

了解Catalyst 9000交换机上的应用托管

目录

[简介](#)

[先决条件](#)

[要求](#)

[使用的组件](#)

[背景信息](#)

[应用托管框架的限制](#)

[应用托管架构](#)

[应用托管支持的Catalyst型号](#)

[应用的硬件资源](#)

[容器网络](#)

[Docker运行时选项](#)

[不受支持的Docker运行时选项列表](#)

[配置](#)

[故障排除](#)

[相关信息](#)

简介

本文档介绍如何在Catalyst 9000系列平台上实施应用托管并对其进行故障排除。

先决条件

要求

Cisco 建议您了解以下主题：

- LAN交换基础。
- 熟悉 Cisco IOS® XE和许可。
- 了解交换架构。
- 熟悉基本的Linux命令。

使用的组件

本文档中的信息基于以下软件和硬件版本：

- Catalyst 9300
- Catalyst 9400
- Catalyst 9500
- Cisco IOS® XE和 16.12.X或 17.X软件

- 有关详细信息，请参阅支持的Catalyst型号应用托管



注意：有关在其他思科平台上启用这些功能所使用的命令，请参阅相应的配置指南。

本文档中的信息都是基于特定实验室环境中的设备编写的。本文档中使用的所有设备最初均采用原始（默认）配置。如果您的网络处于活动状态，请确保您了解所有命令的潜在影响。

背景信息

企业网络中用于解决各种关键业务使用案例的应用程序至关重要，包括协议分析器等管理工具以及入侵检测服务等安全解决方案。传统上，这些应用运行在外部物理或基于云的虚拟服务器上，但现代需求凸显了此类方法的局限性。

Cisco Catalyst 9000系列交换机通过集成专为现代数字时代设计的高级功能使企业网络发生了革命性的变化。这些交换机由Cisco IOS® XE操作系统和x86 CPU支持，不仅仅是传统网络设备；它们作为应用托管的平台，使企业能够整合基础设施并为边缘计算创造新的可能性。这样，安全代理、IoT传感器和流量监控工具等应用可以直接在交换机上运行，无需外部计算硬件。

应用托管框架的限制

要在Catalyst 9000上启用应用托管框架，需要满足以下要求：

- 交换机必须运行版本16.12。只有版本16.12支持Docker应用，因为该版本支持本地Docker引擎。
- 在Catalyst 9000上启用应用托管之前，必须在设备后面板USB3.0端口中安装思科认证的USB3.0闪存驱动器。应用托管仅在后面板USB3.0上工作。
- Catalyst 9000平台需要思科智能许可。要启用应用托管，需要思科DNA优势许可。
- 仅支持Docker容器。Docker应用可以安装在思科支持的SSD存储上。
- 内部闪存不支持第三方应用程序，只有思科签名的应用程序，如ThousandEyes Agents。
- 第三方容器必须位于Catalyst 9000交换机提供的其中一个外部固态硬盘存储选项中。Catalyst 9300系列交换机支持SSD存储。Catalyst 9400系列和Catalyst 9500系列高性能型号以及Catalyst 9600系列交换机支持SATA存储。交换机前面板USB端口不能使用SSD存储或SATA存储。
- 思科不提供任何预打包的第三方不支持的应用。客户必须自己打包。某些第三个与思科合作的方供应商可以共享特定于思科安装的链接。
- 仅Catalyst 9300LM型号支持用于应用托管的前面板USB端口。

应用托管架构

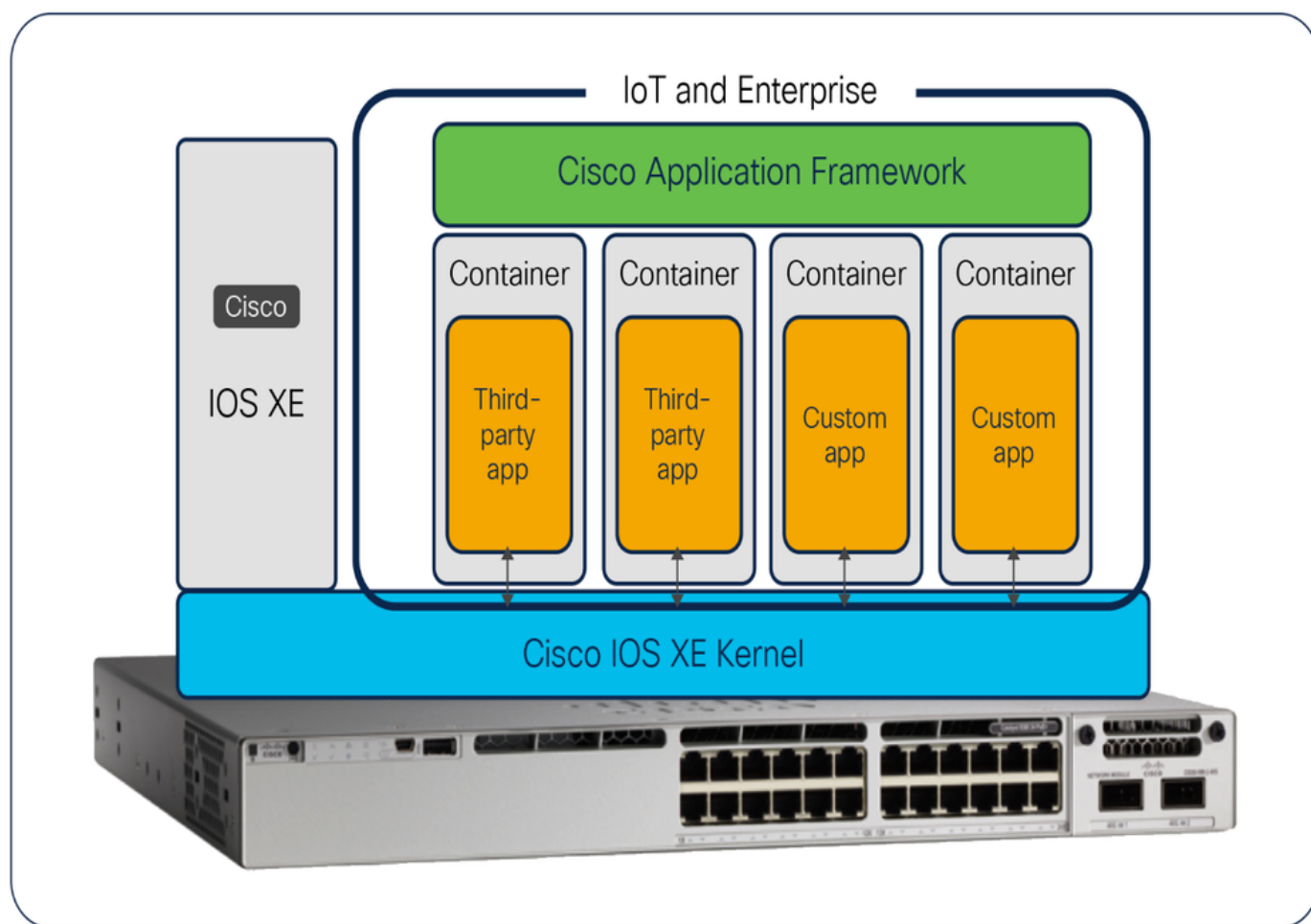
为了支持Cisco Catalyst 9000交换机上的应用托管功能，该交换机提供硬件资源，使应用可以驻留和执行。在Cisco Catalyst 9000交换机上运行的Cisco IOS XE为应用托管预留专用内存和CPU资源，以便为用户应用提供单独的执行空间，而不会影响交换机的完整性和性能。

