部署 Amazon AppStream 2.0 的最佳实践

部署 Amazon AppStream 2.0 的最佳实践:

Table of Contents

Amazon AppStream 2.0 部署最佳实践

发布日期：2022 年 1 月 19 日 (文档修订)

摘要

本白皮书概述了一系列 Amazon AppStream 2.0 部署的最佳实践。本白皮书涵盖了 Amazon 虚拟私有云 (VPC) 设计、映像创建和管理、实例集定制以及实例集自动扩缩策略。它包括用户连接方法、身份验证以及与 Microsoft Active Directory 的集成。本白皮书还包括有关设计 AppStream 2.0 安全、监控和成本优化的建议。

编写本白皮书是为了便于快速访问相关信息。它适用于网络工程师、应用程序交付专家、目录工程师或安全工程师。

简介

Amazon AppStream 2.0 是一种可完全托管的应用程序流式传输服务，使用户能够随时随地即时访问其桌面应用程序。AppStream 2.0 负责管理托管和运行应用程序所需的 AWS 资源。它可以自动扩缩，并根据需要为您的用户提供访问权限。AppStream 2.0 使最终用户能够在其选择的设备上访问所需的应用程序，其响应灵敏的用户体验与本地安装应用程序别无二致。

以下各节详细介绍了 Amazon AppStream 2.0，解释了该服务的工作原理，描述了启动该服务所需的条件，并告诉您有哪些选项和特征可供您使用。在为最终用户部署 AppStream 2.0 时，实施最佳实践非常重要，可以提供卓越的用户体验。此外，各种规模的公司都能从成本优化中受益，从而降低每月的运营成本。

重要概念

要让 AppStream 2.0 充分发挥作用，请熟悉以下概念：

- 映像 — 映像是预先配置的实例模板。映像中包含可流式传输到您的用户的应用程序以及可使用户能快速开始使用这些应用程序的默认 Windows 和应用程序设置。AWS 提供基本映像，您可以使用这些映像来创建包含您自己应用程序的映像。您在创建映像后无法更改映像。要添加其他应用程序、更新现有应用程序或更改映像设置，您必须创建新映像。您可以将映像复制到其他 AWS 区域，或与相同区域中的其他 AWS 账户 共享这些映像。
- 映像生成器 — 映像生成器是用来创建映像的虚拟机。您可以使用 AppStream 2.0 控制台启动并连接映像生成器。连接到映像生成器之后，您可以安装、添加和测试您的应用程序，然后使用映像生成器创建映像。您可以使用您拥有的私有映像启动新映像生成器。
- 实例集 — 实例集由运行您指定的映像的实例集实例（也称为流式处理实例）组成。您可以为实例集设置所需数量的流式处理实例，并配置策略来根据需求自动缩放实例集。请注意，一个用户需要一个实例。
- 堆栈 — 堆栈由关联的实例集、用户访问策略和存储配置组成。您通过设置堆栈开始将应用程序流式传输给用户。
- 流式处理实例 — 流式处理实例（也称为实例集实例）是 Amazon Elastic Compute Cloud (Amazon EC2) 实例，可供单个用户用于应用程序流式传输。用户会话完成后，Amazon EC2 将终止该实例。

VPC 设计

设计指南

将 AppStream 2.0 部署到专用 VPC 中。在设计 AppStream 2.0 VPC 时，请考虑预计会增长的大小。为新的使用案例以及稍后可能添加的其他可用区 (AZ) 保留 IP 地址容量。AppStream 2.0 的一个基本设计要点是，一个 AppStream 2.0 实例只能由一个用户使用。在分配 IP 空间时，请考虑每个 AppStream 2.0 实例一个 IP 地址，对应一个用户。在 AppStream 2.0 中，用户可能使用多个 AppStream 2.0 实例。因此，规划 IP 空间时还必须考虑到需要额外的 AppStream 2.0 实例的使用案例。

尽管采用无类别域间路由 (CIDR) 的 VPC 的最大大小为 /16，但 AWS 建议不要过度分配私有 IP 地址。可以通过其他 CIDR 扩展 VPC 的大小，但这是有限的；因此，请从一开始就分配所需的 IP 地址。

如果 AppStream 2.0 部署已加入 Active Directory 域，则为 VPC 设置的 DHCP 选项必须配置域 DNS。域名服务器应指定对 Active Directory 域具有权威性的 DNS IP 地址，或者 DNS 应将 DNS 请求转发到对 Active Directory 域有权威性的 DNS 实例。此外，VPC 必须已配置 `enableDnsHostnames` 和 `EnableDnsSupport`。

可用区

可用区 (AZ) 是 AWS 区域 中一个或多个具有冗余电源、网络和连接的离散数据中心。与传统的单个或多个数据中心基础设施相比，可用区具有更高的可用性、容错性和可扩展性。

Amazon AppStream 2.0 只需要一个子网即可启动实例集。最好是至少配置两个可用区，每个唯一可用区一个子网。要优化实例集自动扩缩，请使用两个以上的可用区。横向扩展的另一个好处是可以增加子网中的 IP 空间，以满足增长的需要，本文档的以下子网大小调整部分对此进行了介绍。使用 AWS 管理控制台创建实例集期间，只能指定两个子网。使用 AWS Command Line Interface (AWS CLI) 或 AWS CloudFormation 可允许指定超过两个子网 ID。

子网大小调整

为 AppStream 2.0 实例集指定子网，可实现路由策略和网络访问控制列表的灵活性。堆栈可能会有单独的资源需求。例如，AppStream 2.0 堆栈可能有隔离要求，从而建立了单独的规则集。当多个 Amazon AppStream 2.0 实例集使用相同的子网时，请确保所有实例集的最大容量之和不超过可用 IP 地址的总数。

如果同一子网中所有实例集的最大容量可能或已经超过可用 IP 地址的总数，请将实例集迁移到专用子网。这样可以防止自动扩缩事件耗尽分配的 IP 空间。如果实例集的总容量超过分配子网分配的 IP 空间，请使用 API 或 AWS CLI“更新实例集”来分配更多子网。有关更多信息，请参阅 Amazon VPC 配额以及如何提高配额。

最好是扩展子网的数量，相应地调整子网的大小，同时在 VPC 中预留容量以供增长。此外，请确保 AppStream 2.0 实例集的最大值不超过子网分配的 IP 空间总量。在计算 IP 空间总量时，为 AWS 中的每个子网保留五个 IP 地址。使用两个以上的子网并进行横向扩展具有多种好处，例如：

- 提高可用区的故障恢复能力
- 自动扩缩实例集实例时吞吐量更高
- 更有效地使用私有 IP 地址，避免 IP 地址流失

在调整 Amazon AppStream 2.0 的子网规模时，请考虑子网的总数以及利用率峰值期间的预计峰值并发量。可以使用 (`InUseCapacity`) 加上实例集的预留容量 (`AvailableCapacity`) 进行监控。在 Amazon AppStream 2.0 中，已使用和可使用的 AppStream 2.0 实例集实例的总和标记为 `ActualCapacity`。要正确调整总 IP 空间的大小，请预测所需的 `ActualCapacity`，然后除以分配给实例集的子网数量，减去一个子网以满足恢复需要。

例如，如果预计峰值时实例集实例的最大数量为 1000，并且业务要求是在一次可用区故障中保持恢复能力，则 3 个 x/23 子网可以满足技术和业务需求。

- /23 = 512 台主机 — 5 个预留 = 每个子网 507 个实例集实例
- 3 个子网 — 1 个子网 = 2 个子网
- 2 个子网 x 每个子网 507 个实例集实例 = 峰值时为 1014 个实例集实例

子网大小调整示例

虽然 2 个 /22 子网也能满足恢复的需求，但请考虑以下几点：

- 使用两个可用区保留的不是 1536 个 IP 地址，而是保留了 2048 个 IP 地址，从而浪费了本来可以用于其他功能的 IP 地址。
- 如果有一个可用区无法访问，则扩展实例集实例的能力将受到可用区吞吐量的限制。这可能延长 `PendingCapacity` 的持续时间。

子网路由

最佳实践是为 AppStream 2.0 实例创建私有子网，通过集中式 VPC 将出站流量路由到公共互联网。可通过 Amazon AppStream 2.0 服务凭借流式传输网关处理 AppStream 2.0 会话流式传输的入站流量：您无需为此配置公有子网。

区域内连接

对于加入 Active Directory 域的 AppStream 2.0 实例集实例，请在每个 AWS 区域 的共享服务 VPC 中配置 Active Directory 域控制器。Active Directory 的来源可以是基于 Amazon EC2 的域控制器或 AWS 托管的 Microsoft AD。共享服务与 AppStream 2.0 VPC 之间的路由可以通过 VPC 对等连接或中转网关进行。尽管中转网关解决了大规模路由的复杂性，但 VPC 对等连接在大多数设置中更受青睐的原因有很多：

- VPC 对等连接是两个 VPC 之间的直接连接（无需额外跳跃）。
- 不收取每小时费用，仅按可用区之间的标准数据传输费率收费。
- 对带宽没有限制。
- 支持在 VPC 之间访问安全组。

如果 AppStream 2.0 实例连接到共享服务 VPC 中具有大型数据集的应用程序基础设施和/或文件服务器，则尤其如此。因为可以优化通向这些常用资源的路径，VPC 对等连接是首选，即使在所有其他 VPC 和互联网路由都通过中转网关执行的设计中也是如此。

出站互联网流量

虽然直接路由到共享服务主要通过对等连接进行优化，但 AppStream 2.0 的出站流量可以通过使用 AWS 中转网关从多个 VPC 创建单个互联网出口点来设计。在多 VPC 设计中，标准做法是使用专用 VPC 来控制所有传出的互联网流量。通过这种配置，中转网关可以更灵活地控制通过连接到子网的标准路由表进行路由。该设计还支持传递路由，而不会增加复杂性，并且无需在每个 VPC 中使用冗余网络地址转换 (NAT) 网关或 NAT 实例。

将所有出站互联网流量集中到单个 VPC 后，NAT 网关或 NAT 实例就是常见的设计选择。要确定哪个最适合您的组织，请查看比较 NAT 网关和 NAT 实例的管理指南。AWSNetwork Firewall 可以在路由级别进行保护，并在 OSI 模型 中提供第 3 至第 7 层的无状态和有状态规则，从而将保护范围扩展到安全组和网络访问控制级别之外。有关更多信息，请参阅 AWS Network Firewall 的部署模型。如果您的组织选择了执行 URL 筛选等高级特征的第三方产品，请将该服务部署到您的出站互联网 VPC 中。这可以取代 NAT 网关或 NAT 实例。请遵循第三方供应商提供的指南。

本地

当需要连接到本地资源时，尤其是对于加入 Active Directory 的 AppStream 2.0 实例，请通过 AWS Direct Connect 建立恢复能力极高的连接。

VPC 端点

Amazon S3 VPC 端点

许多 Amazon AppStream 2.0 部署都需要通过主文件夹和应用程序设置保持用户状态。启用与这些 Amazon Simple Storage Service (Amazon S3) 位置的私密通信，因为这样可以避免使用公共互联网。您可以通过 VPC 端点网关实现此目的。对于 Amazon S3 来说，VPC 端点网关比 AWS PrivateLink 更受青睐，因为：

- 它已针对 AppStream 2.0 网络访问要求进行了成本优化
- 无需从本地资源访问 Amazon S3 存储桶
- 自定义策略文档可以用于限制单独从 AppStream 2.0 实例发起的访问

创建 VPC 端点网关后，最好是通过创建自定义策略来保护私有化连接。自定义策略以 AppStream 2.0 服务 Identity and Access Management 角色的 Amazon 资源名称 (ARN) 开头。明确指定保持用户状态所需的 S3 操作。

