存储服务。丢失单个存储服务器时，服务不应中断，也不应导致数据丢失。

您可以使用 AWS Management Console 和[CloudWatch 指标](#)来监控 Outposts 上的 Outpost EBS 和 [S3 的容量利用率](#)。

## 数据保护

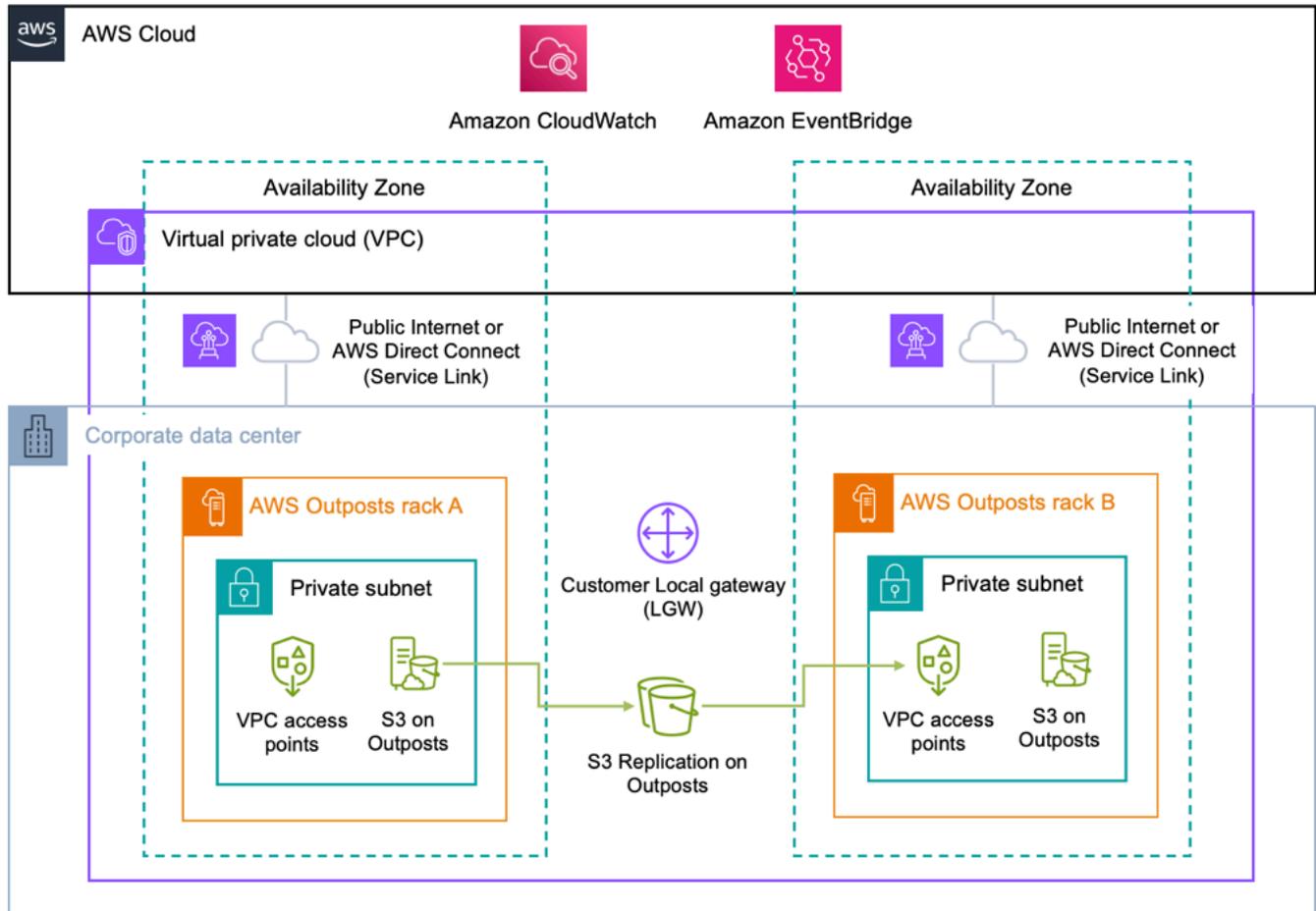
对于 EBS 卷：AWS Outposts 机架支持 EBS 卷快照，以提供简单安全的数据保护机制来保护您的块存储数据。快照是 EBS 卷的 point-in-time 增量备份。默认情况下，Outpost 上的 [Amazon EBS 卷快照](#)存储在区域内的 Amazon S3 上。如果已将 Outpost 配置为 S3 on Outposts 容量，则可以借助 [EBS Local Snapshots on Outposts](#)，将快照本地存储在使用 S3 on Outposts 存储的 Outpost 上。

