

# VMware vRealize Automation 8.7 Release Notes

vRealize Automation 8.7

您可以从 VMware 网站下载最新的技术文档：

<https://docs.vmware.com/cn/>。

**VMware, Inc.**  
3401 Hillview Ave.  
Palo Alto, CA 94304  
[www.vmware.com](http://www.vmware.com)

**威睿信息技术(中国)有限公司**  
北京办公室  
北京市  
朝阳区新源南路 8 号  
启皓北京东塔 8 层 801  
[www.vmware.com/cn](http://www.vmware.com/cn)

上海办公室  
上海市  
淮海中路 333 号  
瑞安大厦 804-809 室  
[www.vmware.com/cn](http://www.vmware.com/cn)

广州办公室  
广州市  
天河路 385 号  
太古汇一座 3502 室  
[www.vmware.com/cn](http://www.vmware.com/cn)

版权所有 © 2022 VMware, Inc. 保留所有权利。 [版权和商标信息](#)

# 目录

- 1 发行版本 4**
- 2 关于 vRealize Automation 8.7 5**
- 3 开始之前 6**
- 4 新增功能 7**
- 5 API 文档和版本控制 10**
- 6 已解决的问题 13**
- 7 已知问题 16**
- 8 已更改和已弃用的功能 18**

# 发行版本

1

vRealize Automation 8.7 | 2022 年 3 月 22 日

- vRA Easy Installer (ISO) 内部版本 19527797
- vRA 产品设备内部版本 19508505
- SaltStack Config 内部版本 19359172

## 本文更新内容

日期	更新描述	类型
2022 年 3 月 22 日	初次发布。	

# 关于 vRealize Automation 8.7

# 2

vRealize Automation 8.7 对 vRealize Automation 8.6.2 的功能进行了完善，新增了各种功能，包括能够更改已置备部署的项目、改进了 ABX 内部部署引擎、SaltStack Config 可用作云模板中的资源、使用 API 通过自定义表单对目录项进行自定义验证、对 SaltStack SecOps 进行自定义修复等。

# 开始之前

# 3

熟悉支持文档。

- [使用 vRealize Easy Installer 安装 vRealize Automation](#)
- [在 vRealize Automation 中管理用户](#)
- [vRealize Automation 转换指南](#)

安装 vRealize Automation 并设置用户后，您可以使用与包含的每个服务对应的“入门”指南和“使用和管理”指南。“入门”指南包括端到端概念证明。“使用和管理”指南提供更深入的信息，支持您探索可用功能。有关其他信息，还可以参见 [vRealize Automation 8.7 产品文档](#)。

- [vRealize Automation Cloud Assembly 入门](#)
- [使用和管理 vRealize Automation Cloud Assembly](#)
- [vRealize Automation Code Stream 入门](#)
- [使用和管理 vRealize Automation Code Stream](#)
- [vRealize Automation Service Broker 入门](#)
- [使用和管理 vRealize Automation Service Broker](#)

有关 vRealize Orchestrator 8.7 功能和限制的信息，请参考 [vRealize Orchestrator 8.7 发行说明](#)。

vRealize Automation 8.7 具备诸多优势，其中包括：

## ABX 内部部署引擎的演进

ABX 内部部署现在使用下一代内部部署引擎，该引擎具有高级性能和可扩展性。新的 FaaS 速度非常快，修复了许多内存限制问题，并引入基于内存的限制。您还可以使用其他日志记录功能更轻松地对操作运行进行故障排除。

新的内部部署引擎包括以下改进：

- ABX 内部部署操作使用更快、更稳定且可扩展的新 FaaS 引擎。它还修复了使用旧 FaaS 引擎时发现的许多问题。
- ABX 内部部署操作支持更快的部署时间。这样，您可以更快地开发操作。
- ABX 内部部署操作内存限制现在是针对每个操作运行的，而不是共享的。因此，由于基于内存的限制，所有现有操作的内存限制都会重置为默认值。如果我们允许通过共享并行操作运行创建的具有高内存限制的现有操作，则在不需要时，这些操作将消耗大量容量。
- ABX 内部部署操作运行日志现在包含已完成操作运行的其他信息。日志末尾有一个额外的日志行，显示操作运行所消耗的近似内存，以便您可以设置适当的内存限制。
- ABX 内部部署操作现在会在操作运行超时的情况下显示日志。您可以更轻松地调查超时的操作运行。
- ABX 内部部署现在可以更好地隔离同一操作的操作运行，因为现在每个操作运行都在单独的容器中运行。

