

使用 vRealize Log Insight 代理

2021 年 9 月 07 日
vRealize Log Insight 8.2

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

**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

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

广州办公室
广州市
天河路 385 号
太古汇一座 3502 室
www.vmware.com/cn

目录

使用 vRealize Log Insight 代理 5

1 vRealize Log Insight 代理概览 6

2 日志轮换方案的类型 8

3 安装或升级 vRealize Log Insight 代理 9

下载代理安装文件 10

安装 Windows 代理 11

使用安装向导安装或更新 vRealize Log Insight Windows 代理 11

从命令行中安装或更新 vRealize Log Insight Windows 代理 12

将 Log Insight Windows Agent 部署到多台计算机 13

安装或更新 vRealize Log Insight Linux 代理 RPM 软件包 17

安装或更新 vRealize Log Insight Linux 代理 DEB 软件包 18

自定义 Debian Linux 的代理安装 19

安装 Log Insight Linux Agent 二进制软件包 21

Linux 上用于安装 vRealize Log Insight 代理的命令行选项 22

vRealize Log Insight 代理的自动更新 23

为各个代理禁用或启用自动更新 23

4 配置 vRealize Log Insight 代理 25

配置 Log Insight Windows Agent 26

Log Insight Windows Agent 的默认配置 26

从 Windows 事件通道收集事件 28

从日志文件收集事件 32

将事件转发到 Log Insight Windows Agent 37

配置 Log Insight Linux Agent 37

vRealize Log Insight Linux 代理的默认配置 38

从日志文件收集事件 39

筛选来自 vRealize Log Insight 代理的事件 46

从 vRealize Log Insight 代理转发信息 48

设置目标 vRealize Log Insight 服务器 48

指定代理的目标 51

vRealize Log Insight 代理的集中式配置 54

配置合并示例 55

使用公用值进行代理配置 57

分析日志 58

配置日志分析程序 58

5 卸载 vRealize Log Insight 代理 86

- 卸载 Log Insight Windows Agent 86
- 卸载 Log Insight Linux 代理 RPM 软件包 86
- 卸载 Log Insight Linux 代理 DEB 软件包 87
- 卸载 Log Insight Linux 代理 bin 软件包 87
- 手动卸载 Log Insight Linux 代理 bin 软件包 88

6 vRealize Log Insight 代理故障排除 89

- 为 Log Insight Windows Agent 创建支持包 89
- 为 Log Insight Linux Agent 创建支持包 90
- 在 Log Insight Agents 中定义日志详细信息级别 90
- 管理 UI 不显示 Log Insight Agents 91
- vRealize Log Insight 代理不发送事件 92
- 为 Log Insight Windows Agent 添加出站例外规则 93
- 在 Windows 防火墙中允许来自 Log Insight Windows Agent 的出站连接 94
- Log Insight Windows Agent 批量部署失败 94
- Log Insight Agents 拒绝自签名证书 95
- vRealize Log Insight 服务器拒绝非加密流量的连接 96

使用 vRealize Log Insight 代理

《使用 vRealize Log Insight 代理》介绍了如何安装和配置 vRealize™Log Insight™ Windows 和 Linux 代理。还包括故障排除提示。

这些信息供需要对 Log Insight Agents 进行安装、配置或故障排除的任何用户使用。这些信息是为熟悉虚拟机技术和数据中心操作并具有丰富经验的 Windows 或 Linux 系统管理员编写的。

有关如何通过 vRealize Log Insight 服务器为代理创建配置类的信息，请参阅《管理 vRealize Log Insight》。

vRealize Log Insight 代理概览

1

vRealize Log Insight 代理会从日志文件中收集事件，并将这些事件转发到 vRealize Log Insight 服务器或任何第三方 syslog 目标。

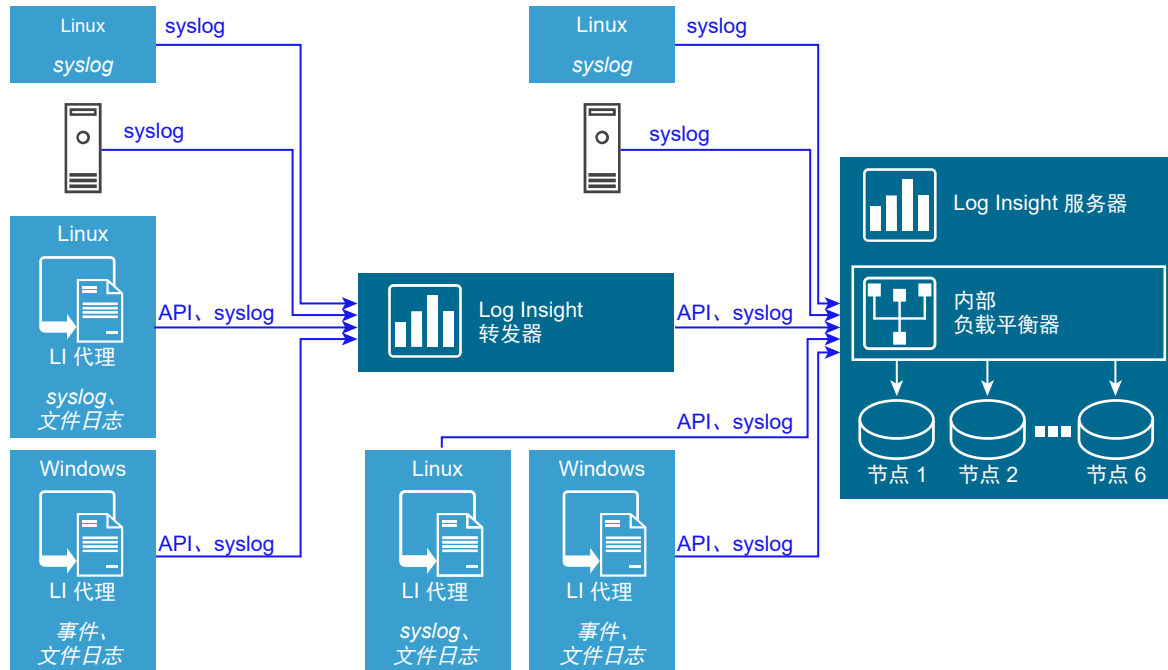
代理支持 syslog 和 vRealize Log Insight 数据获取 API（cfapi 协议），可以将其用于 Linux 或 Windows 平台。您可以通过 Web 界面配置代理（在服务器和客户端中包含 liagent.ini 文件），也可以在安装过程中进行配置。

代理具备以下特点：

- 单个或组部署
- 自动升级
- 可对日志消息进行分析并提取结构化数据。可以为 FileLog 和/或 WinLog 收集器配置分析程序。
- 支持多行消息
- 本机支持多个日志轮换方案
- 广泛的数据获取 API，包括客户端压缩、加密，以及向事件添加元数据的功能

vRealize Log Insight 服务器支持集中式的配置管理，以及创建和管理多组代理。

下图显示了代理部署配置的元素。



vRealize Log Insight 转发器是一个专用 vRealize Log Insight 服务器的实例，其主要职责是将事件转发到远程目标。通常，不会使用用作转发器的服务器实例进行查询。转发器使用内部负载均衡器，另外其结构与 vRealize Log Insight 服务器类似。

代理写入自己的运行日志。对于 **Windows**，这些日志位于 `C:\ProgramData\VMware\Log Insight Agent\logs` 目录中。对于 **Linux**，该操作日志的路径为 `/var/log/loginsight-agent/liagent_*.log`。在重新启动代理或日志文件大小达到 **10 MB** 时，将轮换这些文件。轮换的文件组合限制为 **50 MB**。不能使用 vRealize Log Insight 代理本身收集代理日志。

代理用于实时日志收集。可以使用 vRealize Log Insight 导入程序导入历史日志集合，包括支持包。

分别提供了适用于 **Windows** 和 **Linux** 操作系统的单独安装下载项。

在 **Windows** 系统上，该代理作为 **Windows** 服务运行，并在安装后立即启动。代理会监控应用程序日志文件和 **Windows** 事件通道，后者是用于收集相关 **Windows** 系统事件的池。收集到的事件会被转发到 vRealize Log Insight 服务器或第三方 **syslog** 目标。

在 **Linux** 系统上，该代理作为守护进程运行，并在安装后立即启动。vRealize Log Insight Linux 代理从 **Linux** 计算机上的日志文件中收集事件，并将这些事件转发到 vRealize Log Insight 服务器或 **syslog** 目标。可以使用 **Debian**、**Red Hat** 和 **Linux** 二进制安装软件包。

vRealize Log Insight 代理支持的日志轮换方案

2

vRealize Log Insight 代理支持多种日志轮换方案。

日志轮换可确保日志文件不会无限增大。有多种日志轮换方案，每一种方案都为了一组特定的用例而设计。vRealize Log Insight 本机支持以下方案。

表 2-1. vRealize Log Insight 代理支持的日志轮换方案

日志轮换方案	描述
<code>create-new</code>	达到大小或时间限制时，创建新的日志文件。日志记录进程将停止写入当前日志文件，并将日志输出定向到新创建的文件。现有文件不会被重命名，也不会以任何其他方式进行处理。
<code>rename-recreate</code>	达到大小或时间限制时，一个外部实用程序（例如 <code>logrotate</code> ）会重命名日志文件。然后，日志记录进程会使用之前的名称创建一个日志文件。
<code>copy-truncate</code>	达到大小或时间限制时，一个外部实用程序（例如 <code>logrotate</code> ）会复制日志文件。日志进程会对所复制的文件进行重命名，并截断原始文件，使其大小变为 0。日志记录进程可以继续将日志写入原始文件。

安装或升级 vRealize Log Insight 代理

3

您可以在 Windows 或 Linux 计算机上安装或升级 vRealize Log Insight 代理，包括具有第三方日志管理系统的计算机。

代理收集事件并将其转发到 vRealize Log Insight 服务器。在安装期间，您可以指定服务器、端口和协议设置的参数，或者选择保留默认设置。

您可以通过在安装时使用的相同方法升级代理，也可以使用自动升级。在部署新的 vRealize Log Insight 版本时，自动升级功能将升级传播到代理。有关详细信息，请参见 [vRealize Log Insight 代理的自动更新](#)。升级不适用于 Linux 二进制软件包。

硬件支持

要安装并运行 vRealize Log Insight 代理，您的硬件必须支持主机/计算机所需的最低参数以支持 x86 和 x86_64 架构以及 MMX、SSE、SSE2 和 SSE3 指令集。

平台支持

操作系统	处理器架构
Windows 7、Windows 8、Windows 8.1 和 Windows 10	x86_64、x86_32
Windows Server 2008、Windows Server 2008 R2	x86_64、x86_32
Windows Server 2012、Windows Server 2012 R2、Windows Server 2016 和 Windows Server 2019	x86_64
RHEL 5、RHEL 6、RHEL 7 和 RHEL 8	x86_64、x86_32
SuSE Enterprise Linux (SLES) 11 SP3 和 SLES 12 SP1	x86_64
Ubuntu 14.04 LTS、Ubuntu 16.04 LTS 和 Ubuntu 18.04	x86_64
VMware Photon 版本 1 修订版本 2、版本 2 和版本 3	x86_64

Linux 说明

如果为没有要使用的 root 特权的用户实施 Log Insight Linux Agent 默认安装，默认配置可能会导致数据收集出现问题。代理不记录以下警告：通道订阅失败，并且集合中的文件没有读取权限。将在日志中反复添加无法访问日志文件 ... 将稍后重试 (Inaccessible log file ... will try later) 消息。您可以注释掉导致问题出现的默认配置，或更改用户权限。

如果您使用 rpm 或 DEB 软件包安装 Linux 代理，将在软件包安装过程中安装名为 liagentd 的 init.d 脚本。bin 软件包添加该脚本，但不会注册该脚本。您可以手动注册脚本。

您可以运行 (/sbin/) 服务 liagentd status 命令以验证安装是否成功。

本章讨论了以下主题：

- [下载代理安装文件](#)
- [安装 Windows 代理](#)
- [安装或更新 vRealize Log InsightLinux 代理 RPM 软件包](#)
- [安装或更新 vRealize Log InsightLinux 代理 DEB 软件包](#)
- [自定义 Debian Linux 的代理安装](#)
- [安装 Log Insight Linux Agent 二进制软件包](#)
- [Linux 上用于安装 vRealize Log Insight 代理的命令行选项](#)
- [vRealize Log Insight 代理的自动更新](#)

下载代理安装文件

设置 vRealize Log Insight 代理的第一步是下载适用于您的平台的代理安装软件包。

从 vRealize Log Insight 服务器代理页面中下载的所有软件包在软件包名称后面附加了目标主机名。在初始安装 MSI、RPM 和 DEB 代理的过程中将应用 server.hostname。如果主机名在配置文件中已存在，或者按主机名参数运行软件包，则会忽略嵌入的服务器主机名。

步骤

- 1 导航到 vRealize Log Insight Web 用户界面的[管理](#)选项卡。
- 2 在“管理”部分中，单击[代理](#)。
- 3 滚动到屏幕底部并单击[下载 Log Insight 代理](#)，
- 4 从弹出菜单中选择一个安装软件包，然后单击[保存](#)以下载该软件包。

选项	描述
Windows MSI	适用于 Windows 平台（32 位/64 位）的安装软件包
Linux RPM	适用于 Linux Red Hat、openSUSE（32 位/64 位）或 VMware Photon Platform 的安装软件包

选项	描述
Linux DEB	适用于 Linux Debian 平台（32 位/64 位）的安装软件包
Linux BIN	适用于 Linux（32 位/64 位）的自安装软件包。不需要使用软件包管理系统。

后续步骤

使用已下载的文件部署 vRealize Log Insight 代理。

安装 Windows 代理

您可以通过安装向导或命令行在单个计算机上安装代理，也可以使用脚本部署多个代理实例。

升级 Windows 代理

您可以使用安装时所用的任何方法应用升级文件，以此来升级 Windows 代理。也可以选择使用自动升级功能在后台升级代理。

使用安装向导安装或更新 vRealize Log Insight Windows 代理

您可以使用安装向导在单个计算机上安装或升级 Windows 代理。

前提条件

- 验证您是否拥有 vRealize Log Insight Windows 代理 .msi 文件的副本。请参见[下载代理安装文件](#)。
- 验证是否已具有在 Windows 计算机上执行安装和启动服务的权限。

步骤

- 1 登录到要在其中安装 vRealize Log Insight Windows 代理的 Windows 计算机。
- 2 转到包含 vRealize Log Insight Windows 代理 .msi 文件的目录。
- 3 双击 vRealize Log Insight Windows 代理 .msi 文件，接受许可协议的条款，然后单击**下一步**。
- 4 输入 vRealize Log Insight 服务器的 IP 地址或主机名，然后单击**安装**。
该向导将 vRealize Log Insight Windows 代理作为使用本地系统服务帐户运行的自动 Windows 服务进行安装或更新。
- 5 单击**完成**。

后续步骤

通过编辑 `liagent.ini` 文件配置 vRealize Log Insight Windows 代理。请参见[配置 Log Insight Windows Agent](#)。

从命令行中安装或更新 vRealize Log Insight Windows 代理

您可以从命令行中安装或更新 Windows 代理。

您可以使用默认服务帐户或指定一个服务帐户，并使用命令行参数指定服务器、端口和协议信息。对于 MSI 命令行选项，请参见 Microsoft Developer Network (MSDN) 库网站，并搜索 MSI 命令行选项。

前提条件

- 验证您是否拥有 vRealize Log Insight Windows 代理 .msi 文件的副本。请参见[下载代理安装文件](#)。
- 验证是否已具有在 Windows 计算机上执行安装和启动服务的权限。
- 如果您使用静默安装选项 /quiet 或 /qn，请确认您以管理员身份运行安装。如果您不是管理员但运行静默安装，安装将不会显示管理员特权提示并将失败。使用日志记录选项和参数 /l xv* *file_name* 执行诊断。

步骤

- 1 登录到要安装或更新 vRealize Log Insight Windows 代理的 Windows 计算机。
- 2 打开**命令提示符**窗口。
- 3 转到包含 vRealize Log Insight Windows 代理 .msi 文件的目录。
- 4 使用命令形式如下的默认值进行安装或更新。将 *version-build_number* 替换为您的版本和内部版本号。

/quiet 选项以静默方式运行命令，/l xv 选项会在当前目录中创建一个日志文件。

```
Drive:\path-to-msi_file>VMware-Log-Insight-Agent-version-build_number.msi /
quiet /l xv* li_install.log
```

- 5 （可选）指定要在其下运行 vRealize Log Insight Windows 代理服务的用户服务帐户。

```
Drive:\path-to-msi_file>VMware-Log-Insight-Agent-*.msi SERVICEACCOUNT=domain\user
SERVICEPASSWORD=user_password
```

注 将为 SERVICEACCOUNT 参数所提供的帐户分配**作为服务登录**权限，以及对 %ProgramData%\VMware\Log Insight Agent 目录的完整写访问权限。如果所提供的帐户不存在，将会进行创建。用户名不能超过 20 个字符。如果未指定 SERVICEACCOUNT 参数，vRealize Log Insight Windows 代理服务将在 LocalSystem 服务帐户下安装或更新。

6 （可选） 您可以根据需要指定以下命令行选项的值。

选项	描述
SERVERHOST=hostname	vRealize Log Insight 虚拟设备的 IP 地址或主机名。
SERVERPROTO=protocol	代理向 vRealize Log Insight 服务器发送事件所使用的协议。可能的值为 cfapi 和 syslog。 默认值为 cfapi。
SERVERPORT=portnumber	代理向 vRealize Log Insight 服务器或第三方服务器发送事件所使用的通信端口。 默认情况下，代理根据为 SSL 和协议设置的选项使用相应的端口。请参见下面列表中提供的默认端口值。仅当端口选项与以下默认值不同时，您才需要指定端口选项。 <ul style="list-style-type: none"> ■ 启用了 SSL 的 cfapi: 9543 ■ 禁用了 SSL 的 cfapi: 9000 ■ 启用了 SSL 的 syslog: 6514 ■ 禁用了 SSL 的 syslog: 514
SERVICEACCOUNT=account-name	运行 Log Insight Windows Agent 服务的用户服务帐户。 注 SERVICEACCOUNT 参数中提供的帐户必须具有 作为服务登录 特权以及对 %ProgramData%\VMware\Log Insight Agent 目录的写访问权限，以便安装程序正确运行。如果未指定 SERVICEACCOUNT 参数，vRealize Log Insight Windows 代理服务将安装在 LocalSystem 服务帐户下。
SERVICEPASSWORD=password	用户服务帐户的密码。
AUTOUPDATE={yes no}	启用或禁用代理自动更新。您还必须从 vRealize Log Insight 服务器中启用自动更新以完全启用自动更新。默认值为 yes。
LIAGENT_SSL={yes no}	启用安全连接。如果启用了 SSL，代理会使用 TLS 1.2 协议与服务器进行通信。默认值为 yes。

结果

该命令将 vRealize Log InsightWindows 代理作为 Windows 服务进行安装或更新。当您启动 Windows 计算机时，vRealize Log InsightWindows 代理服务将启动。

后续步骤

验证您设置的命令行参数是否在 liagent.ini 文件中正确应用。请参见配置 [Log Insight Windows Agent](#)。

将 Log Insight Windows Agent 部署到多台计算机

可以在 Windows 域中的目标计算机上批量部署 Log Insight Windows Agent。

步骤

1 创建转换文件以部署多个 vRealize Log InsightWindows 代理

在部署多个代理的过程中，您必须创建一个转换文件来指定部署的配置参数。.mst 转换文件会在您安装代理时应用于 .msi 文件。参数包括代理的目标服务器，以及用于安装和启动 Log Insight 代理服务的通信协议、端口和用户帐户。

2 部署 vRealize Log Insight Windows 代理的多个实例

您可以在 Windows 域中的目标计算机上部署 vRealize Log Insight Windows 代理的多个实例。

创建转换文件以部署多个 vRealize Log Insight Windows 代理

在部署多个代理的过程中，您必须创建一个转换文件来指定部署的配置参数。.mst 转换文件会在您安装代理时应用于 .msi 文件。参数包括代理的目标服务器，以及用于安装和启动 Log Insight 代理服务的通信协议、端口和用户帐户。

参数包括代理的目标服务器，以及用于安装和启动 Log Insight 代理服务的通信协议、端口和用户帐户。

前提条件

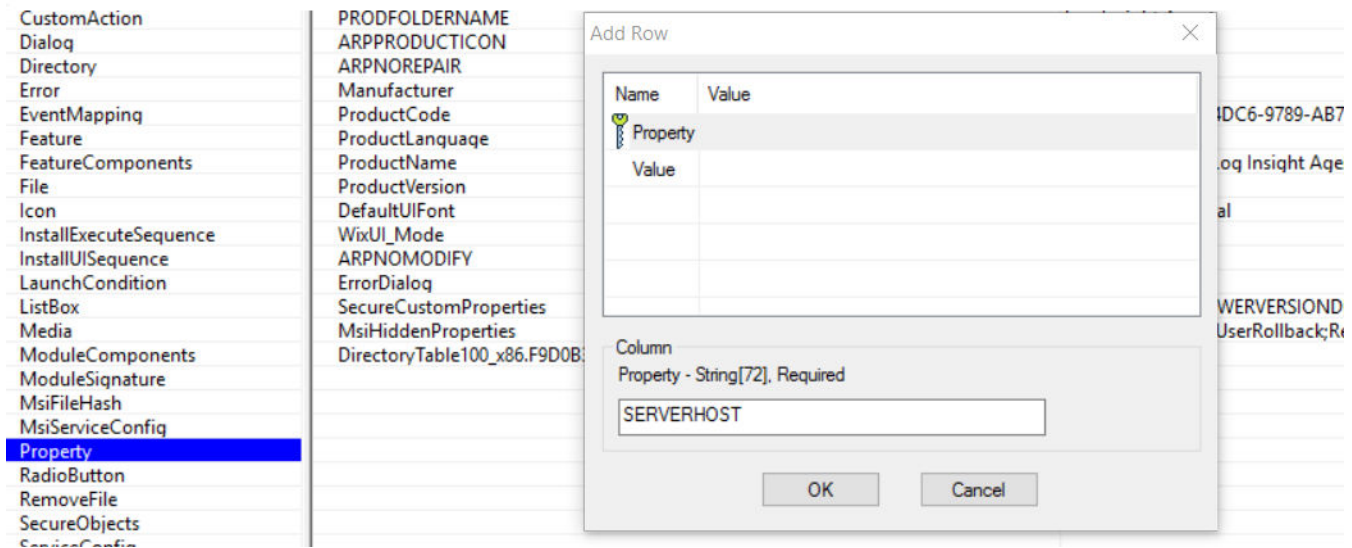
- 验证您是否拥有 vRealize Log Insight Windows .msi 文件的副本。请参见[下载代理安装文件](#)。
- 下载和安装 Orca 数据库编辑器。请参见 <http://support.microsoft.com/kb/255905>。