> **Note**
>
> `Resources` 部分中的以下示例首先指定状态主文件夹路径，然后指定应用程序设置路径。

Example

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Allow-AppStream-to-access-home-folder-and-
application-settings",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:sts::account-id-without-hyphens:assumed-
role/AmazonAppStreamServiceAccess/AppStream2.0"
      },
      "Action": [
        "s3:ListBucket",
        "s3:GetObject",
```

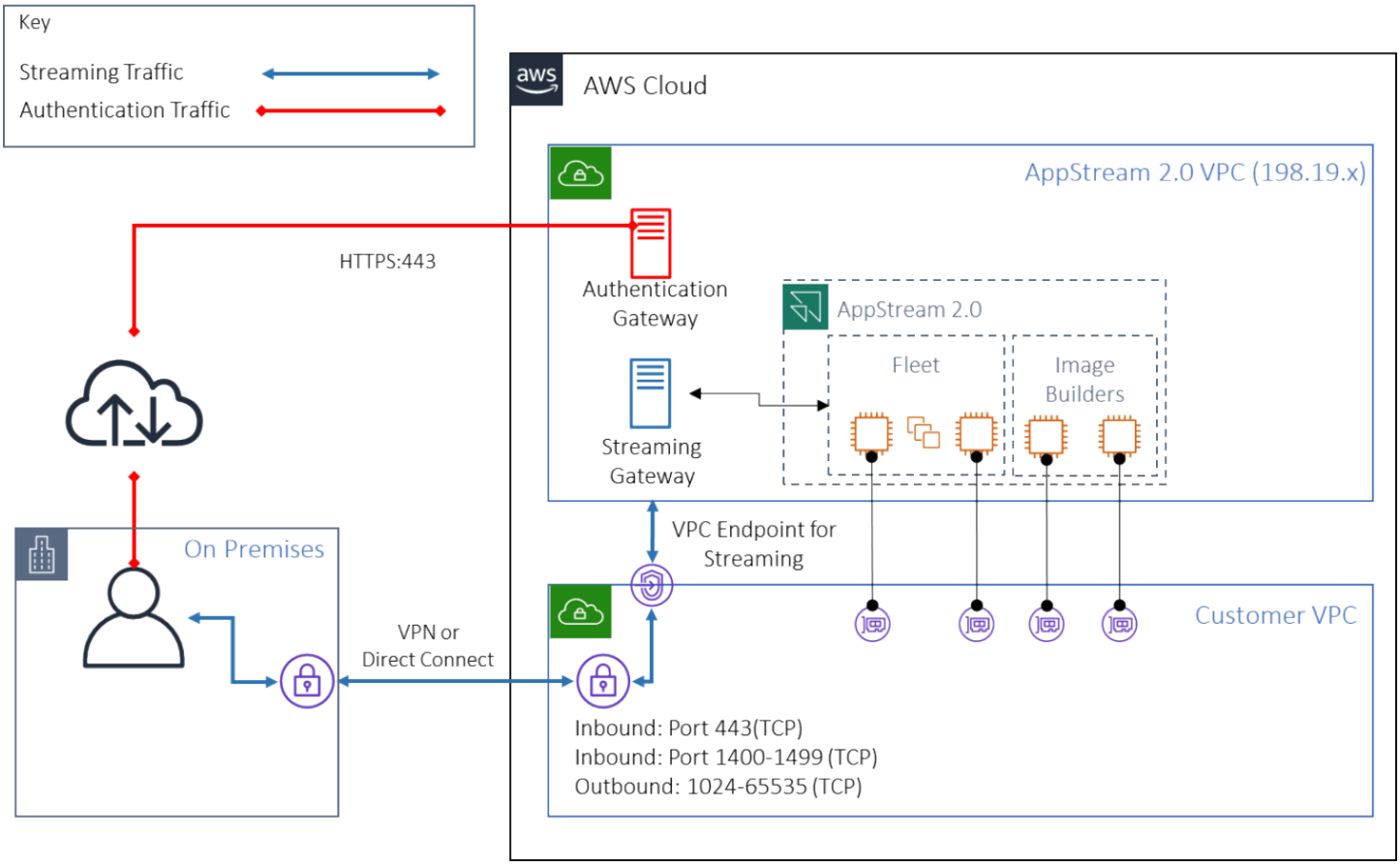
```
        "s3:PutObject",
        "s3:DeleteObject",
        "s3:GetObjectVersion",
        "s3:DeleteObjectVersion"
      ],
      "Resource": [
        "arn:aws:s3:::appstream2-36fb080bb8-*",
        "arn:aws:s3:::appstream-app-settings-*"
      ]
    }
  ]
}
```

Amazon AppStream 2.0 API 接口 VPC 端点

在 Amazon AppStream 2.0 的 API 和 CLI 命令源自 VPC 的设计场景中，请通过接口 VPC 终端节点将这些编程调用私有化。

Amazon AppStream 2.0 流式传输接口 VPC 端点

虽然可以通过接口 VPC 端点路由 Amazon AppStream 2.0 流式传输流量，但请谨慎使用此配置。通过公共互联网的默认流式传输行为是 Amazon AppStream 2.0 流式传输流量最高效、性能最高的传输方式。



Amazon AppStream 2.0 流式传输接口 VPC 端点

如上图所示，公共互联网是通往 Amazon AppStream 2.0 流式传输网关的最高效路径。通过客户托管的 VPC 和网络进行路由会增加复杂性和延迟。它还会增加 Direct Connect 的数据传输费用。

> **Note**
>
> VPC 端点仅支持流式传输，并且身份验证仍必须通过公共互联网进行。要进行 SAML 单点登录 (SSO) 之类的访问，身份提供者 (IdP) 仍然是必要先决条件，只能通过公共互联网访问。

映像创建和管理

在 AppStream 2.0 中启动实例集或映像生成器时，必须选择一个 AppStream 2.0 基础映像。然后，管理员可以在基础映像的基础上进行构建，添加自己的应用程序和配置设置。

在构建映像时，需要考虑一些关键因素，以确保应用程序正常、安全地运行。此外，对于如何维护该映像，还有一些设计方面的考虑。

构建 AppStream 2.0 映像

构建新映像时，您必须要考虑以下各项：

- 操作系统
- 应用程序
- 用户配置文件
- 安全性
- 性能
- 代理版本
- Image Assistant CLI

构建 AppStream 2.0 映像

2021 年 11 月，AppStream 2.0 开始支持 Amazon Linux 2。继这次发布后，AppStream 2.0 现已支持四种类型的平台：

- Windows Server 2012 R2
- Windows Server 2016
- Windows Server 2019
- Amazon Linux 2

您可能必须根据应用程序的要求选择特定的平台（例如，如果您的应用程序需要 Windows，则无法选择 Amazon Linux 2）。除了应用程序要求外，还可以参考以下比较表，可帮助您根据使用案例和环境选择最合适的平台类型：

表 1 — 平台类型、何时使用以及定价

平台类型	何时使用	实例集定价*
Windows Server（2012 R2、2016 或 2019）	您的应用程序只能在 Windows 中运行（而且它不支持 Amazon Linux 2）。您希望流式处理实例加入域。您希望在 AppStream 2.0 流式处理实例上使用现有组策略（Linux 不遵循组策略，但您可以使用会话脚本在会话开始时自动进行配置）。您将使用桌面视图，您的用户更喜欢 Windows 桌面体验。您更喜欢使用 Image Assistant 应用程序，它提供分步向导，用于创建应用程序目录和映像。目前，您必须使用终端命令创建 Amazon Linux 2 映像（有关更多信息，请参阅本教程）。您想使用应用程序设置持久性。基于 Linux 的堆栈目前不支持启用应用程序设置持久性。	每位唯一用户每月 4.19 美元的 RDS SAL（Microsoft 远程桌面服务订阅者访问许可证）许可费**，外加以下费用： 1. Always-On、On-Demand 实例集每小时 0.10 美元 2. Elastic 实例集每小时 0.15 美元
Amazon Linux 2	您想以尽可能更低的成本流式传输实例并避免 RDS SAL 许可费。您的应用程序与 Amazon Linux 2 兼容	与 Windows 实例相比，Linux 实例的价格更低。使用 Linux，您无需支付 RDS SAL 许可法费，只需按小时支付以下费用： 1. Always-On、On-Demand 实例集每小时 0.084 美元： 2. Elastic 实例集每小时 0.112 美元

* 基于弗吉尼亚北部地区的 stream.standard.medium

** 符合条件的客户可以自带许可证，免除 AWS RDS SAL 许可费。如需了解更多详情，请参阅 AppStream 2.0 定价页面。教育领域客户也可能有资格享受特别优惠。学校、大学和某些公共机构可能有资格享受减免的 Microsoft RDS SAL 用户费。

应用程序

在安装应用程序之前，请务必查看应用程序要求，例如应用程序依赖项和硬件要求。在映像生成器实例上成功安装应用程序后，请确保在测试用户环境下切换用户和测试应用程序。

在规划应用程序部署时，请注意服务端点和限额。此外，在创建映像之前，请清理安装程序和帮助文件以优化 C 盘的总空间。提醒一下，AppStream 2.0 实例有一个 200 GB 的固定大小的卷。为确保永远不会超过固定大小的容量，最好是在安装后优化磁盘空间。

如果您想修改用户可以实时访问的应用程序目录，动态应用程序框架提供了 API 操作。动态应用程序提供程序所管理的应用程序可以位于映像中，也可以位于实例外，例如来自 Windows 文件共享或应用程序虚拟化技术。此特征需要已加入 Microsoft Active Directory 域的 AppStream 2.0 实例集。有关更多信息，请参阅将 Active Directory 与 AppStream 2.0 结合使用。

应用程序块

应用程序块代表启动用户将要使用的应用程序所需的安装脚本和应用程序文件。虚拟硬盘 (VHD) 可以是从 Amazon S3 下载的任何对象。建议将此对象的容量小于 1.5GB，因为用户必须将其完全下载后才能访问应用程序。

优化应用程序块

对于基于 Windows 的实例集，建议您创建一个 VHDX 文件来包含您的应用程序。对于基于 Linux 的实例集，建议您创建一个映像 (IMG)。这些虚拟磁盘应创建得尽可能小，以便存放应用程序文件。可以对虚拟磁盘进行压缩，进一步减小其大小。在安装脚本中，您需要先解压缩磁盘，然后再进行安装。示例 Windows PowerShell 安装脚本包含解压缩功能。在存档 (zip) 解压速度和下载速度之间需要权衡一下。可能需要进行一些测试，才能找到能够提供最快应用程序启动时间的平衡点。

更新应用程序

应用程序更新可以分为次要和主要更新。要进行次要更新，请在托管应用程序块文件的 Amazon S3 存储桶上使用启用版本控制。此设置允许管理员通过更改相关应用程序 VHD 对象的版本，回滚到特定应用程序的先前版本，而无需更改应用程序块配置。进行主要更新时，请为更新后的 VHD 创建一个新的应用程序块。这可以让管理员在应用程序块级别而非版本控制级别单独进行主要的应用程序更新，这为管理员管理应用程序提供了一种更有条理的方法。

用户配置文件自定义

Amazon AppStream 2.0 是一种非持久性应用程序和桌面解决方案，这是故意设计的。当用户会话终止时，系统和用户更改也会终止。仅在需要时才启用应用程序设置持久性。这可能会增加登录过程的开销，以及所需的 S3 存储的成本。

在需要应用程序设置持久性的情况下，AWS 建议通过自定义策略和 S3 VPC 网关端点保护该连接。衡量应用程序设置的总体大小，并尽量减少应用程序设置持久性中保存的设置，从而优化成本和性能。

可以在 AppStream 2.0 映像生成器实例上进行用户配置文件的自定义配置。这包括添加和修改注册表项、添加文件以及其他特定于用户的配置。在 AppStream 2.0 Image Assistant 中，有一个创建用户配置文件的选项。它可以将模板用户配置文件复制到默认用户配置文件中。将映像部署到实例集后，从实例集流式传输会话的最终用户将根据默认用户配置文件创建其用户配置文件。请务必考虑最大限度减少用户配置文件的大小，尤其是在启用“应用程序设置持久性”的情况下。默认情况下，用户配置文件的最大 VHDx 大小为 1 GB。每次启动流会话时，都需要从 S3 存储桶下载用户配置文件 VHDx 文件。这会延长流会话的准备时间，并带来超限风险，导致使用 VHDx 文件挂载用户配置文件失败。

对于所需用户配置文件大于 1 GB 的使用案例，AWS 建议使用其他方法来存储配置文件。例如，在共享存储上使用漫游配置文件或 FSLogix 配置文件容器，例如适用于 Windows File Server 的 Amazon FSx。有关更多信息，请参阅使用适用于 Windows File Server 的 Amazon FSx 和 FSLogix 来优化 Amazon AppStream 2.0 上的应用程序设置持久性。

安全性

开发人员需要考虑不同的安全措施。AppStream 管理员负责安装和维护 Windows 操作系统、应用程序及其依赖项的更新。有关使基础映像保持最新状态的其他指导，请参阅将 AppStream 2.0 映像保持最新，以获取有关使基础映像保持最新状态的更多指导。

默认情况下，AppStream 2.0 让用户或应用程序可以在实例上启动任意程序，超出映像应用程序目录中指定的范围。当您的应用程序在工作流程中需要依赖其他应用程序，但您不希望用户能够直接启动该依赖应用程序时，这很有用。例如，您的应用程序需要启动浏览器以提供来自应用程序供应商网站的帮助说明，但您不希望用户直接启动浏览器。在某些情况下，您可能需要控制可以在流式处理实例上启动哪些应用程序。Microsoft AppLocker 是一种应用程序控制软件，它使用明确的控制策略来启用或禁用用户可以运行的应用程序。

防病毒软件可能会对流会话和映像生成器实例产生不利影响。AWS 建议您不要为防病毒软件启用自动更新。有关 Windows Defender 的更多信息，请参阅防病毒软件。

性能

在创建新映像之前，必须以测试用户身份测试应用程序。以测试用户身份进行测试可以确保应用程序可以在非管理员用户环境下运行。此外，还可使用任务管理器和性能监视器等内置工具查看应用程序性能和用户体验。最好可以监控 CPU、内存和 GPU 内存等资源的使用率。如果存在 CPU、内存或 GPU 内存资源限制，请考虑升级实例类型。要提高性能，请执行以下操作：

- 禁用浏览器弹出窗口
- 禁用 IE 增强的安全配置

AppStream 2.0 代理版本选择

创建新映像时，您可以选择使用最新的 AppStream 2.0 代理软件，也可以选择不更新。每个版本的 AppStream 2.0 代理软件都包含错误修复和增强特征。请使用最新的软件保留您的映像。请在本文档的映像更新部分中查看相关机制。

您可以选择使用最新代理选项。此选项可确保在启动时始终安装最新的 AppStream 2.0 代理。但是，计划外的更新可能会影响用户体验，代理更新可能会延长启动实例的时间。更新基础映像需要重新创建该映像。同样重要的是，在将更新的映像部署到生产环境之前进行测试，可以最大限度地缩短启动时间。

Image Assistant 命令行界面（CLI）

对于想要自动或以编程方式创建 AppStream 2.0 映像的开发人员，请使用 Image Assistant CLI。2019 年 7 月 26 日或之后发布的 AppStream 2.0 代理软件的映像生成器上提供该功能。以下概述描述了以编程方式创建 AppStream 2.0 映像的过程。

1. 使用应用程序安装自动化通过映像生成器安装所需的应用程序。此安装可能包括用户将启动的应用程序、所有依赖项和后台应用程序。