此外，应用程序必须驻留在其中一个外部固态驱动器(SSD)存储选项（USB或M2 SATA）中，具体

取决于特定的Cisco Catalyst 9000平台。应用程序无权访问内部设备闪存，该闪存为Cisco IOS XE保留以保护其完整性。

为了获得最大的灵活性以及与主操作系统的完全隔离，Cisco Catalyst 9000交换机上的Cisco IOS XE内核和Cisco Application Framework通过利用控制组(Cgroups)和用户命名空间来支持容器化应用。Cgroups限制对物理资源的访问，例如应用程序的CPU和内存。思科应用框架检查是否有足够的资源来激活和安装应用。如果硬件资源不可用于该应用程序，则它无法激活该应用程序，并且会向管理员发送相关消息。

图 1. 显示了Cisco Catalyst 9000平台上思科应用框架的直观表示



此外，SSD存储通过在SSD存储上提供AES-256硬件加密，并在SSD存储和交换机上提供密码身份验证，从而提供一流的安全性。

AES-256加密完全在硬件中完成。使用密码身份验证时，必须在SSD和交换机上设置密码。将预配置了密码身份验证的SSD插入没有匹配密码配置的Catalyst 9000交换机时，身份验证失败，因为交换机没有配置正确的密码。SSD存储和交换机上的密码必须匹配，才能成功取消身份验证，如图2所示。

如果从Catalyst 9000交换机移除已配置密码的SSD存储并将其插入非Catalyst交换机中，则内容是安全的，不可访问。只有在Catalyst交换机中使用正确的密码解锁后，才能访问任何敏感数据。

图2.显示SSD存储上的验证码:



- 例如，可以使用以下命令启用或禁用安全性并设置密码：

```
<#root>
Cat9k#
hw-module switch 1 usbflash1 security ?

    disable    disable security on USB3.0
    enable     Enable security on USB3.0
    unlock     Unlock USB3.0

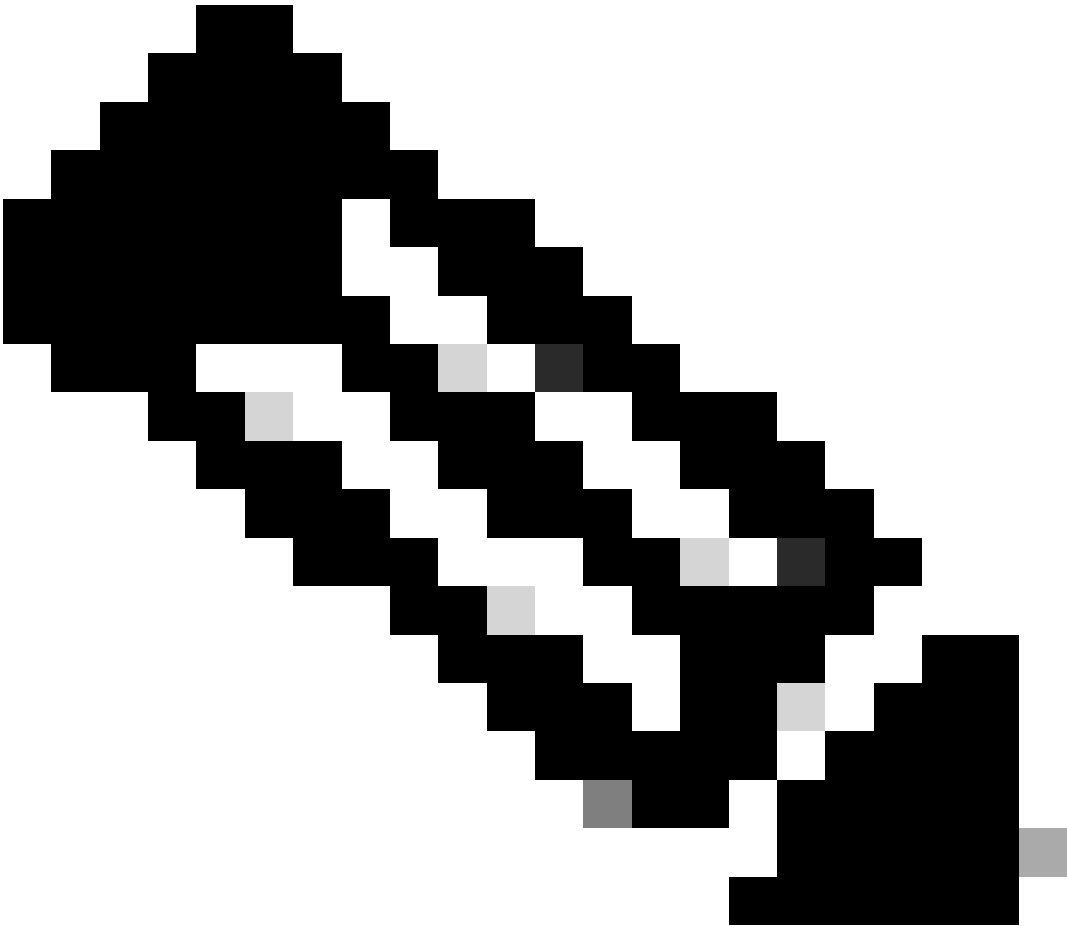
Cat9k(config)#hw-module switch 1 usbflash1-password
```