步骤

- 1 在 Orca 编辑器中打开 vRealize Log Insight Windows 代理 .msi 文件，选择**转换 > 新建转换**。

2 编辑“属性”表，添加自定义安装或升级所需的参数和值。

图 3-1. 属性表



- 单击**属性**。
- 从**表**下拉菜单中，选择**添加行**。
- 在“添加行”对话框中输入一个属性名称和值。

参数如下表中所示。

参数	描述
SERVERHOST	vRealize Log Insight 虚拟设备的 IP 地址或主机名。 默认值为 loginsight 。
SERVERPROTO	代理向 vRealize Log Insight 服务器发送事件所使用的协议。可能的值为 cfapi 和 syslog 。 默认值为 cfapi 。
SERVERPORT	代理向 vRealize Log Insight 服务器或第三方服务器发送事件所使用的通信端口。 默认情况下，代理根据为 SSL 和协议设置的选项使用相应的端口。请参见下面列表中提供的默认端口值。仅当端口选项与以下默认值不同时，您才需要指定端口选项。 <ul style="list-style-type: none"> ■ 启用了 SSL 的 cfapi: 9543 ■ 禁用了 SSL 的 cfapi: 9000 ■ 启用了 SSL 的 syslog: 6514 ■ 禁用了 SSL 的 syslog: 514
SERVICEACCOUNT	运行 Log Insight Windows Agent 服务的用户服务帐户。 注 SERVICEACCOUNT 参数中提供的帐户必须具有 作为服务登录 特权以及对 %ProgramData%\VMware\Log Insight Agent 目录的写访问权限，以便安装程序正确运行。如果未指定 SERVICEACCOUNT 参数，vRealize Log Insight Windows 代理服务将安装在 LocalSystem 服务帐户下。

参数	描述
SERVICEPASSWORD	用户服务帐户的密码。
AUTOUPDATE	启用或禁用代理自动更新。您还必须从 vRealize Log Insight 服务器中启用自动更新以完全启用自动更新。默认值为 yes 。
LIAGENT_SSL	启用安全连接。如果启用了 SSL，代理会使用 TLS 1.2 协议与服务器进行通信。默认值为 yes 。

3 选择**转换 > 生成转换**，然后保存 .mst 转换文件。

后续步骤

使用 .msi 和 .mst 文件部署 vRealize Log Insight Windows 代理。

部署 vRealize Log Insight Windows 代理的多个实例

您可以在 Windows 域中的目标计算机上部署 vRealize Log Insight Windows 代理的多个实例。

有关为何您需要重新引导两次客户机的详细信息，请参见 <http://support.microsoft.com/kb/305293>。

前提条件

- 确认您在域控制器上具有管理员帐户或拥有管理权限的帐户。
- 验证您是否拥有 vRealize Log Insight Windows 代理 .msi 文件的副本。请参见[下载代理安装文件](#)。
- 熟悉 <http://support.microsoft.com/kb/887405> 和 <http://support.microsoft.com/kb/816102> 中描述的过程。

步骤

- 1 以管理员身份或具有管理权限的用户身份登录域控制器。
- 2 创建一个分发点，并将 vRealize Log Insight Windows 代理 .msi 文件复制到该分发点。
- 3 打开组策略管理控制台并创建一个组策略对象以部署 vRealize Log Insight Windows 代理 .msi 文件。
- 4 编辑用于软件部署的组策略对象，并分配软件包。
- 5 （可选）如果您在部署之前生成 .mst 文件，请在 **GPO 属性**窗口的**修改**选项卡上选择 .mst 配置文件，并使用“高级”方法编辑组策略对象以部署 .msi 软件包。
- 6 （可选）升级 vRealize Log Insight Windows 代理。
 - a 将升级 .msi 文件复制到分发点。
 - b 单击组策略对象**属性**窗口中的**升级**选项卡。
 - c 在此软件包将升级的“软件包”部分中，添加最初安装的 .msi 文件版本。
- 7 将 vRealize Log Insight Windows 代理部署到包含域用户的特定安全组。
- 8 关闭域控制器上的所有组策略管理控制台和组策略管理编辑器窗口并重新启动客户机。
如果启用快速登录优化，请重新引导客户机两次。

9 确认已在客户机上将 vRealize Log Insight Windows 代理作为一项本地服务安装。

如果将 SERVICEACCOUNT 和 SERVICEPASSWORD 参数配置为使用 .mst 文件部署 vRealize Log Insight Windows 代理的多个实例，请确认 vRealize Log Insight Windows 代理已安装在您指定的用户帐户下的客户机上。

后续步骤

如果多个 vRealize Log Insight Windows 代理实例失败，请参见 [Log Insight Windows Agent 批量部署失败](#)。

安装或更新 vRealize Log InsightLinux 代理 RPM 软件包

您可以作为 root 或非 root 用户安装或更新 vRealize Log InsightLinux 代理，并且可以在安装期间设置配置参数。安装后，您可以验证安装的版本。

前提条件

- 阅读 [Linux 上用于安装 vRealize Log Insight 代理的命令行选项](#) 以了解安装默认设置以及如何更改这些设置。
- 以 **root** 用户身份登录，或使用 sudo 运行控制台命令。
- vRealize Log InsightLinux 代理需要访问 syslog 和网络服务才能正常工作。安装 vRealize Log InsightLinux 代理并以运行级别 3 和 5 运行。如果希望 vRealize Log InsightLinux 代理在其他运行级别下工作，请相应地配置系统。

步骤

1 您可以从控制台中安装或升级代理。

- 要使用默认配置设置安装 vRealize Log InsightLinux 代理，请打开控制台并运行以下命令。

```
rpm -i VMware-Log-Insight-Agent-<version-and-build-number>.rpm
```

- 要升级代理而不更改当前配置设置，请打开控制台并运行以下命令。

```
rpm -Uvh VMware-Log-Insight-Agent-<version-and-build-number>.rpm
```

2 （可选）在更新期间，您可以覆盖默认安装配置值或当前配置值。您可以将一些选项指定为安装或升级命令的一部分以执行该操作。

```
sudo <OPTION=value> rpm -i <version-and-build-number>.rpm
```

3 （可选）运行以下命令以检查安装的版本。

```
rpm -qa | grep Log-Insight-Agent
```

示例：Linux 代理安装和更新示例

- 以下命令安装适用于 RPM Linux 发行版的 vRealize Log Insight 代理。该命令在单独的服务器上安装代理，分配非默认端口号以及创建 vRealize Log Insight 代理用户。

```
sudo SERVERHOST=myagentserver SERVERPORT=1234 LIAGENTUSER=liagent rpm -i VMware-Log-Insight-Agent-44.1234.rpm
```

- 以下命令使用给定的 rpm 文件更新代理。当前代理配置保持不变。

```
rpm -Uvh VMware-Log-Insight-Agent-44.1234.rpm
```

安装或更新 vRealize Log InsightLinux 代理 DEB 软件包

您可以从命令行或通过 debconf 数据库安装或更新 vRealize Log Insight Linux 代理 DEB (Debian) 软件包。安装后，您可以验证安装的版本。

前提条件

- 请阅读 [Linux 上用于安装 vRealize Log Insight 代理的命令行选项](#)，了解安装默认设置以及如何对这些设置进行更改。
- 以 **root** 用户身份登录，或使用 `sudo` 运行控制台命令。
- 确认 vRealize Log InsightLinux 代理具有正常工作所需的 `syslog` 和网络服务访问权限。默认情况下，vRealize Log InsightLinux 代理在运行级别 2、3、4 和 5 上运行，并在运行级别 0、1 和 6 上停止。
- 有关详细信息和示例，请参见[自定义 Debian Linux 的代理安装](#)。

步骤

- 1 要安装或更新 vRealize Log Insight Linux 代理，请打开控制台并运行 `dpkg -i package_name` 命令。

`package_name` 由前缀 **vmware-log-insight-agent-** 和您下载版本的内部版本号组成。

以下命令形式将使用默认值安装此软件包。

```
dpkg -i vmware-log-insight-agent-VERSION-BUILD_NUMBER_all.deb
```

- 2 （可选）运行以下命令以检查安装的版本：

```
dpkg -l | grep -i vmware-log-insight-agent
```

示例

- 从命令行配置连接协议。

要覆盖默认连接协议，请使用 `SERVERPROTO` 变量，如以下示例中所示：

```
sudo SERVERPROTO=syslog dpkg -i vmware-log-insight-agent-<version-and-build-number>_all.deb
```

自定义 Debian Linux 的代理安装

您可以通过使用命令选项来覆盖当前的安装配置值，或通过配置 `debconf` 数据库来自定义安装。

从命令行自定义

要从命令行配置安装，请使用以下形式的命令：

```
sudo <OPTION=value> dpkg -i vmware-log-insight-agent-<version-and-build-number>_all.deb
```

有关完整的选项列表，请参见 [Linux 上用于安装 vRealize Log Insight 代理的命令行选项](#)。

以下示例显示了一些从命令行完成的典型配置。

- 指定一个目标 vRealize Log Insight 服务器。
- 要在安装期间设置目标，请运行 `sudo` 命令，并将 `hostname` 替换为 vRealize Log Insight 服务器的 IP 地址或主机名，如以下示例中所示：

```
sudo SERVERHOST=hostname dpkg -i vmware-log-insight-agent-<version-and-build-number>_all.deb
```

除非您在安装期间启用了 `--force-confold` 标记，否则只要您更新到新版本，系统就会提示您保留或替换 `liagent.ini` 配置文件。将显示以下系统消息：

```
Configuration file `/var/lib/loginsight-agent/liagent.ini'
==> Modified (by you or by a script) since installation.
==> Package distributor has shipped an updated version.
What would you like to do about it ? Your options are:
  Y or I : install the package maintainer's version
  N or O : keep your currently-installed version
  D      : show the differences between the versions
  Z      : start a shell to examine the situation
The default action is to keep your current version.
*** liagent.ini (Y/I/N/O/D/Z) [default=N] ?
```

要保留现有配置，请使用 `[default=N]`。仍会应用从命令行中传递的其他参数。

- 配置连接协议。

要覆盖默认连接协议，请使用 `SERVERPROTO` 变量，如以下示例中所示：

```
sudo SERVERPROTO=syslog dpkg -i vmware-log-insight-agent-<version-and-build-number>_all.deb
```

■ 配置连接端口。

要覆盖默认连接端口，请为安装程序提供 `SERVERPORT` 变量值，如以下示例中所示：

```
sudo SERVERPORT=1234 dpkg -i vmware-log-insight-agent-<version-and-build-number>_all.deb
```

■ 以非 root 用户身份运行代理。

要以非 **root** 用户身份运行 vRealize Log Insight Linux 代理，请运行 `sudo` 命令。

```
sudo LIAGENTUSER=liagent dpkg -i vmware-log-insight-agent-<version-build-number>_all.deb
```

如果指定的用户不存在，vRealize Log InsightLinux 代理会在安装期间创建用户帐户。创建的帐户在卸载后不会被删除。如果您使用 `LIAGENTUSER=non_root_user` 参数安装 Linux 代理并尝试使用 `LIAGENTUSER=non_root_user2` 参数进行升级，将会出现冲突，并且会显示警告，因为 `non_root_user2` 用户没有 `non_root_user` 用户的权限。

debconf 数据库的 DEB 软件包自定义选项

还可以通过 debconf 数据库来配置代理 DEB 软件包。下表显示了支持的 debconf 选项和相应的 vRealize Log Insight 代理 DEB 安装程序选项：

命令行选项	Debconf 选项	描述
<code>SERVERHOST=hostname</code>	<code>vmware-log-insight-agent/serverhost</code>	vRealize Log Insight 虚拟设备的 IP 地址或主机名。 默认值为 loginsight 。
<code>SERVERPROTO={cfapi syslog}</code>	<code>vmware-log-insight-agent/serverproto</code>	代理向 vRealize Log Insight 服务器发送事件所使用的协议。可能的值为 <code>cfapi</code> 和 <code>syslog</code> 。 默认值为 <code>cfapi</code> 。
<code>SERVERPORT=portnumber</code>	<code>vmware-log-insight-agent/serverport</code>	代理向 vRealize Log Insight 服务器或第三方服务器发送事件所使用的通信端口。默认情况下，代理根据为 SSL 和协议设置的选项使用相应的端口。请参见下面列表中提供的默认端口值。仅当端口选项与以下默认值不同时，您才需要指定端口选项。 <ul style="list-style-type: none"> ■ 启用了 SSL 的 <code>cfapi</code>：9543 ■ 禁用了 SSL 的 <code>cfapi</code>：9000 ■ 启用了 SSL 的 <code>syslog</code>：6514 ■ 禁用了 SSL 的 <code>syslog</code>：514
<code>LIAGENT_INITSYSTEM={init systemd}</code>	<code>log-insight-agent/init_system</code>	在安装期间，代理会自动检测安装该代理的计算机的初始化系统类型。您可以使用此选项指定系统类型值，以覆盖此行为。支持的初始化系统有以下两种类型： <code>init</code> 和 <code>systemd</code> 。
<code>LIAGENT_AUTOUPDATE={yes no}</code>	<code>vmware-log-insight-agent/auto_update</code>	启用或禁用代理自动更新。您还必须从 vRealize Log Insight 服务器中启用自动更新以完全启用自动更新。默认值为 yes 。 Linux BIN 软件包不支持自动更新。

命令行选项	Debconf 选项	描述
LI_AGENT_RUNSERVICES	vmware-log-insight-agent/init_system	默认情况下，在安装后，会立即启动 liagentd（代理）和 liupdaterd（更新程序）服务。可以将 LIAGENT_RUNSERVICES debconf 参数设置为 no 以禁止它们启动。默认值为 yes 。只接受值 yes 和 no ；不支持值 1 或 0 。
LIAGENT_SSL	vmware-log-insight-agent/ssl	C
LIAGENTUSER= <i>user-account-name</i>	vmware-log-insight-agent/liagentuser	<p>指定在其下运行代理的帐户。如果用户不存在，安装程序会创建该用户以作为常规用户。如果指定的用户帐户不存在，vRealize Log Insight Linux 代理会在安装期间创建该用户帐户。创建的帐户在卸载后不会被删除。</p> <p>默认情况下，代理在安装后会以 root 用户身份运行。</p> <p>如果您使用 LIAGENTUSER=<i>non_root_user</i> 参数安装代理并尝试使用 LIAGENTUSER=<i>non_root_user2</i> 进行升级，将会出现冲突，并且会显示警告，因为 <i>non_root_user2</i> 用户没有 <i>non_root_user</i> 用户的权限。</p> <p>创建的用户在卸载期间不会被移除。可以通过手动方式将其移除。此参数仅适用于代理服务。更新程序服务将始终以 root 用户身份运行。</p>

安装 Log Insight Linux Agent 二进制软件包

安装二进制软件包涉及将 .bin 文件更改为可执行文件和安装代理。

不正式支持升级 .bin 软件包。如果您使用 .bin 软件包安装现有的 Log Insight Linux Agent，请对位于 /var/lib/loginsight-agent 目录的 liagent.ini 文件创建一个备份副本以保存本地配置。创建备份副本后，手动卸载 Log Insight Linux Agent。请参见[手动卸载 Log Insight Linux 代理 bin 软件包](#)。

如果您使用 .bin 软件包安装 Linux 代理，名为 liagentd 的 init.d 脚本将在软件包安装过程中进行安装，但软件包不会注册该脚本。您可以手动注册脚本。

您可以通过运行 (/sbin/)service liagentd status 命令，确认安装是否成功。

前提条件

- 下载 Log Insight Linux Agent .bin 软件包并将其复制到目标 Linux 计算机。
- 确认 Log Insight Linux Agent 具有 syslog 和网络服务的访问权限。
- 阅读有关默认配置值以及如何在安装期间更改这些值的内容。请参见[Linux 上用于安装 vRealize Log Insight 代理的命令行选项](#)。

步骤

- 1 打开控制台并运行 chmod 命令，将 .bin 文件更改为可执行文件。

使用相应的版本替换 *filename-version*。

```
chmod +x filename-version.bin
```

- 2 从命令提示符中，运行 `./filename-version.bin` 命令以安装代理。

使用相应的版本替换 *filename-version*。

```
./filename-version.bin
```

- 3 （可选）要在安装期间设置目标 vRealize Log Insight 服务器，请运行 `sudo SERVERHOST=hostname ./filename-version.bin` 命令。

将 *hostname* 替换为 vRealize Log Insight 服务器的 IP 地址或主机名。

```
sudo SERVERHOST=hostname ./filename-version.bin
```

- 4 （可选）要覆盖默认连接协议，请使用 `SERVERPROTO` 变量，如以下示例中所示：

```
sudo SERVERPROTO=syslog ./filename-version.htm
```

- 5 （可选）要覆盖默认连接端口，请为安装程序提供 `SERVERPORT` 变量值，如以下示例中所示：

```
sudo SERVERPORT=1234 ./filename-version.htm
```

- 6 （可选）要以非 **root** 用户身份运行 Log Insight Linux Agent，请运行 `sudo` 命令。

```
sudo LIAGENTUSER=liagent ./filename-version.bin
```

如果指定的用户不存在，Log Insight Linux Agent 会在安装期间创建用户帐户。创建的帐户在卸载后不会被删除。如果您使用 `LIAGENTUSER=non_root_user` 参数安装 Log Insight Linux Agent 并尝试使用 `LIAGENTUSER=non_root_user2` 参数进行升级，将会出现冲突并显示警告，因为 *non_root_user2* 用户没有 *non_root_user* 用户的权限。

Linux 上用于安装 vRealize Log Insight 代理的命令行选项

从命令行安装 vRealize Log Insight 代理时，您可以包括相应的选项以在安装期间配置您的部署。这些选项对应于 `liagent.ini` 文件中的设置。

在安装期间可以使用以下选项来配置在 Linux 系统上运行的 vRealize Log Insight 代理。

选项	描述
<code>SERVERHOST=hostname</code>	vRealize Log Insight 虚拟设备的 IP 地址或主机名。 默认值为 loginsight 。
<code>SERVERPROTO={cfapi syslog}</code>	代理向 vRealize Log Insight 服务器发送事件所使用的协议。 可能的值为 <code>cfapi</code> 和 <code>syslog</code> 。 默认值为 <code>cfapi</code> 。

选项	描述
<code>SERVERPORT=portnumber</code>	<p>代理向 vRealize Log Insight 服务器或第三方服务器发送事件所使用的通信端口。默认情况下，代理根据为 SSL 和协议设置的选项使用相应的端口。请参见下面列表中提供的默认端口值。仅当端口选项与以下默认值不同时，您才需要指定端口选项。</p> <ul style="list-style-type: none"> ■ 启用了 SSL 的 cfapi: 9543 ■ 禁用了 SSL 的 cfapi: 9000 ■ 启用了 SSL 的 syslog: 6514 ■ 禁用了 SSL 的 syslog: 514
<code>LIAGENT_INITSYSTEM={init systemd}</code>	<p>在安装期间，代理会自动检测安装该代理的计算机的初始化系统类型。您可以使用此选项指定系统类型值，以覆盖此行为。支持的初始化系统有以下两种类型：init 和 systemd。</p>
<code>LIAGENT_AUTOUPDATE={yes no}</code>	<p>启用或禁用代理自动更新。您还必须从 vRealize Log Insight 服务器中启用自动更新以完全启用自动更新。默认值为 yes。</p> <p>Linux BIN 软件包不支持自动更新。</p>
<code>LIAGENT_SSL={yes no}</code>	<p>启用安全连接。如果启用了 SSL，代理会使用 TLS 1.2 协议与服务器进行通信。默认值为 yes。</p>
<code>LIAGENTUSER=user-account-name</code>	<p>指定在其下运行代理的帐户。如果用户不存在，安装程序会创建该用户以作为常规用户。如果指定的用户帐户不存在，vRealize Log Insight Linux 代理会在安装期间创建该用户帐户。创建的帐户在卸载后不会被删除。</p> <p>默认情况下，代理在安装后会以 root 用户身份运行。</p> <p>如果您使用 <code>LIAGENTUSER=non_root_user</code> 参数进行安装，并尝试使用 <code>LIAGENTUSER=non_root_user2</code> 进行升级，则会出现冲突，并且会显示警告，因为 <code>non_root_user2</code> 用户没有 <code>non_root_user</code> 用户的权限。</p> <p>创建的用户在卸载期间不会被移除。可以通过手动方式将其移除。此参数仅适用于代理服务。更新程序服务将始终以 root 用户身份运行。</p>

vRealize Log Insight 代理的自动更新

通过使用 vRealize Log Insight 代理的自动更新功能，活动代理可以根据 vRealize Log Insight 服务器中的代理安装软件包检查、下载并自动安装更新。

您可以从服务器中为所有代理启用自动更新，也可以从客户端中为各个代理实例启用自动更新。代理必须处于活动状态，并且版本为 4.3 或更高版本。

Linux BIN 软件包不支持自动更新。

为各个代理禁用或启用自动更新

您可以通过编辑各个代理的客户端配置文件来为它们启用或禁用自动更新。

默认情况下，将从代理的客户端中启用自动更新。

前提条件

代理必须为 4.3 或更高版本。

步骤

1 在编辑器中打开本地 liagent.ini 文件。

2 找到 [update] 部分。

它类似于以下示例。

```
[update]
; Do not change this parameter
package_type=msi
; Enable automatic update of the agent. If enabled:
; the agent will silently check for updates from the server and
; if available will automatically download and apply the update.
; auto_update=yes
```

3 要禁用自动更新，请取消注释 auto_update=yes，并将其更改为 auto_update=no。

注 默认情况下，代理自动更新处于启用状态。因此，auto_update 的默认值为 “yes”，即使已注释也是如此。

4 保存并关闭 liagent.ini 文件。

配置 vRealize Log Insight 代理

4

部署代理后，您可以对其进行配置以便将事件发送到您选择的 vRealize Log Insight 服务器，指定通信协议和设置其他参数。

使用这些说明，根据您的要求配置代理。

- **配置 Log Insight Windows Agent**

安装 Log Insight Windows Agent 后，您可以对其进行配置。编辑 `liagent.ini` 文件可配置要将事件发送到 vRealize Log Insight 的 Log Insight Windows Agent、设置通信协议和端口、添加 Windows 事件通道，以及配置平面文件日志收集。该文件位于 `%ProgramData%\VMware\Log Insight Agent` 目录中。

- **配置 Log Insight Linux Agent**

安装 Log Insight Linux Agent 后，您可以对其进行配置。

- **筛选来自 vRealize Log Insight 代理的事件**

您可以使用本地 `liagent.ini` 文件中 `[server|<dest_id>]` 部分的筛选器选项，指定代理发送到目标的信息。

- **从 vRealize Log Insight 代理转发信息**

您可以将代理收集的事件转发到最多三个目标。目标可以包括 vRealize Log Insight 服务器或转发器，或者第三方日志管理解决方案。

- **vRealize Log Insight 代理的集中式配置**

您可以配置多个 vRealize Log Insight 代理。

- **使用公用值进行代理配置**

您可以使用适用于 Windows 或 Linux 代理的每个代理配置部分的公用参数值覆盖代理配置文件的默认值。

- **分析日志**

代理端日志分析程序从原始日志中提取结构化数据，然后传送到 vRealize Log Insight 服务器。使用日志分析程序，vRealize Log Insight 可以分析日志，从日志中提取信息，并在服务器上显示结果。对于 Windows 和 Linux vRealize Log Insight 代理，均可配置日志分析程序。