2. 确定要优化的文件和文件夹。
3. 如果适用，请使用 Image Assistant `add-application` CLI 操作为 AppStream 2.0 映像指定应用程序元数据和优化清单。
4. 要为 AppStream 2.0 映像指定其他应用程序，请根据需要为每个应用程序重复步骤 1 到 3。
5. 如果适用，请使用 Image Assistant `update-default-profile` CLI 操作覆盖默认 Windows 配置文件，并为用户创建默认应用程序和 Windows 设置。
6. 使用 Image Assistant `create-image` CLI 操作来创建映像。

有关更多信息，请参阅使用 Image Assistant CLI 操作以编程方式创建 AppStream 2.0 映像。

管理用户的流式传输体验

使用会话脚本进行自定义

AppStream 2.0 提供实例上的会话脚本。当用户的流会话中发生特定事件时，您可以使用这些脚本来运行自己的自定义脚本。例如，您可以在用户的流会话开始之前使用自定义脚本来准备 AppStream 2.0 环境。您还可以在用户完成其流会话后使用自定义脚本清理流实例。

在 AppStream 2.0 映像中指定会话脚本。有关配置会话脚本的更多信息，请查看管理指南中关于使用会话脚本管理用户体验的部分。与网络共享或 AWS Identity and Access Management (IAM) 配置文件配合使用，您可以使用会话脚本从存储位置检索其他脚本。有了这个额外的脚本，您可以进一步优化用户体验。这可以最大限度地减少向用户交付应用程序环境所需的映像和实例集数量。

使用 Active Directory 组策略

如果您计划在 Active Directory 域中使用 AppStream 2.0 实例集，则可以使用组策略对象 (GPO) 来管理用户体验。可以将 GPO 分配给在其中创建 AppStream 2.0 实例的组织单位 (OU)。要简化映像创建，请在禁止继承的 OU 中启动 AppStream 2.0 基础映像。这样可以防止其他域策略影响 AppStream 2.0 用户体验。将每个实例集部署到其专门的 OU 中，使用独特的 GPO 来建立环境，从而获得 AppStream 2.0 映像管理的一对多整合优势。

要通过映像集为每个 AppStream 2.0 实例集指定不同的 Internet Explorer 主页，可以使用组策略。

映像更新

进行软件修补对于确保计算资源的安全性和性能至关重要。在 Well-Architected Framework 的安全性支柱中，定期修补被列为最佳实践之一。

在构建和部署映像时，有四类软件需要通过 AppStream 2.0 映像进行修补：

- 应用程序和依赖项 — 负责修补映像中的应用程序和依赖项。
- 微软 Windows 操作系统 — 负责安装和维护 Windows 的更新。
- 软件组件 — 这些是 AppStream 2.0 操作所需的驱动程序、代理和其他软件（例如，Amazon CloudWatch 代理）。AppStream 2.0 会定期发布包含新代理和驱动程序的新基础映像。您可以使用最新的基础映像重建映像，使其映像上的软件组件达到最新基准。当有许多应用程序或安装复杂的应用程序时，在最新基础映像上重建映像的过程可能既耗时又繁琐。

- AppStream 2.0 代理 — 您可以在 Image Assistant 中选择始终使用最新代理版本。使用此选项，从映像启动的流式处理实例会自动使用最新版本的代理。

您可以通过执行以下任一操作来使 AppStream 2.0 映像保持最新状态：

- 使用托管 AppStream 2.0 映像更新来更新映像 — 此更新方法提供最新的 Windows 操作系统更新和驱动程序更新以及最新的 AppStream 2.0 代理软件。此托管方法支持更新服务和 Microsoft 操作系统组件，但它不支持您更新应用程序组件。当应用程序安装很复杂或需要手动配置时，最好使用此方法。
- 使用托管 AppStream 2.0 映像版本更新 AppStream 2.0 代理软件 — 此更新方法提供了最新的 AppStream 2.0 代理软件。此方法支持您更新应用程序组件。

实例集定制

实例集类型

创建实例集时，客户必须选择实例集类型。每种实例集类型在优化用户体验、成本和维护开销方面都有不同的优势。无论选择哪种实例集类型，每个选项都支持 Windows 和 Linux 平台类型，以及桌面视图或应用程序视图。

客户现在可以从以下实例集类型中进行选择：

- Always-On — 此实例集类型为用户提供对其应用程序的即时访问权。即使没有任何用户流式传输应用程序，您也需要为实例集中所有正在运行的实例付费。
- On-Demand — 选择此实例集类型可优化您的流式传输成本。使用 On-Demand 实例集，用户开启会话的时间约为一到两分钟。但是，只有在用户连接时才会向您收取流式处理实例费用，而对于实例集中不是流式传输应用程序的每个实例，则需要按小时支付少量费用。
- Elastic — Elastic 实例集可用于不需要安装且可以从虚拟硬盘 (VHD) 运行的应用程序。Elastic 实例集不支持 AppStream 2.0 映像，也不需要扩缩策略。您只需为流会话的持续时间付费。

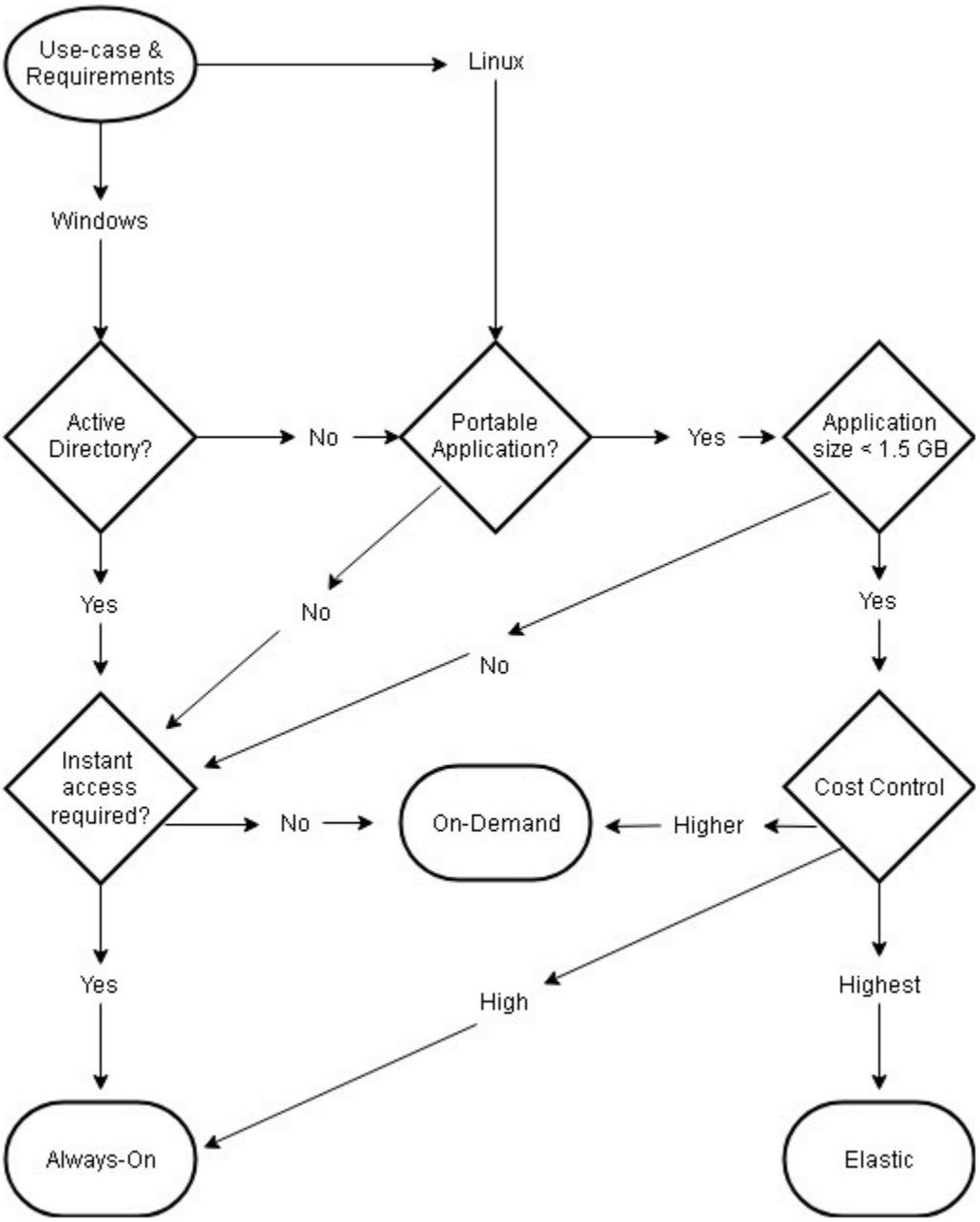
表 2 — Amazon AppStream 2.0 实例集类型

实例集类型	何时使用	用户体验	定价模型	备注
Always-On	您的用户在启动会话时需要即时访问应用程序。也许因为您的使用模式是可预测的，并且您可以通过扩缩策略可靠地控制成本，因此您的实例集不会出现明显的容量过剩。	即时访问应用程序	您需要为实例集中的每个可用实例支付全额费用（无论该实例是否用于会话）。	支持自定义映像和扩缩策略。
On-Demand	您必须在实例集中保持大量过剩	用户在启动会话后要等待一到两	您只需为会话处于活动状态的流	支持自定义映像和扩缩策略。

实例集类型	何时使用	用户体验	定价模型	备注
	容量。您希望将环境成本最优化，并且不想为未使用的容量支付全额费用。您的用户可以在启动会话后等待一到两分钟才能访问其应用程序。您使用的实例类型较多。持续运行的实例的每小时费用比已停止的实例费用昂贵得多。	分钟才能访问其应用程序。	式传输实例支付全额费用，然后为空闲实例按小时支付少量费用。	

实例集类型	何时使用	用户体验	定价模型	备注
Elastic	您的应用程序及其依赖项的大小小于 1.5 GB 时。每当用户使用 Elastic 实例集启动会话时，都必须从 Amazon S3 将您的虚拟硬盘 (VHD) 文件下载到会话中。因此，较大的 VHD 文件（即大小超过 1.5 GB）将导致最终用户体验不佳。您的应用程序是可便携的。也就是说，您的应用程序及其所有依赖项可以放置在 VHD 上并从 VHD 启动。您不需要流式处理实例加入域（Elastic 实例集目前不支持加入域）。您只想为活跃会话付费（即您无需为实例集中未使用的容量付费）。您的用户可以在启动会话后等待 45 秒或更长时间再访问他们的应用程序。您希望	用户在启动会话后等待 45 秒到 3 分钟才能访问应用程序（等待时间取决于虚拟硬盘的大小）。	您只需为流会话的持续时间付费。由于 Elastic 实例集中没有闲置实例的概念，因此对于未使用的实例，您无需支付任何费用。	不支持自定义映像（客户提供 VHD 来启动应用程序）或扩缩策略。目前支持 `stream.standard.small` 和 `stream.standard.medium` 实例。如果您的使用案例需要其他的实例类型，请联系您的 AWS 客户团队。

实例集类型	何时使用	用户体验	定价模型	备注
	AWS 为您管理扩缩（无需扩缩策略来管理）。			



实例集类型使用案例和要求

实例集大小调整

最低容量和计划扩缩

在调整 AppStream 2.0 实例集大小时，有几个考虑因素会直接关系到用户体验和成本。为最小容量输入的值可确保 AppStream 2.0 实例的数量很少会小于此值。AppStream 2.0 会话结束后，如果 AppStream 2.0 实例总数小于最低容量值，则会启动一个新的实例集实例。与往常一样，请务必记住一个 AppStream 2.0 实例直接与一个用户会话相对应，这会直接影响最小容量的值。

为最小容量输入超出预期并发数的值会导致成本增加，但用户体验不会受到影响。值过低可以缩减成本，但如果请求总数超过可用容量，则会影响用户体验。在这种情况下，管理员会发现出现“容量不足”错误。例如，如果一天开始时的预期连接数始终为可预测值，等待 `PendingCapacity` 变成 `AvailableCapacity` 则未能有效利用用户时间。

可以一开始使用可适应典型非高峰时段的最小容量，然后使用计划扩缩策略可在工作日开始之前有效地重置最小容量。不要忘记创建另一个计划扩缩策略，在非高峰时段恢复为最小容量。有关扩缩策略及其实施方法的更多信息，请参阅本文档中的实例集自动扩缩策略部分。

最大容量和服务限额

设置最大容量看似是一个任意值，但是如果预测和设置得当，则可以优化总资源消耗和成本。如果输入的值高于 AWS 账户 中 AppStream 2.0 实例集的服务限额，这可能看起来是有效的，但是，当自动扩缩事件尝试将资源扩展到最大容量时，它们会无法启动，因为最大容量值超过了可用的服务限额。确保针对所需的最大容量提出服务限额请求，以确保自动扩缩按照组织预期的那样运行。

设置最大容量值时的另一个重要考虑因素是成本。有关更多信息，请参阅本文档的通过选择实例集类型来优化成本部分。

选择桌面视图或应用程序视图

决定选择应用程序视图还是桌面视图对性能或成本没有影响。每个 AppStream 2.0 实例集在任何时候都只能访问一个视图。您可以更改流视图选项。请计划在非高峰工作时段进行此更改，因为更改流视图需要重启实例集。

流视图没有单独的最佳实践。流视图选项的影响概括如下：

- 管理员可通过使用情况报告特征报告应用程序的详细使用情况
- 最终用户的整体体验和工作流程（例如，完整的桌面是否可以满足使用案例的需求，还是只查看应用程序就足够了？）。

桌面视图

对于所有用户工作流程都在会话中执行的使用案例，桌面视图可通过将所有应用程序集中在一个环境中来简化用户体验。如需部署超过 3-5 个需要与操作系统 (OS) 集成的应用程序，桌面视图可以提供更一致的用户体验。当维护两个独立且不同的环境时，桌面视图非常实用。例如，用户可以同时访问生产和预生产桌面环境，验证布局、配置和应用程序访问权限的更改。

AppStream 2.0 使用情况报告可为桌面视图创建每日应用程序报告。产生的应用程序报告只是标为“桌面”，直接映射到 AppStream 2.0 会话。有关更多信息，请参阅本文档的监控用户使用情况部分。

纯应用程序视图

当 AppStream 2.0 堆栈旨在提供一些间歇性需要的应用程序时，“仅限应用程序”视图也非常有效。在自助服务终端环境中，通过应用程序视图可提供安全锁定的应用程序交付。在“应用程序视图”中，AppStream 2.0 可使用自定义 shell 替换默认的 Windows shell。此自定义 shell 仅显示正在运行的应用程序，从而最大限度地减少了操作系统的攻击面。

对于使用 AppStream 2.0 来增强现有组织桌面环境的使用情况，首选“纯应用程序”视图。在原生应用程序模式下部署 AppStream 2.0 Windows 客户端，允许充分使用键盘快捷键，从而最大限度地减少用户的困惑。

Amazon 2.0 使用情况报告可为应用程序视图创建每日应用程序报告。要更精细地报告应用程序和运行使用情况，可以考虑使用第三方解决方案从操作系统级别报告。您可以在报告模式下使用 Microsoft AppLocker，也可以考虑 AWS Marketplace 中提供的解决方案，例如 Liquidware 的 Stratusphere UX。

AWS Identity and Access Management 角色配置

如果工作负载要求 AppStream 2.0 最终用户在其会话中访问其他 AWS 服务，则最佳实践是通过使用 AWS Identity and Access Management(IAM) 角色来委派访问权限。通过实例集级别的分配，可以将 IAM 角色直接附加到最终用户的会话。有关在 AppStream 2.0 中使用 IAM 角色的其他最佳实践，请参阅管理员指南的这一部分。

使用静态凭证

某些工作负载可能需要静态输入 IAM 访问密钥，而不是从附加的角色继承而来。获取这些凭证有两种方法。第一种方法是将访问密钥存储在 AWS 服务中，然后向您的最终用户授予明确的 IAM 访问权限，以便从服务中提取该特定值。使用 AWS Secrets Manager 或 AWS SSM 参数存储是访问密钥存储机制的两个示例。第二种方法是使用 AppStream 2.0 凭证提供程序来获取附加角色的访问密钥。这可以通过调用凭证提供程序并解析访问密钥和私有密钥的输出信息来完成。以下是如何在 PowerShell 中执行此操作的示例。

