

# vRealize Log Insight 에이전트 작업

2022년 5월 24일

vRealize Log Insight 8.1

다음 VMware 웹 사이트에서 최신 기술 문서를 확인할 수 있습니다.

<https://docs.vmware.com/kr/>

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

**VMware 코리아**  
서울시 강남구  
영동대로 517  
아셈타워 13층  
(우) 06164  
전화: +82 2 3016 6500  
팩스: +82 2 3016 6501  
[www.vmware.com/kr](http://www.vmware.com/kr)

Copyright © 2022 VMware, Inc. All rights reserved. 저작권 및 상표 정보

# 목차

vRealize Log Insight 에이전트 작업	5
1 vRealize Log Insight 에이전트 개요	6
2 로그 순환 체계 유형	8
3 vRealize Log Insight 에이전트 설치 또는 업그레이드	9
에이전트 설치 파일 다운로드	10
Windows 에이전트 설치	11
설치 마법사를 사용하여 vRealize Log Insight Windows 에이전트 설치 또는 업데이트	11
명령줄에서 vRealize Log Insight Windows 에이전트 설치 또는 업데이트	12
여러 시스템에 Log Insight Windows Agent 배포	14
vRealize Log Insight Linux 에이전트 RPM 패키지 설치 또는 업데이트	17
vRealize Log Insight Linux 에이전트 DEB 패키지 설치 또는 업데이트	18
Debian Linux용 에이전트 설치 사용자 지정	19
Log Insight Linux Agent 이진 패키지 설치	22
Linux에서 vRealize Log Insight 에이전트를 설치하기 위한 명령줄 옵션	24
vRealize Log Insight 에이전트의 자동 업데이트	25
개별 에이전트에 대해 자동 업데이트 사용 안 함 또는 사용	25
4 vRealize Log Insight 에이전트 구성	27
Log Insight Windows Agent 구성	28
Log Insight Windows Agent의 기본 구성	28
Windows 이벤트 채널에서 이벤트 수집	31
로그 파일에서 이벤트 수집	35
Log Insight Windows Agent로 이벤트 전달	40
Log Insight Linux Agent 구성	41
vRealize Log Insight Linux Agent의 기본 구성	41
로그 파일에서 이벤트 수집	43
vRealize Log Insight 에이전트에서 이벤트 필터링	50
vRealize Log Insight 에이전트의 정보 전달	52
대상 vRealize Log Insight 서버 설정	52
에이전트 대상 지정	56
vRealize Log Insight 에이전트의 중앙 집중식 구성	59
구성 병합 예	60
에이전트 구성에 대해 공통 값 사용	61

로그 구문 분석 63

로그 구문 분석기 구성 63

## 5 vRealize Log Insight 에이전트 제거 91

Log Insight Windows Agent 제거 91

Log Insight Linux Agent RPM 패키지 제거 91

Log Insight Linux Agent DEB 패키지 제거 92

Log Insight Linux Agent Bin 패키지 제거 92

Log Insight Linux Agent Bin 패키지 수동 제거 93

## 6 vRealize Log Insight 에이전트 문제 해결 94

Log Insight Windows Agent용 지원 번들 생성 94

Log Insight Linux Agent용 지원 번들 생성 95

Log Insight Agents의 로그 세부 정보 수준 정의 95

관리 UI에 Log Insight Agents가 표시되지 않음 96

vRealize Log Insight 에이전트가 이벤트를 보내지 않음 97

Log Insight Windows Agent에 대한 아웃바운드 예외 규칙 추가 98

Windows 방화벽에서 Log Insight Windows Agent의 아웃바운드 연결 허용 99

Log Insight Windows Agent의 대량 배포가 실패함 100

Log Insight Agents 자체 서명된 인증서 거부 100

vRealize Log Insight 서버가 암호화되지 않은 트래픽의 연결을 거부함 101

# vRealize Log Insight 에이전트 작업

"vRealize Log Insight Agent 작업"에서는 vRealize™Log Insight™ Windows 및 Linux 에이전트를 설치하고 구성하는 방법을 설명합니다. 또한 문제 해결 팁도 포함되어 있습니다.

이 정보는 Log Insight Agents를 설치하거나 구성하거나 문제 해결하려는 사용자를 대상으로 합니다. 이 정보는 가상 시스템 기술과 데이터 센터 운영에 대해 잘 알고 있는 숙련된 Windows 또는 Linux 시스템 관리자를 대상으로 작성되었습니다.

vRealize Log Insight 서버로 에이전트용 구성 클래스를 생성하는 방법에 대해서는 "vRealize Log Insight 관리"를 참조하십시오.

# vRealize Log Insight 에이전트 개요

# 1

vRealize Log Insight 에이전트는 로그 파일에서 이벤트를 수집하고 vRealize Log Insight 서버 또는 모든 타사 syslog 대상으로 전달합니다.

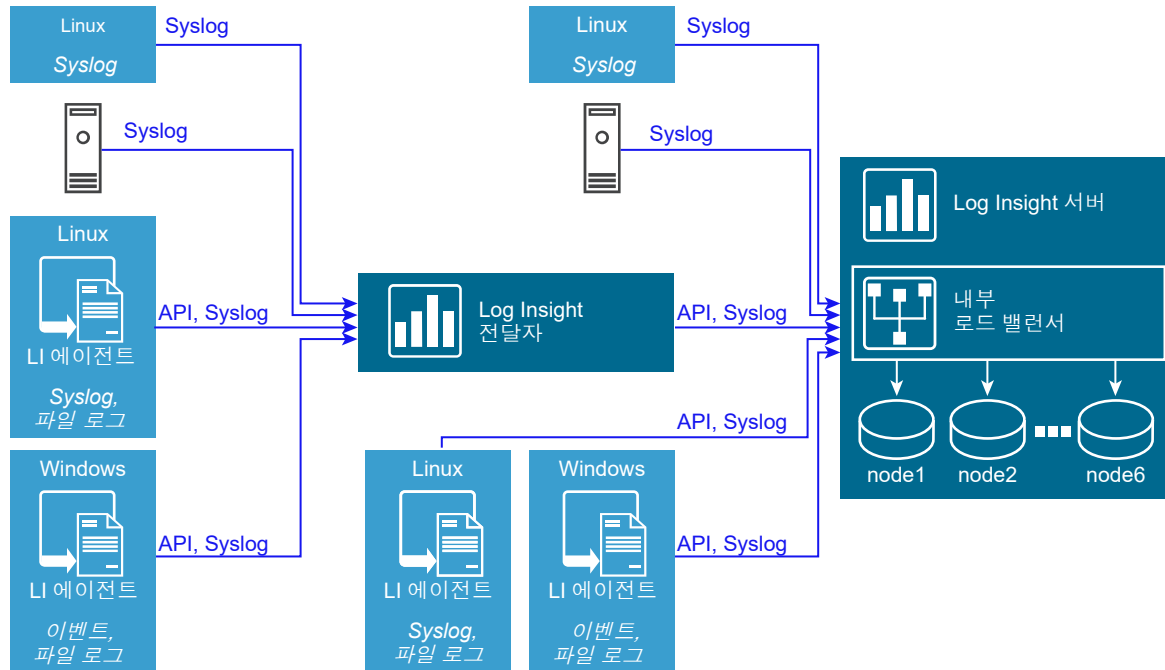
에이전트는 syslog 및 vRealize Log Insight 수집 API(cfapi 프로토콜)를 지원하며 Linux 또는 Windows 플랫폼에서 사용될 수 있습니다. 웹 인터페이스를 통해, 서버 측 및 클라이언트 측의 liagent.ini 파일을 사용하여 또는 설치의 일부로 에이전트를 구성합니다.

에이전트는 다음과 같은 기능을 포함합니다.

- 단일 또는 그룹 배포
- 자동 업그레이드
- 로그 메시지에 작동하고 구조화된 데이터를 추출하는 구문 분석. FileLog 및 WinLog 수집기 또는 둘 모두에 대한 구문 분석기를 구성할 수 있습니다.
- 여러 줄 메시지에 대한 지원
- 여러 로그 순환 체계에 대한 기본적인 지원
- 클라이언트 측 압축, 암호화 및 이벤트에 메타데이터를 추가하는 기능을 포함하는 포괄적인 수집 API

vRealize Log Insight 서버는 에이전트 그룹에 대한 중앙 집중식 구성 관리, 생성 및 관리를 지원합니다.

다음 그림은 에이전트 배포 구성의 요소를 나타냅니다.



vRealize Log Insight 전달자는 기본적으로 원격 대상으로 이벤트를 전달하는 vRealize Log Insight 서버의 전용 인스턴스입니다. 일반적으로, 전달자로 사용되는 서버 인스턴스는 쿼리에 사용되지 않습니다. 전달자는 내부 로드 밸런서를 사용하며, 그렇지 않은 경우 vRealize Log Insight 서버와 같이 구성됩니다.

에이전트는 자체 작업 로그를 씁니다. Windows의 경우 이러한 로그는 C:\ProgramData\VMware\Log Insight Agent\logs 디렉토리에 있습니다. Linux의 경우 작업 로그의 경로는 /var/log/loginsight-agent/liagent\_\*.log입니다. 로그 파일은 에이전트가 다시 시작되거나 파일이 10MB 크기에 도달하면 순환됩니다. 순환에서 50MB의 파일 결합 제한이 유지됩니다. vRealize Log Insight 에이전트 자체를 사용해서 에이전트 로그를 수집할 수는 없습니다.

에이전트는 실시간 로그 수집에 사용됩니다. vRealize Log Insight Importer를 사용하여 지원 번들을 포함하는 기록 로그 수집을 가져옵니다.

Windows 및 Linux 운영 체제용 설치 다운로드가 별도로 제공됩니다.

Windows 시스템에서 에이전트는 Windows 서비스로 실행되며 설치 후 즉시 시작됩니다. 에이전트는 애플리케이션 로그 파일과 Windows 이벤트 채널, 관련 Windows 시스템 이벤트를 수집하기 위한 폴을 모니터링합니다. 수집된 이벤트는 vRealize Log Insight 서버 또는 타사 syslog 대상에 전달됩니다.

Linux 시스템에서 에이전트는 데몬으로 실행되며 설치 후 즉시 시작됩니다. vRealize Log Insight Linux 에이전트는 Linux 시스템의 로그 파일에서 이벤트를 수집하여 이를 vRealize Log Insight 서버 또는 syslog 대상으로 전달합니다. Debian, Red Hat 및 Linux 바이너리 설치 패키지를 사용할 수 있습니다.

# vRealize Log Insight 에이전트에서 지원하는 로그 순환 체계

## 2

vRealize Log Insight 에이전트는 몇 가지 로그 순환 체계를 지원합니다.

로그 순환은 로그 파일이 무한히 커지지 않도록 합니다. 각각 특정 사용 사례 집합에 맞게 디자인된 몇 가지 로그 순환 체계가 있습니다. vRealize Log Insight에는 다음과 같은 체계에 대한 기본적인 지원이 포함되어 있습니다.

표 2-1. vRealize Log Insight 에이전트에서 지원하는 로그 순환 체계

로그 순환 체계	설명
create-new	크기 또는 시간 제한에 도달하면 새 로그 파일이 생성됩니다. 로거 프로세스는 현재 로그 파일에 대한 쓰기를 중지하고 로그 출력을 새로 생성된 파일로 전송합니다. 어떤 경우든 기존 파일은 이름이 바뀌거나 변경되지 않습니다.
rename-recreate	logrotate와 같은 외부 유틸리티는 크기 및 시간 제한에 도달하면 로그 파일의 이름을 바꿉니다. 그런 다음 로거 프로세스는 이전 이름을 사용하여 로그 파일을 생성합니다.
copy-truncate	logrotate와 같은 외부 유틸리티는 크기 및 시간 제한에 도달하면 로그 파일을 복사합니다. 로거 프로세스는 복사한 파일의 이름을 바꾸고, 원래 파일 크기가 0이 되도록 파일을 자릅니다. 로거 프로세스는 원래 파일에 로그를 계속 쓸 수 있습니다.



# vRealize Log Insight 에이전트 설치 또는 업그레이드

## 3

타사 로그 관리 시스템이 있는 시스템을 비롯한 Windows 또는 Linux 시스템에 vRealize Log Insight 에이전트를 설치 또는 업그레이드합니다.

에이전트는 이벤트를 수집한 후 vRealize Log Insight 서버에 전달합니다. 설치 중에 서버, 포트 및 프로토콜 설정에 대한 매개 변수를 지정하거나 기본 설정을 유지하도록 선택할 수 있습니다.

설치에 사용하는 것과 동일한 방법으로 에이전트를 업그레이드하거나 자동 업그레이드를 사용할 수 있습니다. 자동 업그레이드는 새 버전의 vRealize Log Insight를 배포할 때 에이전트로 업그레이드를 전파합니다. 자세한 내용은 [vRealize Log Insight 에이전트의 자동 업데이트](#)를 참조하십시오. Linux 바이너리 패키지에는 업그레이드를 사용할 수 없습니다.

## 하드웨어 지원

vRealize Log Insight 에이전트를 설치하고 실행하려는 경우 하드웨어가 x86 및 x86\_64 아키텍처와 MMX, SSE, SSE2 및 SSE3 명령 집합을 지원하는 호스트/시스템에 필요한 최소 매개 변수를 지원해야 합니다.

## 플랫폼 지원

운영 체제	프로세서 아키텍처
Windows 7, Windows 8, Windows 8.1, 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 및 Windows Server 2019	x86_64
RHEL 5, RHEL 6, RHEL 7 및 RHEL 8	x86_64, x86_32
SuSE Enterprise Linux(SLES) 11 SP3 및 SLES 12 SP1	x86_64
Ubuntu 14.04 LTS, Ubuntu 16.04 LTS 및 Ubuntu 18.04	x86_64
VMware Photon, 버전 1 개정 2, 버전 2 및 버전 3	x86_64

## Linux 참고 사항

사용할 루트 권한이 없는 사용자를 위해 Log Insight Linux Agent의 기본 설치를 구현하는 경우, 기본 구성이 데이터 수집 시 문제를 초래할 수 있습니다. 에이전트는 채널 구독이 실패했다는 경고를 로깅하지 않고, 컬렉션의 파일에 대해 읽기 사용 권한이 없습니다. Inaccessible log file ... will try later 메시지가 로그에 반복적으로 추가됩니다. 문제를 초래하는 기본 구성을 주석 처리하거나 사용자 사용 권한을 변경할 수 있습니다.

rpm 또는 DEB 패키지를 사용하여 Linux 에이전트를 설치하는 경우 패키지 설치의 일부로 liagentd라는 이름의 init.d 스크립트가 설치됩니다. bin 패키지는 스크립트를 추가하지만 이를 등록하지는 않습니다. 스크립트를 수동으로 등록할 수 있습니다.

(/Sbin/)service liagentd status 명령을 실행하여 설치가 성공적으로 수행되었는지 확인할 수 있습니다.

본 장은 다음 항목을 포함합니다.

- [에이전트 설치 파일 다운로드](#)
- [Windows 에이전트 설치](#)
- [vRealize Log Insight Linux 에이전트 RPM 패키지 설치 또는 업데이트](#)
- [vRealize Log Insight Linux 에이전트 DEB 패키지 설치 또는 업데이트](#)
- [Debian Linux용 에이전트 설치 사용자 지정](#)
- [Log Insight Linux Agent 이진 패키지 설치](#)
- [Linux에서 vRealize Log Insight 에이전트를 설치하기 위한 명령줄 옵션](#)
- [vRealize Log Insight 에이전트의 자동 업데이트](#)

## 에이전트 설치 파일 다운로드

vRealize Log Insight 에이전트를 설정하는 첫 번째 단계는 플랫폼에 맞는 에이전트 설치 패키지를 다운로드하는 것입니다.

vRealize Log Insight 서버 에이전트 페이지에서 다운로드된 모든 패키지에는 패키지 이름에 대상 호스트 이름이 추가되어 포함됩니다. 서버 호스트 이름은 MSI, RPM 및 DEB 에이전트의 초기 설치 시 적용됩니다. 구성 파일에 호스트 이름이 있는 경우 또는 호스트 이름 매개 변수를 사용하여 패키지를 실행하는 경우에는 포함된 서버 호스트 이름이 무시됩니다.

### 절차

- 1 vRealize Log Insight 웹 사용자 인터페이스의 **관리** 탭으로 이동합니다.
- 2 관리 섹션에서 **에이전트**를 클릭합니다.
- 3 화면 맨 아래로 스크롤한 후 **Log Insight 에이전트 다운로드**를 클릭하고

4 팝업 메뉴에서 설치 패키지를 선택하고 **저장**을 클릭하여 다운로드합니다.

옵션	설명
Windows MSI	Windows 플랫폼용 설치 패키지(32비트/64비트)
Linux RPM	Linux Red Hat, openSuSE(32비트/64비트) 또는 VMware Photon Platform용 설치 패키지
Linux DEB	Linux Debian 플랫폼용 설치 패키지(32비트/64비트)
Linux BIN	Linux용 자동 설치 패키지(32비트/64비트). 패키지 관리 시스템은 필요하지 않습니다.

다음에 수행할 작업

다운로드한 파일을 사용하여 vRealize Log Insight 에이전트를 배포합니다.

## Windows 에이전트 설치

설치 마법사 또는 명령줄을 통해 단일 시스템에 에이전트를 설치하거나 스크립트를 사용하여 한 에이전트에 대해 여러 인스턴스를 배포할 수 있습니다.

## Windows 에이전트 업그레이드

설치에 사용할 수 있는 방법 중 하나를 사용하여 업그레이드 파일을 적용하는 방식으로 Windows 에이전트를 업그레이드할 수 있습니다. 자동 업그레이드 기능을 사용하거나 에이전트를 백그라운드에서 업그레이드하도록 선택할 수도 있습니다.

## 설치 마법사를 사용하여 vRealize Log Insight Windows 에이전트 설치 또는 업데이트

설치 마법사를 사용하여 단일 시스템에서 Windows 에이전트를 설치하거나 업그레이드할 수 있습니다.

### 사전 요구 사항

- vRealize Log Insight Windows 에이전트 .msi 파일의 복사본이 있는지 확인합니다. [에이전트 설치 파일 다운로드](#) 항목을 참조하십시오.
- 설치를 수행할 수 있는 사용 권한이 있으며 Windows 시스템에서 서비스를 시작했는지 확인합니다.

### 절차

- 1 vRealize Log Insight Windows 에이전트를 설치할 Windows 시스템에 로그인합니다.
- 2 vRealize Log Insight Windows 에이전트 .msi 파일이 있는 디렉토리로 변경합니다.
- 3 vRealize Log Insight Windows 에이전트 .msi 파일을 두 번 클릭하고 라이선스 계약 내용에 동의한 후 **다음**을 클릭합니다.

#### 4 vRealize Log Insight 서버의 IP 주소 또는 호스트 이름을 입력하고 **설치**를 클릭합니다.

vRealize Log Insight Windows 에이전트가 로컬 시스템 서비스 계정 아래에 자동 Windows 서비스로 설치되거나 업데이트됩니다.

#### 5 **마침**을 클릭합니다.

#### 다음에 수행할 작업

liagent.ini 파일을 편집하여 vRealize Log Insight Windows 에이전트를 구성합니다. [Log Insight Windows Agent 구성](#) 을 참조하십시오.

## 명령줄에서 vRealize Log Insight Windows 에이전트 설치 또는 업데이트

명령줄에서 Windows 에이전트를 설치하거나 업데이트할 수 있습니다.

기본값을 사용하거나 특정 서비스 계정을 지정하고 명령줄 매개 변수를 사용하여 서버, 포트 및 프로토콜 정보를 지정할 수 있습니다. MSI 명령줄 옵션에 대한 자세한 내용은 MSDN(Microsoft Developer Network) 라이브러리 웹 사이트에서 MSI 명령줄 옵션을 검색하여 참조하십시오.

#### 사전 요구 사항

- vRealize Log Insight Windows 에이전트 .msi 파일의 복사본이 있는지 확인합니다. [에이전트 설치 파일 다운로드](#) 항목을 참조하십시오.
- 설치를 수행할 수 있는 사용 권한이 있으며 Windows 시스템에서 서비스를 시작했는지 확인합니다.
- 자동 설치 옵션 /quiet 또는 /qn을 사용하는 경우 관리자로 설치 프로그램을 실행하는지 확인하십시오. 관리자가 아닌 사용자가 자동 설치를 실행하는 경우 설치가 관리자 권한을 요청하지 않고 실패합니다. 진단 목적으로 로깅 옵션 및 매개 변수 /l xv\* *file\_name*을 사용하십시오.

#### 절차

- 1 vRealize Log Insight Windows 에이전트를 설치 또는 업데이트할 Windows 시스템에 로그인합니다.
- 2 **명령 프롬프트** 창을 엽니다.
- 3 vRealize Log Insight Windows 에이전트 .msi 파일이 있는 디렉토리로 변경합니다.
- 4 다음 형식의 명령을 사용하여 기본값으로 설치하거나 업데이트합니다. *version-build\_number*를 버전 및 빌드 번호로 바꿉니다.

/quiet 옵션은 해당 명령을 자동으로 실행하며 /l xv 옵션은 현재 디렉토리에 로그 파일을 생성합니다.

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

- 5 (선택 사항) vRealize Log Insight Windows 에이전트 서비스를 실행할 사용자 서비스 계정을 지정합니다.

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

**참고** SERVICEACCOUNT 매개 변수에 제공되는 계정에는 **서비스로 로그인** 권한 및 %ProgramData%\VMware\Log Insight Agent 디렉토리에 대한 전체 쓰기 액세스 권한이 부여됩니다. 지원되는 계정이 없으면 새로 생성됩니다. 사용자 이름은 20자를 초과해서는 안 됩니다. SERVICEACCOUNT 매개 변수를 지정하지 않으면 vRealize Log Insight Windows 에이전트 서비스가 LocalSystem 서비스 계정 아래에 설치 또는 업데이트됩니다.

- 6 (선택 사항) 필요에 따라 다음 명령줄 옵션의 값을 지정할 수 있습니다.

옵션	설명
<b>SERVERHOST=hostname</b>	vRealize Log Insight 가상 장치의 IP 주소 또는 호스트 이름입니다.
<b>SERVERPROTO=protocol</b>	에이전트의 이벤트가 vRealize Log Insight 서버로 전송되는 데 사용되는 프로토콜입니다. 가능한 값은 cfapi 및 syslog입니다. 기본값은 cfapi입니다.
<b>SERVERPORT=portnumber</b>	에이전트의 이벤트가 vRealize Log Insight 또는 타사 서버로 전송되는 데 사용되는 통신 포트입니다. 기본적으로 에이전트는 SSL 및 프로토콜에 설정된 옵션을 기반으로 적절한 포트를 사용합니다. 아래 목록에 제공된 기본 포트 값을 참조하십시오. 이러한 기본값과 다른 경우에만 포트 옵션을 지정해야 합니다. <ul style="list-style-type: none"> <li>■ SSL이 사용되도록 설정된 cfapi: 9543</li> <li>■ SSL이 사용되지 않도록 설정된 cfapi: 9000</li> <li>■ SSL이 사용되도록 설정된 syslog: 6514</li> <li>■ SSL이 사용되지 않도록 설정된 syslog: 514</li> </ul>
<b>SERVICEACCOUNT=account-name</b>	Log Insight Windows Agent 서비스가 실행되는 사용자 서비스 계정입니다. <b>참고</b> 설치 관리자가 제대로 실행되려면 SERVICEACCOUNT 매개 변수에 제공된 계정에 <b>서비스로 로그인</b> 권한 및 %ProgramData%\VMware\Log Insight Agent 디렉토리에 대한 쓰기 액세스 권한이 있어야 합니다. SERVICEACCOUNT 매개 변수를 지정하지 않으면 vRealize Log Insight Windows 에이전트 서비스가 LocalSystem 서비스 계정 아래에 설치됩니다.
<b>SERVICEPASSWORD=password</b>	사용자 서비스 계정의 암호입니다.
<b>AUTOUPDATE={yes no}</b>	에이전트에 대한 자동 업데이트를 사용하거나 사용하지 않도록 설정합니다. 또한 자동 업데이트를 완전하게 사용하도록 설정하려면 vRealize Log Insight 서버에서도 자동 업데이트를 사용하도록 설정해야 합니다. 기본값은 yes입니다.
<b>LIAGENT_SSL={yes no}</b>	안전한 연결을 사용합니다. SSL을 사용하는 경우, 에이전트는 TLS 1.2 프로토콜을 사용하여 서버와 통신합니다. 기본값은 yes입니다.

## 결과

명령이 vRealize Log Insight Windows 에이전트를 Windows 서비스로 설치 또는 업데이트합니다. vRealize Log Insight Windows 에이전트 서비스는 Windows 시스템이 시작될 때 시작됩니다.

## 다음에 수행할 작업

설정된 명령줄 매개 변수가 liagent.ini 파일에 올바르게 적용되었는지 확인합니다. [Log Insight Windows Agent 구성](#) 을 참조하십시오.