## 更改已置备部署的部署项目

现在，针对已置备部署启用了更改项目的实施后操作。已置备的部署可以包含任意数量的计算机、磁盘、资源组、负载均衡器、网络、安全组、NAT 和网关。如果将已置备的部署更新为包含未提及的资源，例如 Terraform 配置，或已载入/已迁移的资源，则更改项目操作不可用。删除了资源后，更改项目操作将再次可用。

- 实施后操作仅限于云管理员。
- 计算机和磁盘的云区域必须存在于目标项目中，并遵守设置的配额限制。配额从初始项目中释放并保留在目标项目中。如果出现任何故障，操作不会自动回滚。
- 有关详细信息，请参见[实施后操作](#)。

**SaltStack Config** 在云模板中作为资源类型提供

现在，您可以将 Salt 工作节点直接拖放到画布上以将 SaltStack Config 资源类型附加到一个或多个虚拟机，从而将该节点作为云模板的一部分在实施前操作中进行本机部署和配置。新资源类型位于左侧资源菜单中的 SaltStack 下。

## 更新了适用于 vRealize Orchestrator 版本 8.4.2+ 的 vRealize Automation Plugin

更新的插件版本现在可从 [VMware Marketplace](#) 获得。

插件现在支持：

- 计算机的 IaaS 清单和脚本对象
- 计算机的 CRUD
- EntityFinders

了解有关 vRealize Automation Plugin 的更多信息。有关完整的功能描述，请查看新 vRealize Orchestrator 社区页面中的“文档”部分。

## 现在支持使用 API 通过自定义表单对目录项进行自定义验证

vRealize Automation 现在支持使用 API 进行自定义验证。使用此新功能，您可以通过 API 设计具有自定义表单和外部验证的目录项。当用户通过 API 从目录项创建部署时，将执行验证。如果验证失败，API 响应将包含验证错误消息。

## SaltStack SecOps 的自定义修复

现在，您可以导入不受 SaltStack SecOps 支持的公告。可以将自定义修复文件附加到公告以进行自动修复。了解有关自定义修复的更多信息。

## SaltStack 配置作业的动态作业输入

使用动态作业减少、重用和委派您的 IT 自动化和配置管理成果。

## 属性组中的可选输入

输入属性组现在支持可选输入。在属性组中，默认情况下所有属性都是可选的。要将不含默认值的所有非布尔属性标记为必需，请将以下云模板属性添加到所需属性组：

populateRequiredOnNonDefaultProperties: true。如果忽略上述属性或将其设置为 false，则所有属性都将被视为可选，这是默认行为。

## 保留已迁移部署的部署创建日期

迁移工具移动的部署现在会保留原始创建日期。

## 针对存储的部署限制策略支持

部署限制策略现在支持对实施前置备和实施后操作执行存储限制，包括调整磁盘大小、添加和删除磁盘。有关部署限制策略的更多信息。

## 移除基础架构计算机和卷视图

基础架构计算机和卷视图已替换为“资源”顶级选项卡下“资源中心”的虚拟机和卷视图。计算机视图的权限不再可用。

## 商城停用



vRealize Automation 中的商城集成已停用。

#### ■ SSC SecOps 支持 Ubuntu 20.04 和 20.10

现在，用户能够跨其 Ubuntu 20.04 和 20.10 系统管理合规性。

#### ■ 基于 ABX 的自定义资源的结构定义修改和格式设置

应用程序架构师现在能够编辑自定义资源结构定义参数，因此它们在部署 UI 中更加全面。参数和计算属性现在均可修改。

限制：

- 当前没有用于自定义资源类型详细信息表单的表单设计器。如果您希望能够自定义显示自定义资源类型的表单，则必须手动编辑并保存自定义资源类型。

为此，您必须对表单定义控制器执行 API POST 调用。

```
/form-service/api/forms
```

在正文中指定表单和以下参数：

```
type: deploymentResource
sourceType: resourceType
sourceId: Custom.YourResourceTypeHere
```

以及要设置的任何其他参数。

- vRO 工作流与自定义资源类型之间不会同步。例如，如果将新输入添加到创建工作流，该输入不会自动显示在结构定义中。即使转到结构定义，也无法添加此新输入。这是因为自定义资源类型是使用创建时的工作流版本创建的，并且该定义目前不可更新。