```
$CMD = 'C:\Program Files\Amazon\Photon\PhotonRoleCredentialProvider
\PhotonRoleCredentialProvider.exe'
$role = 'Machine'

$output = & $CMD --role=$role
$parsed = $output | ConvertFrom-Json

$access_key = $parsed.AccessKeyId
$secret_key = $parsed.SecretAccessKey
$session_token = $parsed.SessionToken
```

保护 AppStream 2.0 S3 存储桶

如果您的 AppStream 2.0 工作负载配置了主文件夹和/或应用程序持久性，则最佳实践是保护存储永久数据的 Amazon S3 存储桶免受未经授权的访问或意外删除。第一层保护是添加 Amazon S3 存储桶策略，以避免存储桶被意外删除。第二层保护是添加符合最低权限原则的存储桶策略。要遵循这一原则，可以只允许必要的各方访问存储桶。

实例集自动扩缩策略

了解 AppStream 2.0 实例

AppStream 2.0 实例集实例的用户与实例集例的比例为 1:1。这意味着每个用户都有自己的流式处理实例。您同时连接的用户数量将决定实例集的大小。

扩缩策略

AppStream 2.0 实例集在应用程序自动扩缩组中启动。这使实例集可以根据使用情况进行扩缩，从而满足需求。随着使用量的增加，实例集会扩展，当用户断开连接时，实例集会相应缩减。这是通过设置扩缩策略来控制的。您可以设置基于计划的扩缩、分步扩缩和目标跟踪扩缩策略。有关这些扩缩策略的更多信息，请参阅Amazon AppStream 2.0 的实例集自动扩缩。

分步扩缩

这些策略按当前实例集大小的一定百分比或特定实例数量来增加或减少实例集容量。分步扩缩策略由 AppStream 2.0 `Capacity Utilization`、`Available Capacity` 或 `Insufficient Capacity Errors` 的 CloudWatch 指标 触发。

使用分步扩缩策略时，AWS 建议您添加一定比例的容量，而不是固定数量的实例。这样可以确保您的扩缩操作与您的实例集大小成正比。这将有助于避免出现扩展速度过慢（因为您添加的实例数量相对于实例集规模太少）或实例集规模较小而实例过多的情况。

目标跟踪

通过目标跟踪扩缩，您可以为实例集指定容量利用率级别。应用程序自动扩缩可创建和管理触发扩缩策略的 CloudWatch 警报。这将增加或减少容量，使实例集保持或接近指定的目标值。为了确保应用程序可用性，实例集会以最快速度随此指标按比例扩展，但缩减会逐步进行。配置目标跟踪时，请考虑扩缩冷却时间，以确保在所需的时间间隔内进行扩展和缩减。

目标跟踪对于高流失率的情况非常有效。当大量用户在短时间内开始或结束会话，就会发生流失。您可以通过检查实例集的 CloudWatch 指标来识别流失情况。如果您的实例集的待处理容量不为零，而所需容量没有变化（或变化很小），则表明可能会出现高流失率。在高流失率的情况下，配置目标跟踪策略，让（100 — 目标利用率百分比）在 15 分钟内大于流失率。例如，如果您有 10% 的实例集将因用户流失而会在 15 分钟内终止，请将容量利用率目标设置为 90% 或更低，以抵消高流失率。

基于计划的扩缩

这些策略使您能够根据基于时间的计划设置所需的实例集容量。当您了解登录行为并且可以预测需求变化时，此策略就可以发挥作用。

例如，在工作日的开始，您可能预计有 100 个用户会在上午 9:00 请求流式处理连接。您可以配置基于计划的扩缩策略，在上午 8:40 将最小实例集大小设置为 100。这样就可以在工作日开始时创建并提供实例集实例，并允许 100 个用户同时连接。然后，您可以设置另一项计划策略，在下午 5:00 将实例集缩减到最少 10 个。这样您可以节省成本，因为下班后的会话需求少于工作日时段。

生产中的扩缩策略

您可以选择将不同类型的扩缩策略组合到一个实例集中，以帮助为您的用户行为定义精确的扩展策略。在前面的示例中，您可以将计划扩缩策略与目标跟踪或分步扩缩策略相结合，以保持特定的利用率水平。当紧急需要容量时，计划扩缩和目标跟踪扩缩的组合有助于减少利用率级别急剧增加所产生的影响。

当扩缩策略更改所需的实例数量时，连接到流会话的用户不会受到缩减或扩展的影响。扩缩策略不会让现有的流会话终止。现有会话将不受打扰地持续下去，直到用户结束会话或实例集策略超时。

使用 CloudWatch 指标监控 AppStream 2.0 的使用情况可以帮助您随着时间的推移优化扩缩策略。例如，在初始设置期间通常会超额配置资源，您可能会看到利用率长期处于较低水平。或者，如果实例集配置不足，您可能会看到高容量利用率和“容量不足”错误。查看 CloudWatch 指标有助于调整您的扩缩策略，帮助减少这些错误。有关更多信息以及您可以使用的 AppStream 2.0 扩缩策略的示例，请参阅扩缩 Amazon AppStream 2.0 实例集。

扩缩策略设计的最佳实践

合并扩缩策略

许多客户选择将不同类型的扩缩策略组合到一个实例集中，以提高 AppStream 2.0 中自动扩缩的功能和灵活性。例如，您可以配置一个计划扩缩策略，以便在预计用户开始工作日的上午 6:00 增加实例集的最小值，并在用户停止工作之前的下午 4:00 减少实例集的最小值。您可以将此计划扩缩策略与目标跟踪或分布扩缩策略相结合，以保持特定的利用率水平，并在白天进行缩减或扩展以应对高峰使用量。当紧急需要容量时，计划扩缩和目标跟踪扩缩的组合有助于减少利用率级别急剧增加所产生的影响。

避免扩缩期间的流失

考虑一下您的实例集是否会因为您的使用案例而经历高度的流失。当大量用户在短时间内开始然后结束会话时，就会发生流失。当许多用户同时访问实例集中的某个应用程序几分钟就退出，可能会发生这种情况。

在这种情况下，您的实例集大小可能会远低于所需的容量，因为当用户结束会话时，实例就会结束。分步扩缩策略添加实例的速度可能不足以抵消流失，因此，您的实例集会停留在一定的大小。

您可以通过检查实例集的 CloudWatch 指标来识别流失情况。如果您的实例集的待处理容量不为零，而所需容量没有变化（或变化很小），则表明可能会出现高流失率。要考虑高流失情况，请使用目标跟踪扩缩策略并选择目标利用率，让（100 — 目标利用率百分比）在 15 分钟内大于流失率。例如，如果您的实例集中有 10% 将因用户流失而在 15 分钟内结束会话，请将容量利用率目标设置为 90% 或更低，以抵消高流失率。

了解最大配置速率

为大量用户管理 AppStream 2.0 实例集的客户应考虑配置速率限制。此限制将影响向某个实例集添加实例或向 AWS 账户 中的所有实例集添加实例的速度。

有两个限制需要考虑：

- 对于单个实例集，AppStream 2.0 的最大配置速率为每分钟 20 个实例。
- 对于单个 AWS 账户，AppStream 2.0 的配置速率为每分钟 60 个实例（爆发可达到每分钟 100 个实例）。

如果并行扩展的实例集超过三个，则账户配置速率上限将在这些实例集之间分摊（例如，并行扩展的六个实例集每个每分钟最多可以配置 10 个实例）。此外，还要考虑给定流式处理实例在响应扩展事件时完成配置所需的时间。对于未加入 Active Directory 域的实例集，这通常为 15 分钟。对于加入 Active Directory 域的实例集，这可能需要长达 25 分钟的时间。

考虑到这些限制，请看下面的例子：

- 如果您想将单个实例集从 0 扩展到 1000 个实例，则需要 50 分钟（每分钟 1000 个实例/20 个实例）才能完成配置，然后还需要 15-25 分钟才能使所有实例可供最终用户使用，总共需要 65-75 分钟。
- 如果您想同时将三个实例集从 0 扩展到 333 个实例（AWS 账户 完成），则所有实例集需要大约 17 分钟（每分钟 999/60 个实例）才能完成配置，然后再花 15 分钟使这些实例可供最终用户使用，总共需要 32-42 分钟。

利用多个可用区

在该区域中选择多个可用区进行实例集部署。当您为实例集选择多个可用区时，您的实例集能够添加实例以响应扩展事件的可能性就会增加。CloudWatch 指标 PendingCapacity 是评估大型实例集部署中实例集可用区设计的优化程度的起点。PendingCapacity 的持续高值可能表明需要扩展水平（跨可用区）扩展。有关更多信息，请参阅监控 Amazon AppStream 2.0 资源。

例如，如果自动扩缩尝试预配置实例以增加实例集的大小，而选定的可用区容量不足，则自动扩缩将改为在您为实例集指定的其他可用区中添加实例。有关可用区和 AppStream 2.0 设计的更多信息，请参阅本文档中的可用区。

监控容量不足错误指标

“容量不足错误”是 AppStream 2.0 实例集的 CloudWatch 指标。这一指标制定因缺少容量而被拒绝的会话请求的数量。

当您更改扩缩策略时，创建一个可以在出现任何容量不足错误时通知您的 CloudWatch 警报会很有帮助。这使您能够快速调整扩展策略，优化用户的可用区。管理指南提供了监控您的 AppStream 2.0 资源的详细步骤。

连接方法

流式传输 AppStream 2.0 中的会话时，用户有两种连接方法可供选择：

- Web 浏览器访问 — 支持任何支持 HTML5 的浏览器。无需插件或下载。
- AppStream 2.0 Windows 客户端

作为最佳实践，请考虑用户使用案例的特征和设备要求，以找到最能满足他们需求的浏览器或设备。

> **Note**
>
> 屏幕分辨率低于 1024x768 像素的设备不支持 AppStream 2.0。

特征和设备支持摘要

表 3 — 特征和设备支持摘要

	Web 浏览器访问	AppStream 2.0 Windows 客户端
多台监视器（最高 2K 分辨率）	支持	支持
多台监视器（最高 4K 分辨率）	不适用	支持
绘图板支持	支持*	支持
触摸显示屏设备支持	支持	不适用
USB 透传设备支持	不适用	支持
键盘快捷键	支持	支持
相对鼠标偏移	支持	支持
文件传输	支持	支持

	Web 浏览器访问	AppStream 2.0 Windows 客户端
本地打印机重定向	不适用	支持
本地驱动器重定向	不适用	支持
网络摄像头支持	支持	支持

*仅限 Google Chrome 和 Mozilla Firefox

Web 浏览器访问

AppStream 2.0 Web 浏览器支持访问应用程序，而无需安装专用客户端。用户可以使用支持 HTML5 的浏览器进行连接。不需要任何浏览器插件或扩展程序。

Web 浏览器访问为终端设备操作系统和类型提供了一系列选择。

适用于 Windows 的 AppStream 2.0 客户端

适用于 Windows 的 AppStream 2.0 客户端是可安装在 Windows PC 上的应用程序。此应用程序提供了使用 Web 浏览器访问 AppStream 2.0 时提供不了的其他功能。例如，AppStream 客户端可让您执行以下操作：

- 使用两台以上的监视器或 4K 分辨率。
- 对通过 AppStream 2.0 流式处理的应用程序使用 USB 设备。
- 在流会话期间访问本地驱动器和文件夹。
- 将打印作业从流应用程序重定向到连接到本地计算机的打印机
- 在流会话中使用本地网络摄像头进行视频和音频会议
- 在流会话期间在访问的应用程序中使用键盘快捷键
- 与远程流应用程序进行交互，交互方式与您和本地安装的应用程序交互的方式大致相同。

AppStream 2.0 客户端连接模式

AppStream 2.0 客户端提供了两种连接模式：原生应用程序模式和经典模式。您选择的连接模式将决定您在应用程序流式处理期间可使用的选项，以及流应用程序的工作和显示方式。管理员可以控制用户在原生应用程序模式和经典模式之间切换的能力。

- 经典模式在 AppStream 2.0 会话窗口中对应用程序进行流式传输。这与最终用户在 Web 浏览器中流式传输应用程序的方式类似。如果最终用户更喜欢以与浏览器相同的方式流式传输应用程序，同时使用其他特征，例如本地文件连接和打印机重定向，则使用经典模式。经典模式是推荐的默认连接模式。经典模式是桌面视图支持的唯一模式。
- 原生应用程序模式使最终用户能够像使用其他本地安装的应用程序一样使用远程流应用程序。如果最终用户习惯使用本地安装的应用程序，则原生应用程序模式可提供无缝体验。远程流应用程序的工作方式与本地安装的应用程序的工作方式大致相同。该应用程序的图标将显示在本地 PC 的任务栏中，就像本地应用程序的图标那样。与本地应用程序的图标不同，原生应用程序模式下的流应用程序的图标包含 AppStream 2.0 徽标。当用户想要使用应用程序键盘快捷键并随时要使用键盘快捷键在单个本地应用程序和单个远程应用程序之间切换时，推荐使用原生应用程序模式。

客户端部署和管理

用户可以自己安装 AppStream 2.0 客户端，或者管理员可以通过远程运行 PowerShell 脚本或使用自定义设置重新打包 AppStream 2.0 客户端来为他们安装 AppStream 2.0 客户端。

您必须让希望用户在流会话中使用的 USB 设备符合条件。如果其 USB 设备不合格，则 AppStream 2.0 将无法检测到它，并且无法与会话共享。在其设备符合条件后，用户必须在每次启动新的流会话时与 AppStream 2.0 共享这些设备。

在大规模部署 AppStream 2.0 客户端时，AWS 建议使用企业部署工具。企业部署工具包括 AppStream 客户端安装文件和组策略管理模板。

自定义域

以编程方式部署 AppStream 2.0 时，可以创建一个自定义域，为用户提供熟悉的流会话体验。在 AppStream 2.0 的 SAML 2.0 IdP 部署中，重要的是要强调用户访问始于 IdP，而不是 AppStream 2.0。用户不需要 AppStream 2.0 URL，因为这些是在身份验证后由 IdP 提供的。因此，SAML 2.0 IdP 部署不需要自定义域名。

身份验证

在 AppStream 2.0 中，身份验证可以在亚马逊 AppStream 2.0 之外进行，也可以作为 AppStream 2.0 服务的一部分进行。选择如何为 AppStream 2.0 部署进行身份验证是设计的基本考虑因素。对于一个组织来说，针对不同的用例部署多个 AppStream 2.0 的情况并不少见。每个使用案例可以有不同的身份验证方法。

AppStream 2.0 有三种类型的身份验证方法：

- SAML 2.0
- 用户池
- 编程方式

确定最佳的方法

Amazon AppStream 2.0 的架构非常灵活，可以满足大多数组织设计要求。在确定最佳的身份验证方法时，最佳实践是考虑服务使用者以及组织策略和程序使用者的目标和用途。

以下是一些将使用案例与组织目标相结合的示例。

表 4 — 不同组织目标的使用案例

示例	描述	身份验证
需要加入域的实例集实例	只有加入域的资源才能访问 AppStream 映像上安装的应用程序。	SAML 2.0
与 Microsoft 服务高度集成	组织依赖于正在开发的 Microsoft 组策略和后端基础架构	SAML 2.0
现有企业单点登录（SSO）	所有新服务都必须利用已确立多个报告和安全流程的企业 SSO 解决方案。	SAML 2.0
对应用程序的智能卡支持	智能卡（例如私人身份验证和通用访问卡），用于通过智能	SAML 2.0

示例	描述	身份验证
	卡读卡器对流式应用程序进行会话中身份验证。	
使用临时人员的临时员工队伍	在一年中的几个月中，会为临时工作人员分配一小部分应用程序，其中不包括用于完成活动的内部资源。	用户池
IT 支持有限	用户少于 50 以及 IT 员工有限的小型组织，他们希望消除维护身份提供商 (IdP) 的开销	用户池
独立软件供应商 (ISV)	由您的组织构建的专有解决方案，包括用户授权和身份验证，将 AppStream 2.0 扩展为解决方案的一部分。 *	编程方式
技术展示	完全短暂的环境，在您的解决方案导览展示中展示专有技术，无需存储用户信息。	编程方式
交互式网站体验	让您的网站与 Windows 流式应用程序进行交互。**	编程方式

*有关更多信息，请参阅软件供应商：将应用程序交付给任何用户设备。

**有关更多信息，请参阅 E AppStream mbed 2.0 直播会话。

如果您的组织有未在前面给出的示例中列出的用例或策略，则最佳做法是预测 AppStream 2.0 工作流程消耗的预期最终状态，以确保身份验证解决方案不会与之冲突。

配置身份提供商

SAML 2.0

安全断言标记语言 (SAML) 2.0 是使用户能够使用 AWS 资源的常见部署选项。各种第三方 SAML 2.0 身份提供商支持 AppStream 2.0。无论您的 AppStream 2.0 资源是否加入了域名，SAML 2.0 IdP 都要求您使用 IAM。

由于大多数堆栈会为每个 SAML 应用程序 IdPs 生成具有特定 SAML 属性的唯一 metadata.xml，因此每个 AppStream 2.0 堆栈都需要一个与 SAML IdP 具有可信关系的角色和一个对 AppStream: Stream 具有单一权限的策略，其条件符合 SAML IdP 和 2.0 堆栈的 ARN 的要求。 AppStream

AppStream 2.0 管理指南提供了单个 AppStream 2.0 堆栈设计的示例配置。有关多堆栈部署，请参阅使用 SAML 2.0 多堆栈应用程序目录的可选步骤。

用户池

AppStream 2.0 中的 “用户池” 选项卡是小型概念验证的有效选项。作为最佳实践，对于任何使用 AppStream 2.0 交付生产应用程序的用例和组织，最好避免使用用户池。

关于用户池，需要注意的一点是，用户的电子邮件地址区分大小写；因此，最佳实践是确保用户了解如何正确输入用户凭证。

流式传输 URL

对于从集中式服务（通常是 ISV）调用 AppStream 2.0 资源的部署，编程身份验证依赖于应用程序进行编程调用，AWS以动态传递信息并为其用户创建 AppStream 2.0 会话。使用 URL 操作创建直播网址时，请使用 API 身份验证方法（通常称为 “编程式CreateStreaming”）。进行 `CreateStreamingURL` 调用的用户必须使用具有 `appstream:CreateStreamingURL` 权限的有效用户或角色。

在创建编程访问策略时，最佳做法是通过在 “资源” 部分指定精确的 AppStream 2.0 Stack ARN 来代替默认的 “*” 来保护访问安全。例如：

Example