配置 Log Insight Windows Agent

安装 Log Insight Windows Agent 后，您可以对其进行配置。编辑 `liagent.ini` 文件可配置要将事件发送到 vRealize Log Insight 的 Log Insight Windows Agent、设置通信协议和端口、添加 Windows 事件通道，以及配置平面文件日志收集。该文件位于 `%ProgramData%\VMware\Log Insight Agent` 目录中。

Log Insight Windows Agent 的默认配置

安装完成后，`liagent.ini` 文件中包含 Log Insight Windows Agent 的预配置默认设置。

Log Insight Windows Agent `liagent.ini` 默认配置

如果您使用非 ASCII 的名称和值，请将配置保存为 UTF-8。

如果您使用集中式配置，最终配置是此文件与服务器中的设置合并，从而构成 `liagent-effective.ini` 文件。

您可能会发现从服务器的代理页面配置设置更高效。

```
; Client-side configuration of VMware Log Insight Agent.
; See liagent-effective.ini for the actual configuration used by VMware Log Insight Agent.

[server]
; Log Insight server hostname or ip address
; If omitted the default value is LOGINSIGHT
;hostname=LOGINSIGHT

;Enables or disables centralized configuration from the vRealize Log Insight server.
;When enabled, agent configuration changes made to the liagent.ini file on the server
;are joined with the settings in this file. to this agent. Accepted values are yes or no and
0 or 1.
;The default is yes.
;
;central_config=yes
;

; Set protocol to use:
; cfapi - Log Insight REST API
; syslog - Syslog protocol
; If omitted the default value is cfapi
;
;proto=cfapi

; Log Insight server port to connect to. If omitted the default value is:
; for syslog: 514
; for cfapi without ssl: 9000
; for cfapi with ssl: 9543
;port=9000

;ssl - enable/disable SSL. Applies to cfapi protocol only.
; Possible values are yes or no. If omitted the default value is no.
;ssl=no
```

```

; Time in minutes to force reconnection to the server
; If omitted the default value is 30
;reconnect=30

[storage]
;max_disk_buffer - max disk usage limit (data + logs) in MB:
; 100 - 2000 MB, default 200
;max_disk_buffer=200

[logging]
;debug_level - the level of debug messages to enable:
; 0 - no debug messages
; 1 - trace essential debug messages
; 2 - verbose debug messages (will have negative impact on performance)
;debug_level=0
;
;The interval in minutes to print statistics
;stats_period=15

[update]
; Do not change this parameter
package_type=msi

; Enable automatic update of the agent. If enabled:
; the agent will silently check for updates from the server and
; if available will automatically download and apply the update.
;auto_update=yes

[winlog|Application]
channel=Application
raw_syslog=no

[winlog|Security]
channel=Security

[winlog|System]
channel=System

```

参数	默认值	描述
hostname	LOGINSIGHT	vRealize Log Insight 虚拟设备的 IP 地址或主机名。 默认值为 loginsight 。
central_config	yes	为此代理启用或禁用集中式配置。禁用集中式配置后，代理会忽略由 vRealize Log Insight 服务器提供的配置。接受的值为 yes、no、1 或 0。默认值为 yes。

参数	默认值	描述
proto	cfapi	代理向 vRealize Log Insight 服务器发送事件所使用的协议。可能的值为 cfapi 和 syslog。 默认值为 cfapi。
port	9543、9000、6514 和 514	代理向 vRealize Log Insight 服务器或第三方服务器发送事件所使用的通信端口。 默认情况下，代理根据为 SSL 和协议设置的选项使用相应的端口。请参见下面列表中提供的默认端口值。仅当端口选项与以下默认值不同时，您才需要指定端口选项。 <ul style="list-style-type: none"> ■ 启用了 SSL 的 cfapi: 9543 ■ 禁用了 SSL 的 cfapi: 9000 ■ 启用了 SSL 的 syslog: 6514 ■ 禁用了 SSL 的 syslog: 514
ssl	yes	启用或禁用 SSL。默认值为 yes。 将 ssl 设置为“yes”时，如果未设置端口值，则会自动为端口选取值 9543。
max_disk_buffer	200	由 Log Insight Windows Agent 用于缓冲事件及其日志的最大磁盘空间 (MB)。 达到指定的 max_disk_buffer 时，代理将开始丢弃新的入站事件。
debug_level	0	定义日志详细信息级别。请参见在 Log Insight Agents 中定义日志详细信息级别。
channel	应用程序、安全、系统	默认情况下会注释掉应用程序、安全和系统 Windows 事件日志通道；Log Insight Windows Agent 不会从这些通道收集日志。 请参见从 Windows 事件通道收集事件 。
raw_syslog	no	对于使用 syslog 协议的代理，允许代理收集并发送原始 syslog 事件。默认值为 no，表示收集的事件通过用户指定的 syslog 属性进行转换。启用此选项可在不进行任何 syslog 转换的情况下收集事件。 接受的值为 yes 或 1 以及 no 或 0。

从 Windows 事件通道收集事件

可以将 Windows 事件通道添加到 Log Insight Windows Agent 配置。Log Insight Windows Agent 将收集事件并将其发送到 vRealize Log Insight 服务器。

字段名称受到一定的限制。以下名称是保留名称，因此不能用作字段名称。

- event_type
- hostname

- source
- text

前提条件

登录到已安装 vRealize Log Insight Windows 代理的 Windows 计算机，启动“服务”管理器以验证是否已安装 vRealize Log Insight 代理服务。

步骤

- 1 导航到 vRealize Log Insight Windows 代理的程序数据目录。

```
%ProgramData%\VMware\Log Insight Agent
```

- 2 在任意文本编辑器中打开 liagent.ini 文件。

- 3 添加以下参数，并设置用于您环境的值。

参数	描述
[winlog section_name]	配置部分的唯一名称。
channel	事件通道的完整名称（显示在内置 Windows 应用程序“事件查看器”中的名称）。要复制正确的通道名称，请在事件查看器中右键单击通道，选择 属性 并复制 完整名称 字段的内容。
enabled	用于启用或禁用配置部分的一个可选参数。可能的值为 yes 或 no（不区分大小写）。默认值为 yes。
tags	用于向所收集事件的字段添加自定义标记的可选参数。使用 JSON 表示法定义标记。标记名称可以包含字母、数字和下划线。标记名称只能以字母或下划线开头，并且不能超过 64 个字符。标记名称不区分大小写。例如，如果使用 tags={"tag_name1" : "tag value 1", "Tag_Name1" : "tag value 2"}，则将 Tag_Name1 作为重复项忽略。不能将 event_type 和时间戳用作标记名称。同一个声明中的所有重复项都将被忽略。 如果目标是 syslog 服务器，标记可能会覆盖 APP-NAME 字段。例如，tags={"appname":"VROPS"}。
whitelist, blacklist	显式包括或排除日志事件的可选参数。 注 blacklist 选项仅适用于字段；不可用于块文本。
exclude_fields	（可选）用于从收集排除各个字段的一个参数。可以通过分号分隔的列表的形式提供多个值。例如，exclude_fields=EventId; ProviderName

```
[winlog|section_name]
channel=event_channel_name
enabled=yes_or_no
tags={"tag_name1" : "Tag value 1", "tag_name2" : "tag value 2" }
```

- 4 保存并关闭 liagent.ini 文件。

示例：配置

请参见下面的 [winlog] 配置示例。

```
[winlog|Events_Firewall ]
channel=Microsoft-Windows-Windows Firewall With Advanced Security/Firewall
enabled=no
```

```
[winlog|custom]
channel=Custom
tags={"ChannelDescription": "Events testing channel"}
```

为 Windows 事件通道设置筛选

您可以为 Windows 事件通道设置筛选器以显式包含或排除日志事件。

使用 `whitelist` 和 `blacklist` 参数来评估筛选表达式。筛选表达式是一个包含事件字段和运算符的布尔表达式。

注 `blacklist` 选项仅适用于字段；不可用于块文本。

- `whitelist` 参数仅收集筛选表达式评估为非零的日志事件。如果忽略此参数，则相应的值为隐含的 1。
- `blacklist` 参数将排除筛选表达式评估为非零的日志事件。默认值为 0。

有关 Windows 事件字段和运算符的完整列表，请参见[事件字段和运算符](#)。

前提条件

登录到已安装 vRealize Log Insight Windows 代理的 Windows 计算机，启动“服务”管理器以验证是否已安装 vRealize Log Insight 代理服务。

步骤

- 1 导航到 vRealize Log Insight Windows 代理的程序数据目录。

```
%ProgramData%\VMware\Log Insight Agent
```

- 2 在任意文本编辑器中打开 `liagent.ini` 文件。
- 3 在 [winlog] 部分中添加 `whitelist` 或 `blacklist` 参数。

例如

```
[winlog|unique_section_name]
channel = event_channel_name
blacklist = filter_expression
```

- 4 从 Windows 事件字段和运算符中创建筛选表达式。

例如

```
whitelist = level > WINLOG_LEVEL_SUCCESS and level < WINLOG_LEVEL_INFO
```

5 保存并关闭 liagent.ini 文件。

示例：筛选器配置

您可以将代理配置为仅收集错误事件，例如

```
[winlog|Security-Error]
channel = Security
whitelist = Level == WINLOG_LEVEL_CRITICAL or Level == WINLOG_LEVEL_ERROR
```

您可以将代理配置为从应用程序通道中仅收集 VMware Network 事件，例如

```
[winlog|VMwareNetwork]
channel = Application
whitelist = ProviderName == "VMnetAdapter" or ProviderName == "VMnetBridge" or ProviderName
== "VMnetDHCP"
```

您可以将代理配置为从安全通道中收集除特定事件以外的所有事件，例如

```
[winlog|Security-Verbose]
channel = Security
blacklist = EventID == 4688 or EventID == 5447
```

事件字段和运算符

使用 Windows 事件字段和运算符来构建筛选表达式。

筛选表达式运算符

运算符	描述
==, !=	等于和不等于。与数字字段和字符串字段一起使用。
>=, >, <, <=	大于或等于、大于、小于、小于或等于。仅与数字字段一起使用。
&, , ^, ~	位和、位或、位异或以及补算运算符。仅与数字字段一起使用。
和、或	逻辑和、逻辑或。用于通过合并简单表达式来构建复杂表达式。
否	一元逻辑非运算符。用于对表达式的值取反。
()	在逻辑表达式中使用括号来更改评估顺序。

Windows 事件字段

可以在筛选表达式中使用以下 Windows 事件字段。

字段名称	字段类型
Hostname	字符串
文本	字符串
ProviderName	字符串
EventSourceName	字符串
EventID	数字
EventRecordID	数字

字段名称	字段类型
通道	字符串
UserID	字符串
Level	数字 您可以使用以下预定义常量 <ul style="list-style-type: none"> ■ WINLOG_LEVEL_SUCCESS = 0 ■ WINLOG_LEVEL_CRITICAL = 1 ■ WINLOG_LEVEL_ERROR = 2 ■ WINLOG_LEVEL_WARNING = 3 ■ WINLOG_LEVEL_INFO = 4 ■ WINLOG_LEVEL_VERBOSE = 5
任务	数字
OpCode	数字
关键字	数字 您可以使用以下预定义位掩码 <ul style="list-style-type: none"> ■ WINLOG_KEYWORD_RESPONSETIME = 0x0001000000000000; ■ WINLOG_KEYWORD_WDICONTEXT = 0x0002000000000000; ■ WINLOG_KEYWORD_WDIDIAGNOSTIC = 0x0004000000000000; ■ WINLOG_KEYWORD_SQM = 0x0008000000000000; ■ WINLOG_KEYWORD_AUDITFAILURE = 0x0010000000000000; ■ WINLOG_KEYWORD_AUDITSUCCESS = 0x0020000000000000; ■ WINLOG_KEYWORD_CORRELATIONHINT = 0x0040000000000000; ■ WINLOG_KEYWORD_CLASSIC = 0x0080000000000000;

示例

收集所有严重、错误和警告事件

```
[winlog|app]
channel = Application
whitelist = level > WINLOG_LEVEL_SUCCESS and level < WINLOG_LEVEL_INFO
```

仅收集安全通道中的审核失败事件

```
[winlog|security]
channel = Security
whitelist = Keywords & WINLOG_KEYWORD_AUDITFAILURE
```

从日志文件收集事件

您可以将 vRealize Log Insight Windows 代理配置为从一个或多个日志文件收集事件。

字段名称受到一定的限制。以下名称是保留名称，因此不能用作字段名称。

- event_type
- hostname
- source

- text

您最多可为代理信息指定三个目标，并且可在发送该信息之前对其进行筛选。请参见[从 vRealize Log Insight 代理转发信息](#)。

注

- 如果监控大量文件（如一千个或更多），将导致代理具有较高的资源利用率并影响主机的总体性能。要防止出现这种情况，请配置代理以使用模式和 glob 仅监控所需的文件，或者存档旧日志文件。如果要求监控大量文件，请考虑增加主机参数，例如，CPU 和 RAM。
- 仅当对目录进行加密的用户运行代理时，代理才能从该加密目录收集事件。
- 此代理仅支持静态目录结构。如果已重命名或添加目录，则必须重新启动代理才能开始监控这些目录，前提是配置涵盖这些目录。

前提条件

登录到已安装 vRealize Log InsightWindows 代理的 Windows 计算机，启动“服务”管理器以验证是否已安装 vRealize Log Insight 代理服务。

步骤

- 1 导航到 vRealize Log InsightWindows 代理的程序数据目录。

```
%ProgramData%\VMware\Log Insight Agent
```

- 2 在任意文本编辑器中打开 liagent.ini 文件。

3 找到该文件的 [server|<dest_id>] 部分。添加配置参数并设置环境的值。

```
[filelog|section_name]
directory=path_to_log_directory
include=glob_pattern
...
```

参数	描述
[filelog section_name]	配置部分的唯一名称。
directory=full-path-to-log-file	<p>日志文件目录的完整路径。支持 glob 模式。示例配置：</p> <ul style="list-style-type: none"> ■ 要从 D:\Logs\new_test_logs 目录的所有子目录中收集，请使用 <code>directory=D:\Logs\new_test_logs*</code> ■ 如果子目录具有自己的子目录，请使用以下配置以监控所有子目录： <code>directory=D:\Logs\new_test_logs**</code> <p>注 要限制文件和目录数并避免较高的资源消耗，您不能为第一级或第二级目录定义目录 glob，例如：<code>directory=c:/tmp/*</code> 或 <code>directory=c:\Logs*</code>。目录路径必须至少具有两个级别。</p> <p>您可以定义一个到当前不存在目录的路径，创建该目录及其中的日志文件后，代理即会收集这些日志文件。</p> <p>可以在一个或多个不同配置部分下定义相同的目录，以便多次从同一文件收集日志。利用该过程，可以对相同的事件源应用不同标记和筛选器。</p> <p>注 如果这些部分使用完全相同的配置，则会在服务器端观察到重复的事件。</p>
include=file_name; ...	<p>（可选）要从中收集数据的文件名或文件掩码（glob 模式）。可以通过分号分隔的列表的形式提供值。默认值为 *，表示包括所有文件。该参数区分大小写。</p> <p>可以使用文件掩码（glob 模式）将遵循相同命名约定的文件，以及一个文件下的所有文件分组到一起。例如，包含空格的文件名（例如，vRealize Ops Analytics.log 和 vRealize Ops Collector.log）可以使用 vRealize?Ops?Analytics*.log 或 vRealize*.log 指定。通过使用文件掩码，可以指定 Linux 和 Windows 主机下的代理配置可接受的文件名。</p> <p>默认情况下，.zip 和 .gz 文件会从收集排除。</p> <p>重要事项 如果要收集轮换的日志文件，请使用 include 和 exclude 参数指定同时与主要文件和轮换文件相匹配的 glob 模式。如果 glob 模式仅与主要日志文件相匹配，则 vRealize Log Insight 代理可能会在轮换期间错过事件。vRealize Log Insight 代理会自动确定轮换文件的正确顺序，并将事件以正确的顺序发送到 vRealize Log Insight 服务器。例如，如果您的主要日志文件名为 myapp.log，轮换日志为 myapp.log.1、myapp.log.2 等等，可以使用以下 include 模式：</p> <pre>include= myapp.log;myapp.log.*</pre>
exclude=regular_expression	<p>（可选）要从收集排除的文件名或文件掩码（glob 模式）。可以通过分号分隔的列表的形式提供值。默认值为空，表示不排除任何文件。</p>

参数	描述
event_marker=regular_expression	<p>(可选) 表示日志文件中事件开始的正则表达式。如果省略, 则默认为换行符。键入的表达式必须使用 Perl 正则表达式语法。</p> <p>注 例如, 引号 (" ") 等符号不作为正则表达式的包装器处理。将其视为模式的一部分。</p> <p>由于 vRealize Log Insight 代理针对实时收集进行了优化, 写入时存在内部延迟的部分日志消息可能会拆分到多个事件中。如果日志文件附加操作停止的时间超过 200 毫秒, 而没有观察到新的 event_marker, 部分事件会被视为已完成、已分析且已传送。此计时逻辑不可配置, 且优先级高于 event_marker 设置。日志文件附加程序应刷新完整事件。</p>
enabled=yes no	<p>(可选) 用于启用或禁用配置部分的一个参数。可能的值为 yes 或 no。默认值为 yes。</p>
charset=char-encoding-type	<p>(可选) 代理所监控的日志文件的字符编码。可能的值包括:</p> <ul style="list-style-type: none"> ■ UTF-8 ■ UTF-16LE ■ UTF-16BE <p>默认值为 UTF-8。</p>
tags={"tag-name": "tag-value", ...}	<p>(可选) 用于向所收集事件的字段添加自定义标记的一个参数。使用 JSON 表示法定义标记。标记名称可以包含字母、数字和下划线。标记名称只能以字母或下划线开头, 并且不能超过 64 个字符。标记名称不区分大小写。例如, 如果使用 tags={"tag_name1": "tag value 1", "Tag_Name1": "tag value 2"}, 则将 Tag_Name1 作为重复项忽略。不能将 event_type 和时间戳用作标记名称。同一个声明中的所有重复项都将被忽略。</p> <p>如果目标是 syslog 服务器, 标记可能会覆盖 APP-NAME 字段。例如, tags={"appname": "VROPS"}。</p>
exclude_fields	<p>(可选) 用于从收集中排除各个字段的一个参数。可以通过分号或逗号分隔的列表形式提供多个值。例如,</p> <ul style="list-style-type: none"> ■ exclude_fields=hostname; filepath ■ exclude_fields=type; size ■ exclude_fields=type, size
raw_syslog=Yes No	<p>对于使用 syslog 协议的代理, 该选项允许代理收集和发送原始 syslog 事件。默认值为 No, 表示使用用户指定的 syslog 属性转换收集的事件。启用此选项可在不进行任何 syslog 转换的情况下收集事件。</p>

示例： 配置

```
[filelog|vCenterMain]
directory=C:\ProgramData\VMware\VMware VirtualCenter\Logs
include=vpxd-*.log
exclude=vpxd-alert-*.log;vpxd-profiler-*.log
event_marker=^{\d{4}}-\d{2}-\d{2} [A-Z]\d{2}:\d{2}:\d{2}\.\d{3}
```

```
[filelog|ApacheAccessLogs]
enabled=yes
directory=C:\Program Files (x86)\Apache Software Foundation\Apache2.2\logs
include=*.log
```

```
exclude=*_old.log
tags={"Provider" : "Apache"}
```

```
[filelog|MSSQL]
directory=C:\Program Files\Microsoft SQL Server\MSSQL10.MSSQLSERVER\MSSQL\Log
charset=UTF-16LE
event_marker=[^\s]
```

设置 Windows 日志文件通道筛选

您可以为 Windows 日志文件设置筛选器以显式包含或排除日志事件。

使用 `whitelist` 和 `blacklist` 参数来评估筛选表达式。筛选表达式是一个包含事件字段和运算符的布尔表达式。

注 `blacklist` 选项仅适用于字段；不可用于块文本。

- `whitelist` 参数仅收集筛选表达式评估为非零的日志事件。如果忽略此参数，则相应的值为隐含的 1。
- `blacklist` 参数将排除筛选表达式评估为非零的日志事件。默认值为 0。

有关 Windows 事件字段和运算符的完整列表，请参见[事件字段和运算符](#)。

前提条件

登录到已安装 vRealize Log Insight Windows 代理的 Windows 计算机，启动“服务”管理器以验证是否已安装 vRealize Log Insight 代理服务。

步骤

- 1 导航到 vRealize Log Insight Windows 代理的程序数据目录。

```
%ProgramData%\VMware\Log Insight Agent
```

- 2 在任意文本编辑器中打开 `liagent.ini` 文件。
- 3 在 `[filelog|]` 部分中添加 `whitelist` 或 `blacklist` 参数。

例如：

```
[filelog|apache]
directory = path_to_log_directory
include = glob_pattern
blacklist = filter_expression
```

- 4 从 Windows 事件字段和运算符中创建筛选表达式。

例如

```
whitelist = myServer
```

- 5 保存并关闭 `liagent.ini` 文件。

示例：筛选器配置

您可以配置代理以仅收集 Apache 日志，其中，server_name 为

```
[filelog|apache]
directory=C:\Program Files\Apache Software Foundation\Apache2.4\logs
include=error.log
parser=clf
whitelist = server_name == "sample.com"
blacklist = remote_host == "127.0.0.1"
```

将事件转发到 Log Insight Windows Agent

您可以将事件从 Windows 计算机转发到运行 Log Insight Windows Agent 的计算机。

您可以使用 Windows 事件转发功能将事件从多台 Windows 计算机转发到已安装 Log Insight Windows Agent 的计算机。然后，可以将 Log Insight Windows Agent 配置为收集所有转发事件，并将这些事件发送到 vRealize Log Insight 服务器。

熟悉 Windows 事件转发。请参见 <http://technet.microsoft.com/en-us/library/cc748890.aspx> 和 [http://msdn.microsoft.com/en-us/library/windows/desktop/bb870973\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/bb870973(v=vs.85).aspx)。