## 여러 시스템에 Log Insight Windows Agent 배포

Windows 도메인에 있는 대상 컴퓨터에서 Log Insight Windows Agent의 대량 배포를 수행할 수 있습니다.

### 절차

#### 1 변환 파일을 생성하여 여러 vRealize Log Insight Windows 에이전트 배포

여러 에이전트 배포의 일부로, 배포에 대한 구성 매개 변수를 지정하는 변환 파일을 생성해야 합니다. .mst 변환 파일은 에이전트를 설치할 때 .msi 파일에 적용됩니다. 매개 변수에는 에이전트에 대한 대상 서버, Log Insight 에이전트 서비스를 설치 및 시작하기 위한 통신 프로토콜, 포트 및 사용자 계정이 포함됩니다.

#### 2 vRealize Log Insight Windows 에이전트의 여러 인스턴스 배포

vRealize Log Insight Windows 에이전트의 여러 인스턴스를 Windows 도메인 내의 대상 컴퓨터에 배포할 수 있습니다.

### 변환 파일을 생성하여 여러 vRealize Log Insight Windows 에이전트 배포

여러 에이전트 배포의 일부로, 배포에 대한 구성 매개 변수를 지정하는 변환 파일을 생성해야 합니다. .mst 변환 파일은 에이전트를 설치할 때 .msi 파일에 적용됩니다. 매개 변수에는 에이전트에 대한 대상 서버, Log Insight 에이전트 서비스를 설치 및 시작하기 위한 통신 프로토콜, 포트 및 사용자 계정이 포함됩니다.

매개 변수에는 에이전트에 대한 대상 서버, Log Insight 에이전트 서비스를 설치 및 시작하기 위한 통신 프로토콜, 포트 및 사용자 계정이 포함됩니다.

### 사전 요구 사항

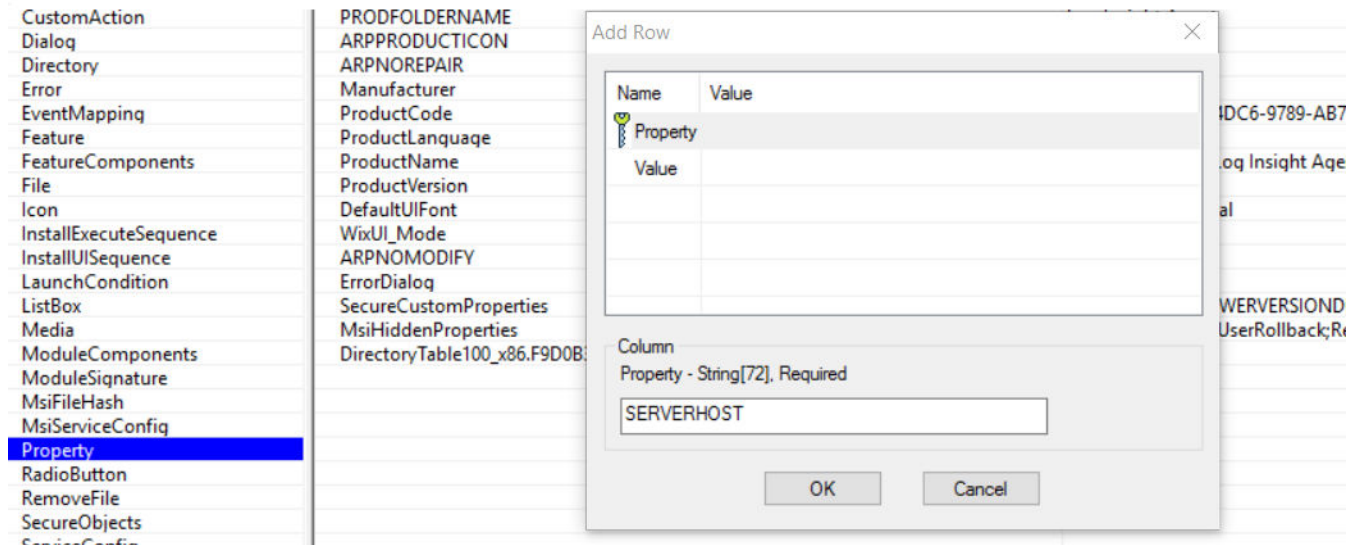
- vRealize Log Insight Windows .msi 파일의 복사본이 있는지 확인합니다. [에이전트 설치 파일 다운로드](#) 항목을 참조하십시오.
- Orca 데이터베이스 편집기를 다운로드하여 설치합니다. <http://support.microsoft.com/kb/255905> 를 참조하십시오.

### 절차

- 1 Orca 편집기에서 vRealize Log Insight Windows 에이전트 .msi 파일을 열고 **변환 > 새 변환**을 선택합니다.

- 2 속성 테이블을 편집하고 사용자 지정된 설치 또는 업그레이드를 위해 필요한 매개 변수 및 값을 추가합니다.

그림 3-1. 속성 테이블



- 속성을 클릭합니다.
- 테이블 드롭다운 메뉴에서 **행 추가**를 선택합니다.
- [행 추가] 대화 상자에 속성 이름 및 값을 입력합니다.

다음 표에 매개 변수가 나와 있습니다.

매개 변수	설명
<b>SERVERHOST</b>	vRealize Log Insight 가상 장치의 IP 주소 또는 호스트 이름입니다. 기본값은 <b>loginsight</b> 입니다.
<b>SERVERPROTO</b>	에이전트의 이벤트가 vRealize Log Insight 서버로 전송되는 데 사용되는 프로토콜입니다. 가능한 값은 <b>cfapi</b> 및 <b>syslog</b> 입니다. 기본값은 <b>cfapi</b> 입니다.
<b>SERVERPORT</b>	에이전트의 이벤트가 vRealize Log Insight 또는 타사 서버로 전송되는 데 사용되는 통신 포트입니다. 기본적으로 에이전트는 SSL 및 프로토콜에 설정된 옵션을 기반으로 적절한 포트를 사용합니다. 아래 목록에 제공된 기본 포트 값을 참조하십시오. 이러한 기본값과 다른 경우에만 포트 옵션을 지정해야 합니다. <ul style="list-style-type: none"> <li>■ SSL이 사용되도록 설정된 <b>cfapi</b>: 9543</li> <li>■ SSL이 사용되지 않도록 설정된 <b>cfapi</b>: 9000</li> <li>■ SSL이 사용되도록 설정된 <b>syslog</b>: 6514</li> <li>■ SSL이 사용되지 않도록 설정된 <b>syslog</b>: 514</li> </ul>

매개 변수	설명
<b>SERVICEACCOUNT</b>	Log Insight Windows Agent 서비스가 실행되는 사용자 서비스 계정입니다.  <b>참고</b> 설치 관리자가 제대로 실행되려면 SERVICEACCOUNT 매개 변수에 제공된 계정에 <b>서비스로 로그인</b> 권한 및 %ProgramData%\VMware\Log Insight Agent 디렉토리에 대한 쓰기 액세스 권한이 있어야 합니다. SERVICEACCOUNT 매개 변수를 지정하지 않으면 vRealize Log Insight Windows 에이전트 서비스가 LocalSystem 서비스 계정 아래에 설치됩니다.
<b>SERVICEPASSWORD</b>	사용자 서비스 계정의 암호입니다.
<b>AUTOUPDATE</b>	에이전트에 대한 자동 업데이트를 사용하거나 사용하지 않도록 설정합니다. 또한 자동 업데이트를 완전하게 사용하도록 설정하려면 vRealize Log Insight 서버에서도 자동 업데이트를 사용하도록 설정해야 합니다. 기본값은 yes입니다.
<b>LIAGENT_SSL</b>	안전한 연결을 사용합니다. SSL을 사용하는 경우, 에이전트는 TLS 1.2 프로토콜을 사용하여 서버와 통신합니다. 기본값은 yes입니다.

### 3 변환 > 변환 생성을 선택하고 .mst 변환 파일을 저장합니다.

#### 다음에 수행할 작업

.msi 및 .mst 파일을 사용하여 vRealize Log Insight Windows 에이전트를 배포합니다.

### vRealize Log Insight Windows 에이전트의 여러 인스턴스 배포

vRealize Log Insight Windows 에이전트의 여러 인스턴스를 Windows 도메인 내의 대상 컴퓨터에 배포할 수 있습니다.

클라이언트 시스템을 두 번 재부팅해야 하는 이유에 대해서는 <http://support.microsoft.com/kb/305293>을 참조하십시오.

#### 사전 요구 사항

- 도메인 컨트롤러에 대한 관리자 계정 또는 관리 권한이 있는 계정을 보유하고 있는지 확인합니다.
- vRealize Log Insight Windows 에이전트 .msi 파일의 복사본이 있는지 확인합니다. **에이전트 설치 파일 다운로드** 항목을 참조하십시오.
- <http://support.microsoft.com/kb/887405> 및 <http://support.microsoft.com/kb/816102>에 설명된 절차를 숙지합니다.

#### 절차

- 1 관리자 계정 또는 관리 권한이 있는 계정으로 도메인 컨트롤러에 로그인합니다.
- 2 배포 지점을 생성하고 해당 배포 지점에 vRealize Log Insight Windows 에이전트 .msi 파일을 복사합니다.
- 3 그룹 정책 관리 콘솔을 열고 vRealize Log Insight Windows 에이전트 .msi 파일을 배포하기 위한 그룹 정책 개체를 생성합니다.
- 4 소프트웨어 배포용 그룹 정책 개체를 편집하고 패키지를 할당합니다.



- 5 (선택 사항) 배포 전에 .mst 파일을 생성한 경우 **GPO 속성** 창의 **수정** 탭에서 .mst 구성 파일을 선택합니다. 그리고 고급 방법을 사용하여 .msi 패키지를 배포할 그룹 정책 개체를 편집합니다.
- 6 (선택 사항) vRealize Log Insight Windows 에이전트를 업그레이드합니다.
  - a 업그레이드 .msi 파일을 배포 지점에 복사합니다.
  - b 그룹 정책 개체 **속성** 창에서 **업그레이드** 탭을 클릭합니다.
  - c 처음 설치된 버전의 .msi 파일을 업그레이드 대상 섹션에 있는 패키지에 추가합니다.
- 7 도메인 사용자가 포함된 특정 보안 그룹에 vRealize Log Insight Windows 에이전트를 배포합니다.
- 8 도메인 컨트롤러에서 모든 그룹 정책 관리 콘솔 및 그룹 정책 관리 편집기 창을 닫고 클라이언트 시스템들을 다시 시작합니다.  
빠른 로그인 최적화가 사용하도록 설정된 경우 클라이언트 시스템을 두 번 재부팅합니다.
- 9 vRealize Log Insight Windows 에이전트가 클라이언트 시스템에 로컬 서비스로 설치되었는지 확인합니다.  
vRealize Log Insight Windows 에이전트의 여러 인스턴스를 배포하기 위해 .mst 파일을 사용하도록 SERVICEACCOUNT 및 SERVICEPASSWORD 매개 변수를 구성한 경우, vRealize Log Insight Windows 에이전트가 지정된 사용자 계정으로 클라이언트 시스템에 설치되었는지 확인합니다.

#### 다음에 수행할 작업

vRealize Log Insight Windows 에이전트의 여러 인스턴스가 제대로 작동하지 않는 경우 [Log Insight Windows Agent](#)의 대량 배포가 실패함을 참조하십시오.

## vRealize Log Insight Linux 에이전트 RPM 패키지 설치 또는 업데이트

vRealize Log Insight Linux 에이전트를 루트 또는 루트가 아닌 사용자로 설치하거나 업데이트할 수 있으며 설치 도중 구성 매개 변수를 설정할 수 있습니다. 설치 후 설치된 버전을 확인할 수 있습니다.

#### 사전 요구 사항

- 설치 기본값에 대해 알아보고 [Linux에서 vRealize Log Insight 에이전트를 설치하기 위한 명령줄 옵션](#)에서 기본값을 변경하는 방법을 알아보십시오.
- **루트**로 로그인하거나 sudo를 사용하여 콘솔 명령을 실행합니다.
- vRealize Log Insight Linux 에이전트가 작동하려면 syslog 및 네트워킹 서비스에 대한 액세스 권한이 있어야 합니다. vRealize Log Insight Linux 에이전트를 설치하고 실행 수준 3 및 5에서 실행합니다. vRealize Log Insight Linux 에이전트를 다른 실행 수준에서 작동하려면 시스템을 그에 맞게 구성합니다.

## 절차

1 콘솔에서 에이전트를 설치하거나 업그레이드할 수 있습니다.

- 기본 구성 설정을 사용하여 vRealize Log Insight Linux 에이전트를 설치하려면 콘솔을 열고 다음 명령을 실행합니다.

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

- 현재 구성 설정을 변경하지 않고 에이전트를 업그레이드하려면 콘솔을 열고 다음 명령을 실행합니다.

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

2 (선택 사항) 업데이트 중에 설치의 기본 구성 값 또는 현재 구성 값을 재정의할 수 있습니다. 설치 또는 업그레이드 명령의 일부로 옵션을 지정하여 이러한 작업을 수행할 수 있습니다.

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

3 (선택 사항) 다음 명령을 실행하여 설치된 버전을 확인합니다.

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

## 예제: Linux 에이전트 설치 및 업데이트 예

- 다음 명령은 Linux RPM 배포에 대한 vRealize Log Insight Agent를 설치합니다. 이 명령은 별도 서버에 에이전트를 설치하고, 기본이 아닌 포트 번호를 할당하고, vRealize Log Insight Agent 사용자를 생성합니다.

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

- 다음 명령은 지정된 rpm 파일로 에이전트를 업데이트합니다. 현재 에이전트 구성은 변경되지 않습니다.

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

## vRealize Log Insight Linux 에이전트 DEB 패키지 설치 또는 업데이트

명령줄 또는 debconf 데이터베이스를 통해 vRealize Log Insight Linux 에이전트 DEB(Debian) 패키지를 설치하거나 업데이트할 수 있습니다. 설치 후 설치된 버전을 확인할 수 있습니다.

## 사전 요구 사항

- **Linux에서 vRealize Log Insight 에이전트를 설치하기 위한 명령줄 옵션** 에서 설치 기본값 및 변경 방법을 숙지하십시오.
- **루트**로 로그인하거나 **sudo**를 사용하여 콘솔 명령을 실행합니다.
- vRealize Log Insight Linux 에이전트에 **syslog** 및 네트워크 서비스에 대한 액세스 권한(에이전트 작동에 필요함)이 있는지 확인합니다. 기본적으로 vRealize Log Insight Linux 에이전트는 실행 수준 2, 3, 4 및 5에서 실행하고 실행 수준 0, 1 및 6에서 중지합니다.
- 자세한 내용 및 예는 **Debian Linux용 에이전트 설치 사용자 지정**을 참조하십시오.

## 절차

- 1 vRealize Log Insight Linux 에이전트를 설치하거나 업데이트하려면 콘솔을 열고 `dpkg -i package_name` 명령을 실행합니다.

`package_name`은(는) **vmware-log-insight-agent-** 접두사와 다운로드 버전의 버전 빌드 번호로 구성됩니다.

다음 명령 형식은 기본값을 사용하여 패키지를 설치합니다.

```
dpkg -i vmware-log-insight-agent-버전-빌드 번호_all.deb
```

- 2 (선택 사항) 다음 명령을 실행하여 설치된 버전을 확인합니다.

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

## 예

- 명령줄에서 연결 프로토콜을 구성합니다.

기본 연결 프로토콜을 재정의하려면 다음 예와 같이 **SERVERPROTO** 변수를 사용합니다.

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

## Debian Linux용 에이전트 설치 사용자 지정

명령 옵션을 사용하여 설치를 위한 현재 구성 값을 재정의하거나 **debconf** 데이터베이스를 구성하여 설치를 사용자 지정할 수 있습니다.

## 명령줄에서 사용자 지정

명령줄에서 설치를 구성하려면 다음 형식의 명령을 사용합니다.

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

전체 옵션 목록은 [Linux에서 vRealize Log Insight 에이전트를 설치하기 위한 명령줄 옵션 항목](#)을 참조하십시오.

다음 예에서는 명령줄에서 수행되는 몇 가지 일반적인 구성을 보여 줍니다.

- 대상 vRealize Log Insight 서버를 지정합니다.
- 설치 도중 대상을 설정하려면 다음 예와 같이 `sudo` 명령을 실행하고 `hostname`을 vRealize Log Insight 서버의 IP 주소 또는 호스트 이름으로 바꿉니다.

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

설치 도중 `--force-confold` 플래그를 사용하도록 설정한 경우 이외에는 새 버전으로 업데이트할 때마다 시스템이 `liagent.ini` 구성 파일을 유지할지 아니면 바꿀지 물어봅니다. 다음의 시스템 메시지가 나타납니다.

```
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] ?
```

기존 구성을 유지하려면 `[default=N]`을 사용합니다. 명령줄에서 전달된 추가적인 매개 변수는 계속 적용됩니다.

- 연결 프로토콜을 구성합니다.

기본 연결 프로토콜을 재정의하려면 다음 예와 같이 `SERVERPROTO` 변수를 사용합니다.

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

- 연결 포트를 구성합니다.

기본 연결 포트를 재정의하려면 다음 예와 같이 `SERVERPORT` 변수의 값을 설치 관리자에 제공합니다.

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

- 루트가 아닌 사용자로 에이전트를 실행합니다.

**루트가 아닌** 사용자로 vRealize Log Insight Linux 에이전트를 실행하려면 `sudo` 명령을 실행합니다.

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

지정한 사용자가 없는 경우 vRealize Log Insight Linux 에이전트가 설치 도중 해당 사용자 계정을 생성합니다. 생성된 계정은 Agent를 제거해도 삭제되지 않습니다. LIAGENTUSER=*non\_root\_user* 매개 변수로 Linux 에이전트를 설치하고 LIAGENTUSER=*non\_root\_user2* 매개 변수로 업그레이드하려고 하면 *non\_root\_user2* 사용자에게는 *non\_root\_user* 사용자의 사용 권한이 없으므로 충돌이 발생하고 경고가 표시됩니다.

## debconf 데이터베이스에 대한 DEB 패키지 사용자 지정 옵션

에이전트 DEB 패키지도 debconf 데이터베이스를 통해 구성할 수 있습니다. 다음 표는 지원되는 debconf 옵션과 해당하는 vRealize Log Insight 에이전트 DEB 설치 관리자 옵션을 나타냅니다.

명령줄 옵션	Debconf 옵션	설명
SERVERHOST= <i>hostname</i>	vmware-log-insight-agent/serverhost	vRealize Log Insight 가상 장치의 IP 주소 또는 호스트 이름입니다. 기본값은 <b>loginsight</b> 입니다.
SERVERPROTO={ <b>cfapi</b>   <b>syslog</b> }	vmware-log-insight-agent/serverproto	에이전트의 이벤트가 vRealize Log Insight 서버로 전송되는 데 사용되는 프로토콜입니다. 가능한 값은 <b>cfapi</b> 및 <b>syslog</b> 입니다. 기본값은 <b>cfapi</b> 입니다.
SERVERPORT= <i>portnumber</i>	vmware-log-insight-agent/serverport	에이전트의 이벤트가 vRealize Log Insight 또는 타사 서버로 전송되는 데 사용되는 통신 포트입니다. 기본적으로 에이전트는 SSL 및 프로토콜에 설정된 옵션을 기반으로 적절한 포트를 사용합니다. 아래 목록에 제공된 기본 포트 값을 참조하십시오. 이러한 기본값과 다른 경우에만 포트 옵션을 지정해야 합니다. <ul style="list-style-type: none"> <li>■ SSL이 사용되도록 설정된 <b>cfapi</b>: 9543</li> <li>■ SSL이 사용되지 않도록 설정된 <b>cfapi</b>: 9000</li> <li>■ SSL이 사용되도록 설정된 <b>syslog</b>: 6514</li> <li>■ SSL이 사용되지 않도록 설정된 <b>syslog</b>: 514</li> </ul>
LIAGENT_INITSYSTEM={ <b>init</b>   <b>systemd</b> }	log-insight-agent/init_system	설치 시간 동안 에이전트에서는 에이전트를 설치 중인 시스템의 <b>init</b> 시스템 유형을 자동으로 감지합니다. 이 옵션으로 시스템 값 유형을 지정하여 이 동작을 재정의할 수 있습니다. 지원되는 init 시스템은 <b>init</b> 및 <b>systemd</b> 의 두 가지 유형이 있습니다.
LIAGENT_AUTOUPDATE={ <b>yes</b>   <b>no</b> }	vmware-log-insight-agent/auto_update	에이전트에 대한 자동 업데이트를 사용하거나 사용하지 않도록 설정합니다. 또한 자동 업데이트를 완전하게 사용하도록 설정하려면 vRealize Log Insight 서버에서도 자동 업데이트를 사용하도록 설정해야 합니다. 기본값은 <b>yes</b> 입니다.  Linux BIN 패키지에는 자동 업데이트가 지원되지 않습니다.

명령줄 옵션	Debconf 옵션	설명
LI_AGENT_RUNSERVICES	vmware-log-insight-agent/init_system	설치 후 즉시 서비스 liagentd(에이전트) 및 liupdaterd(업데이터)가 기본적으로 시작됩니다. LIAGENT_RUNSERVICES debconf 매개 변수를 <b>no</b> 로 설정하여 서비스가 시작되지 않도록 할 수 있습니다. 기본값은 <b>yes</b> 입니다. 허용되는 유일한 값은 <b>yes</b> 와 <b>no</b> 입니다. <b>1</b> 또는 <b>0</b> 은 지원되지 않는 값입니다.
LIAGENT_SSL	vmware-log-insight-agent/ssl	C
LIAGENTUSER= <i>user-account-name</i>	vmware-log-insight-agent/liagentuser	에이전트가 실행 중인 계정을 지정합니다. 사용자가 없으면 설치 관리자가 정규 사용자로 사용자를 생성합니다. 지정한 사용자 계정이 없는 경우 vRealize Log Insight Linux 에이전트가 설치 도중 해당 사용자 계정을 생성합니다. 생성된 계정은 <b>Agent</b> 를 제거해도 삭제되지 않습니다.  기본적으로 에이전트는 루트 사용자로 실행되도록 설치됩니다.  LIAGENTUSER= <i>non_root_user</i> 매개 변수를 사용하여 에이전트를 설치하고 LIAGENTUSER= <i>non_root_user2</i> 를 사용하여 업그레이드하려고 하는 경우 충돌이 발생합니다. <i>non_root_user2</i> 사용자에게 <i>non_root_user</i> 사용자의 사용 권한이 없으므로 경고가 표시됩니다.  생성된 사용자는 제거하는 동안 삭제되지 않습니다. 수동으로 삭제할 수 있습니다. 이 매개 변수는 에이전트 서비스 전용입니다. 업데이터 서비스는 항상 루트 사용자로 실행됩니다.

## Log Insight Linux Agent 이진 패키지 설치

이진 패키지를 설치할 때는 .bin 파일을 실행 파일로 변경한 다음 에이전트를 설치하는 과정이 포함됩니다.

.bin 패키지를 업그레이드하는 것은 공식적으로 지원되지 않습니다. .bin 패키지를 사용하여 기존 Log Insight Linux Agent를 설치한 경우 /var/lib/loginsight-agent 디렉토리에 있는 liagent.ini 파일의 백업 복사본을 만들어 로컬 구성을 유지하십시오. 백업 복사본을 만든 후 Log Insight Linux Agent를 수동으로 제거합니다. [Log Insight Linux Agent Bin 패키지 수동 제거](#) 항목을 참조하십시오.

.bin 패키지를 사용하여 Linux 에이전트를 설치하는 경우 패키지 설치의 일부로 liagentd라는 이름의 init.d 스크립트가 설치되지만 패키지는 스크립트를 등록하지 않습니다. 스크립트를 수동으로 등록할 수 있습니다.

(/sbin/)service liagentd status 명령을 실행하여 성공적으로 설치되었는지 확인할 수 있습니다.

### 사전 요구 사항

- Log Insight Linux Agent .bin 패키지를 다운로드하여 대상 Linux 시스템에 복사합니다.

- Log Insight Linux Agent에 **syslog** 및 네트워킹 서비스에 대한 액세스 권한이 있는지 확인합니다.
- 기본 구성 값과 설치하는 동안 이러한 값을 변경하는 방법을 읽어보십시오. [Linux에서 vRealize Log Insight 에이전트를 설치하기 위한 명령줄 옵션 항목을 참조하십시오.](#)

## 절차

- 1 콘솔을 열고 **chmod** 명령을 실행하여 **.bin** 파일을 실행 파일로 변경합니다.

*filename-version*을 적합한 버전으로 바꿉니다.

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

- 2 명령 프롬프트에서 **./filename-version.bin** 명령을 실행하여 에이전트를 설치합니다.

*filename-version*을 적합한 버전으로 바꿉니다.