```
{
    "Version": "2012-10-17",
    "Statement": [
```

```
        {
            "Effect": "Allow",
            "Action": [
                "appstream:createStreamingURL"
            ],
            "Resource": "arn:aws:appstream:us-east-
1:031421429609:stack/BestPracticesStack"
        }
    ]
}
```

> **Note**
>
> 您可以使用描述堆栈 API 或 AWS CLI 快速检索 AppStream 2.0 堆栈的 ARN。

AppStream 2.0 实例应以通用实例的形式启动。通过从应用程序传递给它的信息， AppStream 2.0 实例使用会话上下文建立环境，为用户提供动态信息。

虽然本地 GPO 可用于在用户登录时指定设置，但在会话中使用`CreateStreamingURL`和传递关键属性（例如客户 ID 或数据库连接设置）时，会话上下文是最佳做法。 AppStream

应用程序授权

AppStream 2.0 可以动态构建呈现给用户的应用程序目录。应用程序授权基于 SAML 2.0 属性，或者使用 AppStream 2.0 动态应用程序框架。

在大多数情况下，建议使用 SAML 2.0 进行基于属性的应用程序授权。要管理应用程序包的交付，建议使用动态应用程序框架。

与 Microsoft Active Directory 集成

Amazon AppStream 2.0 映像生成器和实例集可以与 Microsoft Active Directory 集成。这使您能够提供一种集中的用户身份验证和授权方法，并可以将 Active Directory 组策略应用于已加入域的 AppStream 2.0 实例。使用加入域的 AppStream 实例集可获得与使用本地环境相同的管理优势。这包括集中管理网络文件共享、用户应用程序授权、漫游配置文件、打印机访问权限和其他基于策略的设置。

在将 AppStream 2.0 环境与 Active Directory 集成时，请务必注意，AppStream 2.0 堆栈的初始身份验证仍由 SAML2.0 IdP 管理。用户成功通过 IdP 身份验证后，当用户启动会话时，他们必须输入 Active Directory 域的域密码或进行智能卡身份验证。

在设计将与 AppStream 2.0 一起使用的 Active Directory 域服务 (ADDS) 环境时，有两个服务选项和许多部署方案可供选择。此外，请确保与 Active Directory 站点拓扑所有者一起审查 AppStream 2.0 的联网。

服务选项

Active Directory 也可以使用 AWS 托管的 Microsoft Active Directory (AD) 进行部署。AWS托管的 Microsoft AD 是一项可完全托管的服务，让您可以运行 Microsoft Active Directory。Microsoft Active Directory 也可以在自托管环境中使用，在 EC2 上或本地运行。

部署方案

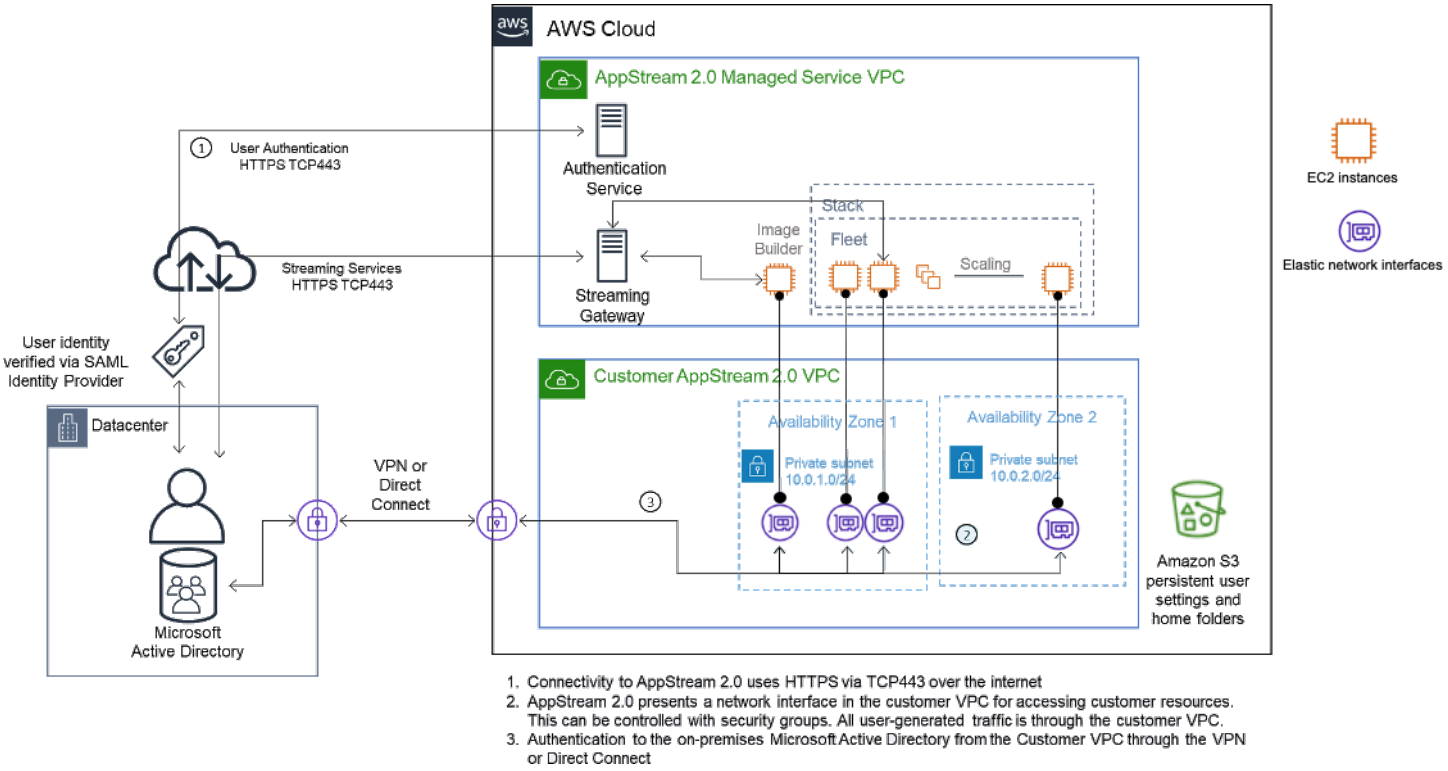
以下列出的部署方案是 AppStream 2.0 与托管的 Microsoft AD 或客户自托管 Active Directory 进行集成的常用推荐选项。下面列出的所有架构图都使用 Amazon 的核心结构。

- Amazon 虚拟私有云 (VPC) — 创建专用于 AppStream 2.0 服务的 Amazon VPC，其中至少有四个私有子网分布在四个可用区。其中两个私有子网用于 AppStream 实例集和映像生成器。其余两个子网用于 EC2 或托管的 Microsoft AD 上的域控制器。
- 动态主机配置协议 (DHCP) 选项集 — 提供将配置信息传递给 AppStream 2.0 实例集和将在 VPC 中配置的映像生成器的标准。DHCP 选项集是在 VPC 级别定义的。它使客户能够定义指定的域名和 DNS 设置，这些设置将在配置后与 AppStream 2.0 实例一起使用。
- AWS目录服务 — Amazon 托管的 Microsoft AD 可以部署到两个私有子网中，这两个子网将与 AppStream 2.0 工作负载配合使用。
- AppStream 2.0 实例集 — AppStream 2.0 实例集或映像生成器托管在 AWS 托管 VPC 中。每个 AppStream 2.0 实例都有两个弹性网络接口 (ENI)。主接口 (`eth0`) 用于管理目的，并通过流网关代

理最终用户与实例的连接。辅助接口 (eth1) 已接入客户 VPC，可用于访问定制 VPC 或本地的其他资源。

场景 1：本地部署的 Active Directory 域服务 (ADDS)

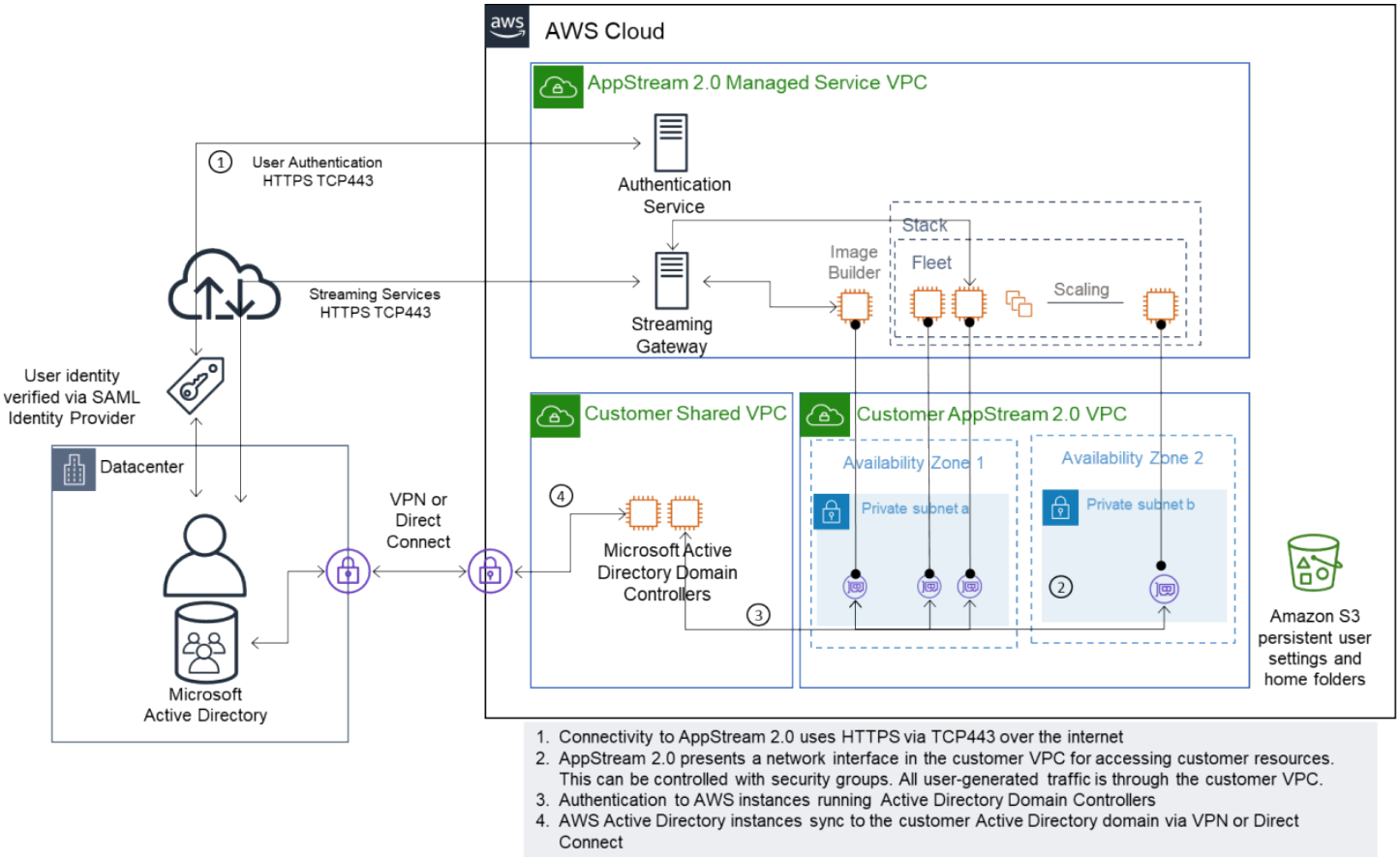
所有身份验证流量都会通过 VPN 或 Direct Connect 连接从客户 VPC 路由到客户网关。此场景的优势在于可以使用可能已经部署的 AD 环境，无需在客户 VPC 中配置其他域控制器。缺点是它只能依赖 VPN 或 Direct Connect 对用户进行身份验证和授权，获取 AppStream 2.0 实例集的访问权。如果存在任何网络连接问题，AppStream 2.0 实例集或映像生成器将受到直接影响。提供双 VPN 隧道或具有不同路径的 Direct Connect 连接可以降低这种潜在风险。



场景 1 — 本地部署的 Active Directory 域服务 (ADDS)

场景 2：将 Active Directory 域服务 (ADDS) 扩展到 AWS 客户 VPC

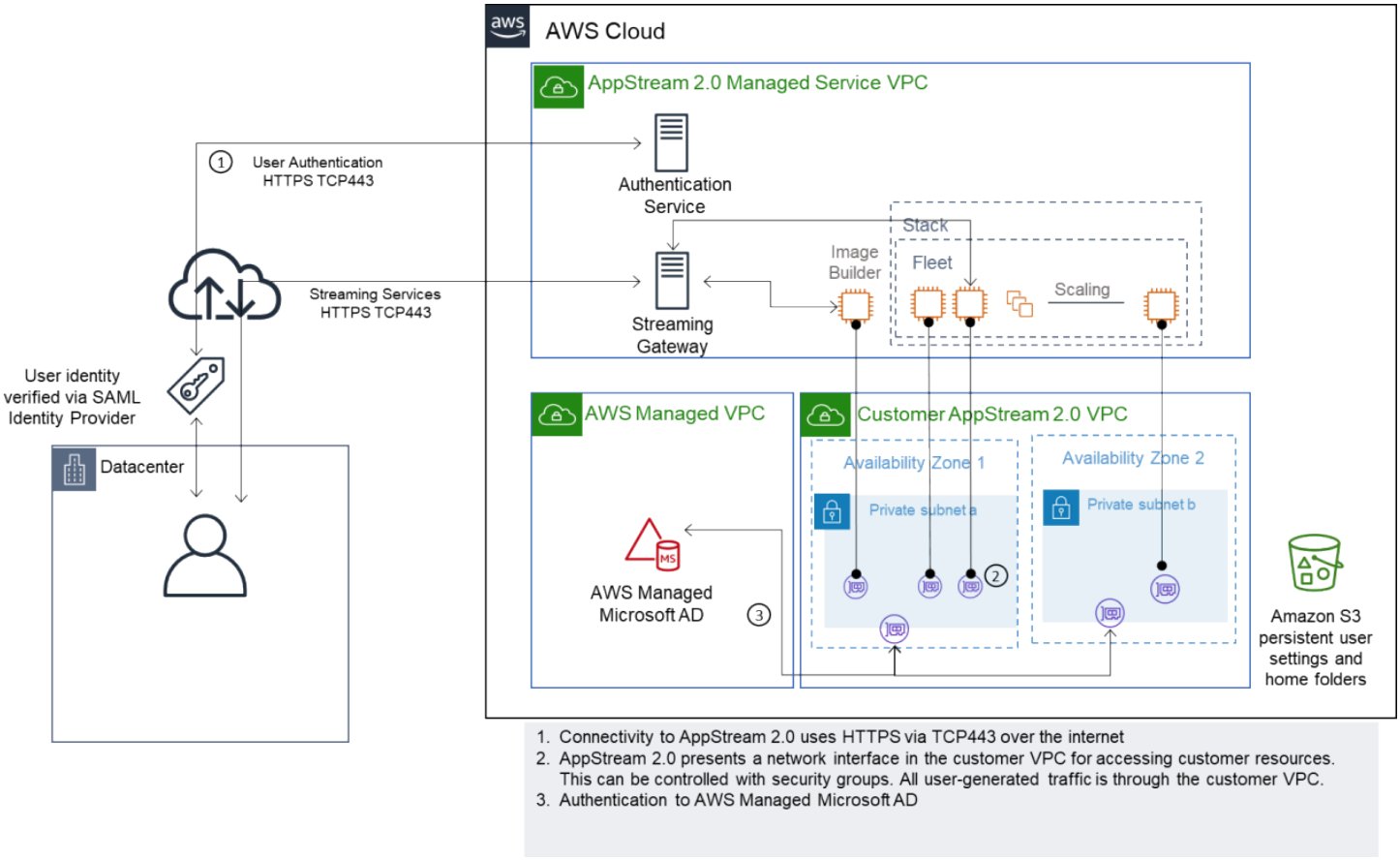
Active Directory 已扩展到客户 VPC。应为客户 VPC 中的新域控制器创建 Active Directory 站点。身份验证流量将路由到 AWS 客户 VPC 中的域控制器，而不是通过 VPN 或 Direct Connect 连接。



场景 2 — 将 Active Directory 域服务扩展到 AWS 客户的虚拟私有云

场景 3：AWS 托管的 Microsoft Active Directory

AWS 托管的 Microsoft AD 部署在 AWS 云 中，用作 AppStream 2.0 实例集和映像生成器的身份和资源域。



场景 3 — AWS 托管的 Active Directory

Active Directory 服务站点拓扑

Active Directory 服务站点拓扑是物理网络的逻辑表示形式。

站点拓扑可以帮助您高效地路由客户端查询和 Active Directory 复制流量。精心设计和维护的站点拓扑可以帮助您的组织实现以下好处：

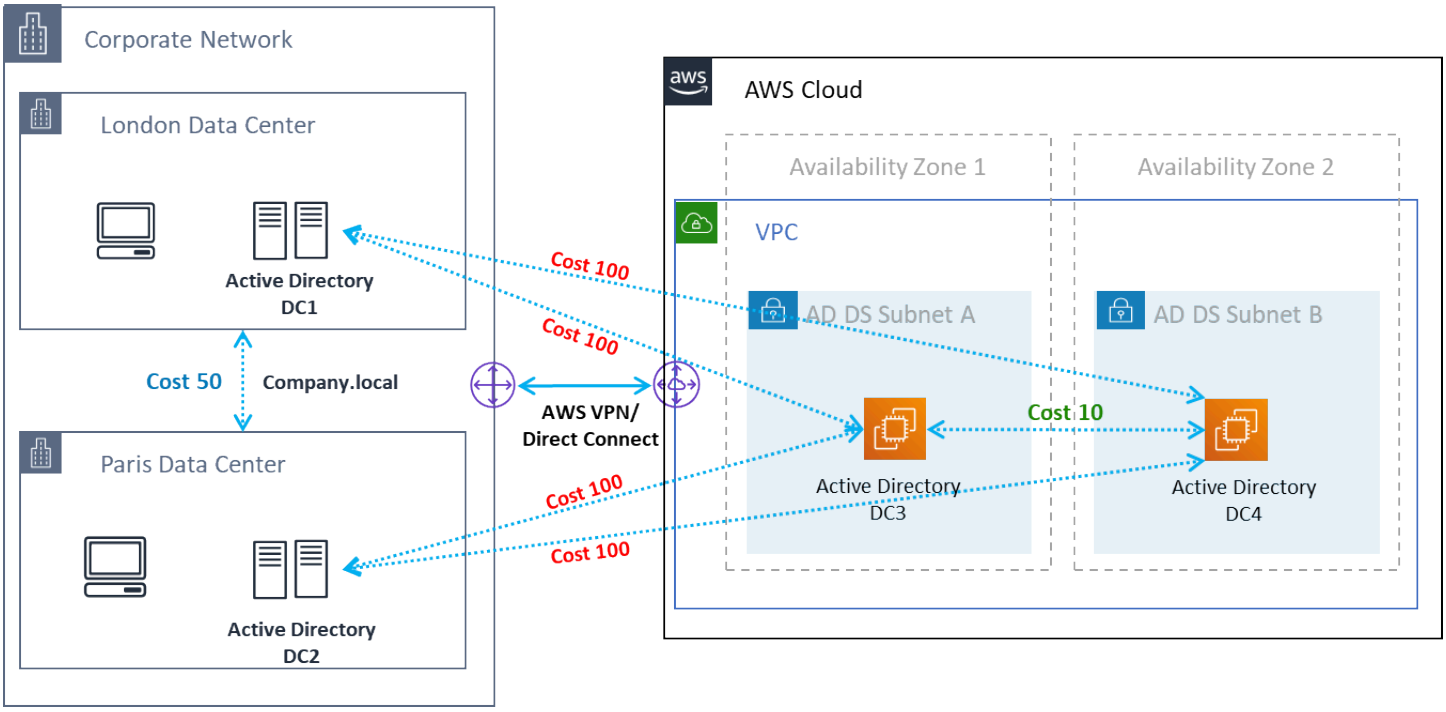
- 在本地和 AWS 云 之间进行同步时，可最大限度地降低复制 Active Directory 数据的成本。
- 可优化客户端计算机定位最近的资源（例如域控制器）的能力。这有助于减少缓慢广域网 (WAN) 链路的网络流量，改善登录和注销流程，并加快资源访问操作。

在引入 AppStream 2.0 服务时，请确保将用于 AppStream 2.0 实例子网的地址范围分配给适合您环境的正确站点。

对于场景 1 和场景 2，要获得登录时间和 Active Directory 资源访问时间方面的最佳用户体验，站点和服务是关键。

站点拓扑负责控制同一站点内和跨站点边界的域控制器之间的 Active Directory 复制。

定义正确的站点拓扑可确保客户端亲和性，这意味着客户端（在本例中为 AppStream 2.0 流式传输实例）可以使用其首选的本地域控制器。



Active Directory 站点和服务 — 客户端亲和性

> **Tip**
>