前提条件

请参见从 [Windows 事件通道收集事件](#)。

步骤

- 1 向 Log Insight Windows Agent 配置添加一个新部分，用于从接收转发事件的 Windows 事件通道中收集事件。

默认通道名称为 ForwardedEvents。

- 2 设置 Windows 事件转发。

后续步骤

转至 vRealize Log Insight Web 用户界面，并确认转发事件已到达。

配置 Log Insight Linux Agent

安装 Log Insight Linux Agent 后，您可以对其进行配置。

您可以使用[集中式代理配置](#)，可设置要将事件发送到 vRealize Log Insight 服务器的代理、指定通信协议和端口，以及配置平面文件日志收集。有关 liagent.ini 文件的位置，请参见在 [Log Insight Agents](#) 中定义日志详细信息级别。

vRealize Log InsightLinux 代理的默认配置

安装完成后，liagent.ini 文件中包含 Log Insight Windows Agent 的预配置默认设置。

vRealize Log InsightLinux 代理 liagent.ini 默认配置

如果您使用非 ASCII 的名称和值，请将配置保存为 UTF-8。

如果您使用集中式配置，最终配置是将此文件与服务器中的设置合并，从而构成 liagent-effective.ini 文件。

您可能会发现从服务器的代理页面配置设置更高效。

```
[server]
; Hostname or IP address of your Log Insight server / cluster load balancer. Default:
;hostname=LOGINSIGHT

;Enables or disables centralized configuration from the vRealize Log Insight server.
;When enabled, agent configuration changes made to the liagent.ini file on the server
;are joined with the settings in this file. to this agent. Accepted values are yes or no and
0 or 1.
;The default is yes.
;
;central_config=yes
;
;
; Protocol can be cfapi (Log Insight REST API), syslog. Default:
;proto=cfapi

; Log Insight server port to connect to. Default ports for protocols (all TCP):
; syslog: 514; syslog with ssl: 6514; cfapi: 9000; cfapi with ssl: 9543. Default:
;port=9543

; SSL usage. Default:
;ssl=yes
; Example of configuration with trusted CA:
;ssl=yes
;ssl_ca_path=/etc/pki/tls/certs/ca.pem

; Time in minutes to force reconnection to the server.
; This option mitigates imbalances caused by long-lived TCP connections. Default:
;reconnect=30

[logging]
; Logging verbosity: 0 (no debug messages), 1 (essentials), 2 (verbose with more impact on
performance).
; This option should always be 0 under normal operating conditions. Default:
;debug_level=0

[storage]
; Max local storage usage limit (data + logs) in MBs. Valid range: 100-2000 MB.
;max_disk_buffer=200

; Uncomment the appropriate section to collect system logs
```

```
; The recommended way is to enable the Linux content pack from LI server
;[filelog|syslog]
;directory=/var/log
;include=messages;messages.?.syslog;syslog.?
```

参数	默认值	描述
hostname	LOGINSIGHT	vRealize Log Insight 虚拟设备的 IP 地址或主机名。 默认值为 loginsight 。
central_config	yes	为此代理启用或禁用集中式配置。禁用集中式配置后，代理会忽略由 vRealize Log Insight 服务器提供的配置。接受的值为 yes、no、1 或 0。默认值为 yes。
proto	cfapi	代理向 vRealize Log Insight 服务器发送事件所使用的协议。可能的值为 cfapi 和 syslog。 默认值为 cfapi。
port	9543、9000、6514 和 514	代理向 vRealize Log Insight 服务器发送事件所使用的通信端口。默认值如下：启用 SSL 的 cfapi 为 9543，禁用 SSL 的 cfapi 为 9000，启用 SSL 的 syslog 为 6514，禁用 SSL 的 syslog 为 514。
ssl	yes	启用或禁用 SSL。默认值为 yes。 将 ssl 设置为 “yes” 时，如果未设置端口值，则会自动为端口选取值 9543。
max_disk_buffer	200	由 Log Insight Windows Agent 用于缓冲事件及其日志的最大磁盘空间 (MB)。 达到指定的 max_disk_buffer 时，代理将开始丢弃新的入站事件。
debug_level	0	定义日志详细信息级别。请参见在 Log Insight Agents 中定义日志详细信息级别。

从日志文件收集事件

您可以将 vRealize Log Insight Linux 代理配置为从一个或多个日志文件收集事件。

默认情况下，vRealize Log Insight Linux 代理会收集由应用程序或编辑器创建的隐藏文件。隐藏文件的文件名以句点开头。您可以通过添加排除参数 **exclude=.***，禁止 vRealize Log Insight Linux 代理收集隐藏文件。

字段名称受到一定的限制。以下名称是保留名称，因此不能用作字段名称。

- event_type
- hostname
- source

■ text

您最多可为代理信息指定三个目标，并且可以在发送该信息之前进行筛选。请参见[从 vRealize Log Insight 代理转发信息](#)

注 如果监控大量文件（如 1000 个或更多），将导致 vRealize Log Insight 代理具有较高的资源利用率并影响主机的总体性能。要防止出现这种情况，请配置代理以使用模式和 **glob** 仅监控所需的文件，或者存档旧日志文件。如果要求监控大量文件，请考虑增加主机参数，例如，CPU 和 RAM。

前提条件

- 以 **root** 用户身份登录，或使用 **sudo** 运行控制台命令。
- 确认 vRealize Log Insight Linux 代理已安装且正在运行。登录到已安装 vRealize Log Insight Linux 代理的 Linux 计算机，打开控制台，然后运行 **pgrep liagent**。

步骤

- 1 在任意文本编辑器中打开 `/var/lib/loginsight-agent/liagent.ini` 文件。
- 2 找到该文件的 `[server|<dest_id>]` 部分。添加配置参数并设置环境的值。

```
[filelog|section_name]
directory=path_to_log_directory
include=glob_pattern
...
```

参数	描述
<code>[filelog section_name]</code>	配置部分的唯一名称。
<code>directory=full-path-to-log-file</code>	<p>日志文件目录的完整路径。支持 glob 模式。示例配置：</p> <ul style="list-style-type: none"> ■ 要从 <code>D:\Logs\new_test_logs</code> 目录的所有子目录中收集，请使用 <code>directory=D:\Logs\new_test_logs*</code> ■ 如果子目录具有自己的子目录，请使用以下配置以监控所有子目录： <code>directory=D:\Logs\new_test_logs**</code> <p>注 要限制文件和目录数并避免较高的资源消耗，您不能为第一级或第二级目录定义目录 glob，例如：<code>directory=c:/tmp/*</code> 或 <code>directory=c:\Logs*</code>。目录路径必须至少具有两个级别。</p> <p>您可以定义一个到当前不存在目录的路径，创建该目录及其中的日志文件后，代理即会收集这些日志文件。</p> <p>可以在一个或多个不同配置部分下定义相同的目录，以便多次从同一文件收集日志。利用该过程，可以对相同的事件源应用不同标记和筛选器。</p> <p>注 如果这些部分使用完全相同的配置，则会在服务器端观察到重复的事件。</p>

参数	描述
include= <i>file_name</i> ; ...	<p>(可选) 要从中收集数据的文件名或文件掩码 (glob 模式)。可以通过分号分隔的列表的形式提供值。默认值为 <code>*</code>, 表示包括所有文件。该参数区分大小写。</p> <p>可以使用文件掩码 (glob 模式) 将遵循相同命名约定的文件, 以及一个文件名下的所有文件分组到一起。例如, 包含空格的文件名 (例如, <code>vRealize Ops Analytics.log</code> 和 <code>vRealize Ops Collector.log</code>) 可以使用 <code>vRealize?Ops?Analytics*.log</code> 或 <code>vRealize*.log</code> 指定。通过使用文件掩码, 可以指定 Linux 和 Windows 主机下的代理配置可接受的文件名。</p> <p>默认情况下, <code>.zip</code> 和 <code>.gz</code> 文件会从收集排除。</p> <p>重要事项 如果要收集轮换的日志文件, 请使用 <code>include</code> 和 <code>exclude</code> 参数指定同时与主要文件和轮换文件相匹配的 glob 模式。如果 glob 模式仅与主要日志文件相匹配, 则 vRealize Log Insight 代理可能会在轮换期间错过事件。vRealize Log Insight 代理会自动确定轮换文件的正确顺序, 并将事件以正确的顺序发送到 vRealize Log Insight 服务器。例如, 如果您的主要日志文件名为 <code>myapp.log</code>, 轮换日志为 <code>myapp.log.1</code>、<code>myapp.log.2</code> 等等, 可以使用以下 <code>include</code> 模式:</p> <pre>include= myapp.log;myapp.log.*</pre>
exclude= <i>regular_expression</i>	<p>(可选) 要从收集排除的文件名或文件掩码 (glob 模式)。可以通过分号分隔的列表的形式提供值。默认值为空, 表示不排除任何文件。</p>
event_marker= <i>regular_expression</i>	<p>(可选) 表示日志文件中事件开始的正则表达式。如果省略, 则默认为换行符。键入的表达式必须使用 Perl 正则表达式语法。</p> <p>注 例如, 引号 (" ") 等符号不作为正则表达式的包装器处理。将其视为模式的一部分。</p> <p>由于 vRealize Log Insight 代理针对实时收集进行了优化, 写入时存在内部延迟的部分日志消息可能会拆分到多个事件中。如果日志文件附加操作停止的时间超过 200 毫秒, 而没有观察到新的 <code>event_marker</code>, 部分事件会被视为已完成、已分析且已传送。此计时逻辑不可配置, 且优先级高于 <code>event_marker</code> 设置。日志文件附加程序应刷新完整事件。</p>
enabled= <i>yes no</i>	<p>(可选) 用于启用或禁用配置部分的一个参数。可能的值为 <code>yes</code> 或 <code>no</code>。默认值为 <code>yes</code>。</p>
charset= <i>char-encoding-type</i>	<p>(可选) 代理所监控的日志文件的字符编码。可能的值包括:</p> <ul style="list-style-type: none"> ■ UTF-8 ■ UTF-16LE ■ UTF-16BE <p>默认值为 UTF-8。</p>
tags= <i>{"tag-name" : "tag-value", ...}</i>	<p>(可选) 用于向所收集事件的字段添加自定义标记的一个参数。使用 JSON 表示法定义标记。标记名称可以包含字母、数字和下划线。标记名称只能以字母或下划线开头, 并且不能超过 64 个字符。标记名称不区分大小写。例如, 如果使用 <code>tags={"tag_name1" : "tag value 1", "Tag_Name1" : "tag value 2" }</code>, 则将 <code>Tag_Name1</code> 作为重复项忽略。不能将 <code>event_type</code> 和时间戳用作标记名称。同一个声明中的所有重复项都将被忽略。</p> <p>如果目标是 syslog 服务器, 标记可能会覆盖 <code>APP-NAME</code> 字段。例如, <code>tags={"appname":"VROPS"}</code>。</p>

参数	描述
exclude_fields	<p>(可选) 用于从收集中排除各个字段的一个参数。可以通过分号或逗号分隔的列表形式提供多个值。例如，</p> <ul style="list-style-type: none"> ■ exclude_fields=hostname; filepath ■ exclude_fields=type; size ■ exclude_fields=type, size
raw_syslog=Yes No	<p>对于使用 syslog 协议的代理，该选项允许代理收集和发送原始 syslog 事件。默认值为 No，表示使用用户指定的 syslog 属性转换收集的事件。启用此选项可在不进行任何 syslog 转换的情况下收集事件。</p>

3 保存并关闭 liagent.ini 文件。

示例：配置

```
[filelog|messages]
directory=/var/log
include=messages;messages.?

[filelog|syslog]
directory=/var/log
include=syslog;syslog.?

[filelog|Apache]
directory=/var/log/apache2
include=*
```

筛选事件

您可以在 vRealize Log Insight Linux 代理上根据字段值对所有已收集的事件进行筛选，以便指定要选取或丢弃的日志事件。您可以使用 whitelist 和 blacklist 收集器选项来定义筛选器。

提示 默认情况下，vRealize Log Insight Linux 代理会收集由程序或编辑器创建的隐藏文件。隐藏文件的名称以句点开头。可以防止 vRealize Log Insight Linux 代理收集隐藏文件，方法是通过添加排除 **exclude=.*** 参数。

对于每个事件，收集器均会计算 whitelist 和 blacklist 筛选器表达式的值。如果 whitelist 表达式的值为 true，而 blacklist 表达式的值为 false 或无法计算，则事件将移至相应队列以便进一步处理。在任何其他情况下，收集器均会丢弃事件。whitelist 表达式的默认值为 true，而 blacklist 表达式的默认值为 false。

提示 Filelog 收集器提供的可进行筛选的字段较少。要获取可进行筛选的字段，可以解析日志。有关详细信息，请参见[分析日志](#)。

`whitelist` 或 `blacklist` 筛选器是由变量、文本和运算符构成的集合，其计算结果为一个逻辑值或整数值。您可以将事件字段用作变量，并将带双引号的字符串和数字用作文本。有关可在筛选器表达式中使用的运算符的信息，请参见[事件字段和运算符](#)。

注

- 如果将数字与字符串进行比较，或者如果执行的比较涉及数字字符串，则每个字符串均会转换为一个数字，并且将以数字形式执行比较。例如：
 - 表达式 `whitelist = 123.0 == "000123"` 的值为 **true**。
 - 表达式 `whitelist = "00987" == "987.00"` 的值为 **true**。
 - 在表达式 `whitelist = response_size >= "12.12"` 中，如果 `response_size` 字段具有数字值，则该表达式将以数字形式进行计算。如果响应大小大于 **12.12**，则表达式的值为 **true**，否则为 **false**。
 - 在表达式 `whitelist = "09123" < "234"` 中，字符串文本将转换为数值，且表达式的值为 **false**。
- 如果其中某一字符串操作数无法转换为数值，则两个操作数都会转换为字符串。对于这种情况，将简单地按字典顺序进行比较，且比较时区分大小写。例如：
 - 表达式 `whitelist = "1234a" == "1234A"` 是一种字符串比较，其值为 **false**。
 - 表达式 `whitelist = 4 < "four"` 会将 **4** 转换为 **"4"**，其值为 **true**。
 - 在表达式 `whitelist = response_size > "thousand"` 中，`response_size` 字段的值将转换为一个字符串值，因此该表达式的值为 **false**。
- 如果筛选器表达式的值为一个整数值，那么当值为 **0** 时，可将其视为 **false**，否则视为 **true**。
 例如，在表达式 `whitelist = some_integer & 1` 中，如果 `some_integer` 字段具有最低有效位组，则该表达式的值为 **true**，否则为 **false**。

有关事件字段和运算符的完整列表，请参见[从日志文件收集事件](#)。

在此示例中，您将从文件 `/var/log/httpd/access` 中收集 **Apache** 访问日志。该文件中的一些示例日志如下所示：

- `127.0.0.1 - frank [10/Oct/2016:13:55:36 +0400] "GET /apache_pb.gif HTTP/1.0" 200 2326`
- `198.51.100.56 - john [10/Oct/2016:14:15:31 +0400] "GET /some.gif HTTP/1.0" 200 8270`
- `198.51.100.12 - smith [10/Oct/2016:14:15:31 +0400] "GET /another.gif HTTP/1.0" 303 348`
- `198.51.100.32 - test [10/Oct/2016:15:22:55 +0400] "GET /experimental_page.gif HTTP/1.0" 400 46374`
- `127.0.0.1 - test [10/Oct/2016:15:22:57 +0400] "GET /experimental_page2.gif HTTP/1.0" 301 100`

前提条件

- 以 **root** 用户身份登录，或使用 `sudo` 运行控制台命令。
- 登录到已安装 vRealize Log Insight Linux 代理的 Linux 计算机，打开控制台，然后运行 `pgrep liagent` 以验证 vRealize Log Insight Linux 代理是否已安装且正在运行。

步骤

- 1 为日志定义一个解析器，如以下代码段中所示：

```
[parser|apache-access]
base_parser=clf
format=%h %l %u %t \"%r\" %s %b
```

对于从文件 `/var/log/httpd/access` 中收集的每个事件，您定义的解析器都会提取 `remote_host`、`remote_log_name`、`remote_auth_user`、`timestamp`、`request`、`status_code` 和 `response_size` 字段。可以使用这些字段来筛选事件。

- 2 在任意文本编辑器中打开 `/var/lib/loginsight-agent/liagent.ini` 文件。
- 3 在该文件中定义一个 `Filelog` 部分以收集和解析日志，如以下代码段中所示：

```
[filelog|apache-access]
directory = /var/log/httpd/
include = access
parser = apache-access
```

- 4 根据您的要求筛选事件。

- 要收集 HTTP 状态为 200 的日志，您可以在 `Filelog` 部分中定义一个 `whitelist`，如以下代码段中所示：

```
[filelog|apache-access]
directory = /var/log/httpd/
include = access
parser = apache-access
whitelist = status_code == 200
```

只有对于示例日志中的第一个和第二个事件，`whitelist` 表达式的值才为 `true`，因此收集器会选取这两个事件。

如果事件中不存在 `status_code` 字段（原因是该字段不在日志中或未进行解析），那么将无法计算 `whitelist` 表达式的值，这意味着该表达式的值将为 `false`，并且收集器将丢弃事件。

- 要丢弃不感兴趣的事件，您可以使用 `blacklist` 选项。例如，如果您对本地流量不感兴趣，则可以阻止本地 IP，如以下代码段中所示：

```
[filelog|apache-access]
directory = /var/log/httpd/
include = access
parser = apache-access
blacklist = remote_host == "127.0.0.1"
```

收集器会从示例日志中选取第二个、第三个和第四个事件。

- 要根据多个谓词筛选事件，您可以使用 `or` 和 `and` 运算符。例如，您可以丢弃从本地 IP 生成的事件或由不需要的任何主机上的测试用户生成的事件，如以下代码段中所示：

```
[filelog|apache-access]
directory = /var/log/httpd/
include = access
parser = apache-access
blacklist = remote_host == "127.0.0.1" or remote_auth_user == "test"
```

使用 `or` 运算符可将 `blacklist` 表达式的值计算为 `true`，以跳过不需要的事件。如果 `remote_host` 字段值为“127.0.0.1”，或者 `remote_auth_user` 字段值为“test”，则该表达式将指示收集器丢弃事件。

收集器会从示例日志中选取第二个和第三个事件。

- 要丢弃由测试用户从本地 IP 生成的事件，您可以在 `blacklist` 表达式中使用 `and`，如以下代码段中所示：

```
[filelog|apache-access]
directory = /var/log/httpd/
include = access
parser = apache-access
blacklist = remote_host == "127.0.0.1" and remote_auth_user == "test"
```

收集器会从示例日志中丢弃第五个事件。

- 您可以将 `whitelist` 与 `blacklist` 筛选器一起使用。例如，如果您需要响应大小大于 1024 字节的事件，但不需要源自本地主机的事件，则可以使用以下代码段：

```
[filelog|apache-access]
directory = /var/log/httpd/
include = access
parser = apache-access
whitelist = response_size > 1024
blacklist = remote_host == "127.0.0.1" or remote_host == "localhost"
```

收集器会从示例日志中选取第二个事件。

从 journald 中收集事件

从 vRealize Log Insight 4.6 开始，在运行 `systemd` 的 Linux 分发版本中，代理可以从来自 `journald` 系统服务的日志中读取日志数据。`journald` 现在是基于 `systemd` 的 Linux 平台中默认的日志记录标准。

`journald` 配置部分支持以下选项：

journal_files

要监控的日志文件。支持以下值：

值	描述
all	打开并监控所有可用的日志文件。
local	监控并仅读取本地计算机上生成的日志文件。
runtime	监控并仅读取易失性日志文件，不包括永久存储中的文件。
system	监控并仅读取系统服务和内核日志文件。
user	监控并仅读取当前用户的日志文件。

fetch_fields

要从日记日志条目中的消息提取的字段。此选项的值是以逗号分隔的字段名称列表（不区分大小写）。支持以下值：

值	描述
pri_severity,pri_facility,syslog_identifier	此选项的默认值。
*	提取所有字段。
all	不提取任何字段。

筛选来自 vRealize Log Insight 代理的事件

您可以使用本地 `liagent.ini` 文件中 `[server|<dest_id>]` 部分的筛选器选项，指定代理发送到目标的信息。

此选项的格式如下：

```
filter = {collector_type; collector_filter; event_filter}
```

筛选器类型	描述
collector_type	逗号分隔的列表，用于定义收集器类型。支持的值为 <code>filelog</code> 或 <code>winlog</code> 。如果未提供任何值，则将使用所有收集器类型。
collector_filter	以正则表达式的格式指定收集器部分的名称。例如， <code>vcops_*</code> 是指以 “ <code>vcops_</code> ” 开头的所有收集器部分。
event_filter	事件字段筛选器使用的语法与收集器部分中接受列表或拒绝列表使用的语法相同。代理只发送将表达式计算结果为 <code>True</code> 或非零值的事件。空 <code>event_filter</code> 的计算结果始终为 <code>True</code> 。要对事件使用 <code>event_filter</code> ，必须在相应的收集器部分中定义一个用于字段提取的分析程序。如果由于所收集的事件中缺少字段而无法对表达式进行计算，则会丢弃该事件。

通过使用逗号将筛选器表达式分隔开，可以指定多个筛选器表达式，如下例所示：

```
filter=
{winlog;Micr.*;},{filelog;apache-access;level=="error"}
```

如果消息满足目标的多组筛选标准，只发送一次。

表 4-1. 语法示例

筛选	含义
filter= {winlog;Microsoft.*;}	仅当事件名称以“Microsoft”开头时，才发送来自 winlog 收集器的事件。
filter= {winlog;Microsoft.*; eventid == 1023}	仅当事件名称以“Microsoft”开头，并且事件 ID 等于 1023 时，才发送来自 winlog 收集器的事件。
filter= {.*;}	默认筛选器值。发送来自所有来源的全部事件。
filter= {winlog;.*;}	发送来自 winlog 部分的所有事件。
filter= {filelog;syslog;facility<5}	如果 facility 小于 5，将发送来自 [filelog syslog] 部分的事件。[filelog syslog] 部分必须包含一个能够提取 facility 字段的分析程序。否则，会跳过所有事件。
filter= {;;}	不匹配任何事件。可以使用此语法禁用事件转发。

以下示例向上一示例中第二个目标的配置添加一个筛选器。

```
; The second destination receives just syslog events through the plain syslog protocol.
[server|syslog-audit]
hostname=third_party_audit_management.eng.vmware.com
proto=syslog
ssl=no
filter= {filelog; syslog; }
```

下一个示例使用更复杂的筛选器表达式。

```
; This destination receives vRealize Operations Manager events if they have the level field
equal
;to "error" or "warning" and they are collected by sections whose name begins with "vrops-"

[server|licf-prod1]
hostname=vrops-errors.licf.vmware.com
filter= {; vrops-.*; level == "error" || level == "warning"}
```

通过使用逗号将筛选器表达式分隔开，可以指定多个筛选器表达式，如下例所示。

```
filter= e.
{winlog;Micr.*;},{filelog;apache-access;level=="error"}
```

从 vRealize Log Insight 代理转发信息

您可以将代理收集的事件转发到最多三个目标。目标可以包括 vRealize Log Insight 服务器或转发器，或者第三方日志管理解决方案。

例如，您可能需要将审核或系统日志发送到您安全团队的服务器，将应用程序日志发送到开发运营团队的服务器，将衡量指标日志发送到 IT 管理系统。您可以使用筛选器指定将哪些信息转到哪个目标。您可以将信息从单个 vRealize Log Insight 代理转发到最多三个目标。