对于 S3 on Outposts 桶 ( 数据驻留使用场景 )：

- 您可以使用 [Outpost 上的 S3 版本控制功能](#)来保存所有更改和对象历史记录。启用后，S3 版本控制功能将对象的多个不同副本保存到同一个存储桶中。对于 Outpost 桶中存储的每个对象，您可以使用 S3 版本控制功能来保留、检索和还原其每个版本。S3 版本控制功能可帮助您从用户意外操作和应用程序故障中恢复。
- 您可以使用 [Outpost 上的 S3 复制功能](#)来创建和配置复制规则，以将 S3 对象自动复制到其他 Outpost 或同一 Outpost 上的其他桶。在复制过程中，S3 on Outposts 对象通过客户的本地网关

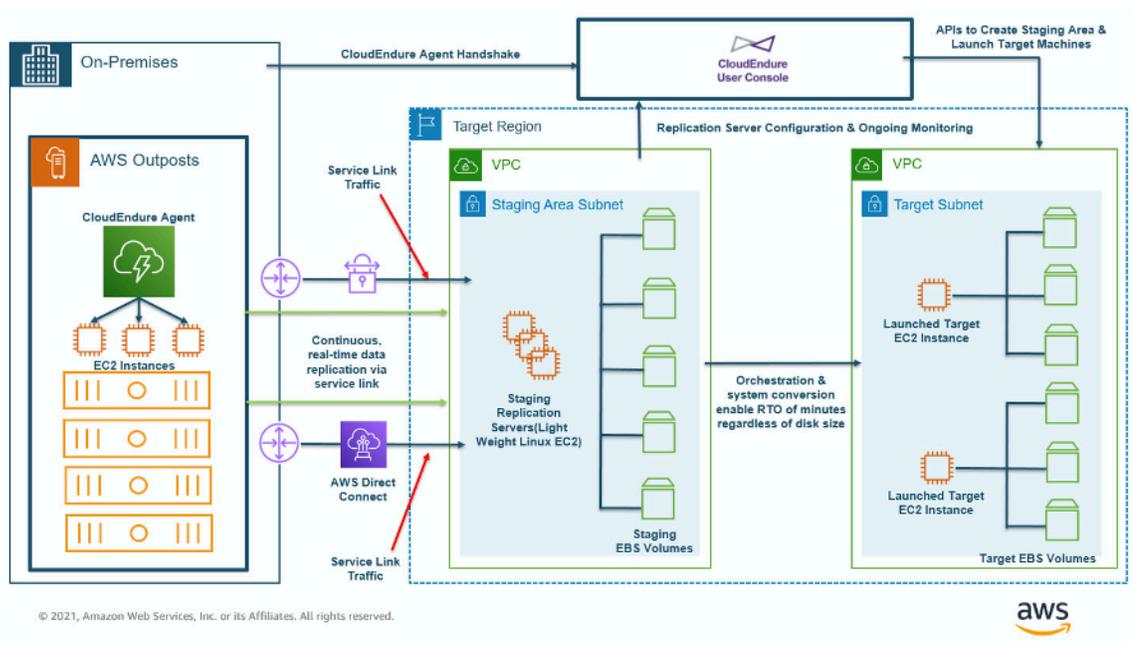
( LGW ) 发送，并且对象不会返回到 AWS 区域。Outpost 上的 S3 复制功能提供了一种简单灵活的方式，可自动复制特定数据边界内的数据，以满足数据冗余和合规性需求。