应用托管支持的Catalyst型号

支持以下应用托管功能：

支持的平台	Cisco IOS XE版本
Catalyst 9300/L	16.12.1版本或更高版本
Catalyst 9300X	17.5.1版本或更高版本
Catalyst 9300LM	17.9.1版本或更高版本
Catalyst 9404 和 9407	17.1.1版本或更高版本
Catalyst 9410	17.5.1版本或更高版本
Catalyst 9400X	17.8.1版本或更高版本

Catalyst 9500高性能和9600系列	17.5.1版本或更高版本
Catalyst 9500X和9600X系列	17.8.1版本或更高版本



注意：Cisco Catalyst 9200系列和Catalyst 9500（基于UADP 2.0）：C9500-40X、C9500-16X、C9500-24Q、C9500-12Q)型号不支持应用托管。

应用的硬件资源

表1.包含应用的Cisco Catalyst 9000平台硬件资源。

Cisco Models	Networking (AppGig Port)	Memory (RAM)	CPU	Storage
Catalyst 9300X	2x10G	Up to 8 GB	50% of total CPU (2 CPU Cores)	120/240 GB (USB 3.0 SSD)
Catalyst 9300/L/LM	1x1G	2 GB	25% of total CPU (1 CPU Core)	120/240 GB (USB 3.0 SSD)
Catalyst 9400X	2x10G	Up to 8 GB	25% of total CPU (1 CPU Core)	240 - 960 GB (SATA)
Catalyst 9400	1x1G	Up to 8 GB	25% of total CPU (1 CPU Core)	240 - 960 GB (SATA)
Catalyst 9500X	2x10G	Up to 8 GB	25% of total CPU (1 CPU Core)	240 - 960 GB (SATA)
Catalyst 9500 High performance	Management Port*	Up to 8 GB	25% of total CPU (1 CPU Core)	240 - 960 GB (SATA)
Catalyst 9600X	Management Port*	Up to 8 GB	25% of total CPU (1 CPU Core)	240 - 960 GB (SATA)
Catalyst 9600	Management Port*	Up to 8 GB	25% of total CPU (1 CPU Core)	240 - 960 GB (SATA)

如果默认选项不足，可以使用自定义资源配置文件保留资源、CPU、内存和vCPU。有关设备资源限制，请参阅以下信息：

Platform	Memory (GB)	vCPUs	CPU Units	USB Back Storage (GB)	M2 SATA Storage (GB)
Catalyst 9300	2	2	7400	120	NA
Catalyst 9400	4-10	2	7400	NA	960
Catalyst 9500	8	2	7400	120	NA
Catalyst 9500 High-performance	8	2	7400	NA	960

- vCPU:这表示单个应用程序可以同时使用的虚拟CPU的最大数量。
- CPU单元：表示分配给应用程序托管的CPU负载资源总数。每个应用都指定其确保可靠运行所需的最低CPU负载。
- USB背存储：指后面板凹式USB 3.0插槽。只有此后面板USB 3.0插槽支持应用托管。
- M.2 SATA存储：表示用于应用数据存储的内部固态硬盘(SSD)。
- 不适用：表示功能或资源对于指定的配置不可用。
- 此示例说明如何创建自定义配置文件：

```
<#root>
```

```
Cat9k(config)#
```

```
app-hosting appid MYAPP
```

```
Cat9k(config-app-hosting)#
```

```
app-resource profile custom
```

```
Cat9k(config-app-resource-profile-custom)#
```

```
cpu 7400
```

```
Cat9k(config-app-resource-profile-custom)#
```

```
memory 2048
```

```
Cat9k(config-app-resource-profile-custom)#
```

```
vcpu 2
```

应用高可用性

Catalyst 9000交换机支持应用自动重启功能，该功能可在系统切换或重启时保留应用上次配置的运行状态。此功能默认启用，并且主用和备用交换机上都需要相同的存储类型。

表 2. Cisco Catalyst 9000平台应用自动重启功能

Supported Platforms	IOS XE Release
Catalyst 9300/L StackWise 480/360 (1+1 mode only)	17.2.1
Catalyst 9300LM StackWise 360 (1+1 mode only)	17.9.1
Catalyst 9300X StackWise 1T (1+1 mode only)	17.6.1
Catalyst 9400 Dual Sup (Single Chassis and StackWise Virtual)	17.5.1
Catalyst 9400X Dual Sup (Single Chassis and StackWise Virtual)	17.9.1
Catalyst 9500 High Performance StackWise Virtual	17.5.1
Catalyst 9600 Dual Sup (Single Chassis and StackWise Virtual)	17.5.1
Catalyst 9600X Dual Sup (Single Chassis)	17.9.1

应用托管框架

默认情况下未启用应用托管框架。思科应用框架(CAF)由思科构建，用于管理在任何网络设备上运行的容器化应用。CAF也称为IOx。Cisco IOx可在网络边缘执行IoT应用（雾计算），同时确保与Cisco IOS软件的安全连接。它可为应用程序托管保留4 GB的磁盘空间。通过CLI禁用IOx或IOx基础设施使用SSD时，会清除分区和应用数据。

- 这是用于启用应用托管基础设施的Cisco IOS XE CLI配置：

```
Cat9k(config)#iox
```



注意：从Cisco IOS XE 16.12.1开始，IoX打包不再为必填项。可以按原样安装Docker应用。这样，用户无需额外打包即可构建并自带应用。应用开发人员可以找到有关在[Cisco DevNet](#)站点上托管应用的[更多信息](#)