代理配置通过您本地 `liagent.ini` 文件的 `[server|<dest_id>]` 部分来完成。将 `cfapi` 协议用于 vRealize Log Insight 服务器或转发器，将 `syslog` 用于其他目标。

为代理指定多个目标时，第一个目标使用默认的 `loginsight` 位置。您必须指定其他目标的位置信息。

下一个示例显示一个指定两个目标的 `liagent.ini` 文件的部分内容。由于默认情况下默认服务器名称 `loginsight` 将隐式应用于第一个目标，因此未指定。第二个 `[server|<dest_id>]` 部分指定一个目标。

```
; The first (default) destination receives all collected events.
[server]
ssl=yes

; The second destination receives just syslog events through the plain syslog protocol.
[server|syslog-audit]
hostname=third_party_audit_management.eng.vmware.com
proto=syslog
ssl=no
```

有关为代理创建筛选器的信息，请参见[筛选来自 vRealize Log Insight 代理的事件](#)。

设置目标 vRealize Log Insight 服务器

您可以为在 Windows 上运行的 vRealize Log Insight 代理设置或更改目标 vRealize Log Insight 服务器。您可以将事件发送到最多三个目标，并可按目标筛选输出。

可以通过 `liagent.ini` 文件的 `[server]` 部分配置默认目标。默认目标始终存在，并且默认情况下主机名设置为 `loginsight`。要添加更多目标，请为每个目标创建 `[server|<dest_id>]` 部分。对于每个额外连接，必须指定一个唯一的主机名作为目标 ID。您可以对其他目标使用与默认 `[server]` 部分相同的选项。请勿将其他目标配置为自动升级，也不要使用它们来配置代理。您可以指定两个其他目标。

默认情况下，代理会将所有收集的事件发送到所有目标。您可以通过 `file` 选项筛选事件以将不同的事件发送到不同的目标。有关详细信息，请参见[筛选来自 vRealize Log Insight 代理的事件](#)。

前提条件

- 登录到已安装 vRealize Log Insight Windows 代理的 Windows 计算机，启动“服务”管理器以验证是否已安装 vRealize Log Insight 代理服务。
- 如果 vRealize Log Insight 群集具有启用的集成负载均衡器，请参见[启用集成负载均衡器](#)以了解自定义 SSL 证书特定的要求。

步骤

- 1 导航到 vRealize Log Insight Windows 代理的程序数据目录。

```
%ProgramData%\VMware\Log Insight Agent
```

- 2 在任意文本编辑器中打开 liagent.ini 文件。

- 3 修改以下参数，并设置用于您环境的值。

参数	描述
proto	代理向 vRealize Log Insight 服务器发送事件所使用的协议。可能的值为 cfapi 和 syslog。 默认值为 cfapi。
hostname	vRealize Log Insight 虚拟设备的 IP 地址或主机名。 可以指定 IPv4 或 IPv6 地址。指定 IPv6 地址时可以使用方括号，也可以不使用。 例如： <pre>hostname = 2001:cdba::3257:9652 or hostname = [2001:cdba::3257:9652]</pre> 如果主机同时支持 IPv4 和 IPv6 堆栈，并且域名被指定为主机名，代理将根据名称解析程序返回的 IP 地址选择 IP 堆栈。如果解析程序同时返回 IPv4 和 IPv6 地址，代理将尝试按给定的顺序依次连接这两个地址。
max_disk_buffer	Log Insight Windows 代理可用于缓冲为此特定服务器所收集事件的最大磁盘空间 (MB)。此选项将覆盖该服务器的 [storage].max_disk_buffer 值。 默认值为 150 MB，您可以将缓冲区大小设置为 50 MB 到 8000 MB。
port	代理向 vRealize Log Insight 服务器或第三方服务器发送事件所使用的通信端口。 默认情况下，代理根据为 SSL 和协议设置的选项使用相应的端口。请参见下面列表中提供的默认端口值。仅当端口选项与以下默认值不同时，您才需要指定端口选项。 <ul style="list-style-type: none"> ■ 启用了 SSL 的 cfapi: 9543 ■ 禁用了 SSL 的 cfapi: 9000 ■ 启用了 SSL 的 syslog: 6514 ■ 禁用了 SSL 的 syslog: 514
ssl	启用或禁用 SSL。默认值为 yes。 将 ssl 设置为 yes 时，端口将设置为 9543，除非您另行指定。
reconnect	强制重新连接到服务器的时间（分钟）。默认值为 30。
filter	指定代理发送到目标位置的信息。此选项采用以下三个参数： <pre>{collector_type; collector_filter; event_filter}</pre>

```
[server]
hostname=LOGINSIGHT
; Hostname or IP address of your Log Insight server / cluster load balancer. Default:
;hostname=LOGINSIGHT

; Protocol can be cfapi (Log Insight REST API), syslog. Default:
;proto=cfapi
```

```
; Log Insight server port to connect to. Default ports for protocols (all TCP):
; syslog: 514; syslog with ssl: 6514; cfapi: 9000; cfapi with ssl: 9543. Default:
;port=9543

; SSL usage. Default:
;ssl=yes
```

4 保存并关闭 liagent.ini 文件。

示例

以下配置示例将设置一个使用受信证书颁发机构的目标 vRealize Log Insight 服务器。

```
[server]
proto=cfapi
hostname=LOGINSIGHT
port=9543
ssl=yes;
ssl_ca_path=/etc/pki/tls/certs/ca.pem
```

以下示例显示了一个包含每个目标的筛选消息的多目标配置。

```
; The first (default) destination receives all collected events.
[server]
hostname=prod1.licf.vmware.com

; The second destination receives just syslog events through the plain syslog protocol.
[server|syslog-audit]
hostname=third_party_audit_management.eng.vmware.com
proto=syslog
ssl=no
filter={filelog; syslog; }

; The third destination receives vRealize Operations Manager events if they have the level
field equal to "error" or "warning"
; and they are collected by sections whose name begins with "vrops-"

[server|licf-prod1]
hostname=vrops-errors.licf.vmware.com
filter={; vrops-.*; level == "error" || level == "warning"}

; Collecting syslog messages.
[filelog|syslog]
directory=/var/log
include=messages

; various vROPs logs. Note that all section names begin with a "vrops-" prefix, which is used
in third destination filter.
[filelog|vrops-ANALYTICS-analytics]
directory=/data/vcops/log
include=analytics*.log*
exclude=analytics*-gc.log*
```

```

parser=auto

[filelog|vrops-COLLECTOR-collector]
directory=/data/vcops/log
include=collector.log*
event_marker=^\\d{4}-\\d{2}-\\d{2} [\\s]\\d{2}:\\d{2}:\\d{2}\\.\\d{3}
parser=auto

[filelog|vrops-COLLECTOR-collector_wrapper]
directory=/data/vcops/log
include=collector-wrapper.log*
event_marker=^\\d{4}-\\d{2}-\\d{2} [\\s]\\d{2}:\\d{2}:\\d{2}\\.\\d{3}
parser=auto

```

后续步骤

您可以为 vRealize Log Insight 代理配置其他 SSL 选项。请参见[在服务器和 Log Insight 代理之间配置 SSL 连接](#)。

指定代理的目标

您可以指定最多三个 vRealize Log Insight Linux 代理可以将事件发送到的目标。

多个目标连接通过 `li-agent.ini` 文件的 `[server|<dest_id>]` 部分进行定义，其中，`<dest_id>` 是每个配置连接的唯一 ID。您可以对其他目标使用与默认 `[server]` 部分相同的选项。但是，请勿将其他目标配置为自动升级，也不要使用它们来配置代理。您可以指定两个其他目标。

您定义的第一个目标可以使用默认服务器值 `loginsight`。定义其他目标时，必须在后续目标的 `[server]` 部分中指定一个主机名。如果不筛选，代理会将所有收集的事件发送到所有目标。这是默认行为。不过，您可以通过筛选事件将不同的事件发送到不同的目标。

前提条件

- 以 **root** 用户身份登录，或使用 `sudo` 运行控制台命令。
- 登录到已安装 vRealize Log InsightLinux 代理的 Linux 计算机，打开控制台，然后运行 `pgrep liagent` 以验证 vRealize Log Insight Linux 代理是否已安装且正在运行。
- 如果 vRealize Log Insight 集群具有启用的集成负载均衡器，请参见[启用集成负载均衡器](#)以了解自定义 SSL 证书特定的要求。

步骤

- 1 在任意文本编辑器中打开 `/var/lib/loginsight-agent/liagent.ini` 文件。

2 修改以下参数，并设置用于您环境的值。

参数	描述
proto	代理向 vRealize Log Insight 服务器发送事件所使用的协议。可能的值为 <code>cfapi</code> 和 <code>syslog</code> 。 默认值为 <code>cfapi</code> 。
hostname	vRealize Log Insight 虚拟设备的 IP 地址或主机名。 可以指定 IPv4 或 IPv6 地址。指定 IPv6 地址时可以使用方括号，也可以不使用。 例如： <pre>hostname = 2001:cdba::3257:9652 or hostname = [2001:cdba::3257:9652]</pre> 如果主机同时支持 IPv4 和 IPv6 堆栈，并且域名被指定为主机名，代理将根据名称解析程序返回的 IP 地址使用 IP 堆栈。如果解析程序同时返回 IPv4 和 IPv6 地址，代理将尝试按给定的顺序依次连接这两个地址。
max_disk_buffer	Log Insight Linux 代理可用于缓冲为此特定服务器所收集事件的最大磁盘空间 (MB)。此选项将覆盖该服务器的 <code>[storage].max_disk_buffer</code> 值。 默认值为 150 MB，您可以将缓冲区大小设置为 50 MB 到 8000 MB。
port	代理向 vRealize Log Insight 服务器或第三方服务器发送事件所使用的通信端口。 默认情况下，代理根据为 SSL 和协议设置的选项使用相应的端口。请参见下面列表中提供的默认端口值。仅当端口选项与以下默认值不同时，您才需要指定端口选项。 <ul style="list-style-type: none"> ■ 启用了 SSL 的 <code>cfapi</code>: 9543 ■ 禁用了 SSL 的 <code>cfapi</code>: 9000 ■ 启用了 SSL 的 <code>syslog</code>: 6514 ■ 禁用了 SSL 的 <code>syslog</code>: 514
ssl	启用或禁用 SSL。默认值为 <code>yes</code> 。 将 <code>ssl</code> 设置为 “yes” 时，如果未设置端口值，则会自动为端口选取值 9543。
reconnect	强制重新连接到服务器的时间（以分钟为单位）。默认值为 30。

```
[server]
hostname=LOGINSIGHT
; Hostname or IP address of your Log Insight server / cluster load balancer. Default:
;hostname=LOGINSIGHT

; Protocol can be cfapi (Log Insight REST API), syslog. Default:
;proto=cfapi

; Log Insight server port to connect to. Default ports for protocols (all TCP):
; syslog: 514; syslog with ssl: 6514; cfapi: 9000; cfapi with ssl: 9543. Default:
;port=9543

; SSL usage. Default:
;ssl=yes
```

3 保存并关闭 `liagent.ini` 文件。

示例

以下配置示例将设置一个使用受信证书颁发机构的目标 vRealize Log Insight 服务器。

```
[server]
proto=cfapi
hostname=LOGINSIGHT
port=9543
ssl=yes;
ssl_ca_path=/etc/pki/tls/certs/ca.pem
```

以下示例显示了一个多目标配置。

- 第一个（默认）目标接收所有收集的事件。

```
[server]
hostname=prod1.licf.vmware.com
```

- 第二个目标通过纯 syslog 协议仅接收 syslog 事件。

```
[server|syslog-audit]
hostname=third_party_audit_management.eng.vmware.com
proto=syslog
ssl=no
filter= {filelog; syslog; }
```

- 第三个目标接收 vRealize Operations Manager 事件，但前提是这些事件具有等同于“错误”或“警告”的级别字段，并且由名称以“vrops-”开头的部分收集。

```
[server|licf-prod1]
hostname=vrops-errors.licf.vmware.com
filter= {; vrops-.*; level == "error" || level == "warning"}

;Collecting syslog messages.
[filelog|syslog]
directory=/var/log
include=messages

;various vRops logs. Note that all section names begin with "vrops-" prefix, which is used in
third destination filter.
[filelog|vrops-ANALYTICS-analytics]
directory=/data/vcops/log
include=analytics*.log*
exclude=analytics*-gc.log*
parser=auto
[filelog|vrops-COLLECTOR-collector]
directory=/data/vcops/log
include=collector.log*
event_marker=^\d
{4}-\d{2}-\d{2}[\s]\d{2}:\d{2}:\d{2}\,\d{3}
parser=auto
[filelog|vrops-COLLECTOR-collector_wrapper]
directory=/data/vcops/log
```

```
include=collector-wrapper.log*
event_marker=^\d{4}
-\d
{2}-\d{2}
[\s]\d
{2}:\d{2}
:\d
{2}
\.\d
{3}
parser=auto
```

后续步骤

您可以为 vRealize Log InsightLinux 代理配置其他 SSL 选项。请参见在[服务器和 Log Insight 代理之间配置 SSL 连接](#)。

vRealize Log Insight 代理的集中式配置

您可以配置多个 vRealize Log Insight 代理。

每个 vRealize Log Insight 代理都具有一个本地配置和一个服务器端配置。本地配置存储在安装有 vRealize Log Insight 代理的虚拟机或物理机上的 liagent.ini 文件中。可以访问和编辑服务器端配置，例如，可在 Web 用户界面中从**管理 > 代理**执行此操作。每个 vRealize Log Insight 代理的配置由若干部分和键组成。键具有可配置的值。

vRealize Log Insight 代理会定期轮询 vRealize Log Insight 服务器并接收服务器端配置。服务器端配置和本地配置进行合并，将得到有效的配置。每个 vRealize Log Insight 代理都使用有效配置作为其工作配置。配置将按逐个部分和逐个键进行合并。服务器端配置中的值将覆盖本地配置中的值。合并规则如下：

- 如果某个部分只存在于本地配置中，或只存在于服务器端配置中，则该部分及其所有内容将成为有效配置的一部分。
- 如果某个部分同时存在于本地配置和服务器端配置中，则该部分中的键将按以下规则合并：
 - 如果某个键只存在于本地配置中，或只存在于服务器端配置中，则该键及其值将成为有效配置中该部分的一部分。
 - 如果某个键同时存在于本地配置和服务器端配置中，则该键将成为有效配置中该部分的一部分，并且系统会使用服务器端配置中的值。

vRealize Log Insight 管理员用户可以将集中式配置应用于所有 vRealize Log Insight 代理。例如，您可以导航到**管理**页面，然后在**管理**部分中单击**代理**。在**代理配置**框中输入配置设置，然后单击**保存所有代理的配置**。配置将在下一个轮询周期内应用于所有可配置的活动代理。

vRealize Log Insight 管理员用户还可以在代理组中使用特定的筛选器（例如，按操作系统、代理版本、主机名或 IP 范围），并将配置应用于特定的 vRealize Log Insight 代理。有关代理组的信息，请参见《使用代理组》。

注

- 只能将集中式配置应用于采用 cfapi 协议的 vRealize Log Insight 代理。
- 在以下任一情况下，vRealize Log Insight 代理不可配置：
 - 当前 vRealize Log Insight 服务器不是主目标。有关配置多个目标的信息，请参见[指定代理的目标](#)。
 - 在代理配置中使用参数 `central_config = no`。有关适用于 Windows 的默认代理配置的信息，请参见 [Log Insight Windows Agent 的默认配置](#)。

配置合并示例

Log Insight Windows Agent 本地和服务器端配置合并的示例。

本地配置

您可以进行如下的 Log Insight Windows Agent 本地配置。

```
[server]
proto=cfapi
hostname=HOST
port=9000

[winlog|Application]
channel=Application

[winlog|Security]
channel=Security

[winlog|System]
channel=System

[filelog|ApacheAccessLogs]
enabled=yes
directory=C:\Program Files (x86)\Apache Software Foundation\Apache2.2\logs
include=*.log
exclude=*_old.log
event_marker=^(\d{1,3}\.){3}\d{1,3} - -
```

服务器端配置

您可以使用 Web 用户界面的[管理 > 代理](#)页面，将集中配置应用于所有代理。例如，您可以排除和添加收集通道，并更改默认重新连接设置。

```
[server]
reconnect=20
```

```
[winlog|Security]
channel=Security
enabled=no

[winlog|Microsoft-Windows-DeviceSetupManagerOperational]
channel=Microsoft-Windows-DeviceSetupManager/Operational
```

有效配置

合并本地和服务端配置即可生成有效配置。Log Insight Windows Agent 配置为:

- 每 20 分钟重新连接到 vRealize Log Insight 服务器
- 继续收集应用程序和系统事件通道
- 停止收集安全事件通道
- 开始收集 Microsoft-Windows-DeviceSetupManager/Operational 事件通道
- 继续收集 ApacheAccessLogs

```
[server]
proto=cfapi
hostname=HOST
port=9000
reconnect=20

[winlog|Application]
channel=Application

[winlog|Security]
channel=Security
enabled=no

[winlog|System]
channel=System

[winlog|Microsoft-Windows-DeviceSetupManagerOperational]
channel=Microsoft-Windows-DeviceSetupManager/Operational

[filelog|ApacheAccessLogs]
enabled=yes
directory=C:\Program Files (x86)\Apache Software Foundation\Apache2.2\logs
include=*.log
exclude=*_old.log
event_marker=^(\d{1,3}\.){3}\d{1,3} - -
```


使用公用值进行代理配置

您可以使用适用于 Windows 或 Linux 代理的每个代理配置部分的公用参数值覆盖代理配置文件的默认值。

公用选项

在 `liagent.ini` 配置文件的 `[common|global]` 部分中指定的选项将传播到所有部分，在 `[common|filelog]` 部分中指定的选项仅传播到所有 `filelog` 部分，而 `[common|winlog]` 选项仅传播到所有 `winlog` 部分。

您可以在公用部分中定义以下参数：`tags`、`include`、`exclude`、`event_marker`、`charset`、`exclude_fields` 和 `parser`，如以下示例中所示。该示例适用于 Windows 代理：

```
[common|global]

tags = {"log_source_vm":"win-2008r2-64"}
exclude_fields = test_tag;some_other_tag
parser = auto

[common|filelog]
tags = {"collector_type":"filelog"}
exclude = *.trc

[filelog|channel_1]
directory = C:\app\log
include = *.log

...
```

此示例指定以下行为：

- `filelog` 部分中的所有日志都具有 `log_source_vm` 和 `collector_type` 标记及其对应的值。
- 发送的所有日志中将不包含 `test_tag` 和 `some_other_tag` 标记。
- 对收集的所有日志应用 `auto` 分析程序。
- 默认情况下，所有 `filelog` 收集器不对 `*.trc` 文件进行监控。

`[common|global]` 中的选项也会应用于所有的 `winlog` 部分。

合并和覆盖条件

如果在多个部分中定义了选项，则会合并或覆盖其值，在合并/覆盖时，具有较小范围的部分具有较高优先级。即，`[common|global]` 中的值会与 `[common|filelog]` 中的值合并或被其覆盖，而后者又会与 `[filelog|sample_section]` 中的值合并或被其覆盖。

合并和覆盖行为符合以下规则：

- 其值表示值列表的选项（`tags`、`include`、`exclude` 和 `exclude_fields`）会与该选项在优先级较高的部分中的值合并。对于标记，如前面所述，优先级较高的部分中的标记值会覆盖同一标记在优先级较低的部分中的值。

- 可以有单个值的选项（event_marker、charset 和 parser）的值会被该选项在优先级较高的部分中的值覆盖。

这意味着，[filelog|sample_section] 中 charset=UTF-8 的值会覆盖 [common|global] 中 charset=UTF-16LE 的全局值。

例如，如果您在 [common|filelog] 中具有 tags={"app":"global-test"}，在 [filelog|flg_test_section] 中具有 tags={"app":"local-test","section":"flg_test_section"}，则 [filelog|flg_test_section] 部分中 "app" 标记的值会覆盖 [common|filelog] 中的值。通过此 filelog 部分收集的所有日志都将具有一个额外的 "app" 标记和对应的 "local-test" 值，以及 "section" 标记和对应的 "flg_test_section" 值。对于 winlog 部分，将采用同样的优先级链，所有 [winlog|...] 部分具有最高优先级，而 [common|global] 则具有最低优先级。

当在公用部分中指定了无效值时，通常会跳过这些值，并且不会将其与对应的优先级较高的 filelog/winlog 部分中的值合并。如果标记或 exclude_fields 选项中存在无效值，代理会提取尽可能多的有效数据，并在遇到无效数据后跳过该文件的剩余部分。所有异常会在代理日志文件中报告出来。请在遇到异常行为时查阅日志文件，并修复代理报告的所有错误。

如果代理在 filelog 或 winlog 部分中检测到某个选项存在无效值，则它不会将该部分中的选项值与公用部分中的选项值合并，也不会启用该部分。所有错误会在代理日志文件中报告出来。请在遇到异常行为时查阅日志文件，并修复代理报告的所有错误。

分析日志

代理端日志分析程序从原始日志中提取结构化数据，然后传送到 vRealize Log Insight 服务器。使用日志分析程序，vRealize Log Insight 可以分析日志，从日志中提取信息，并在服务器上显示结果。对于 Windows 和 Linux vRealize Log Insight 代理，均可配置日志分析程序。

如果 syslog 协议已被使用，那么根据 RFC5424，由分析程序提取的字段属于 STRUCTURED-DATA 的一部分。

配置日志分析程序

可以同时为 FileLog 和 WinLog 收集器配置分析程序。

前提条件

对于 vRealize Log Insight Linux 代理：

- 以 root 用户身份登录，或使用 sudo 运行控制台命令。
- 登录到已安装 Log Insight Linux 代理的 Linux 计算机，打开控制台，然后运行 pgrep liagent 以验证是否已安装且正在运行 Log Insight Linux 代理。

对于 vRealize Log Insight Windows 代理：

- 登录到已安装 Log Insight Windows 代理的 Windows 计算机，启动“服务”管理器以验证是否已安装 vRealize Log Insight 服务。

步骤

- 1 导航到包含 `liagent.ini` 文件的文件夹。

操作系统	路径
Linux	<code>/var/lib/loginsight-agent/</code>
Windows	<code>%ProgramData%\VMware\Log Insight Agent</code>

- 2 在任意文本编辑器中打开 `liagent.ini` 文件。
- 3 要配置特定的分析程序，请定义一个分析程序部分。`[parser|myparser]`

其中 `myparser` 是分析程序的任意名称，可从日志源引用。分析程序部分应该引用任何内置的（或者任何其他已定义的）分析程序，并根据需要配置该分析程序的必备选项和非必需选项。

例如，`base_parser=csv` 显示 `myparser` 分析程序派生自内置分析程序 `csv`。它期望输入日志包含两个用分号分隔的字段。

```
[parser|myparser]

base_parser=csv

fields=field_name1,field_name2

delimiter=";"
```

- 4 在定义 `myparser` 后，从日志源 `winlog` 或 `filelog` 引用它。

```
[filelog|some_csv_logs]

directory=D:\Logs

include=*.txt;*.txt.*

parser=myparser
```