Outpost 上的 S3 复制功能还提供了详细的指标和通知，以监控对象复制的状态。您可以使用 Amazon 跟踪源和目标 Outposts 存储桶之间的待处理字节、待处理操作以及复制延迟，从而监控复制进度。CloudWatch 您还可以设置 Amazon EventBridge 规则以接收复制失败事件，从而快速诊断和更正配置问题。



对于 Outposts 上的 S3 存储桶（非数据驻留用例），可以 AWS 区域：您可以使用自动在 Outposts 上的 S3 在您的 Outposts 和地区之间传输数据。[AWS DataSync](#) DataSync 允许您选择传输什么、何时传输以及使用多少带宽。通过将本地 S3 on Outposts 桶备份到 AWS 区域中的 S3 桶中，您可以利用 99.999999999%（11 个 9）的数据持久性和其他存储层（标准、不频繁访问和 Glacier）来优化区域 S3 服务的成本。

实例复制：您可以使用将单个实例从本地系统复制[CloudEndure](#)到前哨基地、从前哨基地复制到区域、从区域复制到前哨基地，或者从一个前哨基地复制到另一个前哨基地。[AWS Outposts 带有 DR 架构的 CloudEndure](#)博客文章描述了每种场景以及如何设计解决方案。CloudEndure



## 从 Outpost 到区域的灾难恢复 ( DR )

使用 AWS Outposts 机架作为 CloudEndure 目标 ( 复制目标 ) 需要 Outposts 上的 S3 存储。

针对数据保护的建议实操：

- 使用 EBS 快照将块存储卷 point-in-time 备份到该区域的 Amazon S3 或 Outposts 上的 S3。
- 使用 S3 on Outposts 对象版本控制功能来保留对象的多个版本和历史记录。
- 使用 Outpost 上的 S3 复制功能将对象数据自动复制到其他 Outpost。
- 对于非数据驻留用例，请使用 AWS DataSync 将存储在 Outpost 上 S3 中的对象备份到该地区的 Amazon S3。
- CloudEndure 用于在本地系统、逻辑 Outposts 和区域之间复制实例。