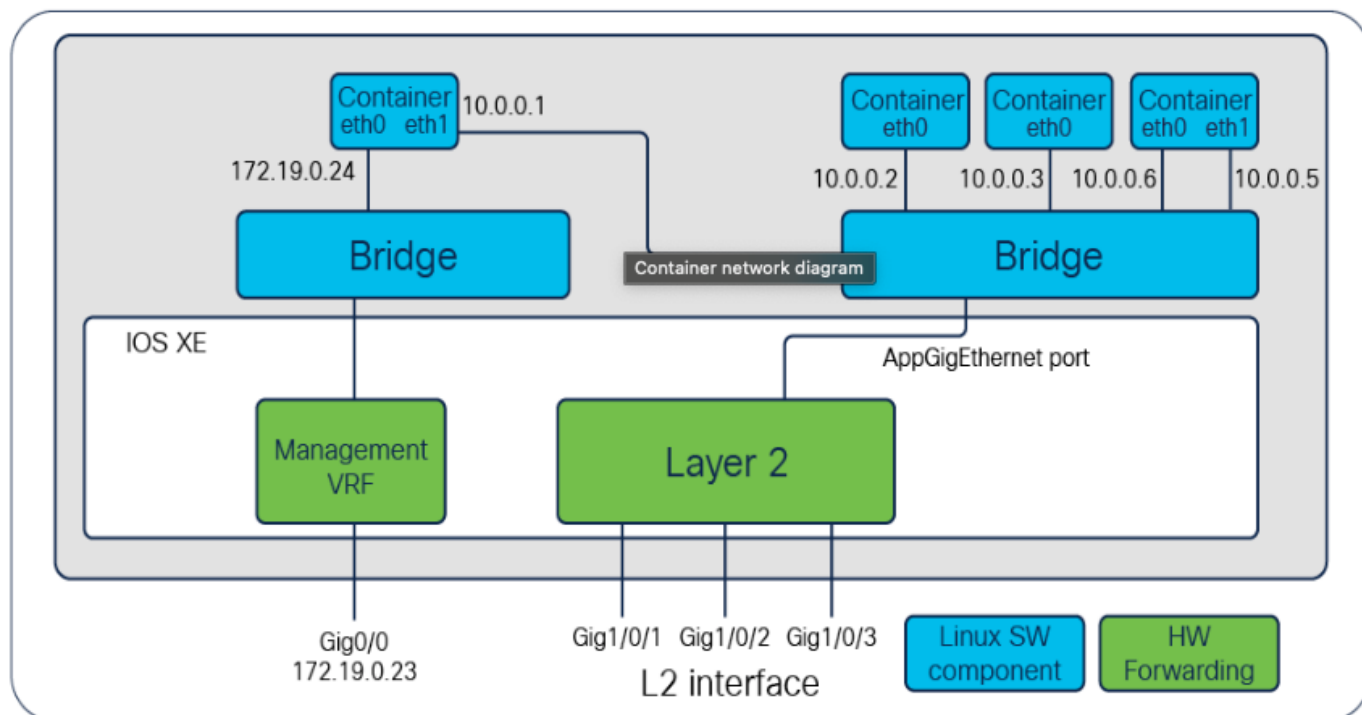
容器网络

该图说明了容器的连接体系结构。它突出显示了Cat9k支持应用托管的所有可能的网络模式。此类设备包括：

- 使用专用内部AppGigabitEthernet接口的前面板数据端口交换
- 管理接口(GigabitEthernet 0/0)

AppGigabitEthernet是一个内部硬件数据端口，硬件交换到前面板数据端口。使用管理接口和前面板数据端口连接容器。在容器中，虚拟网络接口卡(vNIC)显示为标准以太网接口，标记为eth0、eth1等。此设计有助于在容器化环境中实现无缝集成和一致的网络配置。

图 3. 容器网络图



Cisco Catalyst 9300、Catalyst 9400和Catalyst 9500X系列交换机上引入的AppGigabitEthernet接口提供专用的应用流量功能。它是一个内部硬件数据端口，通过硬件交换连接到前面板数据端口。AppGigabitEthernet接口可以配置为中继或VLAN特定接口。对于TRUNK接口，它扩展为第2层TRUNK端口，该端口接收的所有流量都可用于应用。对于VLAN接口，应用程序通过指定VLAN ID号连接到特定VLAN网络。

AppGbEthernet;接口仅在Cisco Catalyst 9300系列、Catalyst 9400系列和Catalyst 9500X交换机上可用。配备管理引擎1的Catalyst 9410机箱需要禁用插槽4端口48（如果适用）以启用AppGigabitEthernet端口。Catalyst 9500高性能和9600系列交换机不支持AppGigabitEthernet接口。这些型号上托管的应用程序的连接是通过管理接口通过环回接口从任何前面板端口实现的。



注意：C9300X、C9400X-Sup-2/2XL和C9500X型号具有2个10G AppGigabitEthernet端口。

应用托管网络

- 对于TRUNK接口，通过端口接收的所有流量均可供应用使用

```
<#root>
```

```
Cat9k(config)#
```

```
interface AppGigabitEthernet 1/0/1
```

```
Cat9k(config)#
```

```
switchport mode trunk
```

```
Cat9k(config-if)#exit
```

```
Cat9k(config)#
```

```
app-hosting appid MYAPP
```

```
Cat9k(config-app-hosting)#
```

```
app-vnic AppGigabitEthernet trunk
```

```
Cat9k(config-config-app-hosting-trunk)#
```

```
guest-interface
```

```
Cat9k(config-config-app-hosting-trunk-mode-guest)#
```

```
end
```

- 对于VLAN接口，应用连接到特定VLAN

```
<#root>
```

```
Cat9k(config)#
```

```
interface AppGigabitEthernet 1/0/1
```

```
Cat9k(config)#
```

```
switchport trunk allowed vlan
```

```
Cat9k(config-if)#
```

```
exit
```

```
Cat9k(config)#
```

```
app-hosting appid MYAPP
```

```
Cat9k(config-app-hosting)#
```

```
app-vnic AppGigabitEthernet trunk
```

```
Cat9k(config-config-app-hosting-trunk)#
```

vlan

guest-interface

```
Cat9k(config-config-app-hosting-trunk-mode-guest)#  
end
```

- 对于管理接口，应用连接到管理端口(GigabitEthernet0/0)

<#root>

```
Cat9k(config)#  
interface gigabitEthernet 0/0
```

```
Cat9k(config-if)#  
ip address
```

```
Cat9k(config-if)#  
exit
```

```
Cat9k(config)#  
app-hosting appid MYAPP
```

```
Cat9k(config-app-hosting)#(config-app-hosting)#  
app-vnic management guest-interface
```

```
Cat9k((config-app-hosting-mgmt-gateway)#  
end
```