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

- 3 (선택 사항) 설치 도중 대상 vRealize Log Insight 서버를 설정하려면 **sudo SERVERHOST=hostname ./filename-version.bin** 명령을 실행합니다.

*hostname*을 vRealize Log Insight 서버의 IP 주소 또는 호스트 이름으로 바꿉니다.

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

- 4 (선택 사항) 기본 연결 프로토콜을 재정의하려면 다음 예와 같이 **SERVERPROTO** 변수를 사용합니다.

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

- 5 (선택 사항) 기본 연결 포트를 재정의하려면 다음 예와 같이 **SERVERPORT** 변수의 값을 설치 관리자에 제공합니다.

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

- 6 (선택 사항) **루트 이외**의 사용자로 Log Insight Linux Agent를 실행하려면 **sudo** 명령을 실행합니다.

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

지정한 사용자가 없는 경우 Log Insight Linux Agent가 설치 도중 해당 사용자 계정을 생성합니다. 생성된 계정은 Agent를 제거해도 삭제되지 않습니다. **LIAGENTUSER=non\_root\_user** 매개 변수로 Log Insight Linux Agent를 설치하고 **LIAGENTUSER=non\_root\_user2** 매개 변수로 업그레이드하려고 하면 **non\_root\_user2** 사용자에게는 **non\_root\_user** 사용자의 사용 권한이 없으므로 충돌이 발생하고 경고가 표시됩니다.

## Linux에서 vRealize Log Insight 에이전트를 설치하기 위한 명령줄 옵션

명령줄에서 vRealize Log Insight 에이전트를 설치할 때 설치 중에 배포를 구성하는 옵션을 포함할 수 있습니다. 이러한 옵션은 `liagent.ini` 파일의 설정에 해당합니다.

Linux 시스템에서 실행되는 vRealize Log Insight 에이전트를 구성하기 위해 설치 중에 다음 옵션을 사용할 수 있습니다.

옵션	설명
<code>SERVERHOST=hostname</code>	vRealize Log Insight 가상 장치의 IP 주소 또는 호스트 이름입니다. 기본값은 <b>loginsight</b> 입니다.
<code>SERVERPROTO={cfapi syslog}</code>	에이전트의 이벤트가 vRealize Log Insight 서버로 전송되는 데 사용되는 프로토콜입니다. 가능한 값은 <code>cfapi</code> 및 <code>syslog</code> 입니다. 기본값은 <b>cfapi</b> 입니다.
<code>SERVERPORT=portnumber</code>	에이전트의 이벤트가 vRealize Log Insight 또는 타사 서버로 전송되는 데 사용되는 통신 포트입니다. 기본적으로 에이전트는 SSL 및 프로토콜에 설정된 옵션을 기반으로 적절한 포트를 사용합니다. 아래 목록에 제공된 기본 포트 값을 참조하십시오. 이러한 기본값과 다른 경우에만 포트 옵션을 지정해야 합니다. <ul style="list-style-type: none"> <li>■ SSL이 사용되도록 설정된 <b>cfapi</b>: 9543</li> <li>■ SSL이 사용되지 않도록 설정된 <b>cfapi</b>: 9000</li> <li>■ SSL이 사용되도록 설정된 <b>syslog</b>: 6514</li> <li>■ SSL이 사용되지 않도록 설정된 <b>syslog</b>: 514</li> </ul>
<code>LIAGENT_INITSYSTEM={init systemd}</code>	설치 시간 동안 에이전트에서는 에이전트를 설치 중인 시스템의 <code>init</code> 시스템 유형을 자동으로 감지합니다. 이 옵션으로 시스템 값 유형을 지정하여 이 동작을 재정의할 수 있습니다. 지원되는 <code>init</code> 시스템은 <code>init</code> 및 <code>systemd</code> 의 두 가지 유형이 있습니다.
<code>LIAGENT_AUTOUPDATE={yes no}</code>	에이전트에 대한 자동 업데이트를 사용하거나 사용하지 않도록 설정합니다. 또한 자동 업데이트를 완전하게 사용하도록 설정하려면 vRealize Log Insight 서버에서도 자동 업데이트를 사용하도록 설정해야 합니다. 기본값은 <b>yes</b> 입니다.  Linux BIN 패키지에는 자동 업데이트가 지원되지 않습니다.



옵션	설명
<code>LIAGENT_SSL={yes no}</code>	안전한 연결을 사용합니다. SSL을 사용하는 경우, 에이전트는 TLS 1.2 프로토콜을 사용하여 서버와 통신합니다. 기본값은 yes입니다.
<code>LIAGENTUSER=user-account-name</code>	<p>에이전트가 실행 중인 계정을 지정합니다. 사용자가 없으면 설치 관리자가 정규 사용자로 사용자를 생성합니다. 지정한 사용자 계정이 없는 경우 vRealize Log Insight Linux 에이전트가 설치 도중 해당 사용자 계정을 생성합니다. 생성된 계정은 Agent를 제거해도 삭제되지 않습니다.</p> <p>기본적으로 에이전트는 루트 사용자로 실행되도록 설치됩니다. <code>LIAGENTUSER=non_root_user</code> 매개 변수를 사용하여 에이전트를 설치하고 <code>LIAGENTUSER=non_root_user2</code>를 사용하여 업그레이드하려고 하는 경우 충돌이 발생합니다. <code>non_root_user2</code> 사용자에게 <code>non_root_user</code> 사용자의 사용 권한이 없으므로 경고가 표시됩니다.</p> <p>생성된 사용자는 제거하는 동안 삭제되지 않습니다. 수동으로 삭제할 수 있습니다. 이 매개 변수는 에이전트 서비스 전용입니다. 업데이트 서비스는 항상 루트 사용자로 실행됩니다.</p>

## vRealize Log Insight 에이전트의 자동 업데이트

vRealize Log Insight 에이전트의 자동 업데이트를 사용하면 활성 에이전트가 vRealize Log Insight 서버의 에이전트 설치 패키지를 기준으로 업데이트를 확인하고 다운로드하고 자동으로 업데이트할 수 있습니다.

모든 에이전트의 경우는 서버에서, 개별 에이전트 인스턴스에 대해서는 클라이언트에서 자동 업데이트를 사용하도록 설정할 수 있습니다. 에이전트는 활성 상태여야 하며 버전 4.3 이상이어야 합니다.

Linux BIN 패키지에는 자동 업데이트가 지원되지 않습니다.

## 개별 에이전트에 대해 자동 업데이트 사용 안 함 또는 사용

해당 에이전트에 대한 클라이언트 측 구성 파일을 편집하여 개별 에이전트에 대해 자동 업데이트를 사용하거나 사용하지 않도록 설정할 수 있습니다.

기본적으로 에이전트에 대한 클라이언트 측에서 자동 업데이트가 사용되도록 설정됩니다.

### 사전 요구 사항

에이전트는 버전 4.3 이상이어야 합니다.

### 절차

- 1 편집기에서 로컬 `liagent.ini` 파일을 엽니다.

## 2 [update] 섹션을 찾습니다.

다음 예와 비슷합니다.

```
[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 자동 업데이트를 사용하지 않도록 설정하려면 auto\_update=yes의 주석을 취소하고 auto\_update=no 항목으로 변경하십시오.

---

**참고** 에이전트에 대한 자동 업데이트는 기본적으로 사용하도록 설정되어 있습니다. 따라서 auto\_update의 기본값은 주석 처리가 된 경우에도 "예"입니다.

---

## 4 liagent.ini 파일을 저장한 후 닫습니다.

# vRealize Log Insight 에이전트 구성

# 4

에이전트를 배포한 후에는 선택한 vRealize Log Insight 서버에 이벤트를 보내도록 구성하고 통신 프로토콜을 지정하고 다른 매개 변수를 설정할 수 있습니다.

필요에 따라 이 지침을 사용하여 에이전트를 필요에 맞게 구성하십시오.

## ■ Log Insight Windows Agent 구성

Log Insight Windows Agent를 설치한 이후에 구성할 수 있습니다. vRealize Log Insight로 이벤트를 보내고 통신 프로토콜 및 포트를 설정하고 Windows 이벤트 채널을 추가하고 플랫폼 파일 로그 수집을 구성하도록 Log Insight Windows Agent를 구성하려면 `liagent.ini` 파일을 편집합니다. 파일은 `%ProgramData%\VMware\Log Insight Agent` 디렉토리에 있습니다.

## ■ Log Insight Linux Agent 구성

Log Insight Linux Agent를 설치한 이후에 구성할 수 있습니다.

## ■ vRealize Log Insight 에이전트에서 이벤트 필터링

로컬 `liagent.ini` 파일의 `[server|<dest_id>]` 섹션에서 필터 옵션을 사용하여 에이전트가 대상으로 전송하는 정보를 제공할 수 있습니다.

## ■ vRealize Log Insight 에이전트의 정보 전달

에이전트에서 수집한 이벤트를 최대 세 개의 대상으로 전달할 수 있습니다. 대상에는 vRealize Log Insight 서버나 전달자 또는 타사 로그 관리 솔루션이 포함될 수 있습니다.

## ■ vRealize Log Insight 에이전트의 중앙 집중식 구성

여러 vRealize Log Insight 에이전트를 구성할 수 있습니다.

## ■ 에이전트 구성에 대해 공통 값 사용

Windows 또는 Linux 에이전트에 대한 각 에이전트 구성 섹션에 적용되는 공통 매개 변수 값으로 에이전트 구성 파일의 기본값을 재정의할 수 있습니다.

## ■ 로그 구문 분석

에이전트 측 로그 구문 분석기는 원시 로그에서 구조화된 데이터를 추출하여 vRealize Log Insight 서버에 제공합니다. vRealize Log Insight는 로그 구문 분석기를 사용하여 로그를 분석하고, 여기서 정보를 추출하고, 해당 결과를 서버에 표시할 수 있습니다. 로그 구문 분석기는 Windows 및 Linux vRealize Log Insight 에이전트 모두에 대해 구성할 수 있습니다.

## Log Insight Windows Agent 구성

Log Insight Windows Agent를 설치한 이후에 구성할 수 있습니다. vRealize Log Insight로 이벤트를 보내고 통신 프로토콜 및 포트를 설정하고 Windows 이벤트 채널을 추가하고 플랫폼 파일 로그 수집을 구성하도록 Log Insight Windows Agent를 구성하려면 liagent.ini 파일을 편집합니다. 파일은 %ProgramData%\VMware\Log Insight Agent 디렉토리에 있습니다.

### Log Insight Windows Agent의 기본 구성

설치를 마치면 liagent.ini 파일에 Log Insight Windows Agent에 대해 미리 구성된 기본 설정이 포함됩니다.

### Log Insight Windows Agent liagent.ini 기본 구성

ASCII가 아닌 이름과 값을 사용하는 경우 구성을 UTF-8로 저장합니다.

중앙 구성을 사용하는 경우, 이 파일의 최종 구성이 서버의 설정과 연결되어 liagent-effective.ini 파일을 형성합니다.

서버의 에이전트 페이지에서 설정을 구성하는 것이 더 효율적일 수 있습니다.

```
; 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

```

매개 변수	기본값	설명
hostname	LOGINSIGHT	vRealize Log Insight 가상 장치의 IP 주소 또는 호스트 이름입니다. 기본값은 <b>loginsight</b> 입니다.
central_config	yes	이 에이전트에 대한 중앙 집중식 구성을 사용하거나 사용하지 않도록 설정합니다. 중앙 집중식 구성이 사용되지 않도록 설정되면 에이전트는 vRealize Log Insight 서버에서 제공하는 구성을 무시합니다. 허용되는 값은 yes, no, 1 또는 0입니다. 기본값은 yes입니다.
proto	cfapi	에이전트의 이벤트가 vRealize Log Insight 서버로 전송되는 데 사용되는 프로토콜입니다. 가능한 값은 cfapi 및 syslog입니다. 기본값은 <b>cfapi</b> 입니다.
port	9543, 9000, 6514 및 514	에이전트의 이벤트가 vRealize Log Insight 또는 타사 서버로 전송되는 데 사용되는 통신 포트입니다. 기본적으로 에이전트는 SSL 및 프로토콜에 설정된 옵션을 기반으로 적절한 포트를 사용합니다. 아래 목록에 제공된 기본 포트 값을 참조하십시오. 이러한 기본값과 다른 경우에만 포트 옵션을 지정해야 합니다. <ul style="list-style-type: none"> <li>■ SSL이 사용되도록 설정된 cfapi: 9543</li> <li>■ SSL이 사용되지 않도록 설정된 cfapi: 9000</li> <li>■ SSL이 사용되도록 설정된 syslog: 6514</li> <li>■ SSL이 사용되지 않도록 설정된 syslog: 514</li> </ul>
ssl	yes	SSL을 사용하거나 사용하지 않도록 설정합니다. 기본값은 <b>yes</b> 입니다. ssl을 [yes]로 설정하고 포트 값을 설정하지 않은 경우 포트 9543이 자동으로 선택됩니다.
max_disk_buffer	200	Log Insight Windows Agent에서 이벤트 및 자체 로그를 버퍼링하는 데 사용하는 MB 단위의 최대 디스크 공간입니다. 지정된 max_disk_buffer에 도달하면 에이전트를 새로 들어오는 이벤트를 삭제하기 시작합니다.
debug_level	0	로그 세부 정보 수준을 정의합니다. <b>Log Insight Agents</b> 의 로그 세부 정보 수준 정의 항목을 참조하십시오.

매개 변수	기본값	설명
channel	애플리케이션, 보안, 시스템	애플리케이션, 보안 및 시스템 Windows 이벤트 로그 채널은 기본적으로 주석 처리되며 Log Insight Windows Agent는 이러한 채널의 로그를 수집하지 않습니다. <a href="#">Windows 이벤트 채널에서 이벤트 수집 항목을 참조하십시오.</a>
raw_syslog	no	syslog 프로토콜을 사용하는 에이전트의 경우 에이전트가 원시 syslog 이벤트를 수집하여 전송하도록 허용합니다. 기본값은 수집된 이벤트가 사용자 지정 syslog 특성으로 변환됨을 의미하는 no입니다. 이 옵션을 사용하도록 설정하여 syslog 변환 없이 이벤트를 수집합니다. 허용되는 값은 [예] 또는 1, [아니오] 또는 0입니다.

## Windows 이벤트 채널에서 이벤트 수집

Windows 이벤트 채널을 Log Insight Windows Agent 구성에 추가할 수 있습니다. 그러면 Log Insight Windows Agent에서 이벤트를 수집하여 vRealize Log Insight 서버로 보냅니다.

필드 이름은 제한됩니다. 다음 이름은 예약된 것이므로 필드 이름으로 사용할 수 없습니다.

- event\_type
- hostname
- source
- text

### 사전 요구 사항

vRealize Log Insight Windows 에이전트가 설치된 Windows 시스템에 로그인하고 서비스 관리자를 시작하여 vRealize Log Insight 에이전트 서비스가 설치되었는지 확인합니다.

### 절차

- 1 vRealize Log Insight Windows 에이전트의 프로그램 데이터 디렉토리로 이동합니다.  
%ProgramData%\VMware\Log Insight Agent
- 2 텍스트 편집기에서 liagent.ini 파일을 엽니다.

### 3 다음 매개 변수를 추가하고 환경에 맞게 값을 설정합니다.

매개 변수	설명
<code>[winlog section_name]</code>	구성 섹션의 고유 이름입니다.
<code>channel</code>	Windows 애플리케이션에 기본 제공되는 이벤트 뷰어에 표시된 이벤트 채널의 전체 이름입니다. 정확한 채널 이름을 복사하려면 이벤트 뷰어에서 채널을 마우스 오른쪽 버튼으로 클릭하고 <b>속성</b> 을 선택한 후 <b>전체 이름</b> 필드의 콘텐츠를 복사합니다.
<code>enabled</code>	구성 섹션을 사용 또는 사용하지 않도록 설정하는 선택적 매개 변수입니다. 가능한 값은 <b>yes</b> 또는 <b>no</b> 입니다(대/소문자를 구분하지 않음). 기본값은 <b>yes</b> 입니다.
<code>tags</code>	수집된 이벤트의 필드에 사용자 지정 태그를 추가하는 선택적 매개 변수입니다. JSON 표기법을 사용하여 태그를 정의해야 합니다. 태그 이름에는 문자, 숫자 및 밑줄을 포함할 수 있습니다. 태그 이름은 문자 또는 밑줄로만 시작할 수 있으며 64자를 초과할 수 없습니다. 태그 이름은 대/소문자를 구분하지 않습니다. 예를 들어 <code>tags={"tag_name1" : "tag value 1", "Tag_Name1" : "tag value 2" }</code> 를 사용하면 <code>Tag_Name1</code> 은 중복 항목으로 무시됩니다. 태그 이름으로 <code>event_type</code> 및 <code>timestamp</code> 를 사용할 수 없습니다. 동일한 선언 내의 중복 항목은 무시됩니다. 대상이 <code>syslog</code> 서버인 경우 태그는 <code>APP-NAME</code> 필드를 재정의할 수 있습니다. 예를 들어 <code>tags={"appname": "VROPS"}</code> 일 수 있습니다.
<code>whitelist, blacklist</code>	로그 이벤트를 명시적으로 포함 또는 제외하는 선택적 매개 변수입니다.  <b>참고</b> <code>blacklist</code> 옵션은 필드에만 작동하며 텍스트를 차단하는 데 사용할 수 없습니다.
<code>exclude_fields</code>	(선택 사항) 수집 대상에서 개별 필드를 제외하는 매개 변수입니다. 여러 값은 세미 콜론으로 구분된 목록으로 제공할 수 있습니다. 예를 들면 다음과 같습니다.  <code>exclude_fields=EventId; ProviderName</code>

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

### 4 liagent.ini 파일을 저장한 후 닫습니다.

## 예제: 구성

다음 [winlog] 구성 예를 참조하십시오.

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

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



## Windows 이벤트 채널에 대한 필터링 설정

Windows 이벤트 채널에 대한 필터를 설정하여 로그 이벤트를 명시적으로 포함하거나 제외할 수 있습니다.

whitelist 및 blacklist 매개 변수를 사용하여 필터 식을 평가할 수 있습니다. 필터 식은 이벤트 필드 및 연산자로 구성된 부울 식입니다.

---

**참고** blacklist 옵션은 필드에만 작동하며 텍스트를 차단하는 데 사용할 수 없습니다.

---

- whitelist 매개 변수는 필터 식이 0 이외의 값으로 평가되는 로그 이벤트만 수집합니다. 이 매개 변수를 생략하면 값은 암시적으로 1이 됩니다.
- blacklist 매개 변수는 필터 식이 0 이외의 값으로 평가되는 로그 이벤트를 제외합니다. 기본값은 0입니다.

Windows 이벤트 필드 및 연산자의 전체 목록은 [이벤트 필드 및 연산자](#) 항목을 참조하십시오.

### 사전 요구 사항

vRealize Log Insight Windows 에이전트가 설치된 Windows 시스템에 로그인하고 서비스 관리자를 시작하여 vRealize Log Insight 에이전트 서비스가 설치되었는지 확인합니다.

### 절차

- 1 vRealize Log Insight Windows 에이전트의 프로그램 데이터 디렉토리로 이동합니다.

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

- 2 텍스트 편집기에서 liagent.ini 파일을 엽니다.

- 3 [winlog|] 섹션에 whitelist 또는 blacklist 매개 변수를 추가합니다.

예를 들면 다음과 같습니다.

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

- 4 Windows 이벤트 필드 및 연산자를 기반으로 필터 식을 생성합니다.

예를 들면 다음과 같습니다.

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

- 5 liagent.ini 파일을 저장한 후 닫습니다.

## 예제: 필터 구성

다음과 같이 오류 이벤트만 수집하도록 에이전트를 구성할 수 있습니다.

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

다음과 같이 애플리케이션 채널에서 VMware 네트워크 이벤트만 수집하도록 에이전트를 구성할 수 있습니다.

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

다음과 같이 보안 채널에서 특정 이벤트를 제외한 모든 이벤트를 수집하도록 에이전트를 구성할 수 있습니다.

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

## 이벤트 필드 및 연산자

Windows 이벤트 필드 및 연산자를 사용하여 필터 식을 작성할 수 있습니다.

### 필터 식 연산자

연산자	설명
==, !=	같음 및 같지 않음. 숫자 필드 및 문자열 필드 모두에 사용할 수 있습니다.
>, >, <, <=	크거나 같음, 보다 큼, 보다 작음, 작거나 같음. 숫자 필드에만 사용할 수 있습니다.
&,  , ^, ~	비트 AND, OR, XOR 및 보수 연산자. 숫자 필드에만 사용할 수 있습니다.
and, or	논리 AND 및 OR. 단순 식을 결합하여 복합 식을 작성하는 데 사용됩니다.
아님	단항 논리 NOT 연산자. 식의 값을 반전하는 데 사용됩니다.
()	연산 순서를 변경하기 위해 논리 식에 괄호를 사용합니다.

### Windows 이벤트 필드

필터 식에 다음 Windows 이벤트 필드를 사용할 수 있습니다.

필드 이름	필드 유형
호스트 이름	문자열
텍스트	문자열
ProviderName	문자열
EventSourceName	문자열
EventID	숫자

필드 이름	필드 유형
EventRecordID	숫자
채널	문자열
UserID	문자열
수준	숫자 다음과 같은 미리 정의된 상수를 사용할 수 있습니다. <ul style="list-style-type: none"> <li>■ WINLOG_LEVEL_SUCCESS = 0</li> <li>■ WINLOG_LEVEL_CRITICAL = 1</li> <li>■ WINLOG_LEVEL_ERROR = 2</li> <li>■ WINLOG_LEVEL_WARNING = 3</li> <li>■ WINLOG_LEVEL_INFO = 4</li> <li>■ WINLOG_LEVEL_VERBOSE = 5</li> </ul>
작업	숫자
OpCode	숫자
키워드	숫자 다음과 같은 미리 정의된 비트 마스크를 사용할 수 있습니다. <ul style="list-style-type: none"> <li>■ WINLOG_KEYWORD_RESPONSETIME = 0x0001000000000000;</li> <li>■ WINLOG_KEYWORD_WDICONTEXT = 0x0002000000000000;</li> <li>■ WINLOG_KEYWORD_WDIDIAGNOSTIC = 0x0004000000000000;</li> <li>■ WINLOG_KEYWORD_SQM = 0x0008000000000000;</li> <li>■ WINLOG_KEYWORD_AUDITFAILURE = 0x0010000000000000;</li> <li>■ WINLOG_KEYWORD_AUDITSUCCESS = 0x0020000000000000;</li> <li>■ WINLOG_KEYWORD_CORRELATIONHINT = 0x0040000000000000;</li> <li>■ WINLOG_KEYWORD_CLASSIC = 0x0080000000000000;</li> </ul>

## 예

다음은 모든 중요, 오류 및 경고 이벤트를 수집합니다.

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

다음은 보안 채널에서 감사 실패 이벤트만 수집합니다.

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

## 로그 파일에서 이벤트 수집

하나 이상의 로그 파일에서 이벤트를 수집하도록 vRealize Log Insight Windows 에이전트를 구성할 수 있습니다.

필드 이름은 제한됩니다. 다음 이름은 예약된 것이므로 필드 이름으로 사용할 수 없습니다.

- event\_type

- hostname
- source
- text

에이전트 정보에 대해 최대 3개의 대상을 사용하고 정보를 전송하기 전에 필터링할 수 있습니다. [vRealize Log Insight 에이전트의 정보 전달 항목을 참조하십시오.](#)

---

## 참고

- 1,000개 이상의 파일과 같은 많은 수의 파일을 모니터링하면 에이전트의 리소스 사용량이 더 높아지고 호스트 시스템의 전체 성능에 영향을 미칩니다. 이를 방지하려면 패턴 및 glob을 사용하여 필요한 파일만 모니터링하도록 에이전트를 구성하거나 이전 로그 파일을 아카이브하십시오. 많은 수의 파일을 모니터링해야 하는 경우 CPU 및 RAM과 같은 호스트 매개 변수를 늘려 보십시오.
  - 에이전트는 디렉토리를 암호화한 사용자가 실행한 경우에만 암호화된 디렉토리에서 수집할 수 있습니다.
  - 에이전트는 정적 디렉토리 구조만 지원합니다. 디렉토리의 이름이 변경되었거나 추가된 경우 구성이 디렉토리를 포함하는 경우 디렉토리 모니터링을 시작하려면 에이전트를 다시 시작해야 합니다.
- 

## 사전 요구 사항

vRealize Log Insight Windows 에이전트가 설치된 Windows 시스템에 로그인하고 서비스 관리자를 시작하여 vRealize Log Insight 에이전트 서비스가 설치되었는지 확인합니다.

## 절차

- 1 vRealize Log Insight Windows 에이전트의 프로그램 데이터 디렉토리로 이동합니다.  
`%ProgramData%\VMware\Log Insight Agent`
- 2 텍스트 편집기에서 liagent.ini 파일을 엽니다.

### 3 파일의 [server |<dest\_id>] 섹션을 찾습니다. 구성 매개 변수를 추가하고 환경에 맞게 값을 설정합니다.

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

매개 변수	설명
<b>[filelog section_name]</b>	구성 섹션의 고유 이름입니다.
<b>directory=full-path-to-log-file</b>	<p>로그 파일 디렉토리의 전체 경로입니다. Glob 패턴이 지원됩니다. 구성 예:</p> <ul style="list-style-type: none"> <li>■ D:\Logs\new_test_logs 디렉토리의 모든 하위 디렉토리에서 수집하려면 다음을 사용하십시오. <code>directory=D:\Logs\new_test_logs\*</code></li> <li>■ 하위 디렉토리에 고유한 하위 디렉토리가 있는 경우 다음 구성을 사용하여 모든 하위 디렉토리를 모니터링합니다. <code>directory=D:\Logs\new_test_logs\*\*</code></li> </ul> <p><b>참고</b> 파일 및 디렉토리 수를 제한하고 리소스 소비량이 높아지지 않도록 하려면 첫 번째 또는 두 번째 수준 디렉토리(예: <code>directory=c:/tmp/*</code> 또는 <code>directory=c:\Logs\*</code>)에 대해 디렉토리 glob를 정의하면 안 됩니다. 디렉토리 경로는 두 수준 이상이어야 합니다.</p> <p>존재하지 않는 디렉토리에 대한 경로를 정의할 수 있습니다. 그러면 에이전트는 디렉토리 및 파일이 생성된 후에 해당 디렉토리에 로그 파일을 수집합니다.</p> <p>하나 이상의 다른 구성 섹션 아래에서 동일한 디렉토리를 정의하여 동일한 파일에서 여러 번 로그를 수집할 수 있습니다. 이 프로세스를 통해 여러 태그 및 필터를 동일한 이벤트 소스에 적용할 수 있습니다.</p> <p><b>참고</b> 해당 섹션에 대해 동일한 구성을 사용하는 경우 중복 이벤트가 서버 측에서 발견됩니다.</p>
<b>include=file_name; ...</b>	<p>(선택 사항) 데이터를 수집할 파일 이름 또는 파일 마스크(glob 패턴)의 이름입니다. 값은 세미콜론으로 구분된 목록으로 제공할 수 있습니다. 기본값은 *이며, 이는 모든 파일이 포함된다는 의미입니다. 매개 변수는 대/소문자를 구분합니다.</p> <p>파일 마스크(glob 패턴)를 사용하여 동일한 이름 지정 규칙을 따르는 여러 파일을 단일 파일 이름으로 그룹화할 수 있습니다. 예를 들어 vRealize Ops Analytics.log 및 vRealize Ops Collector.log와 같은 공백을 포함하는 파일 이름을 <code>vRealize?Ops?Analytics*.log</code> 또는 <code>vRealize*.log</code>로 지정할 수 있습니다. 파일 마스크를 사용하여 Linux 및 Windows 호스트의 에이전트 구성에 허용되는 파일 이름을 지정할 수 있습니다.</p> <p>기본적으로 .zip 및 .gz 파일은 수집 대상에서 제외됩니다.</p> <p><b>중요</b> 회전 로그 파일을 수집하는 경우 include 및 exclude 매개 변수를 사용하여 기본 파일 및 회전 파일 모두와 일치하는 glob 패턴을 지정합니다. glob 패턴이 기본 로그 파일과만 일치하면 회전하는 동안 이벤트가 vRealize Log Insight 에이전트에서 누락될 수 있습니다. vRealize Log Insight 에이전트는 회전 파일의 올바른 순서를 자동으로 확인하고 올바른 순서로 이벤트를 vRealize Log Insight 서버로 전송합니다. 예를 들어 기본 로그 파일의 이름이 myapp.log로 지정되고 회전 로그의 이름이 myapp.log.1, myapp.log.2 등으로 지정되는 경우 다음 include 패턴을 사용할 수 있습니다.</p> <pre>include= myapp.log;myapp.log.*</pre>

매개 변수	설명
<b>exclude=regular_expression</b>	(선택 사항) 수집에서 제외할 파일 이름 또는 파일 마스크(glob 패턴)입니다. 값은 세미콜론으로 구분된 목록으로 제공할 수 있습니다. 기본값은 비워 두는 것이며, 이는 제외할 파일이 없다는 의미입니다.
<b>event_marker=regular_expression</b>	(선택 사항) 로그 파일에서 이벤트의 시작을 나타내는 정규식입니다. 이를 생략하면 새 행이 기본값이 됩니다. 입력하는 식은 Perl 정규식 구문을 사용해야 합니다.  <b>참고</b> 따옴표(" ") 등의 기호는 정규식의 래퍼로 처리되지 않습니다. 이러한 기호는 패턴의 일부로 처리됩니다.  vRealize Log Insight 에이전트는 실시간 수집에 최적화되어 있기 때문에 내부 지연과 함께 작성된 부분 로그 메시지는 여러 이벤트로 분할될 수 있습니다. 확인된 새 event_marker 없이 로그 파일 추가가 200ms 이상 중지되어 있으면 부분 이벤트가 완료, 구문 분석 및 전달된 것으로 처리됩니다. 이 시간 논리는 구성할 수 없으며 event_marker 설정보다 우선시 됩니다. 로그 파일 appender가 전체 이벤트를 플러시해야 합니다.
<b>enabled=yes no</b>	(선택 사항) 구성 섹션을 사용 또는 사용하지 않도록 설정하는 매개 변수입니다. 가능한 값은 yes 또는 no입니다. 기본값은 yes입니다.
<b>charset=char-encoding-type</b>	(선택 사항) 에이전트가 모니터링하는 로그 파일의 문자 인코딩입니다. 가능한 값은 다음과 같습니다. <ul style="list-style-type: none"> <li>■ UTF-8</li> <li>■ UTF-16LE</li> <li>■ UTF-16BE</li> </ul> 기본값은 UTF-8입니다.
<b>tags={"tag-name": "tag-value", ...}</b>	(선택 사항) 수집된 이벤트의 필드에 사용자 지정 태그를 추가하는 매개 변수입니다. JSON 표기법을 사용하여 태그를 정의해야 합니다. 태그 이름에는 문자, 숫자 및 밑줄을 포함할 수 있습니다. 태그 이름은 문자 또는 밑줄로만 시작할 수 있으며 64자를 초과할 수 없습니다. 태그 이름은 대/소문자를 구분하지 않습니다. 예를 들어 tags={"tag_name1": "tag value 1", "Tag_Name1": "tag value 2"}를 사용하면 Tag_Name1은 중복 항목으로 무시됩니다. 태그 이름으로 event_type 및 timestamp를 사용할 수 없습니다. 동일한 선언 내의 중복 항목은 무시됩니다.  대상이 syslog 서버인 경우 태그는 APP-NAME 필드를 재정의할 수 있습니다. 예를 들어 tags={"appname":"VROPS"}일 수 있습니다.
<b>exclude_fields</b>	(선택 사항) 수집 대상에서 개별 필드를 제외하는 매개 변수입니다. 여러 값을 세미콜론 또는 쉼표로 구분된 목록으로 제공할 수 있습니다. 예를 들면 다음과 같습니다. <ul style="list-style-type: none"> <li>■ exclude_fields=hostname; filepath</li> <li>■ exclude_fields=type; size</li> <li>■ exclude_fields=type, size</li> </ul>
<b>raw_syslog=Yes No</b>	syslog 프로토콜을 사용하는 에이전트의 경우 이 옵션은 에이전트가 원시 syslog 이벤트를 수집하여 전송하도록 허용합니다. 기본값은 수집된 이벤트가 사용자 지정 syslob 특성으로 변환됨을 의미하는 [아니요]입니다. 이 옵션을 사용하도록 설정하여 syslog 변환 없이 이벤트를 수집합니다.

## 예제: 구성