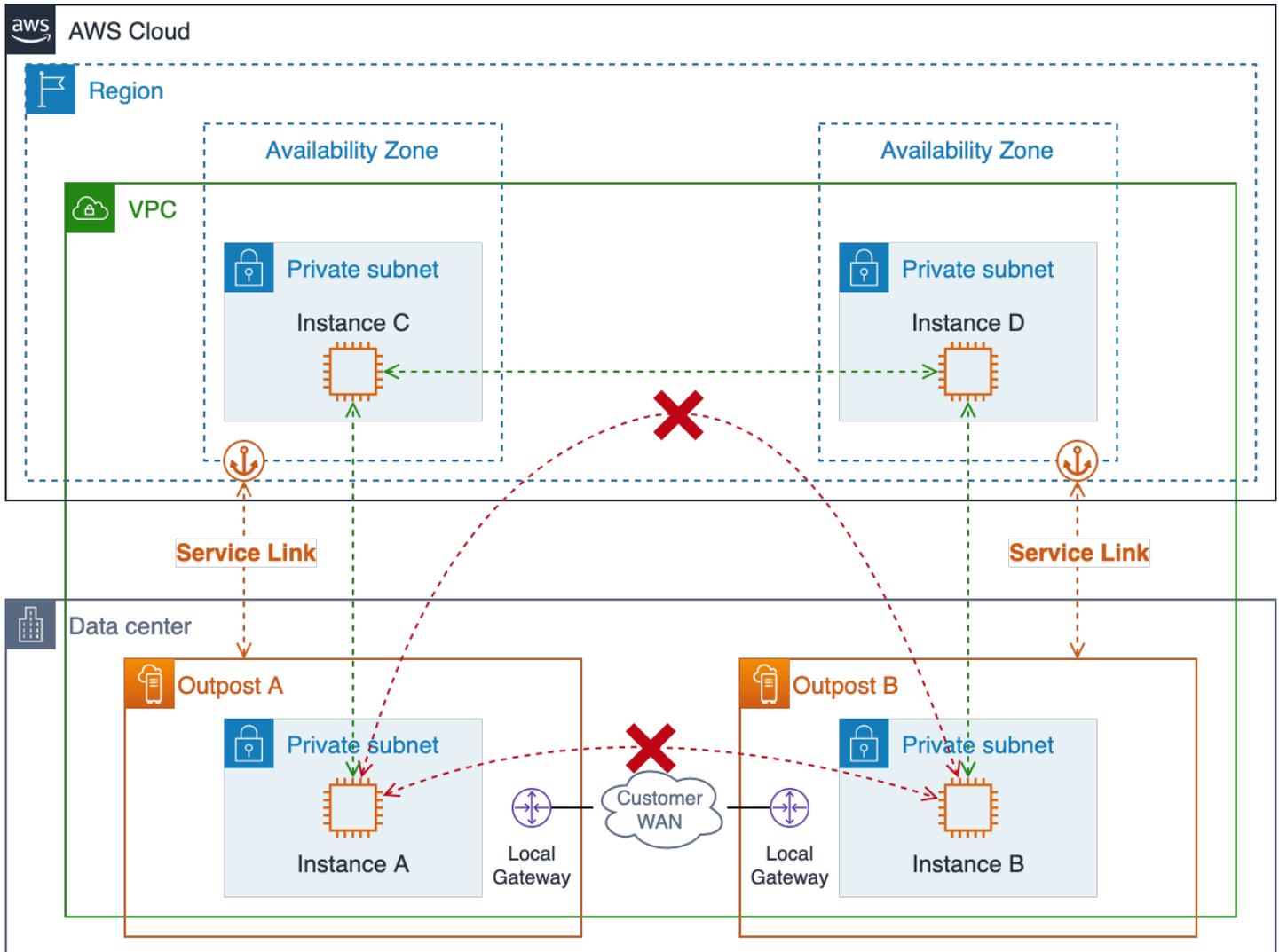
## 更大规模的故障模式

要设计 HA 架构以缓解更大规模的故障模式 [例如机架、数据中心、可用区 ( AZ ) 或区域故障]，您应该在配备独立电源和 WAN 连接的单独数据中心中部署具有充足基础设施容量的多个 Outpost。您可以将 Outpost 锚定到一个 AWS 区域内或跨多个区域的不同可用区 ( AZ )。您还应在位置之间配置弹性和足够的 site-to-site 连接，以支持同步或异步数据复制以及工作负载流量重定向。根据应用程序架

构，您可以使用全局可用的 [Amazon Route 53](#) DNS 和区域可用的 [Elastic Load Balancing](#) 服务将流量定向到所需位置，并在发生大规模故障时将流量自动重定向到剩余的位置。

在跨多个 Outpost 设计和部署应用程序工作负载时，应注意联网限制。两个独立的 Outpost 上的资源无法通过经由区域的流量传输相互通信。部署在同一 VPC 中的两个不同 Outpost 上的资源无法跨客户网络相互通信。部署在不同 VPC 中的两个独立 Outpost 上的资源可以跨客户网络相互通信。