分配给应用容器的IP地址：容器接口的IP地址可通过交换机CLI显式分配，也可通过DHCP动态获取。

- 通过AppGigabitEthernet配置应用的静态IP地址：

```
<#root>
```

```
Cat9k(config)#
```

```
app-hosting appid MYAPP
```

```
Cat9k(config-app-hosting)#
```

```
app-vnic AppGigabitEthernet trunk
```

```
Cat9k(config-config-app-hosting-trunk)#
```

```
vlan
```

```
guest-interface
```

```
Cat9k(config-config-app-hosting-vlan-access-ip)#
```

```
guest-ipaddress
```

netmask

<-- Container IP Address

```
Cat9k(config-config-app-hosting-vlan-access-ip)#  
exit
```

```
Cat9k(config-config-app-hosting-trunk)#  
exit
```

```
Cat9k(config-app-hosting)#  
app-default-gateway
```

guest-interface

```
Cat9k(config-app-hosting)#  
exit
```

- 通过GigabitEthernet 0/0为应用配置静态IP地址

<#root>

```
Cat9k(config)#  
app-hosting appid MYAPP
```

```
Cat9k(config-app-hosting)#  
app-vnic management guest-interface
```

```
Cat9k(config-app-hosting-mgmt-gateway)#
```

```
guest-ipaddress
```

```
netmask
```

```
Cat9k(config-app-hosting-mgmt-gateway)#
```

```
exit
```

```
Cat9k(config-app-hosting)#
```

```
app-default-gateway
```

```
guest-interface
```

```
Cat9k(config-app-hosting)#
```

```
exit
```

- 配置应用的动态IP地址(DHCP)

```
<#root>
```

```
Cat9k(config)#
```

```
app-hosting appid MYAPP
```

```
Cat9k(config-app-hosting)#
```

```
app-vnic AppGigabitEthernet trunk
```

```
Cat9k(config-config-app-hosting-trunk)#
```

```
vlan
```

```
guest-interface
```

```
Cat9k(config-config-app-hosting-vlan-access-ip)#
```

```
end
```

```
Cat9k#
```

```
Or through Interface
```

```
GigabitEthernet 0/0
```

```
Cat9k(config)#
```

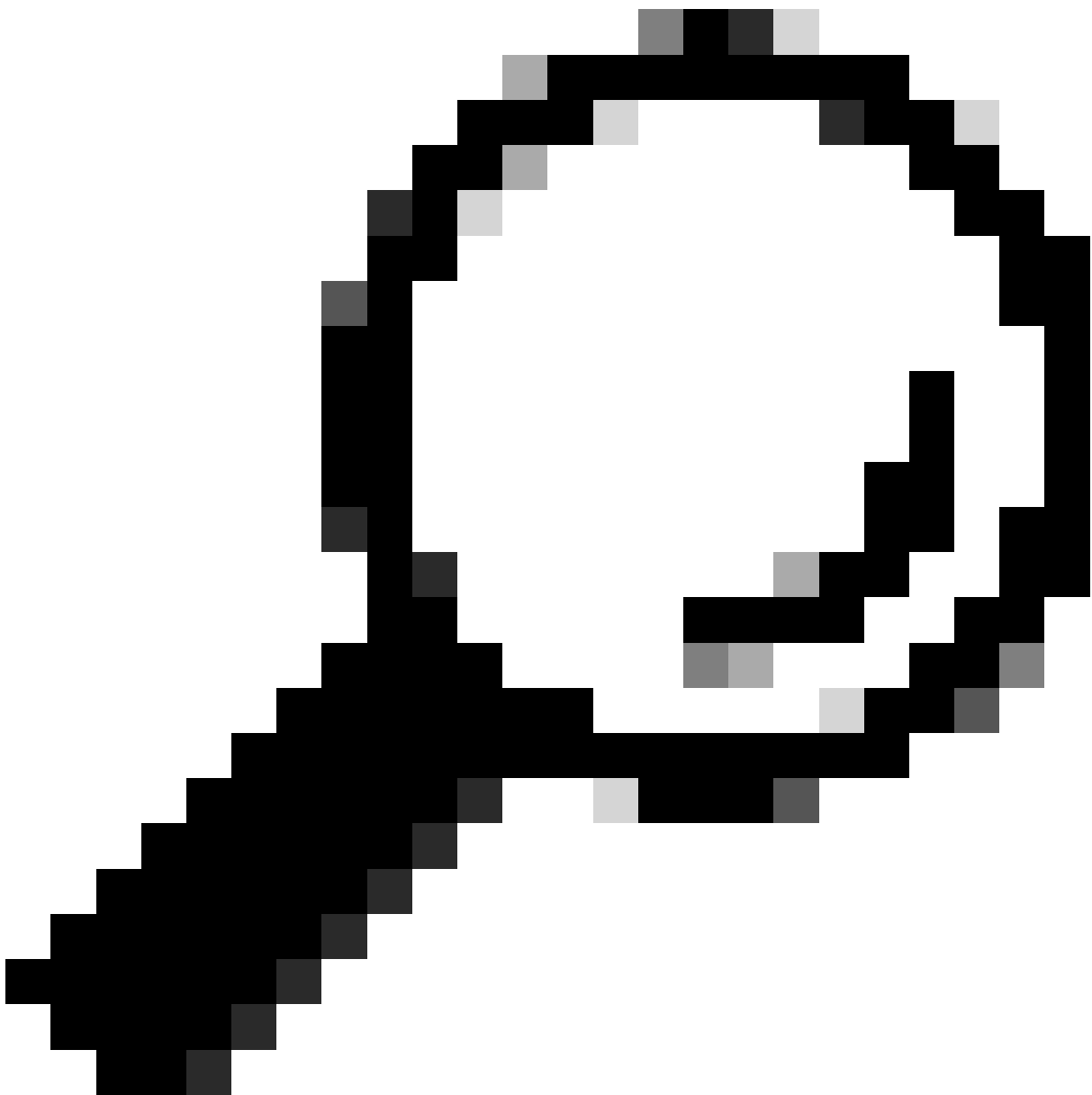
```
app-hosting appid MYAPP
```

```
Cat9k(config-app-hosting)#
```

```
app-vnic management guest-interface
```

```
Cat9k(config-app-hosting-mgmt-gateway)#
```

```
end
```