> 最好是能为本地 AD DS 和 AWS Cloud 之间的站点链接定义高昂的成本。上图举例说明了为确保独立于站点的客户端亲和性，您应该为站点链接分配多少成本（成本 100）。

有关站点拓扑的更多信息，请参阅设计站点拓扑。

Active Directory 组织单位

AWS 建议将配置的组织单位 (OU) 存储在单个 AppStream 2.0 Directory Config 对象中。最好让每个 AppStream 2.0 堆栈都拥有自己的组织单位。这样您就可以灵活地为每个堆栈设置特定的 GPO。确保 AppStream 2.0 计算机对象有专用的组织单位，以避免将特定于 AppStream 2.0 的策略与本地桌面混用。考虑为每个部署了 AppStream 2.0 的 AWS 区域 使用子组织单位。

Active Directory 计算机对象清理

AppStream 2.0 实例是短暂的。随着实例集扩展和缩减，实例集会创建并重复使用 Active Directory 计算机对象。

AWS 建议创建 AD 清理流程来删除在移除 AppStream 实例集后可能存在的陈旧的 Active Directory 计算机对象。

安全性

云安全性一直是 Amazon Web Services (AWS) 的重中之重。安全和合规是客户共同承担 AWS 的责任。有关更多信息，请参阅责任共担模式。作为 an AWS AppStream d 2.0 客户，在堆栈、队列、映像和网络等不同层面上实施安全措施非常重要。

由于其短暂性， AppStream 2.0 通常是应用程序和桌面交付的安全解决方案的首选。对于要预定义并在用户会话结束时清除的环境的使用案例，请考虑在 Windows 部署中常见的防病毒解决方案是否有意义。防病毒软件会增加虚拟化实例的开销，因此最好是减少不必要的活动。例如，在启动时扫描系统卷（临时卷）并不能提高 2.0 的整体安全性。 AppStream

安全 AppStream 2.0 的两个关键问题集中在：

- 是否需要在会话之外保留用户状态？
- 用户在会话中应拥有多少访问权限？

保护持久性数据

AppStream 2.0 的部署可能要求用户状态以某种形式保留。它可能是为了保留个人用户的数据，或者是为了使用共享文件夹进行协作而保留数据。 AppStream2.0 实例存储是临时性的，没有加密选项。

AppStream 2.0 通过 Amazon S3 中的主文件夹和应用程序设置提供用户状态持久性。某些使用案例需要更好地控制用户状态的持久性。对于这些用例， AWS 建议使用服务器消息块 (SMB) 文件共享。

用户状态和数据

由于大多数 Windows 应用程序在与用户创建的应用程序数据位于同一位置时性能最好、最安全，因此最佳做法是将这些数据与 AppStream 2.0 队列保持 AWS 区域 相同。最好能加密这些数据。用户主文件夹的默认行为是使用密钥管理服务中的 Amazon S3 托管加密密钥对静态文件和文件夹进行加密（ ）AWS KMS。 AWS 请务必注意，有权访问 AWS 控制台或 Amazon S3 存储桶的 AWS 管理用户将能够直接访问这些文件。

在需要 Windows 文件共享中的服务器消息块 (SMB) 目标来存储用户文件和文件夹的设计中，该过程要么是自动的，要么需要配置。

表 5 — 保护用户数据的选项

SMB目标	E ncryption-at-rest	E ncryption-in-transit	防病毒软件 (AV)
FSx适用于 Windows 文件服务器	自动通过 AWS KMS	通过SMB加密实现自动	安装在远程实例上的 AV 在映射的驱动器上执行扫描
文件网关、 AWS Storage Gateway	默认情况下，存储在 S3 AWS Storage Gateway 中的所有数据都在服务器端使用 Amazon S3 托管加密密钥 (SSE-S3) 进行加密。您可以选择使用不同的网关类型来加密存储的数据 AWS Key Management Service (KMS)	在任何类型的网关设备和 AWS 存储设备之间传输的所有数据都使用进行加密SSL。	安装在远程实例上的 AV 在映射的驱动器上执行扫描
EC2基于 Windows 文件服务器	启用EBS加密	PowerShell; `Set-SmbServer Configuration - EncryptData $True`	安装在服务器上的 AV 在本地驱动器上执行扫描

端点安全和防病毒

Amazon AppStream 2.0 实例的短暂性质以及数据缺乏持久性，这意味着需要采用不同的方法来确保用户体验和性能不会因为持久性桌面上所需的活动而受到损害。当有组织策略或用于外部数据输入（例如电子邮件、文件传入、外部网页浏览）时，Endpoint Security 代理会安装在 AppStream 2.0 映像中。

移除唯一标识符

Endpoint Security 代理可能具有全局唯一标识符 (GUID)，必须在队列实例创建过程中重置该标识符。供应商有关于在映像中安装产品的说明，这将确保为从映像生成的每个实例生成一个新的GUID镜像。

为确保不会生成，请在运行 AppStream 2.0 Assistant 生成映像之前安装端点安全代理作为最后一个操作。GUID

性能优化

端点安全供应商提供的交换机和设置可优化 AppStream 2.0 的性能。设置因供应商而异，可以在他们的文档中找到，通常在上面的章节中VDI。一些常见设置包括但不限于：

- 关闭启动扫描以确保最大限度地缩短实例创建、启动和登录时间
- 关闭计划的扫描以防止不必要的扫描
- 关闭签名缓存以防止文件枚举
- 启用VDI优化的 IO 设置
- 应用程序为确保性能而需要排除的项目

端点安全供应商提供了虚拟桌面环境的使用说明，用于优化性能。

- 趋势科技 Office Scan 支持虚拟桌面基础架构——Apex One/ OfficeScan (trendmicro.com)
- CrowdStrike 以及如何在数据中心安装 CrowdStrike 猎鹰
- Sophos 和 Sophos Central Endpoint：如何安装在金色映像上以避免重复的身份和 Sophos Central：在虚拟桌面环境中安装 Windows 端点的最佳实践
- McAfee 以及在虚拟桌面基础架构系统上配置和部署McAfee 代理
- 微软端点安全和为非永久VDI计算机配置 Microsoft Defender 防病毒软件——微软科技社区

扫描排除项

如果在 AppStream 2.0 实例中安装了安全软件，则安全软件不得干扰以下进程。

表 6 — AppStream 2.0 进程安全软件不得干扰以下进程。

服务	进程
AmazonCloudWatchAgent	“C:\Program Files\ Amazon\AmazonCloud WatchAgent\ start-amazon-cloudwatch-age nt.exe”
A mazonSSMAgent	“C:\Program Files\ Amazon\SSM\ amazon-ssm-agent .exe”

服务	进程
NICE DCV	“C:\Program Files\NICE\DCV\ Server\ bin\ dcvserver.exe” “C:\Program Files\NICE\DCV\ Server\ bin\ dcvagent.exe”
AppStream 2.0	“C:\ProgramFiles\ Amazon\ AppStream 2\StorageConnector\ StorageConnector .exe” 在文件夹 "C:\Program Files\Amazon\Photon\" “。 \ Agent\ PhotonAgent .exe” “。 \ Agent\ s5cmd.exe” “。 \WebServer\ PhotonAgentWebServer .exe” “。 \CustomShell\ PhotonWindowsAppSwitcher .exe” “。 \CustomShell\ PhotonWindowsCustomShell .exe” “。 \CustomShell\ PhotonWindowsCustomShellBackground .exe”

文件夹

如果在 AppStream 2.0 实例中安装了安全软件，则该软件不得干扰以下文件夹：

Example