从 `some_csv_logs` 源（例如，从 `D:\Logs` 目录）收集的日志由 `myparser` 分析，而且提取的事件将分别以 `field_name1` 和 `field_name2` 显示在服务器上。

注 代理不会将 `D:\Logs` 目录中的静态日志推送到 vRealize Log Insight 中。但是，在 `D:\Logs` 目录中创建的新文件在 vRealize Log Insight 中可用。

- 5 保存并关闭 `liagent.ini` 文件。

分析程序的常见选项

可以为生成命名字段的所有分析程序配置常见选项。

字段名称的保留字

字段名称受到一定的限制。以下名称是保留名称，因此不能用作字段名称。

- event_type
- hostname
- source
- text

常见分析程序选项

下表中的选项可与所有受支持的分析程序一起使用。

选项	描述
base_parser	此自定义分析程序扩展的基本分析程序的名称。它可以是内置分析程序名称或其他自定义分析程序名称。此配置键是必备键。
field_decoder	<p>指定为 JSON 字符串的嵌套分析程序。键是嵌套分析程序要应用到的字段的名称，值是用于该字段的分析程序的名称。每个嵌套分析程序将应用于由基本分析程序解码的相应字段。字段的值是复杂值（例如时间戳）时，字段解码器很有用。</p> <p>field_decoder 选项还支持将更复杂的 JSON 对象作为参数，以允许您在应用嵌套分析程序之前检查的特定字段值中使用条件。</p> <p>注 有关用法和条件配置的详细信息，请参见下面的“field_decoder 选项的条件配置”部分。</p>
field_rename	重命名提取的字段。使用一个 JSON 字符串，其中键是字段的原始名称，值是字段所需的新名称。field_decoder 选项将始终在 field_rename 之前应用。这些选项在 INI 文件中的顺序并不重要。为明确起见，请先指定 field_decoder。
next_parser	<p>要运行的下一个分析程序的名称。允许对同一输入按顺序运行多个分析程序。</p> <p>注 分析程序处理由 next_parser 关键字定义的所有后续分析程序，并且可能会替换由上一个分析程序提取的字段值。</p>
exclude_fields	要在将事件传送到服务器之前从其中删除的以分号分隔的字段名称列表。在执行事件筛选之前移除字段名称，从而使解析期间排除的字段无法在筛选条件中使用。
debug	<p>用于启用特定分析程序的调试的 Yes 或 No 选项。启用调试后，分析程序将执行它接收的输入、它执行的操作以及它产生的结果的详细日志记录。此选项按部分应用，即仅应用于特定部分定义的分析程序。</p> <p>对于分析程序，调试的默认值是 debug=no。</p>

field_decoder 选项的条件配置

对于具有相同通用格式但存在与特定字段值相关的明显差异的日志（例如，具有 **info** 和 **error** 严重性的日志），您可以使用条件嵌套分析程序避免将不需要的分析程序应用于已分析的日志的相应字段。

例如，使用以下日志：

```
2019-03-29T11:00:54.858Z host-FQDN Hostd: error hostd[2099230] [Originator@6876 sub=Default
opID=1983bdbe-cl-800f user=admin.user] AdapterServer caught exception: SSLExceptionE(SSL
Exception: error:140000DB:SSL routines:SSL routines:short read: The connection was closed by
the remote end during handshake.)
```

```
2019-03-29T11:00:55.477Z host-FQDN Hostd: info hostd[6D620B70] ['commonhost' opID=5759adcc-
cf] [transportConnector] -- FINISH task-internal-5726666 -- -- Completed connection restart --
```

您可以使用以下配置对其进行分析：

```
[parser|clf_parser]
base_parser=clf
format=%t %{generator_host}i %i: %{log_severity}i %i[%{thread_id}i]%M
field_decoder={"log_message" : {"log_severity" : {"error" : "error_parser", "info" :
"info_parser"}}}
exclude_fields=log_message

[parser|info_parser]
base_parser=clf
format=[%{common_info}i] [%{process}i] %M
field_rename={"log_message" : "info_log_content"}

[parser|error_parser]
base_parser=clfformat=[%{common_info}i] %{exception_handler}i %i:%{exception_type}i:%i:%
{error_id}i:%i:%i:%i: %M
field_rename={"log_message" : "exception_content"}
```

该配置生成以下结果：

```
timestamp=2019-03-29T11:00:54.858000 generator_host="host-FQDN" log_severity="error"
thread_id="2099230" common_info=Originator@6876 sub=Default opID=1983bdbe-cl-800f
user=admin.user exception_handler="AdapterServer" exception_type="SSLExceptionE(SSL
Exception" error_id="140000DB" exception_content="The connection was closed by the remote end
during handshake.)"
```

此外，还会为 **info** 日志分析以下字段：

```
timestamp=2019-03-29T11:00:55.477000 generator_host="host-FQDN" log_severity="info"
thread_id="6D620B70" log_message="['commonhost' opID=5759adcc-cf] [transportConnector] --
FINISH task-internal-5726666 -- -- Completed connection restart --" common_info="'commonhost'
opID=5759adcc-cf" process="transportConnector" info_log_content="-- FINISH task-
internal-5726666 -- -- Completed connection restart --"
```

逗号分隔值日志分析程序

您可以针对 FileLog 和 WinLog 收集器配置逗号分隔值 (CSV) 分析程序。

针对 csv 分析程序的可用选项是 `fields` 和 `delimiter`。

逗号分隔值分析程序选项

请注意有关 csv 分析程序结构的下列信息。

选项	描述
<code>fields</code>	<p><code>fields</code> 选项指定日志中存在的字段的名称。列出的字段名称的总数必须等于日志中逗号分隔字段的总数。</p> <p><code>fields</code> 选项是 CSV 分析程序的必备选项。如果未指定该选项，则无法进行任何分析。字段值两边的双引号是可选的，具体取决于字段内容。</p> <p>字段名称必须用逗号分隔，例如</p> <pre>fields = field_name1, field_name2, field_name3, field_name4</pre> <p>此定义假定名称 <code>field_name1</code>、<code>field_name2</code>、<code>field_name3</code> 和 <code>field_name4</code> 按顺序分配给提取的字段。</p> <p>如果 CSV 分析程序必须省略某些字段，则可以从列表中省略这些字段的名称。例如，</p> <pre>fields = field_name1, , field_name3, field_name4</pre> <p>在这种情况下，分析程序仅从事件中提取第一个、第三个和第四个字段，并将名称 <code>field_name1</code>、<code>field_name3</code> 和 <code>field_name4</code> 按顺序分配给它们。</p> <p>如果 <code>fields</code> 选项未指定日志中的完整字段列表，则分析程序将返回空列表。例如，如果日志文件包含 <code>field1</code>、<code>field2</code>、<code>field3</code>、<code>field4</code> 和 <code>field5</code>，但是仅指定了 <code>fields= field1,field2,field3</code>，则分析程序将返回空字段列表。</p> <p>不能将 <code>fields=*</code> 用于 CSV 分析程序，因为分析程序会返回空字段列表。除非像上述那样需要省略某些字段，否则必须指定完整的字段列表。</p>
<code>delimiter</code>	<p><code>delimiter</code> 选项指定分析程序要使用的分隔符。默认情况下，csv 分析程序使用逗号作为分隔符，但是，您可以将分隔符更改为分号、空格或者其他特殊字符。必须用双引号将定义的分隔符括起来。</p> <p>例如，<code>delimiter=","</code> 和 <code>delimiter=";"</code>。</p> <p>csv 分析程序支持任何字符集作为分隔符（用双引号括起来），例如 <code>" "</code> 或 <code>"asd"</code>。日志中字段值的分隔符必须与分隔符参数定义的模式完全匹配，否则，分析程序将失败。</p> <p>只要 <code>\</code>、<code>\s</code>、<code>\t</code> 优先被用作转义符而非特殊字符，则可定义如空格或 <code>Tab</code> 之类的特殊字符为 csv 分析程序的分隔符。例如，<code>delimiter="\s"</code> 或 <code>delimiter=" "</code>。</p> <p><code>delimiter</code> 为可选选项。</p>

CSV 日志分析程序配置

要分析从 winlog 或 filelog 源收集的日志，请使用以下配置。

```
[filelog|some_csv_logs]
directory=D:\Logs
include=*.txt;*.txt.*
parser=myparser

[parser|myparser]
base_parser = csv
fields = timestamp,field_name1, field_name2, field_name3
```

```

delimiter = ";"
field_decoder={"timestamp": "tsp_parser"}
[parser|tsp_parser]
; timestamp is a built-in parser
base_parser=timestamp
; "format" is an option of timestamp parser
format=%Y-%m-%d %H:%M:%S

```

对于此配置，从 `some_csv_logs` 源（例如，从 `directory=D:\Logs` 目录）收集的日志由 `myparser` 进行分析。如果收集的日志包含三个用分号分隔的值，则分析的事件将按顺序收到 `field_name1`、`field_name2` 和 `field_name3` 名称。

要解析以下 CSV 日志，请执行以下操作：

```

"United States","USA","North America","High income: OECD","Fiscal year end: September 30;
reporting period for national accounts data: CY."

```

定义 CSV 分析程序配置：

```

[parser|csv_log_parser]
base_parser=csv
fields=country_name, country_code, region, income_group, special_notes

```

CSV 分析程序返回以下字段：

```

country_name=United States
country_code=USA
region=North America
income_group=High income: OECD
special_notes=Fiscal year end: September 30; reporting period for national accounts data: CY.

```

常见日志格式 (Apache) 日志分析程序

您可以为 FileLog 和 WinLog 收集器配置常见日志格式 (CLF) Apache 分析程序。

常见日志格式 (Apache) 分析程序

默认 CLF 分析程序会定义字段的以下顺序和名称。

```

host ident authuser datetime request statuscode bytes

```

分析程序名称：`clf`

CLF 分析程序特定的选项是 `format`。

format 选项

`format` 选项指定生成 Apache 日志所用的格式。该选项不是必备选项。

如果未指定格式，则会使用以下默认的常见日志格式。

```

%h %l %u %t \"%r\" %s %b

```

CLF 分析程序格式字符串不接受正则表达式。例如，指定空格而不是表达式 `\s+`。

要分析其他日志格式，请在代理的配置中指定该格式。服务器端会显示以下名称的已分析字段。

注 在需要变量的情况下，如果配置中未提供 {VARNAME}，则会忽略这些字段。

字段	值
'%a':	"remote_ip"
'%A':	"local_ip"
'%B', '%b':	"response_size"
'%C':	取决于在格式中指定的变量名称
'%c':	取决于在格式中指定的变量名称
'%D':	"request_time_mcs"
'%E':	"error_status"
'%e':	取决于在格式中指定的变量名称
'%F', '%f':	"file_name"
'%h':	"remote_host"
'%H':	"request_protocol"
'%i':	取决于在格式中指定的变量名称
'%k':	"keepalive_request_count"
'%l':	"remote_log_name"
'%L'	"request_log_id"
'%M':	"log_message" (分析程序在到达此说明符后停止分析输入日志)
'%m':	"request_method"
'%n':	取决于在格式中指定的变量名称
'%o':	取决于在格式中指定的变量名称
'%p':	"server_port" 在以下说明符中可以使用一些额外的格式: %(format)p。支持的格式包括 "canonical"、"local" 或 "remote"。如果使用 "canonical" 格式，字段名称保留为 "server_port"；如果使用 "local" 格式，字段名称将为 "local_server_port"；如果使用 "remote" 格式，字段名称将为 "remote_server_port"。
'%P':	"process_id" 在以下说明符中可以使用一些额外的格式: %(format)P。支持的格式包括 "pid"、"tid" 和 "hexid"。如果使用 "pid" 格式，字段名称将为 "process_id"；而 "tid" 和 "hexid" 格式将生成名为 "thread_id" 的字段
'%q':	"query_string"

字段	值
'%r':	"request"
'%R':	"response_handler"
'%s':	"status_code", 生成请求的最终状态。
'%t':	<p>"timestamp", 作为载入事件的时间戳, 参与时间戳分析程序。要覆盖时间戳自动检测, 可以用大括号指定日期和时间格式: <code>%{Y-%m-%d %H:%M:%S}t</code>, 请参见时间戳分析程序以了解更多信息。</p> <p>CLF 分析程序的时间戳格式可以采用 "begin:" 或 "end:" 前缀开头。如果格式以 begin: 开头 (默认), 则时间为请求处理开始时所用的时间。如果格式以 end: 开头, 则时间为写入日志条目的时间, 此时请求处理临近结束。例如, CLF 分析程序支持如下格式: <code>%h %l %u [%{begin:%d/%b/%Y %T}t.%{msec_frac}t] \"%r\" %>s %b</code></p> <p>CLF 分析程序的时间戳格式说明符还支持以下格式标记:</p> <p>sec</p> <p>自新纪元时间以来的秒数。这等同于时间戳分析程序的 <code>%s</code> 说明符。</p> <p>msec</p> <p>自新纪元时间以来的毫秒数</p> <p>usec</p> <p>自新纪元时间以来的微秒数</p> <p>msec_frac</p> <p>毫秒分数 (等同于时间戳分析程序的 <code>%f</code> 说明符)</p> <p>usec_frac</p> <p>微秒分数 (等同于时间戳分析程序的 <code>%f</code> 说明符)</p> <p>要分析其时间戳以格式标记显示的日志, 可以在配置中使用以下格式:</p> <pre>format=%h %l %u %{sec}t \"%r\" %s %b format=%h %l %u %{msec}t \"%r\" %s %b format=%h %l %u %{usec}t \"%r\" %s %b</pre> <p>这些标记不能彼此组合使用, 也不能与采用相同格式字符串的时间戳分析程序格式组合使用。您可以改用多个 <code>%{format}t</code> 标记。例如, 要使用包含毫秒的时间戳, 可以使用以下组合的时间戳 (时间戳分析程序的 <code>%f</code> 说明符除外): <code>%{%d/%b/%Y %T}t.%{msec_frac}t</code>。</p>
'%T':	"request_time_sec"
'%u':	"remote_auth_user"
'%U':	"requested_url"
'%v':	"server_name"
'%V':	"self_referential_server_name"
'%X':	"connection_status", 取决于在格式中指定的变量名称
'%x':	取决于在格式中指定的变量名称

字段	值
'%I':	"received_bytes"
'%O':	"sent_bytes"
'%S':	"transferred_size"

例如，要用 CLF 分析程序分析从 winlog 或 filelog 源收集的日志，请指定以下配置：

```
[filelog|clflogs]
directory=D:\Logs
include=*.txt
parser=myclf

[parser|myclf]
debug=yes ;Note: use this option only while debugging and set it to 'no' when used in
production.
base_parser=clf
format=%h %l %u %b %t \"%r\" %s
```

使用此配置，从 clflogs 源（例如，从 directory=D:\Logs 目录）收集的日志由 myclf 分析。myclf 分析程序仅分析以配置中描述的格式生成的那些日志。

对于分析程序，调试的默认值是 debug=no。

分析使用 CLF 生成的日志

要分析使用 CLF 生成的日志，必须在配置中定义相应的格式。例如，

```
format=%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User_Agent}i\"
```

使用说明符 %{Referer}i 和 %{User_Agent}i 的非空字段将分别以名称 referer 和 user_agent 显示在 vRealize Log Insight 服务器上。

将时间戳分析程序与 CLF 分析程序集成

可以分析具有自定义时间格式的 Apache 日志。

访问具有如下所示的自定义时间格式的日志。

```
format = %h %l %u %{a, %d %b %Y %H:%M:%S}t \"%r\" %>s %b
```

如果未指定自定义时间，则 CLF 分析程序将尝试通过运行自动时间戳分析程序来自动推断时间格式，否则将使用自定义时间格式。

错误日志所支持的自定义时间格式如下：

自定义时间格式	描述	配置格式
%{u}t	包括微秒的当前时间	format=[%{u}t] [%l] [pid %P] [client %a] %M
%{cu}t	采用精简 ISO 8601 格式的当前时间，包括微秒	format=[%{cu}t] [%l] [pid %P] [client %a] %M

有关支持的时间戳说明符的完整列表，请参见[时间戳分析程序](#)。

示例：用于 Windows 的 Apache 默认访问日志配置

示例：用于 Windows 的 Apache 默认错误日志配置

此示例说明如何为用于 Windows 的 Apache v2.4 访问日志配置设置格式。

```
;ACCESS LOG
;127.0.0.1 - - [13/May/2015:14:44:05 +0400] "GET /xampp/navi.php HTTP/1.1" 200 4023 "http://localhost/xampp/" "Mozilla/5.0 (Windows NT 6.1; WOW64; rv:37.0) Gecko/20100101 Firefox/37.0"
;format=%h %l %u %{d/%b/%Y:%H:%M:%S %z}t \"%r\" %>s %b \"%{Referer}i\" \"%{User_agent}i\"

; Section to collect Apache ACCESS logs
[filelog|clflogs-access]
    directory=C:\xampp\apache\logs
    include=acc*
    parser=clfparsers_apache_access
    enabled=yes

;Parser to parse Apache ACCESS logs
[parser|clfparsers_apache_access]
    debug=yes
    base_parser=clf
    format=%h %l %u %{d/%b/%Y:%H:%M:%S %z}t \"%r\" %>s %b \"%{Referer}i\" \"%{User_agent}i\"
```

定义访问日志格式：

1 为访问日志格式 (httpd.conf) 配置 Apache：

```
LogFormat "%h %l %u %{d-%b-%Y:%H:%M:%S}t \"%r\" %a %A %e %k %l %L %m %n %T %v %V %>s %b \"%{Referer}i\" \"%{User-Agent}i\" combined"
```

2 定义 CLF 分析程序配置：

```
;ACCESS LOG
;127.0.0.1 unknown - 21-May-2015:13:59:35 "GET /xampp/navi.php HTTP/1.1" 127.0.0.1 127.0.0.1
- 0 unknown - GET - 1 localhost localhost 200 4023 "http://localhost/xampp/" "-"
[filelog|clflogs-access]
    directory=C:\xampp\apache\logs
    include=acc*,_myAcc*
    parser=clfparsers_apache_access
    enabled=yes
; Parser to parse Apache ACCESS logs
[parser|clfparsers_apache_access]
    debug=yes
    base_parser=clf
    format=%h %l %u %{d-%b-%Y:%H:%M:%S}t \"%r\" %a %A %e %k %l %L %m %n %T %v %V %>s %b \"%{Referer}i\" \"%{User-Agent}i\"
```

CLF 分析程序返回以下结果：

```
remote_host=127.0.0.1
timestamp=2015-05-13T10:44:05
request=GET /xampp/navi.php HTTP/1.1
status_code=200
response_size=4023
referer=http://localhost/xampp/
user_agent=Mozilla/5.0 (Windows NT 6.1; WOW64; rv:37.0) Gecko/20100101 Firefox/37.0
```

此示例说明如何为用于 Windows 的 Apache v2.4 错误日志配置设置格式。

```
;ERROR LOG
;[Wed May 13 14:37:17.042371 2015] [mpm_winnt:notice] [pid 4488:tid 272] AH00354: Child:
Starting 150 worker threads.
;[Wed May 13 14:37:27.042371 2015] [mpm_winnt:notice] [pid 5288] AH00418: Parent: Created
child process 3480
;format=[%{a %b %d %H:%M:%S%f %Y}t] [%m:%{severity}i] [pid %P:tid %{thread_id}i] %E: %M
;format=[%{a %b %d %H:%M:%S%f %Y}t] [%m:%{severity}i] [pid %P] %E: %M

; Section to collect Apache ERROR logs
[filelog|clflogs-error]
    directory=C:\xampp\apache\logs
    include=err*
    parser=clfparser_apache_error
    enabled=yes

;Parser to parse Apache ERROR logs
[parser|clfparser_apache_error]
    debug=yes
    base_parser=clf
    format=[%{a %b %d %H:%M:%S%f %Y}t] [%m:%{severity}i] [pid %P:tid %{thread_id}i] %E: %M
    next_parser=clfparser_apache_error2

;Parser to parse Apache ERROR logs
[parser|clfparser_apache_error2]
    debug=yes
```

```
base_parser=clf
format=[%{ %a %b %d %H:%M:%S%f %Y)t] [%m:%{severity}i] [pid %P] %E: %M
```

注 提供的名称与组合日志格式相对应。另外还介绍了使用上述格式设置键（而不是 Apache 错误日志格式）的 Apache 错误日志。

定义错误日志格式：

1 为错误日志格式 (httpd.conf) 配置 Apache：

```
LogFormat "%h %l %u {%d-%b-%Y:%H:%M:%S}t \"%r\" %a %A %e %k %l %L %m %n %T %v %V %>s %b\n\"%{Referer}i\" \"%{User-Agent}i\" combined"
```

2 定义 CLF 分析程序配置：

```
;Parser to parse Apache ERROR logs
[parser|clfparser_apache_error]
  debug=yes
  base_parser=clf
  format=[%{a %b %d %H:%M:%S%f %Y}t] [%m:%{severity}i] [pid %P] %E: %M
  next_parser=clfparser_apache_error2

;Parser to parse Apache ERROR logs
[parser|clfparser_apache_error2]
  debug=yes
  base_parser=clf
  format=[%{a %b %d %H:%M:%S%f %Y}t] [%m:%{severity}i] [pid %P:tid %{thread_id}i] %E: %M
```

日志条目：

```
[Wed May 13 14:37:17.042371 2015] [mpm_winnt:notice] [pid 4488:tid 272] AH00354: Child:
Starting 150 worker threads.
```

CLF 分析程序针对日志条目返回以下字段（如果使用 +0400 时区的分析程序）：

```
timestamp=2015-05-13T10:37:17.042371
request_method=mpm_winnt
severity=notice
process_id=4488
thread_id=272
error_status=AH00354
log_message=Child: Starting 150 worker threads.
```

日志条目：

```
[Wed May 13 14:37:27.042371 2015] [mpm_winnt:notice] [pid 5288] AH00418: Parent: Created
child process 3480
```

CLF 分析程序针对日志条目返回以下字段（如果使用 +0400 时区的分析程序）：

```
timestamp=2015-05-13T10:37:27.042371
request_method=mpm_winnt
severity=notice
process_id=5288
error_status=AH00418
log_message=Parent: Created child process 3480
```

键/值对分析程序

您可以为 FileLog 和 WinLog 收集器配置键/值对 (Key/Value Pair, KVP) 分析程序。

键/值对 (KVP) 分析程序

kvp 分析程序从任意日志消息文本中查找和提取所有 key=value 匹配项。以下示例显示了 kvp 分析程序格式。

```
[parser|kvp_parser]
base_parser=kvp
fields=*
```

例如，键值日志的格式可以为：scope=local; abstract=false; lazyInit=false; autowireMode=0; dependencyCheck=0;

在 kvp 分析程序中，您必须指定要从其中提取值的字段。例如，如果配置中存在定义 fields=name,lastname,country，则仅会分析指定键的值并将其发送到服务器。