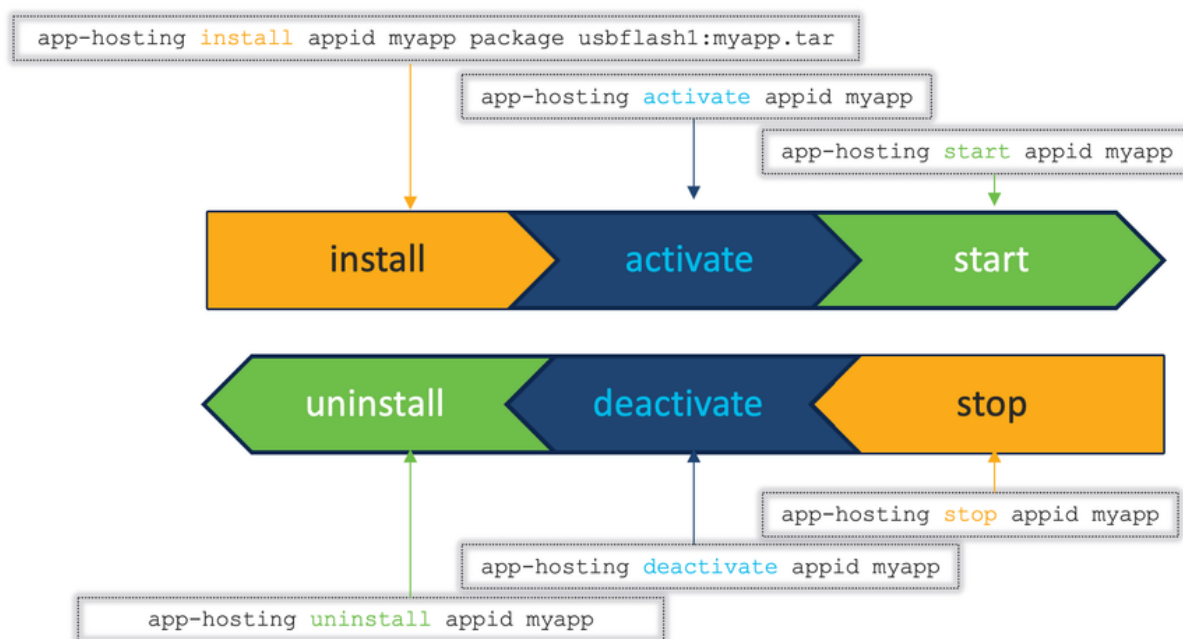
提示：在安装应用之前，请确保网络配置正确。如果需要更改，则停止、停用和卸载，然后重新启动

和应用程序的生命周期

Cisco Catalyst 9000交换机上的应用生命周期包括三个阶段，如图4所示：

- 安装:在此阶段，应用将安装在设备上。但是，应用程序所需的资源尚未分配。
- Activate:应用所需的硬件资源用于确保其功能。
- 开始:应用转换到运行状态，变为完全活动且运行正常。

图4.应用生命周期管理



Docker运行时选项

如果容器应用需要具有运行时选项，这些选项作为命令行选项（如控制器IP、数据目录等）传递，则可以使用“app-resource docker”命令配置这些选项。

系统支持多行运行选项字符串配置。以下是用户需要考虑的主要事项：

- 用户最多可以输入/复制粘贴30行docker运行选项。
- 系统按该顺序从第1行到第30行的字符串生成串连字符串。
- 每行run-option中的每个字符串最多可以包含235个字符。
- 每个字符串可以有多个“完整”docker run-option，只要它们全部包含在235个字符中。
- 请注意，任何行中的运行选项字符串都不能拆分为下一行。
- 无需使用空格终止每行中的字符串。当系统将所有现有run-option字符串串连成单个字符串时，系统会自动为每行生成一个空格。
- 如果用户对“运行”选项做出任何更改，用户需要停止、停用、激活并重新启动应用程序才能使新的运行选项生效。

以下是Docker选项及其在Cat9k交换机上可能的等效配置的示例：

```
docker run -v $(APP_DATA):/data --entrypoint startup.sh
```

Catalyst 9000交换机上的配置：

```
<#root>
```

```
Cat9k(config)#
```

```
app-hosting appid
```

MYAPP

```
Cat9k(config-app-hosting)#
```

```
app-resource docker
```

```
Cat9k(config-app-hosting-docker)#
```

```
run-opts 1 "-v ${APP_DATA}:/data"
```

```
Cat9k(config-app-hosting-docker)#
```

```
run-opts 2 "--entrypoint startup.sh"
```

对于外部永久数据存储，请使用命令：`run-opts 1 "-v /vol/usb1/iox_host_data_share:/(APP_DATA)"`

<#root>

```
Cat9k(config)#
```

```
app-hosting appid MYAPP
```

```
Cat9k(config-app-hosting)#
```

```
app-resource docker
```

```
Cat9k(config-app-hosting-docker)#
```

```
run-opts 1 "-v /vol/usb1/iox_host_data_share:/(APP_DATA)"
```

要从配置中删除运行选项行，请执行与以下示例类似的“no”命令：

<#root>

```
Cat9k(config)#
```

```
app-hosting appid MYAPP
```

```
Cat9k(config-app-hosting)#
```

```
app-resource docker
```

```
Cat9k(config-app-hosting-docker)#
```

```
no run-opts 1 "-v /vol/usb1/iox_host_data_share:/(APP_DATA)"
```

要删除应用程序配置中的所有运行选项，可以执行“no”配置命令：

<#root>

```
Cat9k(config)#
app-hosting appid MYAPP

Cat9k(config-app-hosting)#
no

app-resource docker
```