```
[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]
```

## Windows 로그 파일 채널 필터링 설정

Windows 로그 파일에 대한 필터를 설정하여 로그 이벤트를 명시적으로 포함하거나 제외할 수 있습니다.

whitelist 및 blacklist 매개 변수를 사용하여 필터 식을 평가할 수 있습니다. 필터 식은 이벤트 필드 및 연산자로 구성된 부울 식입니다.

---

**참고** blacklist 옵션은 필드에만 작동하며 텍스트를 차단하는 데 사용할 수 없습니다.

---

- whitelist 매개 변수는 필터 식이 0 이외의 값으로 평가되는 로그 이벤트만 수집합니다. 이 매개 변수를 생략하면 값은 암시적으로 1이 됩니다.
- blacklist 매개 변수는 필터 식이 0 이외의 값으로 평가되는 로그 이벤트를 제외합니다. 기본값은 0입니다.

Windows 이벤트 필드 및 연산자의 전체 목록은 [이벤트 필드 및 연산자](#) 항목을 참조하십시오.

### 사전 요구 사항

vRealize Log Insight Windows 에이전트가 설치된 Windows 시스템에 로그인하고 서비스 관리자를 시작하여 vRealize Log Insight 에이전트 서비스가 설치되었는지 확인합니다.

### 절차

- 1 vRealize Log Insight Windows 에이전트의 프로그램 데이터 디렉토리로 이동합니다.  
`%ProgramData%\VMware\Log Insight Agent`
- 2 텍스트 편집기에서 `liagent.ini` 파일을 엽니다.

- 3 [filelog|] 섹션에 whitelist 또는 blacklist 매개 변수를 추가합니다.

예:

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

- 4 Windows 이벤트 필드 및 연산자를 기반으로 필터 식을 생성합니다.

예를 들면 다음과 같습니다.

```
whitelist = myServer
```

- 5 liagent.ini 파일을 저장한 후 닫습니다.

#### 예제: 필터 구성

server\_name이 있는 Apache 로그만 수집하도록 에이전트를 구성할 수 있습니다.

```
[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"
```

## Log Insight Windows Agent로 이벤트 전달

Windows 시스템의 이벤트를 Log Insight Windows Agent가 실행 중인 시스템으로 전달할 수 있습니다.

Windows 이벤트 전달을 사용하여 여러 Windows 시스템의 이벤트를 Log Insight Windows Agent가 설치된 단일 시스템으로 전달할 수 있습니다. 그런 다음 전달된 모든 이벤트를 수집하고 vRealize Log Insight 서버에 보내도록 Log Insight Windows Agent를 구성할 수 있습니다.

Windows 이벤트 전달 기능을 숙지합니다. <http://technet.microsoft.com/en-us/library/cc748890.aspx> 및 [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)를 참조하십시오.

#### 사전 요구 사항

Windows 이벤트 채널에서 이벤트 수집을 참조하십시오.

#### 절차

- 1 전달된 이벤트를 수신하는 Windows 이벤트 채널로부터 이벤트를 수집하도록 Log Insight Windows Agent 구성에 새 섹션을 추가합니다.

기본 채널 이름은 ForwardedEvents입니다.

- 2 Windows 이벤트 전달을 설정합니다.



## 다음에 수행할 작업

vRealize Log Insight 웹 사용자 인터페이스로 이동하고 전달된 이벤트가 도착했는지 확인합니다.

## Log Insight Linux Agent 구성

Log Insight Linux Agent를 설치한 이후에 구성할 수 있습니다.

중앙 집중식 에이전트 구성을 사용하여 vRealize Log Insight 서버로 이벤트를 보내고, 통신 프로토콜 및 포트를 지정하고, 플랫폼 파일 로그 수집을 구성하도록 에이전트를 설정할 수 있습니다. `liagent.ini` 파일의 위치는 [Log Insight Agents의 로그 세부 정보 수준 정의](#)를 참조하십시오.

## vRealize Log Insight Linux Agent의 기본 구성

설치를 마치면 `liagent.ini` 파일에는 Log Insight Windows Agent에 대해 미리 구성된 기본 설정이 포함됩니다.

### vRealize Log Insight Linux Agent `liagent.ini` 기본 구성

ASCII가 아닌 이름과 값을 사용하는 경우 구성을 UTF-8로 저장합니다.

중앙 구성을 사용하는 경우, 이 파일의 최종 구성이 서버의 설정과 연결되어 `liagent-effective.ini` 파일을 형성합니다.

서버의 에이전트 페이지에서 설정을 구성하는 것이 더 효율적일 수 있습니다.

```
[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.

```

매개 변수	기본값	설명
hostname	LOGINSIGHT	vRealize Log Insight 가상 장치의 IP 주소 또는 호스트 이름입니다. 기본값은 <b>loginsight</b> 입니다.
central_config	yes	이 에이전트에 대한 중앙 집중식 구성을 사용하거나 사용하지 않도록 설정합니다. 중앙 집중식 구성이 사용되지 않도록 설정되면 에이전트는 vRealize Log Insight 서버에서 제공하는 구성을 무시합니다. 허용되는 값은 yes, no, 1 또는 0입니다. 기본값은 yes입니다.
proto	cfapi	에이전트의 이벤트가 vRealize Log Insight 서버로 전송되는 데 사용되는 프로토콜입니다. 가능한 값은 cfapi 및 syslog입니다. 기본값은 cfapi입니다.
port	9543, 9000, 6514 및 514	에이전트의 이벤트가 vRealize Log Insight 서버로 전송되는 데 사용되는 통신 포트입니다. 기본값은 SSL이 사용되는 cfapi의 경우 9543, SSL이 사용되지 않는 cfapi의 경우 9000, SSL이 사용되는 syslog의 경우 6514, SSL이 사용되지 않는 syslog의 경우 514입니다.
ssl	yes	SSL을 사용하거나 사용하지 않도록 설정합니다. 기본값은 yes입니다. ssl을 [yes]로 설정하고 포트 값을 설정하지 않은 경우 포트 9543이 자동으로 선택됩니다.

매개 변수	기본값	설명
max_disk_buffer	200	Log Insight Windows Agent에서 이벤트 및 자체 로그를 버퍼링하는 데 사용하는 MB 단위의 최대 디스크 공간입니다. 지정된 max_disk_buffer에 도달하면 에이전트를 새로 들어오는 이벤트를 삭제하기 시작합니다.
debug_level	0	로그 세부 정보 수준을 정의합니다. <a href="#">Log Insight Agents의 로그 세부 정보 수준 정의 항목</a> 을 참조하십시오.

## 로그 파일에서 이벤트 수집

하나 이상의 로그 파일에서 이벤트를 수집하도록 vRealize Log Insight Linux 에이전트를 구성할 수 있습니다.

기본적으로 vRealize Log Insight Linux 에이전트는 애플리케이션 또는 편집기가 생성한 숨겨진 파일을 수집합니다. 숨겨진 파일 이름은 마침표로 시작합니다. 제외 매개 변수 **exclude=.**를 추가하여 vRealize Log Insight Linux 에이전트가 숨겨진 파일을 수집하지 않도록 방지할 수 있습니다.

필드 이름은 제한됩니다. 다음 이름은 예약된 것이므로 필드 이름으로 사용할 수 없습니다.

- event\_type
- hostname
- source
- text

에이전트 정보에 대해 최대 3개의 대상을 지정하고 전송하기 전에 정보를 필터링할 수 있습니다. [vRealize Log Insight 에이전트의 정보 전달 항목](#)을 참조하십시오.

**참고** 1,000개 이상의 파일과 같은 많은 수의 파일을 모니터링하면 vRealize Log Insight 에이전트의 리소스 사용량이 더 높아지고 호스트 시스템의 전체 성능에 영향을 미칩니다. 이를 방지하려면 패턴 및 glob을 사용하여 필요한 파일만 모니터링하도록 에이전트를 구성하거나 이전 로그 파일을 아카이브하십시오. 많은 수의 파일을 모니터링해야 하는 경우 CPU 및 RAM과 같은 호스트 매개 변수를 늘려 보십시오.

### 사전 요구 사항

- 루트로 로그인하거나 sudo를 사용하여 콘솔 명령을 실행합니다.
- vRealize Log Insight Linux 에이전트가 설치되고 실행되고 있는지 확인합니다. vRealize Log Insight Linux 에이전트를 설치한 Linux 시스템에 로그인하고 콘솔을 연 후 pgrep liagent를 실행합니다.

### 절차

- 1 텍스트 편집기에서 /var/lib/loginsight-agent/liagent.ini 파일을 엽니다.

## 2 파일의 [server |<dest\_id>] 섹션을 찾습니다. 구성 매개 변수를 추가하고 환경에 맞게 값을 설정합니다.

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

매개 변수	설명
<b>[filelog section_name]</b>	구성 섹션의 고유 이름입니다.
<b>directory=full-path-to-log-file</b>	<p>로그 파일 디렉토리의 전체 경로입니다. Glob 패턴이 지원됩니다. 구성 예:</p> <ul style="list-style-type: none"> <li>■ D:\Logs\new_test_logs 디렉토리의 모든 하위 디렉토리에서 수집하려면 다음을 사용하십시오. <code>directory=D:\Logs\new_test_logs\*</code></li> <li>■ 하위 디렉토리에 고유한 하위 디렉토리가 있는 경우 다음 구성을 사용하여 모든 하위 디렉토리를 모니터링합니다. <code>directory=D:\Logs\new_test_logs\*\*</code></li> </ul> <p><b>참고</b> 파일 및 디렉토리 수를 제한하고 리소스 소비량이 높아지지 않도록 하려면 첫 번째 또는 두 번째 수준 디렉토리(예: <code>directory=c:/tmp/*</code> 또는 <code>directory=c:\Logs\*</code>)에 대해 디렉토리 glob를 정의하면 안 됩니다. 디렉토리 경로는 두 수준 이상이어야 합니다.</p> <p>존재하지 않는 디렉토리에 대한 경로를 정의할 수 있습니다. 그러면 에이전트는 디렉토리 및 파일이 생성된 후에 해당 디렉토리에 로그 파일을 수집합니다.</p> <p>하나 이상의 다른 구성 섹션 아래에서 동일한 디렉토리를 정의하여 동일한 파일에서 여러 번 로그를 수집할 수 있습니다. 이 프로세스를 통해 여러 태그 및 필터를 동일한 이벤트 소스에 적용할 수 있습니다.</p> <p><b>참고</b> 해당 섹션에 대해 동일한 구성을 사용하는 경우 중복 이벤트가 서버 측에서 발견됩니다.</p>
<b>include=file_name; ...</b>	<p>(선택 사항) 데이터를 수집할 파일 이름 또는 파일 마스크(glob 패턴)의 이름입니다. 값은 세미콜론으로 구분된 목록으로 제공할 수 있습니다. 기본값은 *이며, 이는 모든 파일이 포함된다는 의미입니다. 매개 변수는 대/소문자를 구분합니다.</p> <p>파일 마스크(glob 패턴)를 사용하여 동일한 이름 지정 규칙을 따르는 여러 파일을 단일 파일 이름으로 그룹화할 수 있습니다. 예를 들어 vRealize Ops Analytics.log 및 vRealize Ops Collector.log와 같은 공백을 포함하는 파일 이름을 <code>vRealize?Ops?Analytics*.log</code> 또는 <code>vRealize*.log</code>로 지정할 수 있습니다. 파일 마스크를 사용하여 Linux 및 Windows 호스트의 에이전트 구성에 허용되는 파일 이름을 지정할 수 있습니다.</p> <p>기본적으로 .zip 및 .gz 파일은 수집 대상에서 제외됩니다.</p> <p><b>중요</b> 회전 로그 파일을 수집하는 경우 include 및 exclude 매개 변수를 사용하여 기본 파일 및 회전 파일 모두와 일치하는 glob 패턴을 지정합니다. glob 패턴이 기본 로그 파일과만 일치하면 회전하는 동안 이벤트가 vRealize Log Insight 에이전트에서 누락될 수 있습니다. vRealize Log Insight 에이전트는 회전 파일의 올바른 순서를 자동으로 확인하고 올바른 순서로 이벤트를 vRealize Log Insight 서버로 전송합니다. 예를 들어 기본 로그 파일의 이름이 myapp.log로 지정되고 회전 로그의 이름이 myapp.log.1, myapp.log.2 등으로 지정되는 경우 다음 include 패턴을 사용할 수 있습니다.</p> <pre>include= myapp.log;myapp.log.*</pre>

매개 변수	설명
<b>exclude=regular_expression</b>	(선택 사항) 수집에서 제외할 파일 이름 또는 파일 마스크(glob 패턴)입니다. 값은 세미콜론으로 구분된 목록으로 제공할 수 있습니다. 기본값은 비워 두는 것이며, 이는 제외할 파일이 없다는 의미입니다.
<b>event_marker=regular_expression</b>	(선택 사항) 로그 파일에서 이벤트의 시작을 나타내는 정규식입니다. 이를 생략하면 새 행이 기본값이 됩니다. 입력하는 식은 Perl 정규식 구문을 사용해야 합니다.  <b>참고</b> 따옴표(" ") 등의 기호는 정규식의 래퍼로 처리되지 않습니다. 이러한 기호는 패턴의 일부로 처리됩니다.  vRealize Log Insight 에이전트는 실시간 수집에 최적화되어 있기 때문에 내부 지연과 함께 작성된 부분 로그 메시지는 여러 이벤트로 분할될 수 있습니다. 확인된 새 event_marker 없이 로그 파일 추가가 200ms 이상 중지되어 있으면 부분 이벤트가 완료, 구문 분석 및 전달된 것으로 처리됩니다. 이 시간 논리는 구성할 수 없으며 event_marker 설정보다 우선시 됩니다. 로그 파일 appender가 전체 이벤트를 플러시해야 합니다.
<b>enabled=yes no</b>	(선택 사항) 구성 섹션을 사용 또는 사용하지 않도록 설정하는 매개 변수입니다. 가능한 값은 yes 또는 no입니다. 기본값은 yes입니다.
<b>charset=char-encoding-type</b>	(선택 사항) 에이전트가 모니터링하는 로그 파일의 문자 인코딩입니다. 가능한 값은 다음과 같습니다. <ul style="list-style-type: none"> <li>■ UTF-8</li> <li>■ UTF-16LE</li> <li>■ UTF-16BE</li> </ul> 기본값은 UTF-8입니다.
<b>tags={"tag-name": "tag-value", ...}</b>	(선택 사항) 수집된 이벤트의 필드에 사용자 지정 태그를 추가하는 매개 변수입니다. JSON 표기법을 사용하여 태그를 정의해야 합니다. 태그 이름에는 문자, 숫자 및 밑줄을 포함할 수 있습니다. 태그 이름은 문자 또는 밑줄로만 시작할 수 있으며 64자를 초과할 수 없습니다. 태그 이름은 대/소문자를 구분하지 않습니다. 예를 들어 tags={"tag_name1": "tag value 1", "Tag_Name1": "tag value 2"}를 사용하면 Tag_Name1은 중복 항목으로 무시됩니다. 태그 이름으로 event_type 및 timestamp를 사용할 수 없습니다. 동일한 선언 내의 중복 항목은 무시됩니다.  대상이 syslog 서버인 경우 태그는 APP-NAME 필드를 재정의할 수 있습니다. 예를 들어 tags={"appname":"VROPS"}일 수 있습니다.
<b>exclude_fields</b>	(선택 사항) 수집 대상에서 개별 필드를 제외하는 매개 변수입니다. 여러 값을 세미콜론 또는 쉼표로 구분된 목록으로 제공할 수 있습니다. 예를 들면 다음과 같습니다. <ul style="list-style-type: none"> <li>■ exclude_fields=hostname; filepath</li> <li>■ exclude_fields=type; size</li> <li>■ exclude_fields=type, size</li> </ul>
<b>raw_syslog=Yes No</b>	syslog 프로토콜을 사용하는 에이전트의 경우 이 옵션은 에이전트가 원시 syslog 이벤트를 수집하여 전송하도록 허용합니다. 기본값은 수집된 이벤트가 사용자 지정 syslob 특성으로 변환됨을 의미하는 [아니요]입니다. 이 옵션을 사용하도록 설정하여 syslog 변환 없이 이벤트를 수집합니다.

### 3 liagent.ini 파일을 저장한 후 닫습니다.

## 예제: 구성

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

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

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

## 이벤트 필터링

필드 값을 기준으로 vRealize Log Insight Linux 에이전트에서 수집된 모든 이벤트를 필터링하여 선택하거나 삭제할 로그 이벤트를 지정할 수 있습니다. whitelist 및 blacklist 수집기 옵션을 사용하여 필터를 정의할 수 있습니다.

**팁** 기본적으로 vRealize Log Insight Linux 에이전트는 프로그램 또는 편집기가 생성한 숨겨진 파일을 수집합니다. 숨겨진 파일 이름은 마침표로 시작합니다. vRealize Log Insight Linux 에이전트가 숨겨진 파일을 수집하지 않도록 지정하기 위해 제외 **exclude=.\*** 매개 변수를 추가할 수 있습니다.

각 이벤트에 대해 수집기는 whitelist 및 blacklist 필터 식을 평가합니다. whitelist 식이 **true**로 평가되고 blacklist 식이 **false**로 평가되거나 평가할 수 없는 경우 이벤트는 추가 처리를 위해 대기열로 이동합니다. 다른 경우 수집기는 이벤트를 삭제합니다. whitelist 식의 기본값은 **true**이고, blacklist 식의 기본값은 **false**입니다.

**팁** Filelog 수집기는 필터링을 위해 더 적은 수의 필드를 제공합니다. 필터링할 필드를 가져오기 위해 로그를 구문 분석할 수 있습니다. 자세한 내용은 [로그 구문 분석](#) 항목을 참조하십시오.

whitelist 또는 blacklist 필터는 단일 논리 값 또는 정수 값으로 평가되는 변수, 리터럴 및 연산자의 집합입니다. 이벤트 필드는 변수로, 큰따옴표로 묶은 문자열 및 숫자는 리터럴로 사용됩니다. 필터 식에서 사용할 수 있는 연산자에 대한 자세한 내용은 [이벤트 필드 및 연산자](#) 항목을 참조하십시오.

## 참고

- 숫자를 문자열과 비교하거나 비교에 문자열 형식으로 된 숫자를 포함하는 경우, 각 문자열은 하나의 숫자로 변환되어 수치로 비교됩니다. 예:
  - 식 `whitelist = 123.0 == "000123"`은 **true**로 평가됩니다.
  - 식 `whitelist = "00987" == "987.00"`은 **true**로 평가됩니다.
  - 식 `whitelist = response_size >= "12.12"`에서 `response_size` 필드에 숫자 값이 있는 경우 이 식은 수치로 평가됩니다. 응답 크기가 12.12보다 큰 경우 식은 **true**이고, 그렇지 않으면 **false**입니다.
  - 식 `whitelist = "09123" < "234"`에서 문자 리터럴은 모두 숫자 값으로 변환되고 식은 **false**로 평가됩니다.
- 문자열 피연산자 중 하나를 숫자 값으로 변환할 수 없는 경우 두 피연산자 모두 문자열로 변환됩니다. 대/소문자를 구분하는 간단한 사전 비교를 수행합니다. 예:
  - 식 `whitelist = "1234a" == "1234A"`는 문자열 비교이며 **false**로 평가됩니다.
  - 식 `whitelist = 4 < "four"`에서 4가 "4"로 변환되어 **true**로 평가됩니다.
  - 식 `whitelist = response_size > "thousand"`에서 `response_size` 필드의 값이 문자열 값으로 변환되어 식이 **false**로 평가됩니다.
- 필터 식이 정수 값으로 평가되는 경우 0일 때는 **false**이고, 그렇지 않은 경우에는 **true**로 처리됩니다. 예를 들어, `whitelist = some_integer & 1` 식은 `some_integer` 필드에 최하위 비트 집합이 있는 경우 **true**로 평가되고, 그렇지 않으면 **false**로 평가됩니다.

이벤트 필드 및 연산자의 전체 목록은 [로그 파일에서 이벤트 수집](#) 항목을 참조하십시오.

이 예에서는 `/var/log/httpd/access` 파일에서 Apache 액세스 로그를 수집합니다. 파일의 일부 샘플 로그는 다음과 같습니다.

- 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

## 사전 요구 사항

- **루트**로 로그인하거나 **sudo**를 사용하여 콘솔 명령을 실행합니다.
- vRealize Log Insight Linux 에이전트를 설치한 Linux 시스템에 로그인하고, 콘솔을 연 후 `pgrep liagent`를 실행하여 vRealize Log Insight Linux 에이전트가 설치되어 실행 중인지 확인합니다.

## 절차

- 1 다음 코드 조각과 같이 로그에 대한 파서를 정의합니다.

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

정의한 파서는 `/var/log/httpd/access` 파일에서 수집된 모든 이벤트에 대해 `remote_host`, `remote_log_name`, `remote_auth_user`, `timestamp`, `request`, `status_code` 및 `response_size` 필드를 추출합니다. 이러한 필드를 사용하여 이벤트를 필터링할 수 있습니다.

- 2 텍스트 편집기에서 `/var/lib/loginsight-agent/liagent.ini` 파일을 엽니다.
- 3 다음 코드 조각과 같이 파일에서 `Filelog` 섹션을 정의하여 로그를 수집하고 구문을 분석합니다.

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

- 4 요구 사항에 따라 이벤트를 필터링합니다.

- HTTP 상태가 200인 로그를 수집하려면 다음 코드 조각과 같이 `Filelog` 섹션에서 `whitelist`를 정의할 수 있습니다.

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

`whitelist` 식은 샘플 로그의 첫 번째와 두 번째 이벤트에 대해서만 `true`로 평가되고 수집기는 이러한 이벤트를 선택합니다.

`status_code` 필드가 로그에 없거나 구문 분석되지 않았기 때문에 이벤트에 존재하지 않는 경우, `whitelist` 식을 평가할 수 없습니다. 즉, `false`로 평가되며 수집기는 이벤트를 삭제합니다.



- 관심 없는 이벤트를 삭제하려면 blacklist 옵션을 사용할 수 있습니다. 예를 들어 로컬 트래픽에 관심이 없는 경우 다음 코드 조각과 같이 로컬 IP를 차단할 수 있습니다.

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

수집기는 샘플 로그에서 두 번째, 세 번째 및 네 번째 이벤트를 선택합니다.

- 둘 이상의 조건자를 기준으로 이벤트를 필터링하려면 or 및 and 연산자를 사용할 수 있습니다. 예를 들어, 다음 코드 조각과 같이 호스트에서 테스트 사용자가 생성한 이벤트 또는 로컬 IP에서 생성되었으나 필요하지 않은 이벤트를 삭제할 수 있습니다.

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

or 연산자를 사용하면 blacklist 식이 true로 평가되어 원치 않는 이벤트가 무시됩니다. 이 식은 remote\_host 필드 값이 "127.0.0.1"이거나 remote\_auth\_user 필드 값이 "test"인 경우 수집기에 이벤트를 삭제하도록 지시합니다.

수집기는 샘플 로그에서 두 번째 및 세 번째 이벤트를 선택합니다.

- 테스트 사용자가 로컬 IP에서 생성한 이벤트를 삭제하려면 다음 코드 조각과 같이 blacklist 식에서 and를 사용하면 됩니다.

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

수집기는 샘플 로그에서 다섯 번째 이벤트를 삭제합니다.

- whitelist 및 blacklist 필터는 함께 사용할 수 있습니다. 예를 들어, 응답 크기가 1024바이트보다 큰 이벤트가 필요하지만 로컬 호스트에서 생성된 이벤트가 필요하지 않은 경우 다음 코드 조각을 사용할 수 있습니다.