下面的两张图片说明了受阻的和成功的网络路径。

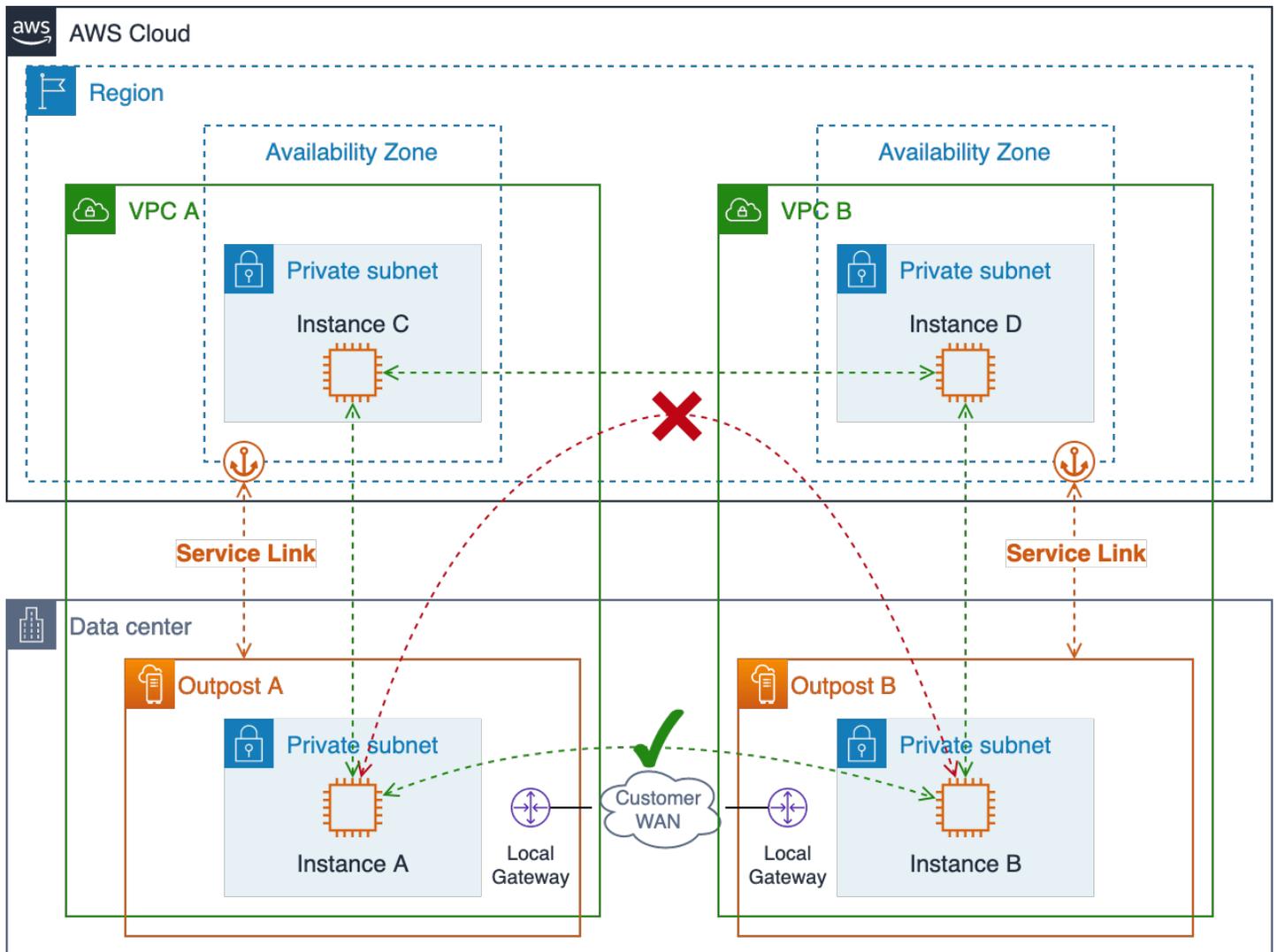


### 单 VPC 多 Outpost 网络路径

由于这是一种反模式，因此经由区域的 Outpost 到 Outpost 流量传输受阻。此类流量将产生双向输出费用，并且可能比仅跨客户 WAN 路由流量产生的延迟更高。