不受支持的Docker运行时选项列表

由于安全原因或选项不适用于Linux平台，因此列出的应用程序托管框架不支持Docker命令行选项。

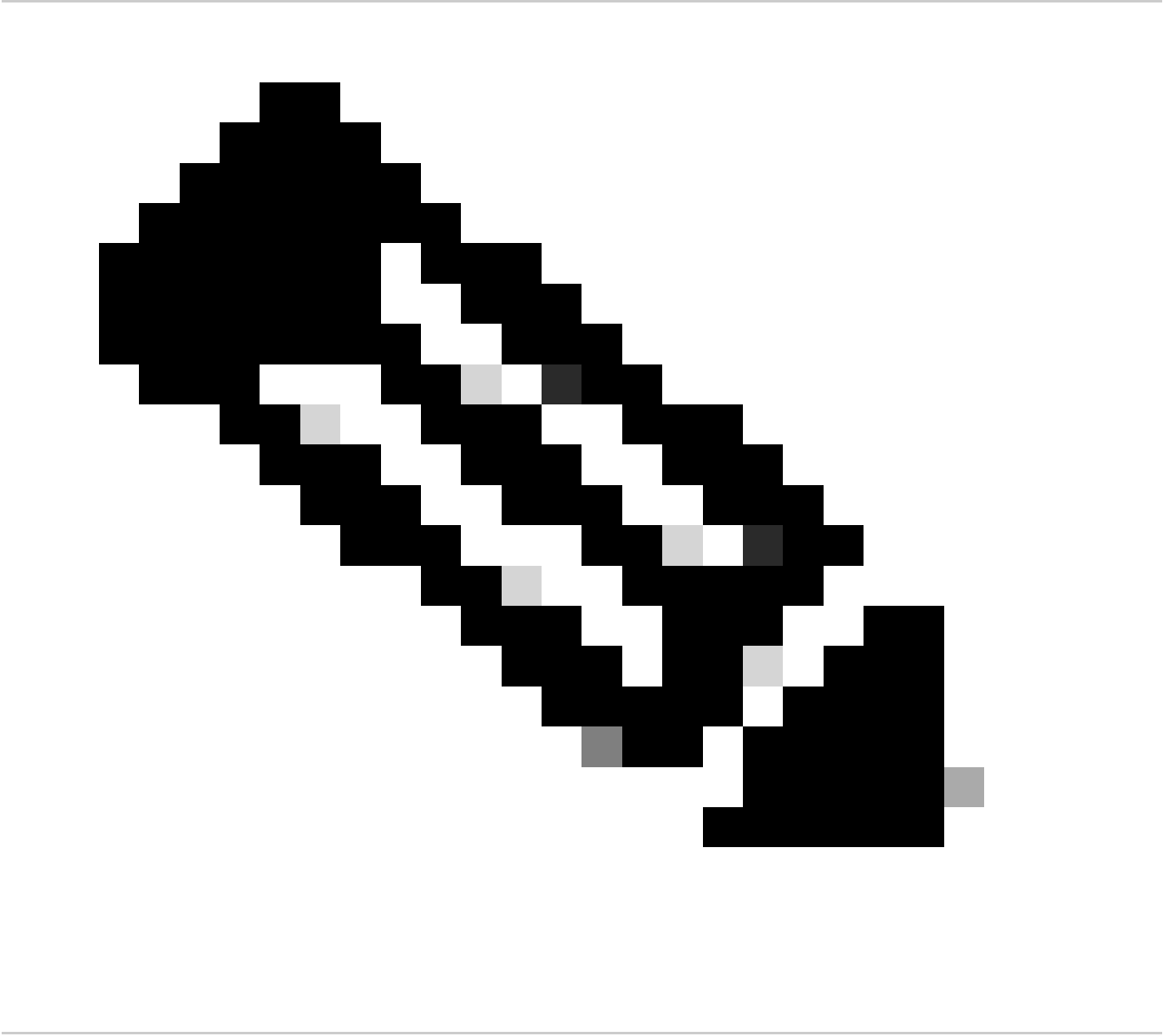
Docker选项	描述
--配售	连接到STDIN、STDOUT或STDERR。
— 无重设备	指定块IO权重（相对设备权重）。
—cgroup-parent	容器的可选父组。
—cidfile	将容器ID写入指定的文件。
—cpu-count	指定CPU计数（仅限Windows）。
—cpu-percent	指定CPU百分比（仅限Windows）。
-cpu	指定CPU数量(API 1.25+)。
—device-cgroup-rule	将规则添加到cgroup allowed devices列表。
—device-read-bps	限制设备的读取速率（每秒字节数）。
— 设备读取IOPS	限制设备的读取速率（I/O每秒）。
—device-write-bps	限制设备的写入速率（每秒字节数）。

—device-write-iops	限制设备的写入速率（ I/O每秒 ）。
—disable-content-trust	跳过图像验证。
-env-file	从文件中加载环境变量。
— 交互式(-i)	即使未连接STDIN，也要保持打开。
—io-maxbandwidth	系统驱动器的最大IO带宽限制（ 仅限 Windows ）。
—io-maxiops	系统驱动器的最大IOps限制（ 仅限Windows ）。
--ip	指定IPv4地址(示例：192.168.100.100)。
—ip6	指定IPv6地址(示例：2001:db8::44)。
— 隔离	指定容器隔离技术。
— 链接	添加指向其他容器的链接。
— 名称	为容器指定名称。
—oomkilldisable	禁用容器的OOM杀手。
--pid	指定要使用的PID命名空间。
--平台	指定平台(实验；API 1.32+)。
— 特权	向容器授予扩展权限。
--runtime	指定用于容器的运行时。
— 存储优化	定义容器的存储驱动程序选项。

—sysctl	指定sysctl选项。
—tty(-t)	分配伪TTY。
—usersns	指定要使用的用户命名空间。
—uts	指定要使用的UTS命名空间。
— 卷驱动程序	为容器指定可选的卷驱动程序。

配置

后续部分介绍如何在Catalyst 9000上配置和部署APP



注意：了解在Catalyst 9000系列交换机上无缝工作的合作伙伴解决方案目录[应用市场](#)

- 下载Docker映像并使用SCP、FTP、TFTP或USB存储复制到Cisco交换机。

```
<#root>
```

```
Cat9k#
```

```
dir usbflash1:/
```

```
Directory of usbflash1:/
```

```
17      -rw-          5843233 Jan 23 2025 20:50:01 +00:00
```

```
MYAPP.tar
```

- 配置网络参数(在本示例中，使用AppGigabitEthernet接口，即192.168.1.10/24，在VLAN 10下并使用Google解析器):

```
<#root>
```

```
Cat9k(config)#
```

```
interface AppGigabitEthernet 1/0/1
```

```
Cat9k(config-if)#
```

```
switchport trunk allowed vlan 10
```

```
Cat9k(config-if)#
```

```
exit
```

```
Cat9k(config)#
```

```
app-hosting appid MYAPP
```

```
Cat9k(config-app-hosting)#
```

```
app-vnic
```

```
AppGigabitEthernet
```

```
trunk
```

```
Cat9k(config-config-app-hosting-trunk)#
```

```
vlan 10 guest-interface 0
```

```
Cat9k(config-config-app-hosting-vlan-access-ip)#  
guest-ipaddress 192.168.1.10 netmask 255.255.255.0
```

```
Cat9k(config-config-app-hosting-vlan-access-ip)#  
exit
```

```
Cat9k(config-config-app-hosting-trunk)#  
exit
```

```
Cat9k(config-app-hosting)#  
app-default-gateway 192.168.1.1 guest-interface 0
```

```
Cat9k(config-app-hosting)#  
name-server 8.8.8.8
```

```
Cat9k(config-app-hosting)#  
exit
```