可以选择使用双引号 “ ” 将键和值括起来以定义空格或其他特殊字符。

使用双引号将键或值括起来时，反斜线字符 “\” 可以用作转义符。反斜线字符后的任何字符按字面定义，其中包括双引号字符或反斜线字符。例如：“\\”

记住以下注意事项。

- 如果键/值对的键后面未跟等号且未提供 VALUE，将跳过该选项，与自定义文本一样。
- 键不能为空，值可以为空。
- 后面未跟值的等号视为自定义文本，将被跳过。
- 值可以用双引号括起的字符串，也可以为空。使用反斜线对属于值的特殊字符进行转义。

KVP 分析程序选项

请注意关于 kvp 分析程序结构的下列信息。

选项	描述
fields	<p>要提取的描述为数据单位的信息。例如 <code>fields=name,lastname,country</code>。</p> <p>如果使用 <code>fields</code> 选项来定义特定的字段名称，则从日志中提取的字段名称中的每个无效字符均会被替换为下划线。例如，如果 <code>fields</code> 选项查找字段 “x-A” 和 “a*(X+Y)”，则分析程序会从日志中提取这些字段，并将它们分别重命名为 “x_a” 和 “a__x_y” 字段。这样可以提取名称中包含任何字符的字段。</p> <p>如果将 <code>fields</code> 选项指定为 “*”（这意味着 <code>kvp</code> 分析程序自动识别字段/值对），则分析程序会查找仅具有 “字母数字+下划线” 字符（受 LI 服务器支持）的字段。所有其他无效字符均会被丢弃，而不是转换为下划线。这样可以防止分析程序将不必要的信息提取到静态字段中。</p>
delimiter	<p>可选。</p> <p>默认分隔符包括空格字符、制表符、换行符、逗号和分号字符。</p> <p>如果在配置中未指定任何分隔符，<code>kvp</code> 分析程序将使用默认的分隔符进行分析。</p> <p>要将默认分隔符更改为特定分隔符，您必须用双引号将特定分隔符括起来进行定义。例如：<code>delimiter = "#^ "</code>。此定义表示将括在双引号中的每个字符用作分隔符。对于 <code>kvp</code> 分析程序，可使用任何字符作为分隔符。您可以在该定义中加入默认分隔符与其他分隔符。</p> <p>例如，<code>delimiter = "#^ \t\r\n\s"</code> 语句使用制表符、换行符和空格作为分隔符。如果使用这些字符作为分隔符，则必须在这些字符前面加上转义符。例如，要将空格定义为分隔符，在将其定义为分隔符时在空格字符前面输入转义符 “\”，例如，<code>delimiter="\s"</code>。</p>
field_decoder	<p>嵌套分析程序指定为 <code>JSON</code> 字符串，其中键为应用于嵌套分析程序的字段的名称，值为要用于该字段的分析程序的名称。</p> <p>每个嵌套分析程序按照基本分析程序的解码应用于相应的字段。</p> <p>如果键值对的值很复杂，例如时间戳或逗号分隔列表，那么字段解码器将很有用。</p>
debug =	<p>可选。 <code>debug =</code> 值可以是 <code>yes</code> 或 <code>no</code>。对于分析程序，调试的默认值是 <code>debug=no</code>。</p> <p>当选项设置为 <code>yes</code> 时，您可以在 <code>liagent_<date>.log</code> 中查看分析程序载入的详细日志。</p>

其他键值选项

键	定义
<code>KVP_MESSAGE = *(MESSAGE_ENTRY [WSPR])</code>	可选空格分隔的消息条目列表
<code>MESSAGE_ENTRY = KVP / FREE_TEXT</code>	条目是键/值对或者自定义文本
<code>KVP = KEY ["=" VALUE]</code>	键/值对。如果 <code>KEY</code> 后面未跟等号和 <code>VALUE</code> ，则将跳过该项，与自定义文本一样。
<code>KEY = BARE_KEY / QUOTED_KEY</code>	
<code>FREE_TEXT = "="</code>	独立的等号将视为自定义文本，将被跳过。
<code>BARE_KEY = *1BARE_KEY_CHAR</code>	至少一个字符
<code>BARE_KEY_CHAR = %0x00-08 / %0x10-19 / %0x21-3C / %3E-%FF</code>	除等号、空格或制表符之外的任何字符
<code>QUOTED_KEY = 0x22 *1(QUOTED_STRING_CHAR / "\" CHAR) 0x22</code>	用双引号字符括起至少一个字符。反斜线用作转义符。
<code>QUOTED_STRING_CHAR = %0x00-21 / %0x23-FF</code>	除双引号之外的任何字符

键	定义
VALUE = BARE_VALUE / QUOTED_VALUE	
BARE_VALUE = *BARE_VALUE_CHAR	零个或多个字符
BARE_VALUE_CHAR = %0x00-08 / %0x10-19 / %0x21-FF	除空格或制表符之外的任何字符
QUOTED_VALUE = 0x22 * (QUOTED_STRING_CHAR / "\" CHAR) 0x22	用双引号字符括起的字符串。可为空值。反斜线用作转义符。

KVP 分析程序配置示例

如有必要，您可以使用 `fields=*` 分析所有字段。

```
[parser|simple_kvp]
base_parser=kvp
fields=*
```

此示例说明如何指定字段解码器。

```
[parser|mykvp]
debug=no
base_parser=kvp
delimiter="#"^|"
fields=*
;OR fields=scope,abstract,lazyInit,autowireMode,dependencyCheck
field_decoder={"field1":"field1_parser1"}

[parser|field1_parser1]
base_parser=clf
format=[%{value1}i]]
field_decoder={"value1":"field_parser2"}
```

要解析以下 KVP 日志，请执行以下操作：

```
Configuring transport... proto = cfapi server_hostname = LOCALHOST ssl = no port = 9000
reconnect = 30
```

定义 KVP 分析程序配置：

```
[parser|kvp_log_parser]
base_parser=kvp
fields=*
```

KVP 分析程序返回以下字段：

```
proto=cfapi
server_hostname=LOCALHOST
ssl=no
port=9000
reconnect=30
```

示例：简单的和复杂的 KVP 分析程序示例

简单的 KVP 分析程序示例

```
[filelog|MyLog]
directory=C:\<folder_name>\Parser_logs
include=*.log
parser=my_KVP_parser

[parser|my_KVP_parser]
base_parser=kvp
fields=*
```

复杂的 KVP 分析程序示例

```
[filelog|MyLog]
directory=C:\<folder_name>\Parser_logs
include=*.log
parser=my_KVP_parser

[parser|my_KVP_parser]
base_parser=kvp
fields=*
field_decoder={"field1":" field1_parser1"}

[parser| field1_parser1]
base_parser=clf
format=[%{value1}i]]
field_decoder={"value1":" field1_parser2"}
```

时间戳分析程序

timestamp 分析程序不产生字段，而是将其输入从字符串形式转换为内部时间戳格式，该格式显示为从 UNIX 纪元开始时间（1970 年 1 月 1 日，UTC/GMT 午夜）起的毫秒数。

唯一受支持的配置选项是 format。例如 format=%Y-%m-%d %H:%M:%S。

与 CLF 分析程序不同，timestamp 分析程序可以在时间说明符之间没有分隔符时分析时间，例如 %A%B%d%H%M%S%Y%z。

timestamp 分析程序使用的格式说明符如下：

```
'%a':   Abbreviated weekday name, for example: Thu
'%A':   Full weekday name, for example: Thursday
'%b':   Abbreviated month name, for example: Aug
'%B':   Full month name, for example: August
```

```

'%d':    Day of the month, for example: 23. strftime() expects zero-padded (01-31) digits
         for this specifier but Log Insight agents can parse space-padded and non-padded
         day numbers, too.
'%e':    Day of the month, for example: 13. strftime() expects space-padded ( 1-31) digits
         for this specifier but Log Insight agents can parse zero-padded and non-padded
         day numbers too.
'%f':    Fractional seconds of time, for example: .036 'f' specifier assumes that '.' or ','
         character should exist before fractional seconds and there is no need to mention
         that character in the format. If none of these characters precedes fractional
seconds,
         timestamp wouldn't be parsed.
'%H':    Hour in 24h format (00-23), for example: 14. Zero-padded, space-padded, and non-
padded hours
         are supported.
'%I':    Hour in 12h format (01-12), for example: 02. Zero-padded, space-padded and non-
padded hours
         are supported.
'%m':    Month as a decimal number (01-12), for example: 08. Zero-padded, space-padded
         and non-padded month numbers are supported.
'%M':    Minute (00-59), for example: 55
'%p':    AM or PM designation, for example: PM
'%S':    Second (00-61), for example: 02
'%s':    Total number of seconds from the UNIX epoch start, for example 1457940799
         (represents '2016-03-14T07:33:19' timestamp)
'%Y':    Year, for example: 2001
'%z':    ISO 8601 offset from UTC in timezone (1 minute=1, 1 hour=100)., for example: +100

```

时间戳分析程序可接受其他说明符，但这些说明符的值都被忽略且不会影响分析时间。

```

'%C':    Year divided by 100 and truncated to integer (00-99), for example: 20
'%g':    Week-based year, last two digits (00-99), for example, 01
'%G':    Week-based year, for example, 2001
'%j':    Day of the year (001-366), for example: 235
'%u':    ISO 8601 weekday as number with Monday as 1 (1-7), for example: 4
'%U':    Week number with the first Sunday as the first day of week one (00-53), for example:
33
'%V':    ISO 8601 week number (00-53), for example: 34
'%w':    Weekday as a decimal number with Sunday as 0 (0-6), for example: 4
'%W':    Week number with the first Monday as the first day of week one (00-53), for example:
34
'%y':    Year, last two digits (00-99), for example: 01

```

如果未定义 format 参数，Timestamp 分析程序将使用默认格式分析时间戳。

自动时间戳分析程序

没有为时间戳分析程序定义格式时将调用自动时间戳分析程序，也可以在没有定义时间戳分析程序的情况下，通过在 `field_decoder` 中使用 `timestamp` 直接调用该分析程序。例如：

```
[parser|mycsv]
base_parser=csv
debug=yes
fields=timestamp,action,source_id,dest
field_decoder={"timestamp": "timestamp"}
```

示例：使用默认配置的时间戳分析程序

此示例显示了使用默认配置的 `timestamp` 分析程序。

```
[parser|tsp_parser]
base_parser=timestamp
debug=no
format=%Y-%m-%d %H:%M:%S%f
```

要将 `timestamp` 分析程序与其他分析程序（例如 `CSV` 分析程序）相集成，请指定以下配置。

```
[parser|mycsv]
base_parser=csv
fields=timestamp,action,source_id,dest
field_decoder={"timestamp": "tsp_parser"}
```

定义此配置时，`mycsv` 分析程序将提取名称为配置中所指定名称的字段，然后对 `timestamp` 字段的内容运行 `tsp_parser`。如果 `tsp_parser` 检索到有效的时间戳，服务器会将该时间戳用于日志消息。

自动日志分析程序

自动分析程序会自动检测行中前 200 个字符内的时间戳。自动检测到的时间戳的格式与 `timestamp` 分析程序的相同。

自动分析程序没有任何选项。除了自动检测时间戳外，键/值分析程序还针对日志条目运行，自动检测日志中现有的任何键/值对，并相应地提取字段。例如，

```
[filelog|some_logs]
directory=/var/log
include=*
parser=auto
```

与其他分析程序一样，可以为自动分析程序定义单独的操作。

```
[filelog|kvplogs]
directory=C:\temp_logs\csv-itbm
include=*.txt
parser=myauto
[parser|myauto]

base_parser=auto
debug=yes
```

如果已为自动分析程序启用 debug，则将输出有关分析的其他信息。例如，有关针对哪些日志运行自动分析程序以及从日志中提取哪些字段的信息。

对于分析程序，调试的默认值是 debug=no。

syslog 分析程序

syslog 分析程序支持 message_decoder 和 extract_sd 选项，并可自动检测以下格式：RFC-6587、RFC-5424 和 RFC-3164。

配置 message_decoder 选项

syslog 分析程序可以使用所有通用选项和 message_decoder 选项。默认情况下，仅提取 timestamp 和 appname 字段。可通过将 liagent.ini 文件中的配置值设置为类似于以下示例的值，来启用 message_decoder 选项：

```
[filelog|data_logs]
directory=D:\Logs
include=*.txt
parser=mysyslog

[parser|mysyslog]
base_parser=syslog
message_decoder=syslog_message_decoder
debug=yes

[parser|syslog_message_decoder]
base_parser=kvp
fields=*
```

示例：使用 message_decoder 选项进行分析

以下示例显示了一个示例事件，以及由配置为使用 message_decoder 选项的 syslog 分析程序添加到该事件的字段：

■ 示例事件：

```
2015-09-09 13:38:31.619407 +0400 smith01 john: Fri Dec 5 08:58:26 2014 [pid 26123]
[jsmith.net] status_code=FAIL oper_
ation=LOGIN: Client "176.31.17.46"
```

■ 由应用了 message_decoder 选项以运行 KVP 分析程序的 syslog 分析程序返回的内容：

```
timestamp=2015-09-09T09:38:31.619407 appname=john status_code=FAIL operation=LOGIN:
```

配置 extract_sd 选项以分析结构化数据

要分析结构化数据，请通过将 liagent.ini 文件中的配置值设置为类似于以下示例的值来启用 extract_sd 选项：

```
[filelog|simple_logs]
directory=/var/log
include=*.txt
```

```
parser=syslog_parser

[parser|syslog_parser]
base_parser=syslog
extract_sd=yes
```

示例：使用 extract_sd 选项进行分析

以下示例显示了一个示例事件，以及由配置为使用 `extract_sd` 选项的 `syslog` 分析程序添加到该事件的字段：

- 示例事件：<165>1 2017-01-24T09:17:15.719Z localhost evntslog - ID47
[exampleSDID@32473 iut="3" eventSource="Application" eventID="1011"]
[examplePriority@32473 class="high"] Found entity IPSet, display name dummy ip set 1411
- `syslog` 分析程序会将以下字段添加到该事件：

```
timestamp=2017-01-24T09:17:15.719000
pri_facility=20
pri_severity=5
procid="-"
msgid="ID47"
iut="3"
eventsource="Application"
eventid="1011"
class="high"
appname="evntslog"
```

由分析程序提取的字段

分析程序会自动从事件中提取以下字段：

RFC 分类	pri_facility	pri_severity	timestamp	appname	procid	msgid
非 RFC			X	X		
RFC-3164	X	X	X	X		
RFC-5424	X	X	X	X	X	X

syslog 分析程序选项

下表介绍了可用的 `syslog` 选项。

选项	描述
<code>message_decoder</code>	定义用于分析事件消息主体的其他分析程序。它可以是内置分析程序（如“自动”），也可以是任何自定义分析程序。
<code>extract_sd</code>	分析结构化数据。 <code>extract_sd</code> 选项仅支持使用值“yes”或“no”。默认情况下，该选项处于禁用状态。启用 <code>extract_sd</code> 选项后，它只是从结构化数据中提取所有键值对。

示例：针对 RFC-5424 标准的分析

以下示例显示了由配置为显示用于收集器的配置的 syslog 实例分析的两个事件、一个示例事件以及由 syslog 分析程序添加到该事件的字段。

■ 配置：

```
[filelog|simple_logs]
directory=/var/log
include=*.txt
parser=syslog
```

■ 在监控的文件中生成的事件：

```
<165>1 2017-01-24T09:17:15.719Z router1 mgd 3046 UI_DBASE_LOGOUT_EVENT
[junos@2636.1.1.1.2.18 username=\"regress\"] User 'regress' exiting configuration
mode - Juniper format
```

■ 由 syslog 分析程序添加到该事件的字段：

```
The following fields will be added to the event by Syslog parser:
timestamp=2017-01-24T09:17:15.719000
pri_facility = 20
pri_severity = 5
procid = 3046
msgid = UI_DBASE_LOGOUT_EVENT
appname = mgd
```

示例：针对 RFC-3164 标准的分析

以下示例显示了用于收集器的配置、一个示例 RFC-3164 事件以及由 syslog 添加到该事件的字段。

■ 配置：

```
[filelog|simple_logs]
directory=/var/log
include=*.txt
parser=syslog
```

■ 在监控的文件中生成的 RFC-3164 事件：

```
<13>2017-01-24T09:17:15.719Z router1 mgd 3046 UI_DBASE_LOGOUT_EVENT User 'regress' exiting
configuration mode - Juniper format
```

■ 由 syslog 分析程序添加到该事件的字段：

```
timestamp=2017-01-24T09:17:15.719000
pri_facility=1
pri_severity=5
appname="mgd"
```

带标签的制表符分隔值分析程序

带标签的制表符分隔值 (LTSV) 格式是制表符分隔值 (TSV) 的变体。

LTSV 文件中的每个记录单独作为一行表示。每个字段以 <TAB> 分隔，具有一个标签和一个值。标签和值通过 : 分隔。借助 LTSV 格式，您可以通过用 <TAB> 拆分行（与 TSV 格式一样）并通过唯一标记扩展任意字段（无特定顺序）来分析每个行。有关 LTSV 定义和格式的详细信息，请参见 <http://ltsv.org/>。

示例：LTSV 分析程序配置

示例：示例 LTSV 日志

LTSV 分析程序不需要特定的配置选项。要使用 LTSV 分析程序，请在配置中指定内置 ltsv 分析程序名称。

```
[parser|myltsv]
base_parser=ltsv
```

LTSV 文件必须是与 ABNF 格式的 LTSV 生产匹配的字节序列。

```
ltsv = *(record NL) [record]
record = [field *(TAB field)]
field = label ":" field-value
label = 1*1byte
field-value = *fbyte

TAB = %x09
NL = [%x0D] %x0A
1byte = %x30-39 / %x41-5A / %x61-7A / "_" / "." / "-" ;; [0-9A-Za-z_.-]
fbyte = %x01-08 / %x0B / %x0C / %x0E-FF
```

```
host:127.0.0.1<TAB>ident:--<TAB>user:frank<TAB>time:[10/Oct/2000:13:55:36 -0700]<TAB>req:GET /
apache_pb.gif HTTP/1.0<TAB>status:200<TAB>size:2326<TAB>referer:http://www.example.com/
start.html<TAB>ua:Mozilla/4.08 [en] (Win98; I ;Nav)
```

通过示例 LTSV 配置，日志的分析应返回下列字段：

```
host=127.0.0.1
ident=--
user=frank
time=[10/Oct/2000:13:55:36 -0700]
req=GET /apache_pb.gif HTTP/1.0
status=200
size=2326
referer=http://www.example.com/start.html
ua=Mozilla/4.08 [en] (Win98; I ;Nav)
```


调试配置

其他调试也可用于 LTSV 分析程序。默认情况下，LTSV 调试处于禁用状态。要打开 LTSV 调试，请输入 `debug=yes`。

```
[parser|myltsv]
base_parser=ltsv
debug=yes
```

当打开调试时，LTSV 分析程序会从日志中提取所有有效标签的值。LTSV 分析程序要求标签名称仅包含字母数字字符、下划线（“_”）、点（“.”）以及短划线（“-”）字符。如果日志中存在至少一个无效标签名称，其解析将失败。即使标签名称有效，代理也将检查字段名称。如果存在无效名称，标签名称应更正为有效的字段名称。

通过 filelog 部分配置 LTSV 分析程序

您还可以直接通过 filelog 部分配置 LTSV 分析程序。

```
[filelog|simple_logs]
directory=/var/log
include=*
parser=ltsv
```

regex 分析程序

regex 分析程序允许对收集的数据使用一些正则表达式。

vRealize Log Insight 代理使用 C++ Boost 库正则表达式，此正则表达式使用 Perl 语法。regex 分析程序可通过指定一个包含已命名的捕获组的正则表达式模式来进行定义。例如：`(?<field_1>\d{4})[-](?<field_2>\d{4})[-](?<field_3>\d{4})[-](?<field_4>\d{4})`

在组内指定的名称（例如：field_1、field_2、field_3 和 field_4）将成为相应的已提取字段的名称。命名需遵循以下要求：

- 在正则表达式模式中指定的名称必须为 vRealize Log Insight 的有效字段名称。
- 名称中只能包含字母数字字符和下划线“_”字符。
- 名称不能以数字字符开头。

如果提供的名称无效，则配置会失败。

regex 分析程序选项

regex 分析程序的唯一必填选项是 format 选项。

在需要其他调试信息时，可以使用 debug 选项。

配置

要创建 regex 分析程序，请使用 regex 作为 base_parser，并提供 format 选项。

示例： regex 配置示例**示例： 分析 Apache 日志示例**

以下示例可用来分析 1234-5678-9123-4567：

```
[parser|regex_parser]
base_parser=regex
format=(?<tag1>\d{4}) [-] (?<tag2>\d{4}) [-] (?<tag3>\d{4}) [-] (?<tag4>\d{4})
[filelog|some_info]
directory=D:\Logs
include=*.txt
parser=regex_parser
```

结果显示为：

```
tag1=1234
tag2=5678
tag3=9123
tag4=4567
```

要使用 regex 分析程序来分析 Apache 日志，请提供适用于 Apache 日志的特定 regex 格式：

```
[parser|regex_parser]
base_parser=regex
format=(?<remote_host>.*)(?<remote_log_name>.*)(?<remote_auth_user>.*)\[(?<log_timestamp>.*)\] "(?<request>.*)" (?<status_code>.*)(?<response_size>.*)
```

结果显示为：

```
127.0.0.1 - admin [10/Oct/2000:13:55:36 -0700] "GET /apache_pb.gif HTTP/1.0" 200 2326
remote_host=127.0.0.1
remote_log_name=-
remote_auth_user=admin
log_timestamp=10/Oct/2000:13:55:36 -0700
request=GET /apache_pb.gif HTTP/1.0
status_code=200
response_size=2326
```

以下代码显示了分析 Apache 日志的另一示例。

```
[parser|regex_parser]
base_parser=regex
format=(?<remote_host>.*(?<remote_log_name>.*))(?<remote_auth_user>.*)\[(?<log_timestamp>.*)\] "(?<request>.*(?<resource>.*)(?<protocol>.*))" (?<status_code>.*)(?<response_size>.*)
127.0.0.1 unknown - [17/Nov/2015:15:17:54 +0400] "\"GET /index.php HTTP/1.1\" 200 4868
remote_host=127.0.0.1 unknown
remote_log_name=unknown
remote_auth_user=-
log_timestamp=17/Nov/2015:15:17:54 +0400
request=GET /index.php HTTP/1.1
```

```
resource=/index.php
protocol=HTTP/1.1
status_code=200
response_size=4868
```

性能注意事项

与其他分析程序（例如 CLF 分析程序）相比，`regex` 分析程序消耗更多资源。如果您能够使用其他分析程序来分析日志，请考虑使用这些分析程序而不使用 `regex` 分析程序以获得更好的性能。