行为：

- 如果保存没有属性值的自定义资源类型，则将根据所选的创建工作流自动生成结构定义。这仅适用于基于 vRO 的自定义资源类型，因为结构定义基于工作流生成。
- 如果尝试保存没有计算属性的自定义资源类型，则会显示错误。此版本之前的行为是隐式生成新的结构定义，对于基于 vRO 的 CRT，并成功保存自定义资源类型。

# API 文档和版本控制

# 5

API 文档随产品提供。要从单个登录页访问所有 Swagger 文档，请转至 <https://<appliance.domain.com>/automation-ui/api-docs>，其中 *appliance.domain.com* 是您的 vRealize Automation 设备。

使用 API 之前，请考虑此版本的最新 API 更新和更改，并注意所用 API 服务的任何更改。如果之前未使用 `apiVersion` 变量锁定 API，则 API 响应可能会发生更改。下表提供了此版本的所有 API 更新和更改。

对于未锁定的 API，默认行为会因 API 而异。

- 对于 Cloud Assembly IaaS API，在不使用 `apiVersion` 参数的情况下执行的所有请求都将重定向到第一个版本，即 2019-01-15。此重定向将允许之前未指定 `apiVersion` 参数的每个用户平稳过渡到最新版本，而不会发生破坏性更改。

注意：对于 Cloud Assembly IaaS API，最新版本为 `apiVersion=2021-07-15`。如果保持未锁定状态，IaaS API 请求将重定向到第一个版本，即 2019-01-15。第一个版本已弃用，将支持 12 个月。为确保平稳过渡到新版本，请将分配了 `apiVersion` 参数的 IaaS API 请求锁定到 2021-07-15。

- 对于其他 API，您的 API 请求将默认为最新版本。如果选择根据 Swagger 规范列出的某个早期版本日期，API 行为将反映从该日期以及下一个最新版本日期之前的任何日期起生效的 API。并非每个 vRealize Automation 版本都对 API 进行版本控制，并非所有 API 都支持 `apiVersion` 参数。

有关 API 版本控制的详细信息，请参见《[vRealize Automation 8.7 API 编程指南](#)》。

服务名称	服务描述	API 更新和更改
ABX	包含特定于 ABX 的所有功能，包括创建和管理操作及其版本以及执行操作和流。	无更改
批准	实施以下策略，控制在置备请求之前必须同意部署或实施后操作的人员。	无更改
蓝图	创建、验证和置备 VMware Cloud Templates，以前称为蓝图。	无更改

服务名称	服务描述	API 更新和更改
CMX	将 Kubernetes 与 vRealize Automation 结合使用时，部署和管理 Kubernetes 集群和命名空间。	<p>新 API 端点</p> <ul style="list-style-type: none"> <li>获取与给定 ID 关联的 K8SInstaller           <ul style="list-style-type: none"> <li>GET /cmx/api/resources/installers/{id}</li> </ul> </li> <li>获取 K8SResources 的资源特定属性           <ul style="list-style-type: none"> <li>GET /cmx/api/resources/installers/{id}/properties</li> </ul> </li> <li>触发安装给定 K8SInstaller 拥有的 K8SResources           <ul style="list-style-type: none"> <li>POST /cmx/api/resources/installers/{id}/install</li> </ul> </li> <li>回滚给定 K8SInstaller 拥有的 K8SResources           <ul style="list-style-type: none"> <li>POST /cmx/api/resources/installers/{id}/uninstall</li> </ul> </li> </ul>
Content Gateway 内容服务	连接到外部内容源，例如 SCM 提供程序中的基础架构即代码内容。	移除了商城和商城下载 API。
自定义表单 表单服务	定义 Service Broker 和 Cloud Assembly VMware 服务中的动态表单呈现和自定义行为。	无更改
部署	访问系统中部署的部署对象和平台或蓝图。	无更改
IaaS	执行基础架构设置任务，包括以迭代方式验证和置备资源。	<p>用于列出数据中心内的文件夹以便可以将资源部署到特定文件夹的新端点</p> <ul style="list-style-type: none"> <li>GET /iaas/api/folders</li> </ul>
迁移	此服务用于根据配置文件中的信息快速设置 vRA 8 实例，也称为零设置。	无更改
项目	包含特定于项目创建、管理和删除的所有功能。	无更改
重新放置	定义对所有云中的现有虚拟机进行管理的策略和计划。	无更改
目录	访问 Service Broker 目录项和目录源，包括内容共享和目录项请求。	无更改
目录服务 策略	与 Service Broker 中创建的策略交互。	无更改
Code Stream 所有管道服务	通过这些 API 可以访问 Code Stream 服务。	无更改