- 在交换机上启用IOx框架，然后等待直到所有服务都在运行：

```
<#root>
```

```
Cat9k(config)#  
iox
```

```
Cat9k#
```

```
show iox-service
```

```
IOx Infrastructure Summary:
```

```
-----  
IOx service (CAF)           : Running  
IOx service (HA)            : Running  
IOx service (IOxman)        : Running  
IOx service (Sec storage)    : Running  
Libvirtd 5.5.0               : Running  
Dockerd v19.03.13-ce        : Running  
Application DB Sync Info    : Available  
Sync Status                  : Disable
```

- 为特定应用设置所需的Docker运行选项：

<#root>

Cat9k(config)#

app-hosting appid MYAPP

Cat9k(config-app-hosting)#

app-resource docker

Cat9k(config-app-hosting-docker)#

run-opts 1 "

"

Cat9k(config-app-hosting-docker)#

run-opts 2 "

"

- 从SSD安装应用并验证其部署：

<#root>

Cat9k#

app-hosting install appid MYAPP package usbflash1:MYAPP.tar

Installing package 'usbflash1:MYAPP.tar' for 'MYAPP'. Use 'show app-hosting list' for progress.

Cat9k#

show app-hosting list

App id	State

MYAPP	
DEPLOYED	

- 激活应用程序并验证其状态：

<#root>

Cat9k#

app-hosting activate appid MYAPP

MYAPP activated successfully

Current state is: ACTIVATED

Cat9k#

show app-hosting list

App id	State

MYAPP	
ACTIVATED	

- 启动应用程序并验证其运行状态

<#root>

Cat9k#

app-hosting start appid MYAPP

MYAPP started successfully

Current state is: RUNNING

Cat9k#

show app-hosting list

App id	State

MYAPP	
RUNNING	

- 保存您的配置更改，以确保这些更改在重新启动后保持不变：

<#root>

Cat9k#

app-hosting start appid MYAPP

MYAPP started successfully

Current state is: RUNNING

```
Cat9k#  
  
show app-hosting list
```

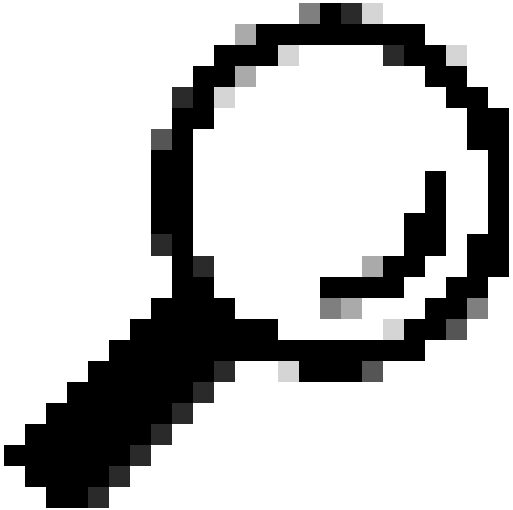
App id	State

MYAPP	
RUNNING	

故障排除

下表列出了可用于对应用托管进行故障排除的各种命令：

命令	目的
show iox-service	显示接口错误计数器
show app-hosting list	显示已安装的设备的列表
show app-hosting detail appid MYAPP	显示有关设备的详细信息
show app-hosting resource	显示可用资源
show app-hosting utilization appid MYAPP	显示设备的利用率信息
应用托管将appid MYAPP日志移动到bootflash:	使用此命令可捕获应用特定跟踪日志（如果存在）。
应用托管将系统技术支持移动到？ bootflash: 目标路径 crashinfo-1: 目标路径 crashinfo: 目标路径 flash-1: 目标路径 flash: 目标路径 webui: 目标路径	使用此命令可将系统技术支持移至备用目录。
show app-hosting infra	使用此命令可检查是否已启用签名验证。

	 提示：仅当应用托管使用 USB/SSD作为介质时，才能禁用 应用签名验证。
应用托管验证？ 禁用应用验证禁用 enable App verification enable	使用此命令可启用/禁用签名验证（形式为 CLI权限模式）。
应用托管连接appid MYAPP会话	使用此命令可访问应用控制台并验证容器 内进程的状态（从CLI权限模式）。

修改Docker容器

如果Docker需要修改，请遵循以下过程：

- 停止应用程序：

```
<#root>  
Cat9k#  
app-hosting stop appid MYAPP  
  
MYAPP stopped successfully  
  
Current state is:  
STOPPED
```

- 停用应用：

<#root>

Cat9k#

app-hosting deactivate appid MYAPP

MYAPP deactivated successfully

Current state is:

DEPLOYED

- 修改Docker选项：

<#root>

Cat9k#

app-hosting start appid MYAPP

Cat9k(config)#

app-hosting appid MYAPP

Cat9k(config-app-hosting)#

app-resource docker

Cat9k(config-app-hosting-docker)#

prepend-pkg-opts

Cat9k(config-app-hosting-docker)#

Cat9k(config-app-hosting-docker)#

exit

Cat9k(config-app-hosting)#

exit

Cat9k(config)#

exit

- 重新激活应用程序：

```
<#root>
```

```
Cat9k#
```

```
app-hosting activate appid MYAPP
```

```
MYAPP activated successfully
```

```
Current state is:
```

```
STOPPED
```

- 启动应用程序：

```
<#root>
```

```
Cat9k#
```

```
app-hosting start appid MYAPP
```

```
MYAPP started successfully
```

```
Current state is:
```

```
RUNNING
```

相关信息

- [Cisco Catalyst 9000系列交换机上的应用托管白皮书](#)
- [应用市场](#)
- [企业中的应用程序托管](#)

关于此翻译

思科采用人工翻译与机器翻译相结合的方式将此文档翻译成不同语言，希望全球的用户都能通过各自的语言得到支持性的内容。

请注意：即使是最好的机器翻译，其准确度也不及专业翻译人员的水平。

Cisco Systems, Inc. 对于翻译的准确性不承担任何责任，并建议您总是参考英文原始文档（已提供链接）。