```
[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"
```

수집기는 샘플 로그에서 두 번째 이벤트를 선택합니다.

## journald에서 이벤트 수집

vRealize Log Insight 4.6부터 에이전트는 systemd를 실행하는 Linux 배포의 로그 데이터에 대해 journald 시스템 서비스에서 로그를 읽을 수 있습니다. 이제 journald는 systemd 기반 Linux 플랫폼에서 로깅을 위한 기본 표준입니다. journald 구성 섹션에서는 다음 옵션을 지원합니다.

### journal\_files

모니터링할 저널 파일입니다. 다음과 같은 값이 지원됩니다.

값	설명
all	사용 가능한 모든 저널 파일을 열고 모니터링합니다.
local	로컬 시스템에서 생성된 저널 파일만 모니터링하고 읽습니다.
runtime	영구 스토리지의 파일을 제외하고 휘발성 저널 파일만 모니터링하고 읽습니다.
system	시스템 서비스 및 커널 저널 파일만 모니터링하고 읽습니다.
사용자	현재 사용자의 저널 파일만 모니터링하고 읽습니다.

### fetch\_fields

저널 로그 항목의 메시지와 함께 가져올 필드입니다. 이 옵션의 값은 대/소문자를 구분하지 않는 쉼표로 구분된 필드 이름 목록입니다. 다음과 같은 값이 지원됩니다.

값	설명
pri_severity,pri_facility,syslog_identifier	이 옵션의 기본값입니다.
*	모든 필드를 가져옵니다.
all	필드를 가져오지 않습니다.

## vRealize Log Insight 에이전트에서 이벤트 필터링

로컬 liagent.ini 파일의 [server|<dest\_id>] 섹션에서 필터 옵션을 사용하여 에이전트가 대상으로 전송하는 정보를 제공할 수 있습니다.

옵션은 다음과 같은 형식입니다.

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

필터 유형	설명
collector_type	수집기 유형을 정의하는 쉼표로 구분된 목록입니다. 지원되는 값은 filelog 또는 winlog입니다. 값을 제공하지 않으면 모든 수집기 유형이 사용됩니다.
collector_filter	수집기 섹션의 이름을 정규식 형식으로 지정합니다. 예를 들어, vcops_.*는 "vcops_"로 시작하는 모든 수집기 섹션을 나타냅니다.
event_filter	이벤트 필드에 대한 필터는 수집기 섹션에서 허용 목록 또는 거부 목록과 동일한 구문을 사용합니다. 에이전트는 표현식을 True 또는 0이 아닌 값으로 평가하는 이벤트만 전송합니다. 비어 있는 event_filter는 항상 True로 평가합니다. 이벤트에 event_filter를 사용하려면 필드 추출에 대한 해당 수집기 섹션에 구문 분석기가 정의되어 있어야 합니다. 수집된 이벤트에 필드가 없어서 식을 평가할 수 없는 경우 이벤트가 삭제됩니다.

다음 예에 표시된 것처럼 둘 이상의 필터 식을 쉼표로 구분하여 지정할 수 있습니다.

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

하나의 메시지가 대상에 대해 둘 이상의 필터 기준을 충족하는 경우 한 번만 전송됩니다.

#### 표 4-1. 구문 예

필터	의미
filter= {winlog;Microsoft.*;}	이벤트 이름이 "Microsoft"로 시작하는 경우에만 winlog 수집기에서 이벤트를 전송합니다.
filter= {winlog;Microsoft.*; eventid == 1023}	이벤트 이름이 "Microsoft"로 시작하고 이벤트 ID가 1023인 경우에만 winlog 수집기에서 이벤트를 전송합니다.
filter= {.*;}	기본 필터 값입니다. 모든 소스에서 모든 이벤트를 전송합니다.
filter= {winlog;.*;}	winlog 섹션에서 모든 이벤트를 전송합니다.
filter= {filelog;syslog;facility<5}	기능이 5보다 작은 경우 [filelog syslog] 섹션에서 이벤트를 전송합니다. [filelog syslog] 섹션에는 기능 필드를 추출하는 구문 분석기가 있어야 합니다. 그렇지 않으면 모든 이벤트를 건너뜁니다.
filter= {;;}	일치하는 이벤트가 없습니다. 이벤트 전달을 사용하지 않도록 설정하려면 이 구문을 사용합니다.

다음 예에서는 이전 예의 두 번째 대상 구성에 필터를 추가합니다.

```
; 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; }
```

다음 예는 좀 더 복잡한 필터 식을 사용합니다.

```
; 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"}
```

다음 예에 표시된 것처럼 둘 이상의 필터 식을 쉼표로 구분하여 지정할 수 있습니다.

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

## vRealize Log Insight 에이전트의 정보 전달

에이전트에서 수집한 이벤트를 최대 세 개의 대상으로 전달할 수 있습니다. 대상에는 vRealize Log Insight 서버나 전달자 또는 타사 로그 관리 솔루션이 포함될 수 있습니다.

예를 들어, 보안 팀의 서버로 감사 또는 시스템 로그를 전송하고, 개발 운영 팀 서버로 애플리케이션 로그를 전송하고, IT 관리 시스템으로 메트릭 로그를 전송할 수 있습니다. 필터를 사용하여 대상으로 전달할 정보를 지정합니다. 단일 vRealize Log Insight 에이전트의 정보를 최대 세 개의 대상으로 전달할 수 있습니다.

에이전트 구성은 로컬 liagent.ini 파일의 [server|<dest\_id>] 섹션을 통해 수행됩니다. 다른 대상의 경우 vRealize Log Insight 서버 또는 전달자, Syslog에 cfapi 프로토콜을 사용합니다.

한 에이전트에 대해 둘 이상의 대상을 지정하는 경우 첫 번째 대상은 기본 loginsight 위치를 사용합니다. 다른 대상에 대한 위치 정보를 지정해야 합니다.

다음 예에서는 두 대상을 지정하는 liagent.ini 파일의 일부를 보여 줍니다. 기본 서버 이름 loginsight는 기본적으로 첫 번째 대상에 대해 암시적이므로 지정되지 않습니다. 두 번째 [server|<dest\_id>] 섹션은 대상을 지정합니다.

```
; 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
```

에이전트에 대한 필터 생성 방법에 대한 자세한 내용은 [vRealize Log Insight 에이전트에서 이벤트 필터링](#)을 참조하십시오.

## 대상 vRealize Log Insight 서버 설정

Windows에서 실행되는 vRealize Log Insight 에이전트에 대해 대상 vRealize Log Insight 서버를 설정하거나 변경할 수 있습니다. 최대 3개의 대상으로 이벤트를 전송하고 대상별로 출력을 필터링할 수 있습니다.

기본 대상은 `liagent.ini` 파일의 `[server]` 섹션을 통해 구성할 수 있습니다. 기본 대상은 항상 존재하며 기본적으로 호스트 이름은 `loginsight`로 설정됩니다. 더 많은 대상을 추가하려면 각 대상에 대해 `[server|<dest_id>]` 섹션을 생성합니다. 각 추가 연결에 대한 대상 ID로 고유한 호스트 이름을 지정해야 합니다. 기본 `[server]` 섹션에 관해 추가 대상에 동일한 옵션을 사용할 수 있습니다. 자동 업그레이드에 대한 추가 대상을 구성하거나 에이전트 구성에 이러한 추가 대상을 사용하지 마십시오. 두 개의 추가 대상을 지정할 수 있습니다.

기본적으로 에이전트는 수집된 모든 이벤트를 모든 대상에 전송합니다. 다양한 이벤트를 다양한 대상에 전송하도록 `file` 옵션을 사용하여 이벤트를 필터링할 수 있습니다. 자세한 내용은 [vRealize Log Insight 에이전트에서 이벤트 필터링 항목을 참조하십시오](#).

### 사전 요구 사항

- vRealize Log Insight Windows 에이전트가 설치된 Windows 시스템에 로그인하고 서비스 관리자를 시작하여 vRealize Log Insight 에이전트 서비스가 설치되었는지 확인합니다.
- 통합된 로드 밸런서를 사용하는 vRealize Log Insight 클러스터가 있는 경우 사용자 지정 SSL 인증서 관련 요구 사항은 [통합된 로드 밸런서 사용](#)을 참조하십시오.

### 절차

- 1 vRealize Log Insight Windows 에이전트의 프로그램 데이터 디렉토리로 이동합니다.

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

- 2 텍스트 편집기에서 `liagent.ini` 파일을 엽니다.

- 3 다음 매개 변수를 수정하고 환경에 맞게 값을 설정합니다.

매개 변수	설명
<b>proto</b>	에이전트의 이벤트가 vRealize Log Insight 서버로 전송되는 데 사용되는 프로토콜입니다. 가능한 값은 <code>cfapi</code> 및 <code>syslog</code> 입니다. 기본값은 <code>cfapi</code> 입니다.
<b>hostname</b>	vRealize Log Insight 가상 장치의 IP 주소 또는 호스트 이름입니다. IPv4 또는 IPv6 주소를 지정할 수 있습니다. IPv6 주소는 대괄호를 사용하거나 사용하지 않고 지정할 수 있습니다. 예: <pre>hostname = 2001:cdba::3257:9652 or hostname = [2001:cdba::3257:9652]</pre> 호스트가 IPv4 스택과 IPv6 스택 모두를 지원하고 도메인 이름을 호스트 이름으로 지정한 경우, 에이전트는 이름 확인 프로그램이 반환하는 IP 주소를 기준으로 IP 스택을 선택합니다. 확인 프로그램이 IPv4 주소와 IPv6 주소 둘 모두 반환하면 에이전트는 주어진 순서대로 두 주소에 순차적으로 연결을 시도합니다.
<b>max_disk_buffer</b>	Log Insight Windows Agent가 이 특정 서버에 대해 수집한 이벤트를 버퍼링하는 데 사용할 수 있는 최대 디스크 공간(MB)입니다. 이 옵션은 이 서버에 대한 <code>[storage].max_disk_buffer</code> 값을 재정의합니다. 기본값은 150MB이고 버퍼 크기를 50~8000MB 범위로 설정할 수 있습니다.

매개 변수	설명
<b>port</b>	<p>에이전트의 이벤트가 vRealize Log Insight 또는 타사 서버로 전송되는 데 사용되는 통신 포트입니다. 기본적으로 에이전트는 SSL 및 프로토콜에 설정된 옵션을 기반으로 적절한 포트를 사용합니다. 아래 목록에 제공된 기본 포트 값을 참조하십시오. 이러한 기본값과 다른 경우에만 포트 옵션을 지정해야 합니다.</p> <ul style="list-style-type: none"> <li>■ SSL이 사용되도록 설정된 cfapi: 9543</li> <li>■ SSL이 사용되지 않도록 설정된 cfapi: 9000</li> <li>■ SSL이 사용되도록 설정된 syslog: 6514</li> <li>■ SSL이 사용되지 않도록 설정된 syslog: 514</li> </ul>
<b>ssl</b>	<p>SSL을 사용하거나 사용하지 않도록 설정합니다. 기본값은 yes입니다.</p> <p>ssl이 yes로 설정된 경우 별도로 지정하지 않은 한, 포트는 9543으로 설정됩니다.</p>
<b>reconnect</b>	서버에 강제로 다시 연결하는 시간(분)입니다. 기본값은 30입니다.
<b>filter</b>	<p>에이전트가 대상으로 전송하는 정보를 지정합니다. 이 옵션은 다음 3개의 인수를 사용합니다.</p> <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 liagent.ini 파일을 저장한 후 닫습니다.

#### 예

다음 구성 예에서는 신뢰할 수 있는 CA(인증 기관)를 사용하는 대상 vRealize Log Insight 서버를 설정합니다.

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

다음 예에서는 대상별 메시지 필터링을 포함하는 다중 대상 구성을 보여줍니다.

```
; 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
```

#### 다음에 수행할 작업

vRealize Log Insight 에이전트에 대한 추가 SSL 옵션을 구성할 수 있습니다. 서버와 Log Insight Agent 사이의 SSL 연결 구성을 참조하십시오.

## 에이전트 대상 지정

이벤트를 전송할 vRealize Log Insight Linux 에이전트의 대상을 3개까지 지정할 수 있습니다.

여러 대상 연결은 `<dest_id>`가 고유한 구성 연결 ID인 `li-agent.ini` 파일의 `[server|<dest_id>]` 섹션을 통해 정의됩니다. 기본 `[server]` 섹션에 관해 추가 대상에 동일한 옵션을 사용할 수 있습니다. 그러나 자동 업그레이드에 대한 추가 대상을 구성하거나 에이전트 구성에 이러한 추가 대상을 사용하지 마십시오. 두 개의 추가 대상을 지정할 수 있습니다.

정의한 첫 번째 대상은 기본 서버 값으로 `loginsight`를 사용할 수 있습니다. 추가 대상을 정의할 경우 후속 대상에 대해 `[server]` 섹션에 호스트 이름을 지정해야 합니다. 필터링을 사용하지 않을 경우 에이전트는 수집된 모든 이벤트를 모든 대상에 전송합니다. 이것이 기본값입니다. 그러나 다양한 이벤트를 다양한 대상에 전송하도록 이벤트를 필터링할 수 있습니다.

### 사전 요구 사항

- 루트로 로그인하거나 `sudo`를 사용하여 콘솔 명령을 실행합니다.
- vRealize Log Insight Linux 에이전트를 설치한 Linux 시스템에 로그인하고, 콘솔을 연 후 `pgrep liagent`를 실행하여 vRealize Log Insight Linux 에이전트가 설치되어 실행 중인지 확인합니다.
- 통합된 로드 밸런서를 사용하는 vRealize Log Insight 클러스터가 있는 경우 사용자 지정 SSL 인증서 관련 요구 사항은 통합된 로드 밸런서 사용을 참조하십시오.

### 절차

- 1 텍스트 편집기에서 `/var/lib/loginsight-agent/liagent.ini` 파일을 엽니다.
- 2 다음 매개 변수를 수정하고 환경에 맞게 값을 설정합니다.

매개 변수	설명
<code>proto</code>	에이전트의 이벤트가 vRealize Log Insight 서버로 전송되는 데 사용되는 프로토콜입니다. 가능한 값은 <code>cfapi</code> 및 <code>syslog</code> 입니다. 기본값은 <code>cfapi</code> 입니다.
<code>hostname</code>	vRealize Log Insight 가상 장치의 IP 주소 또는 호스트 이름입니다. IPv4 또는 IPv6 주소를 지정할 수 있습니다. IPv6 주소는 대괄호를 사용하거나 사용하지 않고 지정할 수 있습니다. 예: <pre>hostname = 2001:cdba::3257:9652 or hostname = [2001:cdba::3257:9652]</pre> 호스트가 IPv4 스택과 IPv6 스택 모두를 지원하고 도메인 이름을 호스트 이름으로 지정한 경우, 에이전트는 이름 확인 프로그램이 반환하는 IP 주소에 기반하여 적절한 IP 스택을 사용합니다. 확인 프로그램이 IPv4 주소와 IPv6 주소 둘 모두 반환하면 에이전트는 주어진 순서대로 두 주소에 순차적으로 연결을 시도합니다.
<code>max_disk_buffer</code>	Log Insight Linux Agent가 이 특정 서버에 대해 수집한 이벤트를 버퍼링하는 데 사용할 수 있는 최대 디스크 공간(MB)입니다. 이 옵션은 이 서버에 대한 <code>[storage].max_disk_buffer</code> 값을 재정의합니다. 기본값은 150MB이고 버퍼 크기를 50~8000MB 범위로 설정할 수 있습니다.



매개 변수	설명
<b>port</b>	<p>에이전트의 이벤트가 vRealize Log Insight 또는 타사 서버로 전송되는 데 사용되는 통신 포트입니다. 기본적으로 에이전트는 SSL 및 프로토콜에 설정된 옵션을 기반으로 적절한 포트를 사용합니다. 아래 목록에 제공된 기본 포트 값을 참조하십시오. 이러한 기본값과 다른 경우에만 포트 옵션을 지정해야 합니다.</p> <ul style="list-style-type: none"> <li>■ SSL이 사용되도록 설정된 cfapi: 9543</li> <li>■ SSL이 사용되지 않도록 설정된 cfapi: 9000</li> <li>■ SSL이 사용되도록 설정된 syslog: 6514</li> <li>■ SSL이 사용되지 않도록 설정된 syslog: 514</li> </ul>
<b>ssl</b>	<p>SSL을 사용하거나 사용하지 않도록 설정합니다. 기본값은 yes입니다.</p> <p>ssl을 [yes]로 설정하고 포트 값을 설정하지 않은 경우 포트 9543이 자동으로 선택됩니다.</p>
<b>reconnect</b>	강제로 서버에 다시 연결하는 시간(분)입니다. 기본값은 30입니다.

```
[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 liagent.ini 파일을 저장한 후 닫습니다.

## 예

다음 구성 예에서는 신뢰할 수 있는 CA(인증 기관)를 사용하는 대상 vRealize Log Insight 서버를 설정합니다.

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

다음 예는 여러 대상 구성을 보여줍니다.

- 첫 번째(기본) 대상은 수집된 모든 이벤트를 수신합니다.

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

- 두 번째 대상은 기본 **syslog** 프로토콜을 통해 **syslog** 이벤트만 수신합니다.

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

- 세 번째 대상은 "오류" 또는 "경고"와 동일한 수준 필드가 있는 경우 vRealize Operations Manager 이벤트를 수신하며 이름이 "vrops-"로 시작하는 섹션에서 수집됩니다.

```
[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
```

#### 다음에 수행할 작업

vRealize Log Insight Linux 에이전트에 대한 추가 SSL 옵션을 구성할 수 있습니다. 서버와 Log Insight Agent 사이의 SSL 연결 구성을 참조하십시오.

## vRealize Log Insight 에이전트의 중앙 집중식 구성

여러 vRealize Log Insight 에이전트를 구성할 수 있습니다.

각 vRealize Log Insight 에이전트마다 로컬 구성과 서버 측 구성이 있습니다. 로컬 구성은 vRealize Log Insight 에이전트가 설치된 가상 또는 물리적 시스템의 `liagent.ini` 파일에 저장되어 있습니다. 서버 측 구성은 액세스 및 편집이 가능합니다. 예를 들어 웹 사용자 인터페이스의 **관리 > 에이전트**에서 액세스하고 편집할 수 있습니다. 각 vRealize Log Insight 에이전트의 구성은 섹션과 키로 구성됩니다. 키에는 구성 가능한 값이 있습니다.

vRealize Log Insight 에이전트는 주기적으로 vRealize Log Insight 서버를 폴링하여 서버 측 구성을 수신합니다. 서버 측 구성 및 로컬 구성은 병합되며, 해당 결과가 유효 구성이 됩니다. 각 vRealize Log Insight 에이전트는 유효 구성을 운영 구성으로 사용합니다. 구성은 섹션별 그리고 키별로 병합됩니다. 서버 측 구성의 값은 로컬 구성의 값을 재정의합니다. 병합 규칙은 다음과 같습니다.