服务名称	服务描述	API 更新和更改
身份服务	身份、帐户和服务管理 API 列表。	<p>新 API 端点</p> <ul style="list-style-type: none"> <li>■ 获取组织内组的角色            GET /csp/gateway/am/api/orgs/{orgId}/groups/{groupId}/roles         </li> <li>■ 从组织中移除组            DELETE /csp/gateway/am/api/orgs/{orgId}/groups         </li> <li>■ 获取组织角色列表            GET /csp/gateway/am/api/orgs/{orgId}/roles         </li> <li>■ 获取特定组织的组            GET /csp/gateway/am/api/orgs/{orgId}/groups         </li> <li>■ 更新组织内组的角色            PATCH /csp/gateway/am/api/orgs/{orgId}/groups/{groupId}/roles         </li> </ul> <p>更新了 API 请求参数，要求在检索组织中的所有服务定义时提供身份验证凭据</p> <p>GET /csp/gateway/slc/api/definitions</p> <p>添加了 API 请求参数</p> <p>“includeGroupIdsInRoles” 用于指示响应中继承的角色是否应显示组信息</p> <ul style="list-style-type: none"> <li>■ GET /csp/gateway/am/api/v2/orgs/{orgId}/users</li> <li>■ GET /csp/gateway/am/api/orgs/{orgId}/users</li> <li>■ GET /csp/gateway/am/api/orgs/{orgId}/users/search</li> </ul>
重新放置服务	对 onboardingBlueprintState 上的 PATCH 操作添加的新限制	无更改

# 已解决的问题

# 6

本版本已解决下列问题。

- **管理员角色缺少权限。**

当 SaltStack Config 与 vIDM 集成且具有管理员角色时，您无法查看工作节点、工作节点密钥或接受工作节点密钥。

- **在 AWS Lambda 上运行的可扩展性操作可能会失败并显示错误。**

由于 AWS Lambda 服务中的微小更改，在 AWS Lambda 上运行的可扩展性操作可能会失败并显示以下错误：

“错误 com.amazonaws.services.lambda.model.ResourceConflictException: 此时无法执行该操作。函数当前处于以下状态: 挂起 (Error com.amazonaws.services.lambda.model.ResourceConflictException: The operation cannot be performed at this time. The function is currently in the following state: Pending)”。

- **从快照置备虚拟机不会将虚拟机放置到在存储配置文件中配置的正确数据存储中。**

使用快照置备虚拟机时，虚拟机不会放置在该快照所在的正确数据存储中，而无论在存储配置文件中配置了何种数据存储。

- **升级后重新配置安全规则失败。**

升级后，用户无法使用新规则，使用 NSX-T 版本低于 3.x 的协议和端口，重新配置安全组。

- **“请求 - 确认删除请求”页面上出现括号位置错误问题。**

存在多个部署资源时，弹出确认屏幕中的括号位置不按预期显示。

- **Azure、AWS 网络标记为缺失，并作为新网络重新收集。**

针对 AWS 和 Azure 云帐户创建且包含发现的网络和安全组的 vRealize Automation 网络配置文件可能开始时有缺少的项目，即网络和/或安全组。缺少的项目会在创建后的几天内显示以及在某些环境中显示。出现缺少项目的情况可能是因为枚举进程在云帐户与置备实体之间找不到对应关系并由此删除了置备实体。

- **vSphere 适配器 - 在没有自定义规范的情况下，对 Windows 计算机进行网络重新配置失败**

更新具有 Windows 操作系统的已部署 vSphere 计算机以连接到其他网络时，如果云帐户中未指定任何自定义规范，则该操作将失败。失败错误消息为：“来自 vCenter 的错误: 指定的参数不正确: spec.identity (Error from vCenter: A specified parameter was not correct: spec.identity)”。出现此错误的原因是 vRA 未检测到这是 Windows 计算机，并创建了适用于 Linux 计算机的自定义。