如果未提供分析程序，且您使用的是 `regex` 分析程序，请尽可能清晰地定义格式。以下示例显示了能够提供更佳性能结果的配置。该示例指定了包含数字值的字段。

```
(?<remote_host>\d+\.\d+\.\d+\.\d+) (?<remote_log_name>.*?) (?<remote_auth_user>.*?) \[(?<log_timestamp>.*?)\] "(?<request>.*?)" (?<status_code>\d+) (?<response_size>\d+)
```

JSON 分析程序

您可以自定义 JSON 分析程序配置，以便有选择地分析 JSON 日志。

您可以为 **FileLog** 和 **WinLog** 收集器配置逗号分隔值 (CSV) 分析程序。仅使用 Log Insight 代理 JSON 分析程序分析有效的 JSON 日志。无效的 JSON 日志分析程序将返回空结果。

对于默认 JSON 分析程序配置，Log Insight 代理将提取 JSON 日志中的所有字段。在 JSON 日志将自身表示为复杂 JSON 对象（也可能包含 JSON 对象）时，分析程序使用下划线 (`_`) 字符连接嵌套和更高层 JSON 对象的名称。这会为相应元素生成内容丰富的字段名称。如果 JSON 日志还包含数组，成员元素名称将包含数组名称，后跟数组中的元素索引。

JSON 分析程序还提供一个称为 **fields** 的特定选项。

JSON 分析程序 “fields” 选项

您可以使用 **fields** 选项指定在配置中分析的字段。该选项的用途是启用选择性 JSON 日志分析。

注 对于选择性分析，您必须指定所需的 JSON 元素的路径。不同层中的 JSON 对象必须使用点 (`.`) 字符隔开。

以下列表提供一些示例配置，以使您能够根据需要有选择地分析 JSON 日志。

- 要分析 JSON 日志中的多个元素，所需的元素必须作为 **fields** 选项的参数列出并以逗号分隔。请参见以下示例：

```
{
  "operation" : {
    "timestamp" : "2018-11-22T15:28:58.094000",
    "thread_id" : "0x05673",
    "initiator" : "connector",
    "log_severity" : "info",
    "log_message" : "Requested connection to the server.",
    "operation_result" : "success"
  }
}
```

- 要仅分析最里面的 JSON 对象（如 **timestamp**、**log_severity** 和 **log_message**），请参见以下示例。该示例配置生成以下字段结果：**operation_timestamp = "2018-11-22T15:28:58.094000"** and **operation_log_severity = "info"**

```
[parser|json_parser]
base_parser=json
fields=operation.timestamp,operation.log_severity, operation.log_message
```

- 要分析整个 JSON 对象，请包含对象路径，后跟一个星号 (*) 字符。

```
{
  "product_name" : "LI Agent",
  "operation" : {
    "timestamp" : "2018-11-22T15:28:58.094000",
    "thread_id" : "0x05673",
    "initiator" : "connector",
    "log_severity" : "info",
    "log_message" : "Requested connection to the server.",
    "operation_result" : "success"
  }
}
```

- 要仅分析 **operation** 对象，请使用以下配置：

```
[parser|json_parser]
base_parser=json
fields=operation.*
```

- 如果 JSON 日志包含数组，并且您希望仅分析数组的特定元素，请在配置中使用数组的元素索引，如下示例配置中所示：

```
{
  "Records": [
    {
      "object": {
        "key": "/events/mykey",
        "size": 502,
        "eTag": "091820398091823",
        "sequencer": "1123123"
      }
    },
    {
      "object": {
        "key": "/events/user_key",
        "size": 128,
        "eTag": "09182039000001",
        "sequencer": "1123231"
      }
    }
  ]
}
```

```

        "object":{
            "key": "/events/admin_key",
            "size": 1024,
            "eTag": "09182039547241",
            "sequencer": "1123213"
        }
    }
}
]
}

```

- 要仅分析同一日志中的 **key** 和 **size** 元素，请使用以下配置生成以下字段：

```
records0_object_key="/events/mykey"
```

```
records0_object_size=502
```

```
records2_object_key="/events/admin_key"
```

```
records2_object_size=1024
```

```

[parser|json_parser]
base_parser=json
fields = Records0.object.key Records0.object.size, Records2.object.key,
Records2.object.size

```

- 要分析所有数组元素的 **key** 字段，请使用以下配置：

```

[parser|json_parser]
base_parser=json
fields=Records.#.object.key

```

- 要分析所有字段，请使用带有星号 (*) 字符的 **fields** 选项。该配置相当于默认 JSON 分析程序配置。

```

[parser|json_parser]
base_parser=json
fields=*

```

卸载 vRealize Log Insight 代理

5

如果您需要卸载 vRealize Log Insight 代理，请按照适用于您安装的代理软件包的说明进行操作。

本章讨论了以下主题：

- 卸载 Log Insight Windows Agent
- 卸载 Log Insight Linux 代理 RPM 软件包
- 卸载 Log Insight Linux 代理 DEB 软件包
- 卸载 Log Insight Linux 代理 bin 软件包
- 手动卸载 Log Insight Linux 代理 bin 软件包

卸载 Log Insight Windows Agent

您可以从 Windows 控制面板的“程序和功能”屏幕中卸载 Log Insight Windows Agent。

前提条件

登录到已安装 vRealize Log Insight Windows 代理的 Windows 计算机，启动“服务”管理器以验证是否已安装 vRealize Log Insight 代理服务。

步骤

- 1 转到**控制面板 > 程序和功能**。
- 2 选择 VMware vRealize Log Insight Windows Agent 并单击**卸载**。

结果

卸载程序会停止 VMware vRealize Log Insight Windows Agent 服务，并将其文件从系统中移除。

卸载 Log Insight Linux 代理 RPM 软件包

可以卸载 Log Insight Linux Agent RPM 软件包。

前提条件

- 以 **root** 用户身份登录，或使用 **sudo** 运行控制台命令。

- 登录到已安装 Log Insight Linux Agent 的 Linux 计算机，打开终端控制台，然后运行 `pgrep liagent` 以验证 VMware Log Insight Linux Agent 是否已安装且正在运行。

步骤

- ◆ 运行以下命令，将 `VERSION` 和 `BUILD_NUMBER` 替换为已安装代理的版本和内部版本号。

```
rpm -e VMware-Log-Insight-Agent-VERSION-BUILD_NUMBER
```

结果

卸载程序将停止 VMware Log Insight Linux Agent 守护程序，并从系统中移除所有文件（日志除外）。

卸载 Log Insight Linux 代理 DEB 软件包

您可以卸载 Log Insight Linux Agent DEB 软件包。

前提条件

- 以 **root** 用户身份登录，或使用 `sudo` 运行控制台命令。
- 登录到已安装 Log Insight Linux Agent 的 Linux 计算机，打开终端控制台，然后运行 `pgrep liagent` 以验证 VMware Log Insight Linux Agent 是否已安装且正在运行。

步骤

- ◆ 运行以下命令

```
dpkg -P vmware-log-insight-agent
```

结果

卸载程序将停止 VMware Log Insight Linux Agent 守护程序，并从系统中移除所有文件（日志除外）。

卸载 Log Insight Linux 代理 bin 软件包

您可以使用 vRealize Log Insight 脚本卸载 Log Insight Linux Agent .bin 软件包。

前提条件

- 以 **root** 用户身份登录，或使用 `sudo` 运行控制台命令。
- 登录到已安装 Log Insight Linux Agent 的 Linux 计算机，打开终端控制台并运行 `pgrep liagent` 以验证 VMware vRealize Log Insight Linux Agent 是否已安装并正在运行。

步骤

- 1 在 Shell 提示符中，输入以下命令以启动脚本。

```
loginsight-agent-uninstall
```

- 2 通过检查以下命令返回的错误代码是否为 0，可验证卸载是否已成功完成。

```
echo $?
```

手动卸载 Log Insight Linux 代理 bin 软件包

如果您选择不使用卸载脚本，则可以手动卸载 Log Insight Linux Agent .bin 软件包。

前提条件

手动卸载 Log Insight Linux 代理 bin 软件包

- 以 **root** 用户身份登录，或使用 `sudo` 运行控制台命令。
- 登录到已安装 Log Insight Linux Agent 的 Linux 计算机，打开终端控制台并运行 `pgrep liagent` 以验证 VMware vRealize Log Insight Linux Agent 是否已安装并正在运行。

步骤

- 1 通过运行以下命令停止 Log Insight Linux Agent 守护程序

旧 Linux 分发版的 `sudo service liagentd stop` 或 `sudo /sbin/service liagentd stop`。

- 2 手动移除 Log Insight Linux Agent 文件

- `/usr/lib/loginsight-agent` - 守护进程二进制和许可证文件目录。
- `/usr/bin/loginsight-agent-support` - 用于生成 Log Insight Linux Agent 的支持包。
- `/var/lib/loginsight-agent` - 配置文件和数据库存储目录。
- `/var/log/loginsight-agent` - Log Insight Linux Agent 的日志目录。
- `/var/run/liagent/liagent.pid` - Log Insight Linux Agent PID 文件。如果未自动删除，请手动移除该文件。
- `/etc/init.d/liagentd` - Log Insight Linux Agent 守护进程的脚本目录。
- `/usr/lib/systemd/system/liagentd.service`

vRealize Log Insight 代理故障排除

6

已知故障排除信息能够帮助您诊断和更正与 vRealize Log Insight 代理操作相关的问题。

本章讨论了以下主题：

- 为 Log Insight Windows Agent 创建支持包
- 为 Log Insight Linux Agent 创建支持包
- 在 Log Insight Agents 中定义日志详细信息级别
- 管理 UI 不显示 Log Insight Agents
- vRealize Log Insight 代理不发送事件
- 为 Log Insight Windows Agent 添加出站例外规则
- 在 Windows 防火墙中允许来自 Log Insight Windows Agent 的出站连接
- Log Insight Windows Agent 批量部署失败
- Log Insight Agents 拒绝自签名证书
- vRealize Log Insight 服务器拒绝非加密流量的连接

为 Log Insight Windows Agent 创建支持包

如果 Log Insight Windows Agent 由于出现问题未按预期运行，可以将日志和配置文件的副本发送到 VMware 支持服务。

步骤

- 1 登录到安装了 Log Insight Windows Agent 的目标计算机。
- 2 单击 Windows **开始**按钮，然后单击 **VMware > Log Insight 代理 - 收集支持包**。
- 3 （可选）如果快捷方式不可用，请导航到 Log Insight Windows Agent 的安装目录，然后双击 `loginsight-agent-support.exe`。

注 默认安装目录为 `C:\Program Files (x86)\VMware\Log Insight Agent`

结果

该支持包已生成，并作为 `.zip` 文件保存在 `My Documents` 中。

后续步骤

可根据请求将支持包转发到 VMware 支持服务。

为 Log Insight Linux Agent 创建支持包

如果 Log Insight Linux Agent 由于出现问题未按预期运行，可以将日志和配置文件的副本发送到 VMware 支持服务。

步骤

1 登录到安装了 Log Insight Linux Agent 的目标计算机。

2 运行下列命令。

```
/usr/lib/loginsight-agent/bin/loginsight-agent-support
```

结果

该支持包已生成，并作为 .zip 文件保存在当前目录中。

后续步骤

可根据请求将支持包转发到 VMware 支持服务。

在 Log Insight Agents 中定义日志详细信息级别

您可以编辑 vRealize Log Insight 代理的配置文件以更改日志记录级别。

前提条件

对于 Log Insight Linux Agent:

- 以 **root** 用户身份登录，或使用 **sudo** 运行控制台命令。
- 登录到已安装 Log Insight Linux Agent 的 Linux 计算机，打开控制台，然后运行 `pgrep liagent` 以验证 VMware vRealize Log Insight Linux Agent 是否已安装且正在运行。

对于 Log Insight Windows Agent:

- 登录到已安装 vRealize Log Insight Windows 代理的 Windows 计算机，启动“服务”管理器以验证是否已安装 vRealize Log Insight 代理服务。

步骤

1 导航到包含 `liagent.ini` 文件的文件夹。

操作系统	路径
Linux	<code>/var/lib/loginsight-agent/</code>
Windows	<code>%ProgramData%\VMware\Log Insight Agent</code>

2 在任意文本编辑器中打开 `liagent.ini` 文件。

- 3 在 liagent.ini 文件的 [logging] 部分中更改日志调试级别。

注 调试级别越高，对 vRealize Log Insight 代理的影响就越大。默认值和推荐值均为 0。调试级别 1 可提供更多信息。建议将该级别用于排除大多数问题。调试级别 2 可提供详细的信息。只有当 VMware 支持人员提出要求时才使用级别 1 和 2。

```
[logging]
; The level of debug messages to enable: 0..2
debug_level=1
```

- 4 保存并关闭 liagent.ini 文件。

结果

日志调试级别已更改。

管理 UI 不显示 Log Insight Agents

有关 Log Insight Agents 实例的信息不显示在管理 UI 的“代理”页面上。

问题

安装 Log Insight Agents 后，在管理 UI 的“代理”页面中看不到 Log Insight Agents。

原因

最常见的原因是出现网络连接问题或 liagent.ini 文件中的 Log Insight Agents 配置得不正确。

解决方案

- ◆ 验证安装了 Log Insight Agents 的 Windows 或 Linux 系统是否已连接到 vRealize Log Insight 服务器。
- ◆ 验证 Log Insight Agents 是否使用 cfapi 协议。
如果使用 syslog 协议，UI 不会显示 Log Insight Windows Agents。
- ◆ 查看位于以下目录中的 Log Insight Agents 日志文件的内容。
 - Windows - %ProgramData%\VMware\Log Insight Agent\log
 - Linux - /var/log/loginsight-agent/
 查找包含以下短语的日志消息：Config transport error: Couldn't resolve host name 和 Resolver failed. No such host is known.
- ◆ 验证 liagent.ini 是否包含目标 vRealize Log Insight 服务器的正确配置。请参见[设置目标 vRealize Log Insight 服务器和指定代理的目标](#)。

vRealize Log Insight 代理不发送事件

不正确的配置可能会阻止 vRealize Log Insight 代理将事件转发到 vRealize Log Insight 服务器。如果未正确配置平面文件收集通道，则您会看到类似如下消息：针对通道 “CHANNEL_NAME” 获取的设置无效。通道 “CHANNEL_NAME” 将保持休眠状态，直到通道配置正确为止 (Invalid settings were obtained for channel 'CHANNEL_NAME'. Channel 'CHANNEL_NAME' will stay dormant until properly configured)。

问题

vRealize Log Insight 代理实例会显示在**管理 > 代理**页面中，但没有事件会显示在 vRealize Log Insight 代理主机名称的“交互式分析”页面中。未正确配置平面文件收集通道。

原因

不正确的配置可能会阻止 vRealize Log Insight 代理将事件转发到 vRealize Log Insight 服务器。

解决方案

- ◆ 定义一个有效的收集通道。验证是否正确配置了平面文件收集通道。请参见[第 4 章 配置 vRealize Log Insight 代理](#)。
- ◆ 对于 vRealize Log Insight Windows 代理，请尝试以下操作。
 - 如果启用了 Windows 通道，请查看位于 %ProgramData%\VMware\Log Insight Agent\log 中的 vRealize Log Insight Windows 代理日志文件的内容。查找与包含短语已订阅通道 CHANNEL_NAME 的通道配置相关的日志消息。常用的通道有 Application、System 和 Security。
 - 如果未正确配置通道，则您会看到类似如下日志消息：无法订阅通道 CHANNEL_NAME 事件。错误代码：15007。找不到指定通道。请检查通道配置。您会看到除 15007 以外的错误代码。
 - 如果未正确配置平面文件收集通道，则您会看到类似如下消息：针对通道 “CHANNEL_NAME” 获取的设置无效。通道 “CHANNEL_NAME” 将保持休眠状态，直到通道配置正确为止
- ◆ 对于 vRealize Log Insight Windows 代理和 vRealize Log Insight Linux 代理，请尝试以下操作。
 - ◆ 如果未配置平面文件收集通道，则您会看到类似如下消息：无法在配置中找到 “filelog” 部分。平面文件日志收集器将保持休眠状态，直到通道配置正确为止

vRealize Log Insight 代理日志文件的内容位于以下目录中。

- Windows - %ProgramData%\VMware\Log Insight Agent\log
- Linux - /var/log/loginsight-agent/

后续步骤

有关配置 vRealize Log Insight 代理的更多信息，请参见[配置 Log Insight Windows Agent](#)和[配置 Log Insight Linux Agent](#)。

为 Log Insight Windows Agent 添加出站例外规则

在 Windows 防火墙中定义用于对 Log Insight Windows Agent 解除阻止的例外规则。

该过程适用于 Windows Server 2008 R2 及更高版本，以及 Windows 7 及更高版本。

前提条件

- 验证是否已具有管理员帐户或具有管理特权的帐户。

步骤

- 1 选择**开始 > 运行**。
- 2 键入 `wf.msc` 并单击**确定**。
- 3 在左窗格中，右键单击**出站规则**，然后单击**新建规则**。
- 4 选择**自定义**，并按照向导设置以下选项。

选项	描述
程序	<code>liwinsvc.exe</code>
服务	LogInsightAgentService
协议和端口	用于 cfapi 的 TCP 9000 和用于 syslog 的 514

- 5 在“指定此规则应用的配置文件”页面上，选择相应的网络类型。

- 域
- 公用
- 专用

注 可以选择所有网络类型，以确保例外规则始终处于活动状态，而不管网络类型如何。

后续步骤

转到 Log Insight Windows Agent 日志目录 `%ProgramData%\VMware\Log Insight Agent\log`，并打开最新的日志文件。如果最近的事件中包含消息 `Config transport error: Couldn't resolve host name` 及 `Resolver failed. No such host is known`，请重新启动 Log Insight Windows Agent 服务和 Windows 计算机。

注 Log Insight Windows Agent 服务可能需要长达 5 分钟来重新连接到服务器。

在 Windows 防火墙中允许来自 Log Insight Windows Agent 的出站连接

配置 Windows 防火墙设置，以允许 Log Insight Windows Agent 到 vRealize Log Insight 服务器的出站连接。

在安装并启动 Log Insight Windows Agent 服务之后，Windows 域或本地防火墙可能会限制与目标 vRealize Log Insight 服务器的连接。

该过程适用于 Windows Server 2008 R2 及更高版本，以及 Windows 7 及更高版本。

前提条件

- 验证是否已具有管理员帐户或具有管理特权的帐户。

步骤

- 1 选择开始 > 运行。
- 2 键入 `wf.msc` 并单击**确定**。
- 3 在“操作”窗格中，单击**属性**。
- 4 在**域配置文件**选项卡上，从**出站连接**下拉菜单中选择**允许(默认)**。

如果计算机没有连接到域，您可以选择**专用配置文件**或**公用配置文件**，具体取决于计算机所连接的网络类型。

- 5 单击**确定**。

后续步骤

在 Windows 防火墙中定义用于对 Log Insight Windows Agent 解除阻止的例外规则。请参见为 [Log Insight Windows Agent](#) 添加出站例外规则。

Log Insight Windows Agent 批量部署失败

在目标计算机上批量部署 Log Insight Windows Agent 的操作失败。

问题

通过使用组策略对象在 Windows 域计算机上执行批量部署后，Log Insight Windows Agent 无法安装为本地服务。

原因

组策略设置可能会导致 Log Insight Windows Agent 无法正确安装。

解决方案

- 1 编辑组策略对象 (GPO) 设置，并重新部署 Log Insight Windows 代理。
 - a 右键单击 GPO，单击**编辑**并导航到**计算机配置 > 策略 > 管理模板 > 系统 > 登录**。
 - b 启用**始终在计算机启动和登录时等待网络策略**。
 - c 导航到**计算机配置 > 策略 > 管理模板 > 系统 > 组策略**。
 - d 启用**启动策略处理等待时间**，并将**等待的时间(秒)**设置为 120。
- 2 在目标计算机上运行 `gpupdate /force /boot` 命令。

Log Insight Agents 拒绝自签名证书

Log Insight Agents 拒绝自签名证书。

问题

vRealize Log Insight 代理拒绝自签名证书，无法与服务器建立连接。

注 如果在使用代理时遇到连接问题，可以通过将代理的调试级别更改为 1 来生成详细日志以供查看。有关详细信息，请参见在 [Log Insight Agents](#) 中定义日志详细信息级别。

原因

在代理日志中显示的消息具有特定原因。

消息	原因
拒绝对等自签名证书。公共密钥与以前存储的证书密钥不匹配。	<ul style="list-style-type: none"> ■ 当 vRealize Log Insight 证书被替换时，可能会发生这种情况。 ■ 如果启用 HA 的群集环境在 vRealize Log Insight 节点上配置了不同的自签名证书，可能会发生这种情况。
拒绝对等自签名证书。以前已接收由可信 CA 签名的证书。	代理端存储有 CA 签名证书。

解决方案

- ◆ 确认目标主机名是可信的 vRealize Log Insight 实例，然后从 vRealize Log Insight 代理的 cert 目录中手动删除以前的证书。
 - 对于 Log Insight Windows Agent，请转至 `C:\ProgramData\VMware\Log Insight Agent\cert`。
 - 对于 Log Insight Linux Agent，请转至 `/var/lib/loginsight-agent/cert`。

注 有些平台可能会使用非标准路径来存储可信证书。Log Insight Agents 提供了一个选项，可以通过设置 `ssl_ca_path=<fullpath>` 配置参数来配置可信证书存储路径。使用可信根证书包文件的路径替换 `<fullpath>`。请参见[配置 Log Insight 代理的 SSL 参数](#)。

vRealize Log Insight 服务器拒绝非加密流量的连接

如果您尝试发送非加密流量，vRealize Log Insight 服务器会拒绝与 Log Insight Agents 连接。

您可以将 vRealize Log Insight 服务器配置为接受非 SSL 连接，或将 Log Insight Agents 配置为通过 SSL cfapi 协议连接来发送数据。

问题

当您尝试使用 cfapi 发送非加密流量时，vRealize Log Insight 服务器会拒绝连接。代理日志中显示以下错误消息之一：403 禁止 (403 Forbidden) 或 403 仅允许 SSL 连接 (403 Only SSL connections are allowed)。

原因

vRealize Log Insight 配置为仅接受 SSL 连接，而 Log Insight Agents 配置为使用非 SSL 连接。

解决方案

- 1 将您的 vRealize Log Insight 服务器配置为接受非 SSL 连接。
 - a 导航到**管理**选项卡。
 - b 在“配置”下，单击 **SSL**。
 - c 在“API 服务器 SSL”标题下，取消选择**需要 SSL 连接**。
 - d 单击**保存**。
- 2 将 vRealize Log Insight 代理配置为通过 SSL cfapi 协议连接来发送数据。
 - a 导航到包含 liagent.ini 文件的文件夹。

操作系统	路径
Linux	/var/lib/loginsight-agent/
Windows	%ProgramData%\VMware\Log Insight Agent

- b 在任意文本编辑器中打开 liagent.ini 文件。
- c 将 liagent.ini 文件的 [server] 部分中的 ssl 键值更改为 **yes**，并将协议更改为 cfapi。

```
proto=cfapi
ssl=yes
```

- d 保存并关闭 liagent.ini 文件。