- 섹션이 로컬 구성에만 제공되거나 서버 측 구성에만 제공되는 경우 해당 섹션 및 관련 콘텐츠 모두가 유효 구성의 일부가 됩니다.
- 섹션이 로컬 구성 및 서버 측 구성 모두에 제공되는 경우 섹션의 키는 다음 규칙에 따라 병합됩니다.
  - 키가 로컬 구성에만 제공되거나 서버 측 구성에만 제공되는 경우 키 및 관련 값이 유효 구성에서 해당 섹션의 일부가 됩니다.
  - 키가 로컬 구성 및 서버 측 구성 모두에 제공되는 경우 키는 유효 구성에서 해당 섹션의 일부가 되고 서버 측 구성의 값이 사용됩니다.

vRealize Log Insight 관리자는 모든 vRealize Log Insight 에이전트에 중앙 집중식 구성을 적용할 수 있습니다. 예를 들어 **관리** 페이지로 이동하고 **관리** 섹션에서 **에이전트**를 클릭합니다. **에이전트 구성** 상자에 구성 설정을 입력하고 **모든 에이전트에 대해 구성 저장**을 클릭합니다. 구성은 다음 폴링 주기 동안 구성 가능한 모든 활성 에이전트에 적용됩니다.

또한 관리자 vRealize Log Insight 사용자는 OS, 에이전트 버전, 호스트 이름 또는 IP 범위와 같은 에이전트 그룹의 특정 필터를 사용하고 특정 vRealize Log Insight 에이전트에 구성을 적용할 수 있습니다. 에이전트 그룹에 대한 자세한 내용은 "에이전트 그룹 작업"을 참조하십시오.

### 참고

- 중앙 집중식 구성은 cfapi 프로토콜을 사용하는 vRealize Log Insight 에이전트에만 적용할 수 있습니다.
- 다음 시나리오에서는 vRealize Log Insight 에이전트를 구성할 수 없습니다.
  - 현재 vRealize Log Insight 서버가 기본 대상이 아닌 경우. 다중 대상을 구성하는 방법은 [에이전트 대상 지정](#)의 내용을 참조하십시오.
  - `central_config = no` 매개 변수가 에이전트 구성에서 사용되는 경우. Windows용 기본 에이전트 구성에 대한 자세한 내용은 [Log Insight Windows Agent의 기본 구성](#)의 내용을 참조하십시오.

## 구성 병합 예

다음은 Log Insight Windows Agent의 로컬 구성과 서버 측 구성의 병합 예입니다.

### 로컬 구성

Log Insight Windows Agent의 로컬 구성은 다음과 같을 수 있습니다.

```
[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} - -
```

### 서버 측 구성

웹 사용자 인터페이스의 **관리 > 에이전트** 페이지를 사용하여 모든 에이전트에 중앙 집중식 구성을 적용할 수 있습니다. 예를 들어 수집 채널을 제외 및 추가할 수 있고 기본 다시 연결 설정을 변경할 수 있습니다.

```
[server]
reconnect=20

[winlog|Security]
channel=Security
enabled=no

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

### 유효 구성

유효 구성은 로컬 구성 및 서버 측 구성을 병합한 결과입니다. Log Insight Windows Agent는 다음을 수행하도록 구성됩니다.

- vRealize Log Insight 서버에 20분마다 다시 연결
- 애플리케이션 및 시스템 이벤트 채널을 계속 수집
- 보안 이벤트 채널 수집 중지

- Microsoft-Windows-DeviceSetupManager/Operational 이벤트 채널 수집 시작
- ApacheAccessLogs 계속 수집

```
[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} - -
```

## 에이전트 구성에 대해 공통 값 사용

Windows 또는 Linux 에이전트에 대한 각 에이전트 구성 섹션에 적용되는 공통 매개 변수 값으로 에이전트 구성 파일의 기본값을 재정의할 수 있습니다.

### 공통 옵션

liagent.ini 구성 파일의 [common|global] 섹션에 지정된 옵션은 모든 섹션으로 전파되고, [common|filelog] 섹션에 지정된 옵션은 모든 filelog 섹션으로만 전파되고, [common|winlog] 옵션은 모든 winlog 섹션으로만 전파됩니다.

다음 예와 같이 공통 섹션에서 tags, include, exclude, event\_marker, charset, exclude\_fields 및 parser와 같은 매개 변수를 정의할 수 있습니다. 이 예는 Windows 에이전트용입니다.

```
[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

...
```

이 예에서는 다음 동작을 지정합니다.

- **filelog** 섹션의 모든 로그는 해당 값에 **log\_source\_vm** 및 **collector\_type** 태그를 둘 다 포함합니다.
- **test\_tag** 및 **some\_other\_tag** 태그는 전송되는 모든 로그에서 제외됩니다.
- **auto** 파서는 수집된 모든 로그에 적용됩니다.
- 기본적으로 모든 **filelog** 수집기는 모니터링에서 **\*.trc** 파일을 제외합니다.

[common|global]의 옵션은 모든 **winlog** 섹션에도 적용됩니다.

## 병합 및 재정의의 기준

옵션이 둘 이상의 섹션에 정의되면 해당 값은 병합되거나 재정의되고, 더 작은 범위를 갖는 섹션이 병합/재정의 시 더 높은 우선 순위를 갖습니다. 즉, [common|global]의 값은 [common|filelog]의 값으로 병합 또는 재정의된 다음 [filelog|sample\_section]의 값으로 결합 또는 재정의됩니다.

병합 및 재정의의 동작은 다음 규칙을 따릅니다.

- 해당 값이 값 목록을 나타내는 옵션(tags, include, exclude 및 exclude\_fields)은 섹션의 해당 옵션 값과 병합될 때 더 높은 우선 순위를 갖습니다. 또한 태그의 경우 앞서 설명한 것처럼 더 높은 우선 순위를 갖는 섹션의 태그 값이 더 낮은 우선 순위를 갖는 섹션의 동일한 태그 값을 재정의합니다.
- 단일 값을 가질 수 있는 옵션(event\_marker, charset 및 parser) 값은 더 높은 우선 순위의 섹션에 있는 해당 옵션의 값으로 재정의됩니다.

즉, [filelog|sample\_section]의 charset=UTF-8 값은 [common|global]의 charset= UTF-16LE 글로벌 값을 재정의합니다.

따라서 예를 들어 [common|filelog]에 tags={"app":"global-test"}가 있고 [filelog|flg\_test\_section]에 tags={"app":"local-test", "section":"flg\_test\_section"}이 있는 경우 [filelog|flg\_test\_section] 섹션의 "app" 태그 값은 [common|filelog]의 값을 재정의합니다. 이 **filelog** 섹션을 통해 수집한 모든 로그는 "local-test" 값을 갖는 "app" 태그와 "flg\_test\_section" 값을 갖는 "section" 태그를 추가적으로 포함합니다. **winlog** 섹션의 경우 우선 순위 체인은 같습니다. 즉, 모든 [winlog|...] 섹션이 가장 높은 우선 순위를 갖고, [common|global] 섹션이 가장 낮은 우선 순위를 갖습니다.

공통 섹션에 잘못된 값이 지정되면 일반적으로 이러한 값은 건너뛰며, 앞에 나온 해당 **filelog/winlog** 섹션의 값과 병합되지 않습니다. 태그 또는 exclude\_fields 옵션에 잘못된 값이 있는 경우 에이전트는 가능한 많은 유효한 데이터를 추출하고, 잘못된 데이터가 나오면 파일의 나머지 부분을 건너뛰니다. 모든 이상 징후는 에이전트 로그 파일에 보고됩니다. 예기치 않은 동작이 발생하는 경우 로그 파일을 확인하고 에이전트에서 보고된 모든 오류를 수정하십시오.

에이전트는 filelog 또는 winlog 섹션에서 잘못된 옵션 값을 감지하면 해당 섹션의 옵션 값을 공통 섹션의 옵션 값과 병합하지 않고 해당 섹션을 사용하도록 설정하지 않습니다. 모든 오류는 에이전트 로그 파일에 보고됩니다. 예기치 않은 동작이 발생하는 경우 로그 파일을 확인하고 에이전트에서 보고된 모든 오류를 수정하십시오.

## 로그 구문 분석

에이전트 측 로그 구문 분석기는 원시 로그에서 구조화된 데이터를 추출하여 vRealize Log Insight 서버에 제공합니다. vRealize Log Insight는 로그 구문 분석기를 사용하여 로그를 분석하고, 여기에서 정보를 추출하고, 해당 결과를 서버에 표시할 수 있습니다. 로그 구문 분석기는 Windows 및 Linux vRealize Log Insight 에이전트 모두에 대해 구성할 수 있습니다.

syslog 프로토콜을 사용하는 경우, 구문 분석기를 통해 추출된 필드는 RFC5424에 따라 STRUCTURED-DATA의 일부입니다.

## 로그 구문 분석기 구성

FileLog 및 WinLog 수집기 모두에 대해 구문 분석기를 구성할 수 있습니다.

### 사전 요구 사항

vRealize Log Insight Linux Agent:

- 루트로 로그인하거나 sudo를 사용하여 콘솔 명령을 실행합니다.
- Log Insight Linux Agent를 설치한 Linux 시스템에 로그인하고, 콘솔을 연 후 pgrep liagent를 실행하여 Log Insight Linux Agent가 설치되어 실행 중인지 확인합니다.

vRealize Log Insight Windows Agent:

- Log Insight Windows Agent가 설치된 Windows 시스템에 로그인하고 서비스 관리자를 시작하여 vRealize Log Insight 서비스가 설치되었는지 확인합니다.

### 절차

- liagent.ini 파일이 포함된 폴더로 이동합니다.

운영 체제	경로
Linux	/var/lib/loginsight-agent/
Windows	%ProgramData%\VMware\Log Insight Agent

- 텍스트 편집기에서 liagent.ini 파일을 엽니다.
- 특정 구문 분석기를 구성하려면 구문 분석기 섹션을 정의합니다. [parser|myparser]

여기서 myparser는 로그 소스에서 참조할 수 있는 구문 분석기의 임의의 이름입니다. 구문 분석기 섹션은 기본 제공(또는 다른 정의된) 구문 분석기를 참조하고 해당 구문 분석기의 필수 옵션과 필요한 경우 선택적 옵션을 구성해야 합니다.

예를 들어 `base_parser=csv`는 `myparser` 구문 분석기가 기본 제공 구문 분석기 `csv`에서 파생되었음을 보여줍니다. 이 경우 입력 로그는 세미콜론으로 구분된 두 개의 필드로 구성되어야 합니다.

```
[parser|myparser]

base_parser=csv

fields=field_name1,field_name2

delimiter= ";"
```

- 4 `myparser`를 정의한 후에는 로그 소스 `winlog` 또는 `filelog`에서 이 구문 분석기를 참조합니다.

```
[filelog|some_csv_logs]

directory=D:\Logs

include=*.txt;*.txt.*

parser=myparser
```

D:\Logs 디렉토리 및 같은 `some_csv_logs` 소스에서 수집된 로그는 `myparser`에 의해 구문 분석되며 추출된 이벤트는 서버에서 각각 `field_name1` 및 `field_name2`로 표시됩니다.

---

**참고** D:\Logs 디렉토리의 정적 로그는 에이전트가 vRealize Log Insight로 가져오지 않습니다. 하지만 D:\Logs 디렉토리에 생성된 새 파일은 vRealize Log Insight에서 사용할 수 있습니다.

---

- 5 `liagent.ini` 파일을 저장한 후 닫습니다.

## 구문 분석기의 공통 옵션

이름 지정된 필드를 생성하는 모든 구문 분석기에 대해 공통 옵션을 구성할 수 있습니다.

### 필드 이름의 예약어

필드 이름은 제한됩니다. 다음 이름은 예약된 것이므로 필드 이름으로 사용할 수 없습니다.

- `event_type`
- `hostname`
- `source`
- `text`

### 일반 구문 분석기 옵션

다음 표에 나와 있는 옵션은 지원되는 모든 구문 분석기에 사용할 수 있습니다.



옵션	설명
base_parser	이 사용자 지정 구문 분석기가 확장하는 기본 구문 분석기의 이름입니다. 기본 제공 구문 분석기 이름이거나 다른 사용자 지정 구문 분석기 이름일 수 있습니다. 이 구성 키는 필수입니다.
field_decoder	<p>중첩된 구문 분석기는 JSON 문자열로 지정됩니다. 이 문자열의 키는 중첩된 구문 분석기가 적용되는 필드의 이름이고, 값은 해당 필드에 사용할 구문 분석기의 이름입니다. 중첩된 구문 분석기 각각은 기본 구문 분석기가 디코딩한 적절한 필드에 적용됩니다. 필드 디코더는 필드의 값이 타임 스탬프와 같은 복잡한 값일 때 유용합니다. 또한 <b>field_decoder</b> 옵션은 중첩된 구문 분석기가 적용되기 전에 확인되는 특정 필드 값에 대해 조건을 사용할 수 있도록 허용하는 인수로 좀 더 복잡한 JSON 개체를 지원합니다.</p> <p><b>참고</b> 사용량 및 조건부 구성에 대한 자세한 내용은 아래의 <b>field_decoder</b> 옵션에 대한 조건부 구성 섹션을 참조하십시오.</p>
field_rename	추출된 필드의 이름을 바꿉니다. 키는 필드의 원래 이름이고 값은 필드의 원하는 새 이름에 해당하는 JSON 문자열을 사용합니다. <b>field_decoder</b> 옵션은 <b>field_rename</b> 전에 항상 적용됩니다. INI 파일에서 이러한 옵션의 순서는 중요하지 않습니다. 확실히 하기 위해 <b>field_decoder</b> 를 먼저 지정하십시오.
next_parser	<p>다음으로 실행할 구문 분석기의 이름입니다. 동일한 입력에 대해 여러 구문 분석기를 순차적으로 실행할 수 있습니다.</p> <p><b>참고</b> 구문 분석기는 <b>next_parser</b> 키워드로 정의된 이후의 모든 구문 분석기를 처리하며 이전 구문 분석기에 의해 이미 추출된 필드 값을 바꿀 수 있습니다.</p>
exclude_fields	서버에 제공되기 전에 이벤트에서 제거할 세미콜론으로 구분된 필드 이름 목록입니다. 필드 이름은 이벤트 필터링이 수행되기 전에 제거되므로 구문 분석 도중 제외된 필드를 필터 조건에 사용할 수 없습니다.
debug	<p>특정 구문 분석기의 디버깅을 사용하도록 설정하는 <b>Yes</b> 또는 <b>No</b> 옵션입니다. 디버깅을 사용하도록 설정하면 구문 분석기는 수신되는 입력, 수행한 작업 및 생성한 결과의 상세 로깅을 수행합니다. 옵션은 섹션별로 적용됩니다. 즉, 특정 섹션으로 정의된 구문 분석기에만 적용됩니다.</p> <p>구문 분석기에 대한 디버깅의 기본 값은 <b>debug=no</b>입니다.</p>

### field\_decoder에 대한 조건부 구성 옵션

동일한 공통 형식을 갖지만 특정 필드 값과 관련된 로그(예: **info** 및 **error** 심각도의 로그)의 경우, 조건부로 중첩된 구문 분석기를 사용하여 불필요한 구문 분석기가 이미 구문 분석된 로그의 해당 필드에 적용되는 경우를 줄일 수 있습니다.

예를 들어 다음 로그를 사용합니다.

```
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 --
```

다음 구성을 사용하여 구문 분석할 수 있습니다.

```
[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"}
```

이 구성은 다음과 같은 결과를 생성합니다.

```
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.)"
```

또한 **info** 로그에 대해 다음 필드가 구문 분석됩니다.

```
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 --"
```

## 선택으로 구분된 값 로그 구문 분석기

FileLog 및 WinLog 수집기 모두에 대해 CSV(선택으로 구분된 값) 구문 분석기를 구성할 수 있습니다.

csv 구문 분석기에 대해 사용할 수 있는 옵션은 fields 및 delimiter입니다.

## 섬표로 구분된 값 구문 분석기 옵션

CSV 구문 분석기의 구조에 대한 다음 정보를 참고하십시오.

옵션	설명
fields	<p>fields 옵션은 로그에 있는 필드의 이름을 지정합니다. 나열된 필드 이름의 전체 수는 로그에 있는 섬표로 구분된 필드의 전체 수와 일치해야 합니다.</p> <p>fields 옵션은 CSV 구문 분석기에 필수입니다. 이 옵션을 지정하지 않으면 아무것도 구문 분석되지 않습니다. 필드 내용에 따라 필드 값 앞뒤의 큰따옴표는 선택 사항입니다.</p> <p>필드 이름은 다음과 같이 섬표로 구분되어야 합니다.</p> <pre>fields = field_name1, field_name2, field_name3, field_name4</pre> <p>이 정의는 추출된 필드에 이름 field_name1, field_name2, field_name3 및 field_name4가 순차적으로 할당되는 것으로 가정합니다.</p> <p>CSV 구문 분석기가 일부 필드를 생략해야 하는 경우에는 해당 이름을 목록에서 생략할 수 있습니다. 예를 들면 다음과 같습니다.</p> <pre>fields = field_name1, , field_name3, field_name4</pre> <p>이 경우 구문 분석기는 첫 번째, 세 번째 및 네 번째 필드만 이벤트에서 추출하며 여기에 이름 field_name1, field_name3 및 field_name4를 순차적으로 할당합니다.</p> <p>fields 옵션이 로그의 전체 필드 목록을 지정하지 않으면 구문 분석기는 빈 목록을 반환합니다. 예를 들어 로그 파일에 field1, field2, field3, field4 및 field5가 포함되어 있지만 fields= field1,field2,field3만 지정된 경우 구문 분석기는 빈 필드 목록을 반환합니다.</p> <p>구문 분석기가 빈 필드 목록을 반환하므로 CSV 구문 분석기에 fields=*를 사용할 수 없습니다. 이미 설명한 것처럼 특정 필드를 생략해야 하는 경우 이외에는 전체 필드 목록을 지정해야 합니다.</p>
delimiter	<p>delimiter 옵션은 구문 분석기에서 사용할 구분 기호를 지정합니다. 기본적으로 csv 구문 분석기는 섬표를 구분 기호로 사용하지만, 구분 기호를 세미콜론, 공백 또는 기타 특수 문자로 변경할 수 있습니다. 정의된 구분 기호는 큰따옴표로 묶어야 합니다.</p> <p>예: delimiter="," 및 delimiter=";"</p> <p>csv 구문 분석기의 경우 " " 또는 "asd" 같이 따옴표로 묶은 문자 집합을 구분 기호로 지원합니다. 로그에 사용된 필드 값의 구분 기호는 구분 기호 매개 변수에 정의된 패턴과 정확하게 일치해야 합니다. 구분 기호가 일치하지 않으면 구문 분석기가 실패합니다.</p> <p>공백 또는 탭 같은 특수 문자도 csv 구문 분석기에 사용할 구분 기호로 정의할 수 있지만, 이 경우에는 해당 특수 문자 앞에 이스케이프 문자가 반드시 있어야 합니다(\", \s, \t). 예를 들어 delimiter="\s" 또는 delimiter=" "와 같습니다.</p> <p>delimiter 옵션은 선택 사항입니다.</p>

## CSV 로그 구문 분석기 구성

winlog 또는 filelog 소스에서 수집된 로그를 구문 분석하려면 다음 구성을 사용합니다.

```
[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
```

이 구성을 사용하면 some\_csv\_logs 소스에서(예: directory=D:\Logs 디렉토리에서) 수집된 로그가 myparser에 의해 구문 분석됩니다. 수집된 로그에 세미콜론으로 구분된 세 개의 값이 포함된 경우, 구문 분석된 이벤트에는 field\_name1, field\_name2 및 field\_name3 이름이 순차적으로 할당됩니다.

다음 CSV 로그를 구문 분석하려면:

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

CSV 구문 분석기 구성을 정의합니다.

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

CSV 구문 분석기가 다음 필드를 반환합니다.

```
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.
```

## CLF(Common Log Format)(Apache) 로그 구문 분석기

FileLog 및 WinLog 수집기 모두에 대해 CLF(Common Log Format) Apache 구문 분석기를 구성할 수 있습니다.

### CLF(Common Log Format)(Apache) 구문 분석기

기본 CLF 구문 분석기는 다음과 같은 순서 및 이름의 필드를 정의합니다.

```
host ident authuser datetime request statuscode bytes
```

구문 분석기 이름: clf

CLF 구문 분석기의 옵션은 format입니다.

#### format 옵션

format 옵션은 Apache 로그 생성에 사용되는 형식을 지정합니다. 이 옵션은 필수 사항이 아닙니다.

형식을 지정하지 않으면 다음의 기본 공통 로그 형식이 사용됩니다.

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

CLF 구문 분석기 형식 문자열은 정규식을 허용하지 않습니다. 예를 들어 식 `\s+` 대신 공백을 지정하십시오.

다른 로그 형식을 구문 분석하려면 에이전트의 구성에 해당 형식을 지정하십시오. 구문 분석된 필드는 서버 측에 다음 이름으로 표시됩니다.

**참고** 변수가 필요한 경우 구성에 {VARNAME}이 제공되지 않으면 필드가 무시됩니다.

필드	값
'%a':	"remote_ip"
'%A':	"local_ip"
'%B', '%b':	"response_size"
'%C':	형식에 지정된 변수 이름에 따라 좌우됩니다.
'%c':	형식에 지정된 변수 이름에 따라 좌우됩니다.
'%D':	"request_time_mcs"
'%E':	"error_status"
'%e':	형식에 지정된 변수 이름에 따라 좌우됩니다.
'%F', '%f':	"file_name"
'%h':	"remote_host"
'%H':	"request_protocol"
'%i':	형식에 지정된 변수 이름에 따라 좌우됩니다.
'%k':	"keepalive_request_count"
'%l':	"remote_log_name"
'%L'	"request_log_id"
'%M':	"log_message"(이 지정자에 도달한 후에 파서는 입력 로그 구문 분석을 중지함)
'%m':	"request_method"
'%n':	형식에 지정된 변수 이름에 따라 좌우됩니다.
'%o':	형식에 지정된 변수 이름에 따라 좌우됩니다.
'%p':	"server_port" 추가 형식을 이 지정자와 함께 사용할 수 있습니다. %(format)p. 지원되는 형식은 "canonical", "local" 또는 "remote"입니다. "canonical" 형식이 사용되면 필드 이름은 "server_port"로 유지됩니다. "local" 형식이 사용되면 필드 이름은 "local_server_port"가 되고 "remote" 형식이 사용되면 필드 이름은 "remote_server_port"가 됩니다.

필드	값
'%P':	"process_id" 추가 형식을 이 지정자와 함께 사용할 수 있습니다. %{format}P. 지원되는 형식은 "pid", "tid" 및 "hextid"입니다. "pid"가 형식으로 사용되면 필드 이름은 "process_id"가 됩니다. 반면에 "tid" 및 "hextid" 형식은 이름이 "thread_id"인 필드를 생성합니다.
'%q':	"query_string"
'%r':	"request"
'%R':	"response_handler"
'%s':	"status_code". 요청의 최종 상태를 생성하는 이 필드도 지원됩니다. 서버에 "status_code"로 나타납니다.
'%t':	<p>"timestamp"은 수집 시 이벤트 타임 스탬프로 작동하고, 타임 스탬프 파서를 활용합니다. 타임 스탬프 자동 감지를 재정의하려면 날짜 및 시간 형식을 %{Y-%m-%d %H:%M:%S}t와 같이 중괄호로 지정할 수 있습니다. 자세한 내용은 <a href="#">타임 스탬프 구문 분석기</a>를 참조하십시오.</p> <p>CLF 파서에 대한 타임 스탬프 형식은 "begin:" 또는 "end:" 접두사로 시작할 수 있습니다. 형식이 begin:으로 시작하면(기본값) 요청 처리가 시작될 때가 시간으로 지정됩니다. end:로 시작될 경우 이 시간은 요청 처리가 끝나는 시간에 가까운, 로그 항목이 기록될 때에 해당합니다. 예를 들어 다음과 같은 형식은 CLF 파서에서 지원됩니다. %h %l %u [%{begin:%d/%b/%Y %T}t.%{msec_frac}t] \"%r\" %&gt;s %b</p> <p>CLF 파서 타임 스탬프 형식 지정자에 대해 다음 형식 토큰도 지원됩니다.</p> <p><b>sec</b></p> <p>Epoch 이후의 시간(초)입니다. 이 시간은 타임 스탬프 파서의 %s 지정자와 같습니다.</p> <p><b>msec</b></p> <p>Epoch 이후의 시간(밀리초)입니다.</p> <p><b>usec</b></p> <p>Epoch 이후의 시간(마이크로초)입니다.</p> <p><b>msec_frac</b></p> <p>밀리초 부분(타임 스탬프 파서의 %f 지정자와 같음)</p> <p><b>usec</b></p> <p>마이크로초 부분(타임 스탬프 파서의 %f 지정자와 같음)</p> <p>타임 스탬프가 형식 토큰으로 표시되는 로그를 구문 분석하려면 구성에 다음 형식을 사용할 수 있습니다.</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>이러한 토큰을 서로 조합하거나 동일한 형식 문자열의 타임 스탬프 파서 형식과 조합할 수 없습니다. 대신 여러 %{format}t 토큰을 사용할 수 있습니다. 예를 들어 타임 스탬프 파서의 %f 지정자를 사용하는 경우를 제외하고 밀리초를 포함하는 타임 스탬프를 사용하려면 다음의 조합된 타임 스탬프를 사용할 수 있습니다. %{d/%b/%Y %T}t.%{msec_frac}t .</p>
'%T':	"request_time_sec"

필드	값
'%u':	"remote_auth_user"
'%U':	"requested_url"
'%v':	"server_name"
'%V':	"self_referential_server_name"
'%X':	"connection_status"는 형식에 지정된 변수 이름에 따라 좌우됩니다.
'%x':	형식에 지정된 변수 이름에 따라 좌우됩니다.
'%I':	"received_bytes"
'%O':	"sent_bytes"
'%S':	"transferred_size"

예를 들어 winlog 또는 filelog 소스에서 수집된 로그를 CLF 구문 분석기로 구문 분석하려면 다음 구성을 지정합니다.