#### ■ 修复了在某些情况下 CSV 值未评估为字符串值的问题

当 CSV 中的对应值为以下类型之一时，类型为字符串/密码的列/字段的“复合”值出现不一致：

- 数字 - 值在表单结构定义中以数字形式写入，即使该值应为字符串。即值 12 而不是值 '12'。
- false - 值以值 false 的形式写入，而不是值 'false'。

#### ■ 更改其他值时，将删除外部值“复合”参数 CSV

由于解析逻辑出错，每当任何其他参数的值发生更改时，表单设计器就会删除“复合”参数的设定值。

#### ■ 在获取所有服务定义端点中添加的授权会破坏某些管道作业

要访问身份服务 API 端点以检索组织中的所有服务定义 (GET /csp/gateway/slc/api/definitions)，必须向请求提供身份验证凭据。

#### ■ RELEASE\_IPADDRESS\_PERIOD\_MINUTES 开关无法识别组织

全局运行以将 IP 地址从“已释放”移至“可用”的任务并非组织感知型任务。在一个或多个租户配置了超时的多组织/多租户环境中，将只选择一个值并将其应用于所有组织。

#### ■ 非受管磁盘类型的 Azure 虚拟机上的添加磁盘实施后操作。

vRA 不支持创建独立的非受管 Azure 磁盘。因此，必须在非受管磁盘类型的 Azure 虚拟机上禁用添加磁盘实施后操作。

#### ■ 作业 ID 为 0 的 Salt 配置创建失败。错误：工作节点部署和/或状态文件在 Windows 虚拟机上运行失败 (Minion deployment and/or state file run failed on Windows VM;s) [Salt 错误：无法启动 Salt (Failed to start Salt)]

工作节点部署在 Windows 虚拟机上失败，且 Salt 端显示以下错误

##### Salt 端错误

```
"return": "Exception occurred in runner deploy.minion: Traceback (most recent call
last):
  File \"/usr/lib/python3.7/site-packages/salt/client/mixins.py", line 390, in
low
  data["return"] = func(*args, **kwargs)
  File \"/usr/lib/python3.7/site-packages/salt/loader.py", line 1241, in _call
  return self.loader.run(run_func, *args, **kwargs)
  File \"/usr/lib/python3.7/site-packages/salt/loader.py", line 2274, in run
  return self._last_context.run(self._run_as, _func_or_method, *args, **kwargs)
  File \"/usr/lib/python3.7/site-packages/salt/loader.py", line 2289, in _run_as
  return _func_or_method(*args, **kwargs)
  File \"/usr/lib/python3.7/site-packages/sseape/runners/deploy.py", line 589, in minion
  raise salt.exceptions.SaltException('Error in installing salt minion - {}'.format(str(ret)))
salt.exceptions.SaltException: Error in installing salt minion - {'salt-vm-windows-test-mcm612-187496514722': {'Error':
{'Not Deployed': 'Failed to start Salt on host salt-vm-windows-test-mcm612-187496514722'}}
```

```
}}\n",  
"master_uuid": "a50dfade-26bf-42a5-be08-0b2d785af2c8",  
"minion_id": "saltstack_enterprise_installer",
```

- **未正确处理读取操作异常。**

如果部署迭代式更新发生后端错误，仅显示一般错误消息。在服务器日志中，显示了详细的错误消息。但是，由于未正确处理异常，UI 中仅显示一般错误消息。

- **请求跟踪器不适用于资源视图。**

在“所有资源”页面上，选择计算机并执行任何实施后操作后，除非启动手动刷新，否则不会显示请求跟踪器。

# 已知问题

# 7

此版本中存在以下已知问题。

- **启动升级到 8.5.1 和 8.6.0 失败。**

在 vRealize Automation 8.5.0 系统上，启动通过 vRSLCM 以迭代方式升级到 vRealize Automation 8.5.1 或更高版本时，大约在启动升级后一分钟左右的该操作在第 1 阶段的“vRealize Automation 升级/修补/内部网络”步骤失败。之前的升级虽然成功完成，但无法删除其运行时数据，导致升级处于“正在进行”状态。因此，无法启动新的升级。这可能会影响已从 vRealize Automation 8.4.x 升级到 8.5.0 且具有长主机名 (FQDN) 的某些系统。