```
C:\Program Files\Amazon\*
C:\ProgramData\Amazon\*
C:\Program Files (x86)\AWS Tools\*
C:\Program Files (x86)\AWS SDK for .NET\*
C:\Program Files\NICE\*
C:\ProgramData\NICE\*
C:\AppStream\*
```

```
    C:\Program Files\Internet Explorer\*
    C:\Program Files\nodejs\
```

端点安全控制台清洁

每当用户在空闲和断开连接超时之后进行连接时，Amazon AppStream 2.0 都会创建新的唯一实例。这些实例将具有唯一的名称，并将在端点安全管理控制台中构建。将超过 4 天或更长时间（或更短，具体取决于 AppStream 2.0 会话超时）的未使用过期计算机设置为删除，可以最大限度地减少控制台中过期实例的数量。

网络排除项

AppStream 2.0 管理网络范围 (198.19.0.0/16) 以及以下端口和地址不应被 AppStream 2.0 实例中的任何安全/防火墙或防病毒解决方案阻止。

表 7 — AppStream 2.0 流媒体实例中的端口安全软件不得干扰

端口	使用情况
8300、3128	它用于建立流式连接
8000	这用于管理 AppStream 2.0 之前的流媒体实例
8443	这用于管理 AppStream 2.0 之前的流媒体实例
53	DNS

表 8 — AppStream 2.0 托管服务地址安全软件不得干扰

端口	使用情况
169.254.169.123	NTP
169.254.169.249	NVIDIAGRID许可服务
169.254.169.250	KMS

端口	使用情况
169.254.169.251	KMS
169.254.169.253	DNS
169.254.169.254	元数据

保护会 AppStream 话

限制应用程序和操作系统的控制

AppStream 2.0 使管理员能够准确指定哪些应用程序可以在应用程序流模式下从网页启动。但是，这并不能保证只能运行指定的应用程序。

Windows实用程序和应用程序可以通过其他方式通过操作系统启动。 AWS 建议使用 Microsoft AppLocker 来确保只有你的组织需要的应用程序才能运行。必须修改默认规则，因为它们授予所有人访问关键系统目录的路径权限。

> **Note**
>
> Windows Server 2016 和 2019 要求运行 Windows 应用程序身份服务才能强制执行 AppLocker 规则。《AppStream 管理指南》中详细介绍了使用 Micro AppLocker soft 从 AppStream 2.0 开始访问应用程序。

对于加入 Active Directory 域的队列实例，请使用组策略对象 (GPOs) 提供用户和系统设置，以保护用户的应用程序和资源访问权限。

防火墙和路由

创建 AppStream 2.0 队列时，必须分配子网和安全组。子网已分配了网络访问控制列表 (NACLs) 和路由表。在启动新的映像生成器或创建新实例集时，您最多可以关联五个安全组。安全组最多可以继承现有安全组的五个指定设置。对于每个安全组，您需要添加规则以控制进出实例的出站和入站网络数据流

A NACL 是您的可选安全层VPC，它充当无状态防火墙，用于控制进出一个或多个子网的流量。您可以使用ACLs与您的安全组相似的规则来设置网络，以便为您的网络添加额外的安全层VPC。有关安全组和网络之间差异的更多信息ACLs，请参阅比较安全组和NACLs页面。

在设计和应用安全组和NACL规则时，请考虑 Well-Architect AWS ed 最佳实践以实现最低权限。最低权限的原则是只授予完成任务所需的权限。

对于拥有将本地环境连接到的高速专用网络AWS（通过 Di AWS rect Connect）的客户，您可以考虑使用VPC终端节点，这意味着流媒体流量将通过您的专用网络连接而不是通过公共互联网进行路由。AppStream有关此主题的更多信息，请参阅本文档的 AppStream 2.0 流媒体接口VPC端点部分。

数据丢失防护

我们将介绍两种数据丢失防护。

客户端到 AppStream 2.0 实例的数据传输控制

表 9 — 数据传入和传出控制指南

设置	Options	指南
剪贴板	• 仅复制并粘贴到远程会话 • 仅复制到本地设备 • 禁用	禁用此设置不会禁用会话中的复制和粘贴。如果需要将数据复制到会话中，请选择“仅粘贴到远程会话”，以最大限度地减少数据泄露的可能性。
文件传输	• 上传和下载 • 仅上传 • 仅下载 • 禁用	避免启用此设置以防止数据泄露。
打印到本地设备	• 已启用 • 禁用	如果需要打印，请使用由您的组织控制和监控的网络映射打印机。

思考一下现有组织数据传输解决方案相对于堆栈设置的优势。这些配置并不是为了取代全面的安全数据传输解决方案而设计的。

控制来自 AppStream 2.0 实例的出口流量

在担心数据丢失的地方，重要的是要掩盖用户进入 AppStream 2.0 实例后可以访问的内容。网络出口（或传出）路径是什么样子？ 通常要求最终用户在其 AppStream 2.0 实例中访问公共 Internet，因此需要考虑在网络路径中放置 WebProxy 或内容过滤解决方案。其他注意事项包括本地防病毒应用程序和 AppStream 实例内部的其他端点安全措施（有关更多信息，请参阅 “端点安全和防病毒” 部分）。

使用 AWS 服务

AWS Identity and Access Management

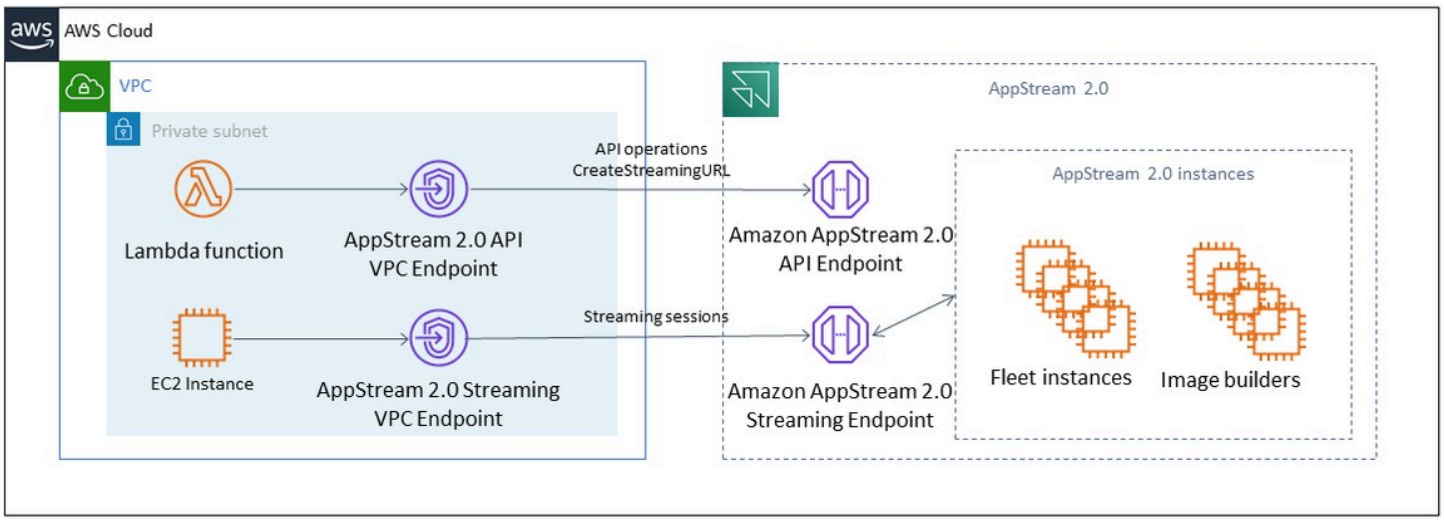
使用IAM角色访问 AWS 服务并在附加到该角色的IAM策略中具体说明是一种最佳实践，即只有 AppStream 2.0 会话中的用户才能访问而无需管理其他凭据。请遵循在 AppStream 2.0 中使用IAM角色的最佳实践。

创建IAM策略来保护 Amazon S3 存储桶，这些存储桶是为了在主文件夹和应用程序设置中保留用户数据而创建的。这会阻止非 AppStream 2.0 管理员进行访问。

VPC端点

VPC终端节点可在您VPC和支持的 AWS 服务以及由提供支持的VPC终端节点服务之间建立私有连接 AWS PrivateLink。 AWS PrivateLink 是一种允许您使用私有 IP 地址私密访问服务的技术。您VPC与其他服务之间的流量不会离开 Amazon 网络。如果仅 AWS 服务需要访问公共互联网，则VPC端点将完全取消对NAT网关和互联网网关的要求。

在自动化例程或开发人员需要API调用 AppStream 2.0 的环境中，为 2.0 API操作创建接口VPC端点。 AppStream 例如，如果私有子网中存在无法访问公网 Internet 的EC2实例，则API可以使用 AppStream 2.0 的VPC终端节点来调用 AppStream 2.0 API 操作，例如CreateStreamingURL。下图显示了一个示例设置，其中 Lambda 函数API和实例使用 AppStream 2.0 和流式传输VPC终端节点。EC2



VPC端点

流媒体VPC终端节点允许您通过终VPC端节点流式传输会话。流媒体接口终端节点将流媒体流量保持在您的内部VPC。流媒体流量包括像素、USB、用户输入、音频、剪贴板、文件上传和下载以及打印机流量。要使用VPC终端节点，必须在 AppStream 2.0 堆栈中启用VPC端点设置。相比从互联网访问能力有限的地点通过公共互联网流式传输用户会话，这可以作为一种替代方案，并且它可以通过 Direct Connect 实例进行访问。通过VPC终端节点流式传输用户会话需要满足以下条件：

- 与接口终端节点关联的安全组必须允许对用户连接的 IP 地址范围内的端口 443 1400-1499 (TCPTCP) 和端口 () 进行入站访问。
- 子网的网络访问控制列表必须允许从临时网络端口 1024-65535 (TCP) 到用户连接的 IP 地址范围的出站流量。
- 需要互联网连接才能对用户进行身份验证并交付 AppStream 2.0 运行所需的网络资产。

要详细了解如何使用 AppStream 2.0 限制 AWS 服务流量，请参阅VPC终端节点创建和流式传输的管理指南。

当需要完全的公共互联网访问时，最好在 Image Builder 上禁用 Internet Explorer 增强安全配置 (ESC)。有关更多信息，请参阅 AppStream 2.0 管理指南以禁用 Internet Explorer 增强型安全配置。

灾难恢复

Amazon AppStream 2.0 内置了跨多达三个可用区的冗余。这意味着，如果用户在已降级的可用区中有一个活动会话，他们只需断开连接并重新连接即可，就能在一个正常的可用区内保留会话，前提是您有足够的容量。虽然这在区域内提供了高可用性，但如果服务在区域层面遇到问题，则无法提供灾难恢复解决方案。

要为您的 AppStream 2.0 用户提供灾难恢复计划，您首先需要在辅助区域构建一个 AppStream 2.0 环境。从设计角度来看，此环境应与您的本地环境建立冗余连接（如果适用），并且不应依赖主区域。例如，如果您的 AppStream 2.0 实例集已加入域，则您应该在辅助区域中部署其他域控制器，并配置站点和服务。从 AppStream 2.0 的角度来看，此环境应包含与主区域相同的实例集和堆栈设置。实例集本身应运行相同的基础映像，该映像可以通过控制台或以编程方式复制到您的辅助区域。如果在 AppStream 2.0 会话中运行的应用程序与您的主区域有后端依赖关系，则也应具有区域冗余，以确保用户仍然可以在主区域出现故障时访问应用程序的后端。您在目标区域的服务级别限制应与您的主区域相匹配。

身份路由

在灾难恢复场景中，有两种不同的方法可以实现对应用程序的访问。总体而言，这两种方法的不同之处在于如何将用户定向到失效转移区域。第一种方法是在 IdP 中使用单个 AppStream 2.0 应用程序配置执行的，第二种方法是使用两个单独的应用程序配置。

方法 1：更改应用程序的中继状态

当用户从身份提供商 (IdP) 登录到 AppStream 2.0 时，在进行身份验证后，他们会被中继到一个特定 URL，该 URL 与他们打算访问的区域和堆栈相一致。有关中继状态 URL 的更多信息，请参阅 Amazon AppStream 2.0 管理指南。管理员可以基于与主区域相同的 AppStream 2.0 映像配置一个跨区域堆栈，供用户进行失效转移。管理员只需更新中继状态 URL，使其指向失效转移堆栈，即可控制此失效转移。要使此方法正常运行，需要让关联的 IAM 策略允许访问两个堆栈（主堆栈和失效转移堆栈）。有关如何配置这些 IAM 策略的更多详细信息，请参阅以下示例策略。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": "appstream:Stream",
```

```
            "Resource": [
            "arn:aws:appstream:PrimaryRegion:190836837966:stack/StackName",
            "arn:aws:appstream:FailoverRegion:190836837966:stack/StackName"
            ],
            "Condition": {
                "StringEquals": {
                    "appstream:userId": "${saml:sub}"
                }
            }
        }
    ]
}
```

方法 2：在 IdP 中配置两个 AppStream 2.0 应用程序

此方法要求管理员在 IdP 中为 AppStream 2.0 构建两个单独的应用程序。然后，他们可以同时展示两个应用程序并让用户选择去哪里，也可以锁定/隐藏应用程序，直到需要进行失效切换。这种方法更适合让全球用户经常四处走动的使用案例。这些用户应该从最近的端点进行流式传输，因此，分配两个应用程序可以让他们选择为最近的区域配置的应用程序。这也可以自动完成，有关更多信息，请参阅此博客文章。

存储持久性

在利用 AppStream 2.0 中包含的数据持久性特征（例如应用程序持久性和主文件夹同步）时，您需要将这些数据复制到失效转移区域。这些特征可将持久性数据存储在给定 AppStream 2.0 区域的 Amazon S3 存储桶中。要跨区域持久保存数据，您需要将源存储桶上的所有更改复制到失效转移区域 AppStream 2.0 存储桶。这可以通过原生 Amazon S3 特征来完成，例如 Amazon S3 跨区域复制。每个用户的持久性数据将存储在经过哈希处理的用户名的文件夹下。由于用户名将统一跨区域进行哈希处理，因此只需复制数据即可确保数据在辅助区域中持久保存。有关 AppStream 2.0 使用的 Amazon S3 存储桶的更多信息，请参阅本指南。

监控

使用控制面板

监控车队利用率是一项常规活动，可以通过 CloudWatch 指标和创建仪表板来执行。或者，在 AppStream 2.0 控制台中，使用 “队列使用情况” 选项卡。请定期监控实例集使用情况，因为用户行为并不总是可以预测的，而且即使提前规划准备得再充分，需求也可能会超出预期。 AppStream 2.0 指标和维度的完整列表 CloudWatch 可在 AppStream 2.0 管理指南的 “监控资源” 下找到。