```
[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
```

이 구성을 사용하면 clflogs 소스에서(예: directory=D:\Logs 디렉토리에서) 수집된 로그가 myclf에 의해 구문 분석됩니다. myclf 구문 분석기는 구성에 설명된 형식으로 생성된 로그만 구문 분석합니다. 구문 분석기에 대한 디버그의 기본 값은 debug=no입니다.

### CLF를 사용하여 생성된 로그 구문 분석

CLF를 사용하여 생성된 로그를 구문 분석하려면 구성에서 해당 형식을 정의해야 합니다. 예를 들면 다음과 같습니다.

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

지정자 %{Referer}i 및 %{User\_Agent}i를 사용하는 비어 있지 않은 필드는 vRealize Log Insight 서버에서 각각 referer 및 user\_agent라는 이름으로 표시됩니다.

### 타임 스탬프 구문 분석기와 CLF 구문 분석기 통합

사용자 지정 시간 형식의 Apache 로그를 구문 분석할 수 있습니다.

다음과 같이 사용자 지정 시간 형식을 가진 로그에 액세스합니다.

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

사용자 지정 시간이 지정되지 않은 경우 CLF 구문 분석기는 자동 타임 스탬프 구문 분석기를 실행하여 자동으로 시간 형식을 추론하려고 시도하며, 그렇지 않은 경우에는 사용자 지정 시간 형식이 사용됩니다.

오류 로그에 대해 지원되는 사용자 지정 시간 형식은 다음과 같습니다.

사용자 지정 시간 형식	설명	구성 형식
%{u}t	마이크로초(ms)를 포함한 현재 시간	format=[%{u}t] [%l] [pid %P] [client %a] %M
%{cu}t	마이크로초(ms)를 포함한 컴팩트 ISO 8601 형식 현재 시간	format=[%{cu}t] [%l] [pid %P] [client %a] %M

지원되는 타임 스탬프 지정자의 전체 목록은 [타임 스탬프 구문 분석기](#)를 참조하십시오.

**예제: Windows에 대한 Apache 기본 액세스 로그 구성**

**예제: Windows에 대한 Apache 기본 오류 로그 구성**

이 예는 Windows에 대한 Apache v2.4 액세스 로그 구성의 형식을 지정하는 방법을 보여줍니다.

```
;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=clfparsed_apache_access
    enabled=yes

;Parser to parse Apache ACCESS logs
[parser|clfparsed_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\"
```

액세스 로그 형식 정의:

1 액세스 로그 형식(httpd.conf)에 대해 Apache를 구성합니다.

```
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 CLF 구문 분석기 구성을 정의합니다.

```
;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|clfflogs-access]
    directory=C:\xampp\apache\logs
    include=acc*;*myAcc*
    parser=clfparsed_apache_access
    enabled=yes
; Parser to parse Apache ACCESS logs
[parser|clfparsed_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 구문 분석기가 다음을 반환합니다.

```
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
```

이 예는 Windows에 대한 Apache v2.4 오류 로그 구성의 형식을 지정하는 방법을 보여줍니다.

```
;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|clfflogs-error]
    directory=C:\xampp\apache\logs
    include=err*
    parser=clfparsed_apache_error
    enabled=yes

;Parser to parse Apache ERROR logs
[parser|clfparsed_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=clfparsed_apache_error2

;Parser to parse Apache ERROR logs
[parser|clfparsed_apache_error2]
    debug=yes
```

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

**참고** 제공되는 이름은 결합된 로그 형식에 해당합니다. Apache 오류 로그는 Apache 오류 로그 형식이 아니라 위의 형식 지정 키를 사용하여 설명되기도 합니다.

오류 로그 형식 정의:

- 1 오류 로그 형식(httpd.conf)에 대해 Apache를 구성합니다.

```
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 CLF 구문 분석기 구성을 정의합니다.

```
;Parser to parse Apache ERROR logs
[parser|clfpaser_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=clfpaser_apache_error2

;Parser to parse Apache ERROR logs
[parser|clfpaser_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
```

로그 항목:

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

CLF 구문 분석기가 로그 항목에 대해 다음 필드를 반환합니다(+0400 표준 시간대에서 구문 분석기를 사용 중인 경우).

```
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.
```

로그 항목:

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

CLF 구문 분석기가 로그 항목에 대해 다음 필드를 반환합니다(+0400 표준 시간대에서 구문 분석기를 사용 중인 경우).

```
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
```

## 키/값 쌍 구문 분석기

FileLog 및 WinLog 수집기 모두에 대해 KVP(키/값 쌍) 구문 분석기를 구성할 수 있습니다.

### KVP(키/값 쌍) 구문 분석기

kvp 구문 분석기는 임의의 로그 메시지 텍스트에서 key=value 일치 항목을 모두 찾아서 추출합니다. 다음 예에서는 kvp 구문 분석기 형식을 보여 줍니다.

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

예를 들어 키-값 로그는 scope=local; abstract=false; lazyInit=false; autowireMode=0; dependencyCheck=0; 형식일 수 있습니다.

kvp 구문 분석기를 사용할 때는 값을 추출할 필드를 지정해야 합니다. 예를 들어 구성에 정의 fields=name,lastname,country가 있는 경우에는 지정된 키를 가진 값만 구문 분석하여 서버로 보냅니다.

필요한 경우 공백 또는 기타 특수 문자를 정의할 때 키와 값 모두 큰따옴표("")로 묶을 수 있습니다.

키 또는 값에 대해 큰따옴표를 사용할 경우 백슬래시 문자("\")를 이스케이프 문자로 사용할 수 있습니다. 백슬래시 다음에 나오는 모든 문자는 리터럴 방식으로 정의됩니다(큰따옴표 문자 또는 백슬래시 문자 포함). 예: "\\

다음 고려 사항에 주의하십시오.

- 키/값 쌍의 키 뒤에 등호 기호가 없고 VALUE가 제공되지 않으면 일반 텍스트의 경우처럼 옵션을 건너뜁니다.
- 키는 비워 둘 수 없으며 값은 비워 둘 수 있습니다.
- 등호 기호 뒤에 값이 없으면 일반 텍스트로 취급하여 건너뜁니다.
- 값은 큰따옴표 문자로 묶인 문자열이거나 비어 있을 수 있습니다. 값에 특수 문자가 포함된 경우 백슬래시를 사용하여 이스케이프합니다.

### KVP 구문 분석기 옵션

kvp 구문 분석기의 구조에 대해 다음 정보를 참조하십시오.

옵션	설명
fields	<p>추출하려는 정보이며 데이터 단위로 정의됩니다. 예를 들면 <code>fields=name,lastname,country</code> 입니다.</p> <p>특정 필드 이름이 fields 옵션에서 정의되면 로그에서 추출한 필드 이름에서 유효하지 않은 문자가 밑줄로 대체됩니다. 예를 들어 fields 옵션에서 필드 “x-A” 및 “a*(X+Y)”를 찾는 경우 구문 분석기는 로그에서 이러한 필드를 추출하고 “x_a” 및 “a__x_y” 필드로 각각 이름을 변경합니다. 이렇게 하면 이름에 문자가 있는 필드를 추출할 수 있습니다.</p> <p>fields 옵션이 “*”로 지정된 경우 이는 kvp 구문 분석기가 필드/값 쌍을 자동으로 인식한 다음 “영숫자+밑줄” 문자만 있는 필드(LI 서버에서 지원)를 검색한다는 의미입니다. 유효하지 않은 다른 모든 문자는 밑줄로 변환되지 않고 삭제됩니다. 이는 구문 분석기가 불필요한 정보를 정적 필드로 추출하는 것을 방지합니다.</p>
delimiter	<p>선택 사항입니다.</p> <p>기본 구분 기호는 공백 문자, 탭, 줄 바꿈 문자, 쉼표 및 세미콜론 문자입니다.</p> <p>구성에 구분 기호가 지정되지 않은 경우 kvp 구문 분석기는 기본 구분 기호를 구문 분석에 사용합니다.</p> <p>기본 구분 기호를 특정 구분 기호로 변경하려면 해당 구분 기호를 큰따옴표로 묶어야 합니다. 예를 들면 <code>delimiter = "#^ "</code>입니다. 이 정의는 큰따옴표 안에 있는 각 문자가 구분 기호로 사용된다는 것을 의미합니다. kvp 구문 분석기의 경우 모든 문자를 구분 기호로 간주할 수 있습니다. 정의에 포함된 다른 구분 기호와 함께 기본 구분 기호를 포함할 수 있습니다.</p> <p>예를 들어 <code>delimiter = "#^ \t\r\n\s"</code> 문에는 탭, 줄 바꿈 문자 및 공백이 구분 기호로 포함됩니다. 이러한 문자를 사용할 경우 해당 문자 앞에 이스케이프 문자가 나와야 합니다. 예를 들어 공백 문자를 구분 기호로 정의하려면 공백 문자 앞에 이스케이프 문자 “\”를 입력합니다(예: <code>delimiter="\s"</code>).</p>
field_decoder	<p>중첩된 구문 분석기는 JSON 문자열로 지정됩니다. 이 문자열의 키는 중첩된 구문 분석기에 적용할 필드의 이름이고, 값은 해당 필드에 사용할 구문 분석기의 이름입니다.</p> <p>중첩된 구문 분석기 각각은 기본 구문 분석기가 디코딩한 적절한 필드에 적용됩니다.</p> <p>필드 디코더는 키-값 쌍의 값이 타임 스탬프 또는 쉼표로 구분된 목록과 같은 복잡한 값일 때 유용합니다.</p>
debug =	<p>선택 사항입니다. debug = 값은 yes 또는 no일 수 있습니다. 구문 분석기에 대한 디버그의 기본 값은 debug=no입니다.</p> <p>옵션을 yes로 설정하면 <code>liagent_&lt;date&gt;.log</code>에서 구문 분석기 수집의 상세 로그를 볼 수 있습니다.</p>

## 추가적인 키 값 옵션

키	정의
<code>KVP_MESSAGE = * (MESSAGE_ENTRY [WSPR])</code>	선택적인 공백으로 구분된 메시지 항목의 목록입니다.
<code>MESSAGE_ENTRY = KVP / FREE_TEXT</code>	항목은 키/값 쌍이거나 일반 텍스트입니다.
<code>KVP = KEY ["=" VALUE]</code>	키/값 쌍입니다. KEY 다음에 등호와 VALUE가 나오지 않으면 일반 텍스트처럼 건너뛩니다.
<code>KEY = BARE_KEY / QUOTED_KEY</code>	
<code>FREE_TEXT = "="</code>	단독으로 나오는 등호는 일반 텍스트로 간주되어 건너뛩니다.

키	정의
BARE_KEY = *1BARE_KEY_CHAR	적어도 한 문자 이상입니다.
BARE_KEY_CHAR = %0x00-08 / %0x10-19 / %0x21-3C / %3E-%FF	등호, 공백 또는 탭을 제외한 모든 문자입니다.
QUOTED_KEY = 0x22 *1(QUOTED_STRING_CHAR / "\" CHAR) 0x22	큰따옴표 문자로 묶은 하나 이상의 문자입니다. 백슬래시를 이스케이프 문자로 사용합니다.
QUOTED_STRING_CHAR = %0x00-21 / %0x23-FF	큰따옴표를 제외한 모든 문자입니다.
VALUE = BARE_VALUE / QUOTED_VALUE	
BARE_VALUE = *BARE_VALUE_CHAR	0개 이상의 문자입니다.
BARE_VALUE_CHAR = %0x00-08 / %0x10-19 / %0x21-FF	공백 또는 탭을 제외한 모든 문자입니다.
QUOTED_VALUE = 0x22 *(QUOTED_STRING_CHAR / "\" CHAR) 0x22	큰따옴표 문자로 묶은 문자열입니다. 이 값은 비워 둘 수 있습니다. 백슬래시를 이스케이프 문자로 사용합니다.

## KVP 구문 분석기 구성 예

필요한 경우 fields=\*를 사용하여 모든 필드를 구문 분석할 수 있습니다.

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

이 예에서는 필드 디코더를 지정하는 방법을 보여 줍니다.

```
[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"}
```

다음 KVP 로그를 구문 분석하려면:

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

KVP 구문 분석기 구성을 정의합니다.

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

KVP 구문 분석기가 다음 필드를 반환합니다.

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

## 예제: 단순한 KVP 구문 분석기 및 복잡한 KVP 구문 분석기의 예

단순한 KVP 구문 분석기 예

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

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

복잡한 KVP 구문 분석기 예

```
[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"}
```

## 타임 스탬프 구문 분석기

timestamp 구문 분석기는 필드를 생성하지는 않지만 대신 문자열의 입력을 내부 타임 스탬프 형식(UNIX epoch 시작인 1970년 1월 1일(UTC/GMT 자정) 이후의 시간을 밀리초로 표시)으로 변환합니다.

유일하게 지원되는 구성 옵션은 format입니다. 예를 들면 format=%Y-%m-%d %H:%M:%S입니다.

CLF 구문 분석기와 달리 timestamp 구문 분석기는 %A%B%d%H%M%S%Y%z와 같이 시간 지정자 사이에 구분 기호가 없을 때 시간을 구문 분석할 수 있습니다.

timestamp 구문 분석기가 사용하는 형식 지정자는 다음과 같습니다.

```
'%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
```

추가적인 지정자는 타임 스탬프 구문 분석기에 서 수락되지만 해당 값은 무시되며 구문 분석된 시간에 영  
향을 미치지 않습니다.

```
'%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
```

format 매개 변수가 정의되지 않은 경우 Timestamp 구문 분석기는 기본 형식을 사용하여 타임 스탬프를  
구문 분석합니다.



## 자동 타임 스탬프 구문 분석기

자동 타임 스탬프 구문 분석기는 타임 스탬프 구문 분석기에 대해 정의된 형식이 없는 경우 호출되거나 `field_decoder`에서 `timestamp`를 사용하여 타임 스탬프 구문 분석기 정의 없이 직접 구문 분석기를 호출할 수 있습니다. 예:

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

### 예제: 기본 구성을 사용하는 타임 스탬프 구문 분석기

이 예에서는 기본 구성을 사용하는 `timestamp` 구문 분석기를 보여줍니다.

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

`timestamp` 구문 분석기를 다른 구문 분석기(예: `CSV` 구문 분석기)와 통합하려면 다음 구성을 지정하십시오.

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

이 구성이 정의되면 `mycsv` 구문 분석기는 구성에 지정된 이름의 필드를 추출하고 `timestamp` 필드의 내용에 대해 `tsp_parser`를 실행합니다. `tsp_parser`가 유효한 타임 스탬프를 검색하면 서버는 로그 메시지에 해당 타임 스탬프를 사용합니다.

## 자동 로그 구문 분석기

자동 구문 분석기는 행의 첫 200자 내에서 타임 스탬프를 자동으로 감지합니다. 자동 감지된 타임 스탬프의 형식은 `timestamp` 구문 분석기의 형식과 동일합니다.

자동 구문 분석기에는 옵션이 없습니다. 타임 스탬프의 자동 감지 이외에도, 로그 항목에 대해 키/값 구문 분석기가 실행되어 로그의 기존 키/값 쌍 모두를 자동으로 감지하고 그에 따라 필드를 추출합니다. 예를 들면 다음과 같습니다.

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

다른 구문 분석기와 마찬가지로 자동 구문 분석기에 대한 별도의 작업을 정의할 수 있습니다.

```
[filelog|kvplogs]
directory=C:\temp_logs\csv-itbm
```

```
include=*.txt
parser=myauto
[parser|myauto]

base_parser=auto
debug=yes
```

자동 구문 분석기에 대해 debug를 사용하도록 설정한 경우에는 구문 분석에 대한 추가 정보가 인쇄됩니다. 예를 들어 자동 구문 분석기가 실행된 로그 및 로그에서 추출된 필드에 대한 정보가 인쇄됩니다.

구문 분석기에 대한 디버그의 기본 값은 debug=no입니다.

## Syslog 구문 분석기

syslog 구문 분석기는 message\_decoder 및 extract\_sd 옵션을 지원하며 3개의 형식(RFC-6587, RFC-5424 및 RFC-3164)을 자동으로 감지합니다.

### message\_decoder 옵션 구성

syslog 구문 분석기에 대해 모든 공통 옵션과 message\_decoder 옵션을 사용할 수 있습니다. 기본적으로 timestamp 및 appname 필드만 추출됩니다. 다음 예와 유사하도록 liagent.ini 파일에서 구성 값을 설정하여 message\_decoder 옵션을 사용하도록 설정합니다.

```
[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=*
```

### 예제: message\_decoder 옵션으로 구문 분석

다음 예는 message\_decoder 옵션을 사용하도록 구성된 syslog 구문 분석기에서 이벤트에 추가한 샘플 이벤트 및 필드를 보여줍니다.

#### ■ 샘플 이벤트:

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

#### ■ KVP 구문 분석기를 실행하도록 message\_decoder 옵션이 적용된 syslog 구문 분석기에서 반환됨:

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

## 구조화된 데이터 구문 분석을 위한 extract\_sd 옵션 구성

구조화된 데이터를 구문 분석하려면 다음 예와 유사하도록 `liagent.ini` 파일에서 구성 값을 설정하여 `extract_sd` 옵션을 사용하도록 설정합니다.

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

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

### 예제: extract\_sd 옵션으로 구문 분석

다음 예는 `extract_sd` 옵션을 사용하도록 구성된 `syslog` 구문 분석기에서 이벤트에 추가한 샘플 이벤트 및 필드를 보여줍니다.

- 샘플 이벤트: <165>1 2017-01-24T09:17:15.719Z localhost evntslg - ID47  
[exampleSDID@32473 iut="3" eventSource="Application" eventID="1011"]  
[examplePriority@32473 class="high"] Found entity IPSet, display name dummy ip set 1411
- `syslog` 구문 분석기에 의해 다음 필드가 이벤트에 추가됨:

```
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="evntslg"
```

### 구문 분석기에서 추출한 필드

구문 분석기는 이벤트에서 자동으로 다음 필드를 추출합니다.

RFC 분류	pri_facility	pri_severity	timestamp	appname	procid	msgid
RFC 이외			X	X		
RFC-3164	X	X	X	X		
RFC-5424	X	X	X	X	X	X

### Syslog 구문 분석기 옵션

다음 표에서는 사용 가능한 `syslog` 옵션에 대해 설명합니다.

옵션	설명
message_decoder	이벤트의 메시지 본문을 구문 분석하는 데 사용하는 추가 구문 분석기를 정의합니다. '자동' 또는 사용자 정의 구문 분석기 등 기본 제공 구문 분석기가 될 수 있습니다.
extract_sd	구조화된 데이터를 구문 분석합니다.  <b>extract_sd</b> 옵션에는 <b>yes</b> 또는 <b>no</b> 값만 지원됩니다. 이 옵션은 기본적으로 사용되지 않도록 설정되어 있습니다. <b>extract_sd</b> 옵션이 사용되도록 설정되면 구조화된 데이터에서 모든 키-값 쌍이 추출됩니다.

### 예제: RFC-5424 표준에 대한 구문 분석

다음 예는 수집기, 샘플 이벤트 및 **syslog** 구문 분석기에서 이벤트에 추가하는 필드에 사용된 구성을 표시하는 구성된 **syslog** 인스턴스에서 구문 분석한 두 개의 이벤트를 나타냅니다.

#### ■ 구성:

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

#### ■ 모니터링된 파일에서 생성된 이벤트:

```
<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
```

#### ■ syslog 구문 분석기에서 이벤트에 추가한 필드:

```
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
```

### 예제: RFC-3164 표준에 대한 구문 분석

다음 예는 수집기, 샘플 RFC-3164 이벤트 및 **syslog**에서 이벤트에 추가한 필드에 사용된 구성을 나타냅니다.

#### ■ 구성:

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

- 모니터링된 파일에서 생성된 RFC-3164 이벤트:

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

- syslog 구문 분석기에서 이벤트에 추가한 필드:

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

## 레이블 지정된 탭으로 구분된 값 구문 분석기

LTSV(레이블 지정된 탭으로 구분된 값) 형식은 TSV(탭으로 구분된 값)의 변형입니다.

LTSV 파일의 각 기록은 한 줄에 나타납니다. 각 필드는 <TAB>으로 구분되어 있으며 레이블과 값이 있습니다. 레이블과 값은 :으로 구분되어 있습니다. LTSV 형식을 사용하는 경우 줄을 <TAB>으로 분할(TSV 형식과 동일)하여 각 줄을 구문 분석하고 특별한 순서 없이 고유한 레이블로 모든 필드를 확장할 수 있습니다. LTSV 정의 및 형식에 대한 자세한 내용은 <http://ltsv.org/>를 참조하십시오.

### 예제: LTSV 구문 분석기 구성

#### 예제: 샘플 LTSV 로그

LTSV 구문 분석기에는 특정 구성 옵션이 필요하지 않습니다. LTSV 구문 분석기를 사용하려면 구성에 기본 제공 ltsv 구문 분석기 이름을 지정합니다.

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

LTSV 파일은 ABNF 형식의 LTSV 프로덕션과 일치하는 바이트 순서여야 합니다.

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

TAB = %x09
NL = [%x0D] %x0A
lbyte = %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)
```

샘플 LTSV 구성을 사용하는 경우 로그의 구문 분석은 다음 필드를 반환합니다.

```
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)
```

## 디버그 구성

LTSV 구문 분석기에 대해 추가 디버깅도 사용할 수 있습니다. 기본적으로 LTSV 디버깅은 사용되지 않도록 설정되어 있습니다. LTSV 디버깅을 사용하도록 설정하려면 debug=yes를 입력합니다.

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

디버깅이 설정되면 LTSV 구문 분석기가 로그에서 모든 올바른 레이블의 값을 추출합니다. LTSV 구문 분석기를 사용하려면 레이블 이름이 영숫자, 밑줄('\_'), 점('.') 및 대시('-')로만 구성되어야 합니다. 잘못된 레이블 이름이 하나라도 있으면 구문 분석이 실패합니다. 레이블 이름이 올바른 경우에도 에이전트가 필드 이름을 확인합니다. 잘못된 이름이 있다면 잘못된 레이블 이름을 올바른 필드 이름으로 수정해야 합니다.

## filelog 섹션에서 LTSV 구문 분석기 구성

filelog 섹션에서 직접 LTSV 구문 분석기를 구성할 수도 있습니다.

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

## 정규식 구문 분석기

regex 구문 분석기를 사용하면 수집된 데이터에 대해 몇 가지 정규식을 사용할 수 있습니다.

vRealize Log Insight 에이전트는 Perl 구문에 해당하는 C++ Boost 라이브러리 정규식을 사용합니다.

regex 구문 분석기는 명명된 캡처 그룹이 포함된 정규식을 지정하는 방법으로 정의할 수 있습니다. 예: (?<field\_1>\d{4}) [-] (?<field\_2>\d{4}) [-] (?<field\_3>\d{4}) [-] (?<field\_4>\d{4})

그룹에 지정된 이름(예: field\_1, field\_2, field\_3 및 field\_4)은 해당하는 추출된 필드의 이름으로 사용됩니다. 이름에 대한 요구 사항은 다음과 같습니다.

- 정규식 패턴에 지정하는 이름은 vRealize Log Insight의 올바른 필드 이름이어야 합니다.
- 이름에는 영숫자와 밑줄("\_") 문자만 포함할 수 있습니다.
- 이름은 숫자로 시작할 수 없습니다.

유효하지 않은 이름이 제공된 경우 구성에 실패합니다.

## 정규식 구문 분석기 옵션

regex 구문 분석기의 유일한 필수 옵션은 format 옵션입니다.

debug 옵션은 추가적인 디버깅 정보가 필요할 때 사용할 수 있습니다.

## 구성

regex 구문 분석기를 생성하려면 regex를 base\_parser로 사용하고 형식 옵션을 제공합니다.

### 예제: 정규식 구성 예

#### 예제: Apache 로그 구문 분석 예

다음 예는 1234-5678-9123-4567을 분석하는 데 사용할 수 있습니다.