**解决办法** 在此版本中，LCM 将执行预检查，并就此问题向您发送通知。有关解决办法步骤的信息，请参见知识库文章 [85965](#)。

- **从 vRealize Automation 8.5 和 8.5.1 升级可能会失败，并显示错误“升级由于严重错误而终止 (Upgrade terminated due to critical error)”。**

从 vRealize Automation 8.5 或 8.5.1 升级可能会失败，并显示错误“升级由于严重错误而终止 (Upgrade terminated due to critical error)”。磁盘空间检查显示 /root 的利用率为 \*或接近\* 100%。

**解决办法** 有关解决办法步骤的信息，请参见知识库文章 [85864](#)。

- **重新载入虚拟机时，不会在内部 IPAM 中分配 IPv4 和 IPv6 地址。**

对于已载入且其 IP 成功分配的虚拟机，取消注册该虚拟机并立即载入时，其 IP 仍将保持已释放状态，而不是回到已分配状态。

**解决办法** 等待 30 分钟，然后再重新载入虚拟机以分配 IP。

- **现在支持使用 API 通过自定义表单对目录项进行自定义验证**

如果客户使用 vRA 8.6 并通过 UI 进行目录项表单外部验证，则升级到 vRA 8.7 后，通过 API 请求目录项时，不会执行外部验证。

**解决办法** 在 Service Broker UI 上，转到目录项具有的自定义表单，然后单击 UI 上的“保存”按钮，重新保存表单。也可以找到目录项 ID 和表单 ID，然后使用 PATCH /catalog/api/admin/items/{catalog-item-id} 在目录项中填充 formId。

- **SSC 主节点身份验证失败。**



RaaS 实例运行时，每隔 24 小时，密钥轮换引擎就会尝试刷新一次 jwt 令牌。在某些情况下，引擎会保留过期的 jwt 令牌，而不是刷新该令牌，从而导致 **salt-master** 服务中出现 401 回溯追踪错误，因为它无法对 RaaS 服务进行身份验证。这将导致 SSC 的某些关键功能失败。

**解决办法** 在运行 **salt-master** 服务的虚拟机上，执行以下操作：

- a 移除 sseapi\_key.pub

```
rm /etc/salt/pki/master/sseapi_key.pub
```

- b 移除 jwt 身份验证令牌

```
rm /var/cache/salt/master/auth_token.jwt
```

- c 重新启动 salt-master 服务

```
systemctl restart salt-master
```

## ■ SSC vRA 8.6.1 Windows 工作节点和状态文件部署支持已终止

**概述** 有时，在 Windows 上安装工作节点会失败。

**问题摘要** 到工作节点安装启动时，必需的 Windows 服务未在主机上运行。

**解决方案** 升级到 SSC 插件版本 8.6.2。该版本引入了 180 秒的默认延迟，以便所有必需 Windows 服务变为活动状态。有关 SSC 插件的安装信息，请访问 <https://docs.vmware.com/cn/VMware-vRealize-Automation-SaltStack-Config/8.6/install-configure-saltstack-config/GUID-E5ECC4A0-ABEF-475D-8BF9-53429C0CB6DB.html>

**影响** 用户需要将 SSC 插件版本更新到 8.6.2

## ■ SSC 目标组搜索包含前导空格会破坏搜索功能

目标组搜索包含前导空格会导致搜索功能被破坏。将出现无限旋转状况且无法查看目标组。

刷新或离开页面。如果不刷新页面，该功能将不可用，并且无法访问目标组数据网格。

## ■ 在云模板中错误地放置元素会破坏 UI 页面。

在 Firefox 中，使用拖放操作有时可能会重定向页面。在拖动资源节点时，将其拖放到画布外部也可能会导致在 Firefox 中重定向页面。

**解决办法** 将资源放置在画布中并将其删除。

## ■ 自定义资源订阅不适用于基于可扩展性操作的自定义资源。

vRealize Automation 8.5.1 引入了基于可扩展性操作的自定义资源，但该功能存在一些限制。例如，云管理员仍然无法在基于事件的订阅中包含基于可扩展性操作的资源。

## ■ 在基于可扩展性操作的自定义资源的部署更新期间出现超时异常。

更新基于可扩展性操作的自定义资源部署时，可能会显示“504 网关超时问题 (504 Gateway Time-out issue)”错误。如果可扩展性操作读取失败，则会出现该错误。

## 已更改和已弃用的功能

# 8

无