预期增长

每当 PendingCapacity 出现大幅激增时，就会发生自动扩缩事件。当新的 AppStream 2.0 队列实例可供托管用户会话使用时，确认这一点AvailableCapacity并PendingCapacity保持反向关系非常重要。InsufficientCapacityError为每个 AppStream 2.0 队列创建 CloudWatch 警报，通知管理员确保自动扩展不会落后于需求。

如果需求超过容量并且 InsufficientCapacityError 指标值很常见，请考虑在工作日开始时通过计划扩缩策略提高最小容量。此外，还要制定第二个计划扩缩策略，以便在需求得到满足后降低最小容量。请记住，降低最小容量的值不会影响现有会话。在工作日结束之前降低最小容量可以通过降低 ActualCapacity 值来按预期进行有效扩缩。这样可以优化成本。

如果需求始终不可预测，请使用 Target Tracking 扩展策略来确保 AppStream 2.0 队列AvailableCapacity中有足够的容量来满足需求，同时确定使用模式。继续监控，因为目标跟踪跟踪的是实例集消耗的百分比。随着实例集实例总数的增长，未使用的实例集实例总数会成倍增加。除非将最大容量设置为保守值，否则这可能会造成浪费。使用多种类型的扩缩策略（例如，计划和目标跟踪）以实现可靠性与成本优化之间的平衡。

监控用户使用情况

监控唯一用户，因为用户需要根据使用情况付费。这笔用户费用是 Image Assistant (RDS) 订阅者访问许可证 (SAL) 的费用。可以通过执行身份验证的 IdP 的报告或使用情况报告来评估唯一用户。

使用情况报告作为单独 .csv 的文件存储在 S3 存储桶中，您可以下载并使用第三方商业智能 (BI) 工具分析这些报告。您 AWS 无需下载报告即可在中分析使用数据，也可以在自定义日期范围内创建报告，而无需连接多个文件。.csv例如，您可以使用 Amazon Athena 和 Amaz QuickSight on 来创建 2.0 使用情况数据的自定义报告和可视 AppStream 化效果。

保留应用程序和 Windows 事件日志

AppStream 2.0 实例会话完成后，实例即告结束。这意味着会话中使用的所有应用程序和 Windows 事件日志都将丢失。如果需要保留这些应用程序和 Windows 事件日志，一种方法是使用 Amazon Data Firehose 将它们实时传输到 S3，然后使用亚马逊 OpenSearch 服务（OpenSearch 服务）进行搜索。如果预计查询不会很频繁，为了优化成本，请使用 Amazon Athena 进行搜索，而不是运行亚马逊服务。 OpenSearch

审计网络和管理活动

如果尚未设置，则最好使用 Amazon AppStream 2.0 AWS CloudTrail进行配置。 AWS 账户 要专门审计 AppStream 2.0 API 调用，请使用值为的过滤器事件源`appstream.amazonaws.com`。

启用 VPC 流日志可审计对客户管理的资源的访问权限。VPC 流日志可以发布到 CloudWatch 日志，以便在需要审计时执行查询。

随着 AppStream 2.0 队列的增长，监控子网 IP 分配非常重要。通过运行 describe-subnets CLI 报告分配给实例集的每个子网中的可用 IP 地址，来报告 IP 分配。确保您的组织有足够的 IP 地址容量，来满足所有以最大容量运行的实例集的需求。

成本优化

成本优化侧重于避免不必要的成本。关键主题包括了解和控制资金的使用去向，以及选择最合适、最正确的资源类型数量。分析一段时间内的支出并进行扩展，以便满足业务需求。以下 AppStream 2.0 资源需要按实际使用量付费：

- Always-On 实例集实例
- On-Demand 实例集实例
- On-Demand 停止的实例费用
- 映像生成器实例
- 用户费用

有关当前定价信息，请参阅 AWS 网站上的 Amazon AppStream 2.0 定价。

设计具有成本效益的 AppStream 2.0 部署

规划和设计 AppStream 2.0 部署的第一步是使用简单的定价工具来估算与 AWS 使用量相关的费用基准。提供您的用户总数、每小时的实际并发使用情况、实例类型和实例集利用率，定价工具会估算您的每用户价格。它还显示了使用 On-Demand 实例集相比使用 Always-On 实例集时预计可节省的价格。

客户喜欢 AppStream 2.0 的定价模式，是因为他们只需要为那些为了满足用户流式处理需求而配置的实例付费。这种模式与他们现有的应用程序流环境不同。这些配置通常基于峰值容量的配置，即使在夜间、周末和节假日负载较低时也是如此。Amazon AppStream 2.0 定价工具仅提供与您使用 AppStream 2.0 相关的 AWS 费用的估算值，不包括任何可能适用的税费。您的实际费用取决于各种因素，包括您对 AWS 服务的实际使用情况。

AppStream 2.0 定价工具以 Microsoft Excel 或 OpenOffice Calc 电子表格形式提供，让您可以输入有关实例集的基本信息，然后根据您的使用模式为 On-Demand 实例集和 Always-On 实例集提供 AppStream 2.0 环境的成本估算。您可以根据历史或预期的使用趋势来拟算成本。Elastic 实例集中内置了这些特征，管理员无需预测使用情况、创建、维护扩展策略和映像。在 Amazon Linux 2 上运行的 Elastic 实例集和实例（所有实例集类型）按流会话的持续时间（以秒为单位）计费，最少为 15 分钟。

通过选择实例类型来优化成本

对于实例集和映像生成器实例，您可以为应用程序选择一系列不同的实例系列和类型。

最终用户测试 — 下一步是向一组试点用户推出 AppStream 2.0 实例集进行测试，以验证我们选择的实例类型。请试点用户测试其所有常规和繁重的工作流程，以捕获有关内存、CPU 和显卡的指标，以便您可以捕获基准性能指标，这一点很重要。试点组应包含使用该应用程序的各种用户角色，以确保您在多个用户体验中对其进行测试。用户验收测试使您能够收集有关流会话体验的反馈。创建或更新堆栈时，可以选择使用自定义反馈 URL。用户选择“发送反馈”链接提交有关其应用程序流式处理体验的反馈后，会重定向到这个 URL。如果存在性能瓶颈，请使用 Windows 性能指标来分析资源限制。例如，如果当前实例集实例类型 stream.standard.medium 显示资源限制，则将该实例类型升级到 stream.standard.large。相反，如果性能指标显示资源远未得到充分利用，则可以考虑降级实例类型。

通过选择实例集类型来优化成本

在创建新的 AppStream 2.0 实例集时，开发者必须选择 Always-On 或 On-Demand 实例集类型。在从定价角度选择实例类型时，了解 AppStream 2.0 如何管理实例集实例非常重要。对于 Always-On 实例集，实例集实例将保持运行状态。因此，当用户尝试流式传输会话时，实例集实例随时准备开始流式传输会话。

对于 On-Demand 实例集，在启动实例集实例后，它们将保持停止状态。已停止的实例费用低于保持运行的实例费用，这有助于降低成本。On-Demand 实例集实例必须从停止状态启动。用户必须等待大约两分钟，他们的流会话才可用。

Elastic 实例集非常适合独立应用程序，可以将其安装在 Amazon Simple Storage Service (Amazon S3) 存储桶中的虚拟硬盘上。因为 Elastic 实例集只会在流式传输期间按秒计费，因此可以进一步降低某些使用案例的成本。价格取决于您在创建实例集时选择的实例类型和大小以及操作系统。

如果最终用户在工作时间需要实例集实例，则最好保持原来的流会话。这是因为实例集实例按小时收费，每次启动新的流会话时，都会产生另一笔实例集实例费用。

表 10 — AppStream 2.0 实例集类型比较

实例集类型	优点	注意事项
Always-On	减少流会话的等待时间	用户需要支付每小时的实例费，因为无法选择让实例保持停止状态。
On-Demand	由于实例保持停止状态，因此可以节省成本	流会话等待时间更长

实例集类型	优点	注意事项
Elastic	对于可以安装在虚拟硬盘上的应用程序有零星使用模式的使用案例，按秒计费可能很有用	随着应用程序虚拟硬盘的规模越来越大，将其安装到流实例所需的时间可能会很长

AppStream 2.0 会监控您的实例集利用率，并自动调整实例集容量，以尽可能低的成本满足用户的需求。容量调整是根据您定义的扩缩策略进行的，要么基于当前利用率，要么基于计划。定期审查实例集使用指标，以确认实例集扩缩策略的剩余容量不会过高。

扩缩策略

实例集自动扩缩能够让您优化实例集资源，无需过度使用资源等待用户登录。管理员可以根据各种利用率调整实例集的大小，满足用户需求。使用 CloudWatch AppStream 2.0 实例集指标或第三方监控工具来了解用户活动并配置扩缩策略，以便根据预期使用情况扩展或缩小 AppStream 2.0 实例集。用户日志是了解实际使用情况的重要机制。自动扩缩可以根据此见解动态更改实例集大小。

在许多情况下，AppStream 2.0 实例集是根据最大用户数量创建的，不会针对一天和一周中的不同时间（例如晚上和周末）进行调整。通常，流式处理的应用程序的并发用户数少于用户总数，尤其是在用户可以灵活地进行远程办公的情况下。在预测使用模式时，必须将这些因素考虑在内。过高估计会导致超额配置 AppStream 2.0 实例，从而产生额外成本。要获得最佳配置，您可能需要将一个或多个计划扩缩策略与扩展策略结合起来。

要了解有关实施扩缩策略的更多信息，请查看扩缩 Amazon AppStream 2.0 实例集。

用户费用

用户通过 AppStream 2.0 实例集实例流式传输应用程序，按每个 AWS 区域 每月向每个用户收取用户费用。为 AppStream 2.0 用户生成的用户 ID 一致。连接到映像生成器时不收取用户费用。

学校、大学和某些公共机构可能有资格享受减免的 Microsoft RDS SAL 用户费，即每位用户每月 0.44 美元。有关资格要求，请参阅 Microsoft 许可条款和文档。

如果您享有 Microsoft 许可证移动性权益，您可能有资格自带 Microsoft RDS 客户端访问许可证 (CAL) 并将其用于 Amazon AppStream 2.0。如果您在自己的许可证覆盖范围内，则无需支付月度用户费。有关是否可以将现有 Microsoft RDS CAL 许可证与 Amazon AppStream 2.0 一起使用的更多信息，请参阅许可证移动性指南或咨询您的 Microsoft 许可代表。AWS

映像生成器使用情况

AppStream 2.0 映像生成器实例按小时收费。映像生成器实例费用包括计算、存储和流式传输协议使用的任意网络流量。所有正在运行的映像生成器实例均需支付适用的运行实例费用。该费用根据实例类型和大小计算，即使在没有连接管理员的情况下也是如此。

作为优化成本的最佳实践，请在不使用映像生成器实例时将其关闭。CloudWatch Events 规则可用于安排日常作业，例如调用 Lambda 函数来停止映像生成器实例。

您可以使用托管 AppStream 2.0 图像更新来更新 AppStream 2.0 映像。此更新方法提供最新的 Windows 操作系统更新和驱动程序更新，以及最新的 AppStream 2.0 代理软件。使用此方法更新映像时，作为托管服务流程的一部分，映像生成器会自动启动和停止。

结论

利用 AppStream 2.0，您可以轻松地将现有的桌面应用程序添加到 AWS，并允许用户立即进行流式处理。Windows 用户可以使用 AppStream 2.0 客户端或支持 HTML5 的 Web 浏览器进行应用程序流式传输。您可以为每个应用程序维护一个单独的版本，从而使应用程序管理更加轻松。您的用户可始终访问其应用程序的最新版本。您的应用程序在 AWS 计算资源上运行，数据不会存储在用户的设备上，这意味着用户始终可以获得安全的高性能体验。

与桌面应用程序流式传输的传统本地解决方案不同，AppStream 提供即用即付定价模式，无需前期投资且没有基础设施需要维护。您可以立即进行全球扩展，从而确保用户始终享受出色体验。

Amazon AppStream 2.0 可集成到现有 IT 系统和流程中，本白皮书描述了实现这一部署的最佳实践。遵循本白皮书中的指导方针最终可以让您实现经济实惠的云桌面部署，该部署可以在 AWS 全球基础架构上随着您的业务安全扩展。

贡献者

本文档的贡献者包括：

- Andrew Wood，Amazon Web Services 高级解决方案架构师
- Andrew Morgan，Amazon Web Services 最终用户计算专家解决方案架构师
- Arun PC，Amazon Web Services 高级最终用户计算专家解决方案架构师
- Asriel Agronin，Amazon Web Services 高级解决方案架构师
- Dustin Shelton，Amazon Web Services 高级最终用户计算专家解决方案架构师
- Jeremy Schiefer，Amazon Web Services 高级解决方案架构师
- Navi Magee，Amazon Web Services 首席解决方案架构师
- Pete Fergus，Amazon Web Services 高级云端支持工程师
- Phil Persson，Amazon Web Services 首席最终用户计算专家解决方案架构师
- Richard Spaven，Amazon Web Services 高级最终用户计算专家解决方案架构师
- Spencer DeBrosse，Amazon Web Services 高级解决方案架构师
- Stephen Stetler，Amazon Web Services 高级解决方案架构师
- Taka Matsumoto，Amazon Web Services 高级云端支持工程师
- Vasant Sirsat，Amazon Web Services 高级最终用户计算专家解决方案架构师

延伸阅读

有关更多信息，请参阅：

- 亚马逊 AppStream 2.0 管理指南
- 亚马逊 AppStream API 参考
- 使用适用于 Windows 文件服务器的 Amazon FSx 和 fsLogix 来优化亚马逊 2.0 上的应用程序设置持久性 AppStream
- 使用亚马逊 ElasticSearch 和亚马逊 Firehose 监控亚马逊 AppStream 2.0
- 使用亚马逊 Athena 和亚马逊分析您的亚马逊 AppStream 2.0 使用报告 QuickSight
- 扩大您的亚马逊 AppStream 2.0 车队规模
- 使用微软 AppLocker 管理亚马逊 AppStream 2.0 上的应用程序体验
- 在 Amazon AppStream 2.0 中使用自定义域名
- 如何在 AppStream 2.0 中使用我自己的 Microsoft RDS CAL？
- 亚马逊 AppStream 2.0 定价工具
- 使用 AppStream 2.0 创建在线软件试用版
- 使用亚马逊 AppStream 2.0 创建 SaaS 门户

文档修订

要获得有关白皮书更新的通知，请订阅 RSS 源。

变更	说明	日期
文档已更新	更新内容将包括弹性实例集、基于属性的应用程序授权、多堆栈应用程序目录、基于 Linux 的实例集、数据传入和传出、灾难恢复和其他更新。	2022 年 6 月 14 日
文档已更新	HTML 版本已发布。	2022 年 1 月 19 日
初次发布	白皮书已发布。	2021 年 6 月 8 日

版权声明

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。