同一 VPC 中的多个 Outpost 上的资源无法相互通信。同一 VPC 中的不同 Outpost 之间的流量将始终沿着本地 VPC CIDR 路由穿过将会受阻的区域。

您应该使用单独的 VPC 在多个 Outpost 上部署资源，以便能够跨本地和 WAN 网络路由 Outpost 到 Outpost 流量。



## 多 VPC 多 Outpost 网络路径

针对防范更大规模的故障模式的建议实操：

- 部署多个锚定到多个 AZ 和区域的 Outpost。
- 在多 Outpost 部署中，为每个 Outpost 使用单独的 VPC。

## 结论

借助 AWS Outposts 机架，您可以使用熟悉的 AWS 工具和服务（例如亚马逊 EC2、亚马逊 EBS、Outposts 上的 Amazon S3、Amazon ECS、Amazon EKS 和 Amazon RDS）构建、管理和扩展高度可用的本地应用程序。工作负载可以在本地运行、为客户端提供服务、访问本地网络中的应用程序和系统，以及访问 AWS 区域中的全套服务。Outpost 机架非常适合需要低延迟访问本地系统、本地数据处理、数据驻留以及迁移具有本地系统相互依赖项的应用程序的工作负载。

当您为 Outpost 部署提供足够的电力、空间、冷却和弹性连接时 AWS 区域，您就可以构建高度可用的单一数据中心服务。而且，为了实现更高级别的可用性和弹性，可以部署多个 Outpost，并跨逻辑和地理边界分发应用程序。

Outposts rack 消除了构建本地计算、存储和应用程序网络池的无差别繁重的工作，并允许您将 AWS 全球基础设施的覆盖范围扩展到您的数据中心和托管设施。现在，您可以将时间和精力集中在实现应用程序现代化、简化应用程序部署和提高 IT 服务的业务影响上。

## 贡献者

本文档的贡献者包括：

- Mallory Gershenfeld，亚马逊云科技 S3 on Outposts 的 Product Manager
- Chris Lunsford，高级专业解决方案架构师 AWS Outposts，亚马逊 Web Services
- Rohan Mathews，Amazon Web Services AWS Outposts 首席架构师

# 文档历史记录

如需获取有关该白皮书更新的通知，请订阅 RSS 信息源。

变更	说明	日期
<a href="#">次要更新</a>	在容量规划中增加了额外的排档指导。	2024 年 2 月 9 日
<a href="#">次要更新</a>	更新以体现自初次发布以来的功能发布。	2023 年 7 月 19 日
<a href="#">次要更新</a>	更新了针对高度可用的网络连接的建议实操。	2023 年 6 月 29 日
<a href="#">初次发布</a>	白皮书首次发布。	2021 年 8 月 12 日

## Note

要订阅 RSS 更新，您必须为当前使用的浏览器启用 RSS 插件。

## 版权声明

客户有责任对本文档中的信息进行单独评测。本文件：(a) 仅供参考，(b) 代表当前 AWS 的产品供应和做法，如有更改，恕不另行通知，以及 (c) 不产生其关联公司、供应商或许可方的任何承诺或保证。AWS 产品或服务“按原样”提供，不附带任何形式的担保、陈述或条件，无论是明示还是暗示。对客户的所有责任和责任由 AWS 协议控制，本文档不属于其客户之间的任何协议，也不会对其 AWS 进行修改。AWS

© 2023 , Amazon Web Services, Inc. 或其附属公司。保留所有权利。

# AWS 词汇表

有关最新 AWS 术语，请参阅《AWS 词汇表 参考资料》中的[AWS 词汇表](#)。

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。