```
[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
```

결과는 다음과 같이 표시됩니다.

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

regex 구문 분석기를 사용하여 Apache 로그를 구문 분석하려면 Apache 로그에 사용할 특정 regex 형식을 제공해야 합니다.

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

결과는 다음과 같이 표시됩니다.

```
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
```

다음 코드는 Apache 로그를 구문 분석하는 다른 예를 보여 줍니다.

```
[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
```

## 성능 고려 사항

regex 구문 분석기는 CLF 구문 분석기 같은 다른 구문 분석기에 비해 리소스를 더 많이 사용합니다. 다른 구문 분석기를 사용하여 로그를 구문 분석할 수 있는 경우에는 해당 구문 분석기를 regex 구문 분석기 대신 사용하여 성능을 개선하는 방법을 고려해 보십시오.

사용할 수 있는 다른 구문 분석기가 없어 regex 구문 분석기를 사용해야 하는 경우에는 형식을 최대한 명확하게 정의하십시오. 다음 예에서는 더 나은 성능을 제공하는 구성을 보여 줍니다. 이 예에서는 숫자 값이 있는 필드를 지정합니다.

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

## JSON 구문 분석기

JSON 구문 분석기 구성을 사용자 지정하여 JSON 로그를 선택적으로 구문 분석할 수 있습니다.

**FileLog** 및 **WinLog** 수집기 둘 다에 대해 CSV(쉼표로 구분된 값) 구문 분석기를 구성할 수 있습니다. 올바른 JSON 로그만 Log Insight 에이전트 JSON 구문 분석기를 통해 구문 분석됩니다. 잘못된 JSON 로그 구문 분석기는 빈 결과를 반환합니다.

기본 JSON 구문 분석기 구성은 Log Insight 에이전트를 통해 JSON 로그에서 모든 필드를 추출합니다. JSON 로그가 자신을 JSON 개체를 포함할 수 있는 복잡한 JSON 개체로 나타내면 구문 분석기는 밑줄(\_) 문자를 사용하여 중첩된 상위 계층 JSON 개체의 이름을 연결합니다. 그러면 해당 요소에 대해 설명이 포함된 필드 이름이 생성됩니다. JSON 로그에 어레이도 포함된 경우 멤버 요소 이름 뒤에 어레이 이름과 어레이의 요소 인덱스가 차례로 포함됩니다.

JSON 구문 분석기는 **fields**라고도 하는 특정 옵션도 제공합니다.



## JSON 구문 분석기 '필드' 옵션

**fields** 옵션을 사용하여 구성에서 구문 분석할 필드를 지정할 수 있습니다. 이 옵션의 목적은 JSON 로그의 선택적 구문 분석을 사용하도록 설정하는 것입니다.

**참고** 선택적 구문 분석을 위해서는 원하는 JSON 요소의 경로를 지정해야 합니다. 다른 계층의 JSON 개체는 점(.) 문자로 구분해야 합니다.

다음 목록에서는 JSON 로그를 원하는 대로 선택적으로 구문 분석할 수 있는 예 구성을 제공합니다.

- JSON 로그에서 두 개 이상의 요소를 구문 분석하려면 원하는 요소를 **fields** 옵션에 대한 매개 변수로 나열하고 쉼표로 구분해야 합니다. 아래 예를 참조하십시오.

```
{
  "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"
  }
}
```

- **timestamp**, **log\_severity** 및 **log\_message**와 같은 맨 안쪽에 있는 JSON 개체만 구문 분석하려면 아래 예를 참조합니다. 이 예 구성에서는 다음 필드 결과를 생성합니다. **operation\_timestamp** = "2018-11-22T15:28:58.094000" 및 **operation\_log\_severity** = "info"

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

- 전체 JSON 개체를 구문 분석하려면 개체에 대한 경로와 별표(\*) 문자를 차례로 포함합니다.

```
{
  "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"
  }
}
```

- **operation** 개체만 구문 분석하려면 다음 구성을 사용합니다.

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

- JSON 로그에 어레이가 포함되어 있고 어레이의 특정 요소만 구문 분석하려는 경우에는 이 예 구성에 나와 있는 것처럼 구성에서 어레이의 요소 인덱스를 사용합니다.

```
{
  "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"
      }
    }
  ]
}

```

- 동일한 로그에서 **key** 및 **size** 요소만 구문 분석하려면 다음 구성을 사용하여 다음 필드를 생성합니다.

```
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

```

- 모든 어레이 요소에 대해 **key** 필드를 구문 분석하려면 다음 구성을 사용합니다.

```

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

```

- 모든 필드를 구문 분석하려면 별표(\*) 문자와 함께 **fields** 옵션을 사용합니다. 이 구성은 기본 JSON 구문 분석기 구성과 동일합니다.

```

[parser|json_parser]
base_parser=json
fields=*

```

# vRealize Log Insight 에이전트 제거

# 5

vRealize Log Insight 에이전트를 제거해야 하는 경우 설치된 에이전트 패키지에 해당하는 지침을 따르십시오.

본 장은 다음 항목을 포함합니다.

- Log Insight Windows Agent 제거
- Log Insight Linux Agent RPM 패키지 제거
- Log Insight Linux Agent DEB 패키지 제거
- Log Insight Linux Agent Bin 패키지 제거
- Log Insight Linux Agent Bin 패키지 수동 제거

## Log Insight Windows Agent 제거

[Windows 제어판]의 [프로그램 및 기능] 화면에서 Log Insight Windows Agent를 제거할 수 있습니다.

### 사전 요구 사항

vRealize Log Insight Windows 에이전트가 설치된 Windows 시스템에 로그인하고 서비스 관리자를 시작하여 vRealize Log Insight 에이전트 서비스가 설치되었는지 확인합니다.

### 절차

- 1 제어판 > 프로그램 및 기능으로 이동합니다.
- 2 VMware vRealize Log Insight Windows Agent를 선택하고 **제거**를 클릭합니다.

### 결과

그러면 VMware vRealize Log Insight Windows Agent 서비스가 중지되고 시스템에서 관련 파일이 제거됩니다.

## Log Insight Linux Agent RPM 패키지 제거

Log Insight Linux Agent RPM 패키지를 제거할 수 있습니다.

### 사전 요구 사항

- 루트로 로그인하거나 sudo를 사용하여 콘솔 명령을 실행합니다.
- Log Insight Linux Agent를 설치한 Linux 시스템에 로그인하고, 터미널 콘솔을 연 후 `pgrep liagent`를 실행하여 VMware Log Insight Linux Agent가 설치되어 실행 중인지 확인합니다.

### 절차

- ◆ 다음 명령을 실행합니다. 여기서 `VERSION` 및 `BUILD_NUMBER`를 설치된 에이전트의 버전 및 빌드 번호로 바꿉니다.

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

### 결과

그러면 VMware Log Insight Linux Agent 대몬이 중지되고 시스템에서 자체 로그를 제외한 모든 파일이 제거됩니다.

## Log Insight Linux Agent DEB 패키지 제거

Log Insight Linux Agent DEB 패키지를 제거할 수 있습니다.

### 사전 요구 사항

- 루트로 로그인하거나 sudo를 사용하여 콘솔 명령을 실행합니다.
- Log Insight Linux Agent를 설치한 Linux 시스템에 로그인하고, 터미널 콘솔을 연 후 `pgrep liagent`를 실행하여 VMware Log Insight Linux Agent가 설치되어 실행 중인지 확인합니다.

### 절차

- ◆ 다음 명령을 실행합니다.

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

### 결과

그러면 VMware Log Insight Linux Agent 대몬이 중지되고 시스템에서 자체 로그를 제외한 모든 파일이 제거됩니다.

## Log Insight Linux Agent Bin 패키지 제거

vRealize Log Insight 스크립트를 사용하여 Log Insight Linux Agent .bin 패키지를 제거할 수 있습니다.

### 사전 요구 사항

- 루트로 로그인하거나 sudo를 사용하여 콘솔 명령을 실행합니다.
- Log Insight Linux Agent를 설치한 Linux 시스템에 로그인하고, 터미널 콘솔을 연 후 `pgrep liagent`를 실행하여 VMware vRealize Log Insight Linux Agent가 설치되어 실행 중인지 확인합니다.

**절차**

- 1 셸 프롬프트에서 다음 명령을 입력하여 스크립트를 시작합니다.

```
loginsight-agent-uninstall
```

- 2 다음 명령에서 반환된 오류 코드가 0인지 확인하여 제거가 완료되었는지 확인할 수 있습니다.

```
echo $?
```

## Log Insight Linux Agent Bin 패키지 수동 제거

제거 스크립트를 사용하지 않으려는 경우 Log Insight Linux Agent .bin 패키지를 수동으로 제거할 수 있습니다.

**사전 요구 사항****Log Insight Linux Agent Bin 패키지 수동 제거**

- 루트로 로그인하거나 sudo를 사용하여 콘솔 명령을 실행합니다.
- Log Insight Linux Agent를 설치한 Linux 시스템에 로그인하고, 터미널 콘솔을 연 후 pgrep liagent를 실행하여 VMware vRealize Log Insight Linux Agent가 설치되어 실행 중인지 확인합니다.

**절차**

- 1 Log Insight Linux Agent 대몬을 중지합니다. 이는

```
sudo service liagentd stop
```

명령 또는 이전 버전의 Linux인 경우 `sudo /sbin/service liagentd stop` 명령을 실행하여 중지하면 됩니다.

- 2 Log Insight Linux Agent 파일을 수동으로 제거합니다.

- /usr/lib/loginsight-agent - 대몬 바이너리 및 라이선스 파일 디렉토리입니다.
- /usr/bin/loginsight-agent-support - Log Insight Linux Agent에 대한 지원 번들을 생성하는 데 사용됩니다.
- /var/lib/loginsight-agent - 구성 파일 및 데이터베이스 스토리지 디렉토리입니다.
- /var/log/loginsight-agent - Log Insight Linux Agent의 로그 디렉토리입니다.
- /var/run/liagent/liagent.pid - Log Insight Linux Agent PID 파일입니다. 자동으로 삭제되지 않은 파일은 수동으로 제거하십시오.
- /etc/init.d/liagentd - Log Insight Linux Agent 대몬의 스크립트 디렉토리입니다.
- /usr/lib/systemd/system/liagentd.service

# vRealize Log Insight 에이전트 문제 해결

## 6

알려진 문제 해결 정보는 vRealize Log Insight 에이전트의 운영과 관련된 문제를 진단하고 수정하는 데 도움이 됩니다.

본 장은 다음 항목을 포함합니다.

- Log Insight Windows Agent용 지원 번들 생성
- Log Insight Linux Agent용 지원 번들 생성
- Log Insight Agents의 로그 세부 정보 수준 정의
- 관리 UI에 Log Insight Agents가 표시되지 않음
- vRealize Log Insight 에이전트가 이벤트를 보내지 않음
- Log Insight Windows Agent에 대한 아웃바운드 예외 규칙 추가
- Windows 방화벽에서 Log Insight Windows Agent의 아웃바운드 연결 허용
- Log Insight Windows Agent의 대량 배포가 실패함
- Log Insight Agents 자체 서명된 인증서 거부
- vRealize Log Insight 서버가 암호화되지 않은 트래픽의 연결을 거부함

## Log Insight Windows Agent용 지원 번들 생성

문제가 발생하여 Log Insight Windows Agent가 예상대로 작동하지 않으면 로그 및 구성 파일의 복사본을 VMware 지원 서비스로 전송할 수 있습니다.

### 절차

- 1 Log Insight Windows Agent가 설치된 대상 시스템에 로그인합니다.
- 2 Windows **시작** 버튼을 클릭한 후 **VMware > Log Insight Agent - Collect 지원 번들**을 클릭합니다.
- 3 (선택 사항) 바로 가기를 사용할 수 없으면 Log Insight Windows Agent의 설치 디렉토리로 이동하고 `loginsight-agent-support.exe`를 두 번 클릭합니다.

---

**참고** 기본 설치 디렉토리는 `C:\Program Files (x86)\VMware\Log Insight Agent`입니다.

---

## 결과

번들이 생성되어 내 문서에 .zip 파일로 저장됩니다.

## 다음에 수행할 작업

요청에 따라 지원 번들을 VMware 지원 서비스로 전달합니다.

# Log Insight Linux Agent용 지원 번들 생성

문제가 발생하여 Log Insight Linux Agent가 예상대로 작동하지 않으면 로그 및 구성 파일의 복사본을 VMware 지원 서비스로 전송할 수 있습니다.

## 절차

- 1 Log Insight Linux Agent가 설치된 대상 시스템에 로그인합니다.
- 2 다음 명령을 실행합니다.

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

## 결과

번들이 생성되어 현재 디렉토리에 .zip 파일로 저장됩니다.

## 다음에 수행할 작업

요청에 따라 지원 번들을 VMware 지원 서비스로 전달합니다.

# Log Insight Agents의 로그 세부 정보 수준 정의

vRealize Log Insight Agent의 구성 파일을 편집하여 로깅 수준을 변경할 수 있습니다.

## 사전 요구 사항

Log Insight Linux Agent의 경우:

- 루트로 로그인하거나 sudo를 사용하여 콘솔 명령을 실행합니다.
- Log Insight Linux Agent를 설치한 Linux 시스템에 로그인하고, 콘솔을 연 후 pgrep liagent를 실행하여 VMware vRealize Log Insight Linux Agent가 설치되어 실행 중인지 확인합니다.

Log Insight Windows Agent의 경우:

- vRealize Log Insight Windows 에이전트가 설치된 Windows 시스템에 로그인하고 서비스 관리자를 시작하여 vRealize Log Insight 에이전트 서비스가 설치되었는지 확인합니다.

## 절차

- 1 `liagent.ini` 파일을 포함하는 폴더로 이동합니다.

운영 체제	경로
Linux	<code>/var/lib/loginsight-agent/</code>
Windows	<code>%ProgramData%\VMware\Log Insight Agent</code>

- 2 텍스트 편집기에서 `liagent.ini` 파일을 엽니다.
- 3 `liagent.ini` 파일의 `[logging]` 섹션에서 로그 디버그 수준을 변경합니다.

**참고** 디버그 수준이 높을수록 vRealize Log Insight Agent에 미치는 영향도 커집니다. 기본값이자 권장 값은 0입니다. 디버그 수준 1은 좀 더 많은 정보를 제공하며 대부분의 문제 해결을 위해 이 수준이 권장됩니다. 디버그 수준 2는 세부 정보를 제공합니다. 수준 1과 수준 2는 VMware 지원팀에서 요청한 경우에만 사용하십시오.

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

- 4 `liagent.ini` 파일을 저장한 후 닫습니다.

## 결과

로그 디버그 수준이 변경됩니다.

## 관리 UI에 Log Insight Agents가 표시되지 않음

Log Insight Agents 인스턴스에 대한 정보가 관리 UI의 에이전트 페이지에 나타나지 않습니다.

### 문제

Log Insight Agents를 설치한 후 관리 UI의 에이전트 페이지에 Log Insight Agents가 표시되지 않습니다.

### 원인

가장 일반적인 원인은 네트워크 연결 문제이거나 `liagent.ini` 파일에서 잘못된 Log Insight Agents 구성 때문입니다.

### 해결책

- ◆ Log Insight Agents가 설치된 Windows 또는 Linux 시스템이 vRealize Log Insight 서버와 연결되었는지 확인합니다.
- ◆ Log Insight Agents가 cfapi 프로토콜을 사용하는지 확인합니다.  
syslog 프로토콜을 사용하면 UI에 Log Insight Windows Agents가 표시되지 않습니다.



- ◆ 다음 디렉토리에 위치한 Log Insight Agents 로그 파일의 콘텐츠를 봅니다.
  - Windows - %ProgramData%\VMware\Log Insight Agent\log
  - Linux - /var/log/loginsight-agent/

Config transport error: Couldn't resolve host name 및 Resolver failed. No such host is known 구를 포함하는 로그 메시지를 찾습니다.
- ◆ liagent.ini에 대상 vRealize Log Insight 서버에 대한 올바른 구성이 포함되어 있는지 확인합니다. 대상 vRealize Log Insight 서버 설정 및 에이전트 대상 지정 을 참조하십시오.

## vRealize Log Insight 에이전트가 이벤트를 보내지 않음

잘못된 구성으로 인해 vRealize Log Insight 에이전트가 vRealize Log Insight 서버로 이벤트를 전달하지 못할 수 있습니다. 플랫폼 파일 수집 채널이 올바르게 구성되지 않은 경우 다음과 같은 메시지를 볼 수 있습니다. "Invalid settings were obtained for channel 'CHANNEL\_NAME'. Channel 'CHANNEL\_NAME' will stay dormant until properly configured."

### 문제

vRealize Log Insight 에이전트 인스턴스가 **관리 > 에이전트** 페이지에 나타나지만 vRealize Log Insight 에이전트 호스트 이름의 대화형 분석 페이지에는 이벤트가 나타나지 않습니다. 플랫폼 파일 수집 채널이 올바르게 구성되지 않았습다.

### 원인

잘못된 구성으로 인해 vRealize Log Insight 에이전트가 vRealize Log Insight 서버로 이벤트를 전달하지 못할 수 있습니다.

### 해결책

- ◆ 올바른 수집 채널을 정의합니다. 플랫폼 파일 수집 채널이 올바르게 구성되었는지 확인합니다. [장 4 vRealize Log Insight 에이전트 구성](#) 항목을 참조하십시오.
- ◆ vRealize Log Insight Windows 에이전트의 경우 다음을 시도합니다.
  - Windows 채널이 사용되도록 설정된 경우 %ProgramData%\VMware\Log Insight Agent\log에 있는 vRealize Log Insight Windows 에이전트 로그 파일의 콘텐츠를 확인합니다. Subscribed to channel CHANNEL\_NAME 구를 포함하는 채널 구성 관련 로그 메시지를 찾습니다. 일반적으로 사용되는 채널은 ApplicationSystem 및 Security입니다.
  - 채널이 올바르게 구성되지 않은 경우 다음과 유사한 로그 메시지를 볼 수 있습니다. Could not subscribe to channel CHANNEL\_NAME events. Error Code: 15007. The specified channel could not be found. Check channel configuration. 15007 이외의 오류 코드 번호를 볼 수도 있습니다.
  - 플랫폼 파일 수집 채널이 올바르게 구성되지 않은 경우 다음과 같은 메시지를 볼 수 있습니다. Invalid settings were obtained for channel 'CHANNEL\_NAME'. Channel 'CHANNEL\_NAME' will stay dormant until properly configured

- ◆ vRealize Log Insight Windows 에이전트 및 vRealize Log Insight Linux 에이전트의 경우 다음을 시도합니다.

- ◆ 구성된 플랫폼 파일 수집 채널이 없는 경우 다음과 유사한 메시지를 볼 수 있습니다. Cannot find section 'filelog' in the configuration. The flat file log collector will stay dormant until properly configured

vRealize Log Insight 에이전트 로그 파일의 콘텐츠는 다음 디렉토리에 위치합니다.

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

다음에 수행할 작업

vRealize Log Insight 에이전트 구성에 대한 자세한 내용은 [Log Insight Windows Agent 구성](#) 및 [Log Insight Linux Agent 구성](#) 항목을 참조하십시오.

## Log Insight Windows Agent에 대한 아웃바운드 예외 규칙 추가

Windows 방화벽에서 Log Insight Windows Agent 차단 해제 예외 규칙을 정의할 수 있습니다.

절차는 Windows Server 2008 R2 이상 및 Windows 7 이상에 적용됩니다.

사전 요구 사항

- 관리자 계정 또는 관리 권한이 있는 계정을 보유하고 있는지 확인합니다.

절차

- 1 시작 > 실행을 선택합니다.
- 2 wf.msc를 입력하고 **확인**을 클릭합니다.
- 3 왼쪽 창에서 **아웃바운드 규칙**을 마우스 오른쪽 버튼으로 클릭하고 **새 규칙**을 클릭합니다.
- 4 **사용자 지정**을 선택하고 마법사에 따라 다음 옵션을 설정합니다.

옵션	설명
프로그램	liwinsvc.exe
서비스	Log Insight Agent 서비스
프로토콜 및 포트	cfapi의 경우 TCP 9000 그리고 syslog의 경우 514

- 5 이 규칙을 적용할 프로파일 지정 페이지에서 적합한 네트워크 유형을 선택합니다.

- 도메인
- 공용

## ■ 전용

**참고** 네트워크 유형에 관계없이 예외 규칙을 활성화 상태로 유지하기 위해 모든 네트워크 유형을 선택할 수 있습니다.

### 다음에 수행할 작업

Log Insight Windows Agent 로그 디렉토리인 %ProgramData%\VMware\Log Insight Agent\log로 이동하고 최신 로그 파일을 엽니다. 최근 이벤트에 Config transport error: Couldn't resolve host name 및 Resolver failed. No such host is known 메시지가 포함되어 있으면 Log Insight Windows Agent 서비스와 Windows 시스템을 다시 시작합니다.

**참고** Log Insight Windows Agent 서비스가 서버에 다시 연결되는 데 최대 5분이 걸릴 수 있습니다.

## Windows 방화벽에서 Log Insight Windows Agent의 아웃바운드 연결 허용

Windows 방화벽 설정을 구성하여 Log Insight Windows Agent에서 vRealize Log Insight 서버로의 아웃바운드 연결을 허용할 수 있습니다.

Log Insight Windows Agent 서비스를 설치하고 시작한 후 Windows 도메인 또는 로컬 방화벽이 대상 vRealize Log Insight 서버로의 연결을 제한할 수도 있습니다.

절차는 Windows Server 2008 R2 이상 및 Windows 7 이상에 적용됩니다.

### 사전 요구 사항

- 관리자 계정 또는 관리 권한이 있는 계정을 보유하고 있는지 확인합니다.

### 절차

- 1 시작 > 실행을 선택합니다.
- 2 wf.msc를 입력하고 **확인**을 클릭합니다.
- 3 작업 창에서 **속성**을 클릭합니다.
- 4 **도메인 프로파일** 탭의 **아웃바운드 연결** 드롭다운 메뉴에서 **허용(기본값)**을 선택합니다.

컴퓨터가 도메인에 연결되어 있지 않으면 컴퓨터가 연결된 네트워크 유형에 따라 **개인 프로파일** 또는 **공개 프로파일**을 선택할 수 있습니다.

- 5 **확인**을 클릭합니다.

### 다음에 수행할 작업

Windows 방화벽에서 Log Insight Windows Agent에 대해 차단 해제 예외 규칙을 정의합니다. [Log Insight Windows Agent에 대한 아웃바운드 예외 규칙 추가](#)를 참조하십시오.

## Log Insight Windows Agent의 대량 배포가 실패함

대상 시스템에서 Log Insight Windows Agent의 대량 배포가 실패합니다.

### 문제

GPO(그룹 정책 개체)를 사용하여 Windows 도메인 시스템에서 대량 배포를 수행한 후 Log Insight Windows Agent의 로컬 서비스로의 설치가 실패합니다.

### 원인

그룹 정책 설정으로 인해 Log Insight Windows Agent가 올바르게 설치되지 않을 수 있습니다.

### 해결책

- 1 GPO(그룹 정책 개체) 설정을 편집하고 Log Insight Windows Agent 에이전트를 다시 배포하십시오.
  - a GPO를 마우스 오른쪽 버튼으로 클릭하고 **편집**을 클릭한 후 **컴퓨터 구성 > 정책 > 관리 템플릿 > 시스템 > 로그인**으로 이동합니다.
  - b **컴퓨터 시작 및 로그인 시 네트워크가 초기화될 때까지 항상 대기** 정책을 사용하도록 설정합니다.
  - c **컴퓨터 구성 > 정책 > 관리 템플릿 > 시스템 > 그룹 정책**으로 이동합니다.
  - d **시작 정책 처리 대기 시간**을 사용하도록 설정하고 **대기 시간(초)**을 120으로 설정합니다.
- 2 대상 시스템에서 `gpupdate /force /boot` 명령을 실행합니다.

## Log Insight Agents 자체 서명된 인증서 거부

Log Insight Agents가 자체 서명된 인증서를 거부합니다.

### 문제

vRealize Log Insight 에이전트는 자체 서명된 인증서를 거부하며 서버와의 연결을 설정할 수 없습니다.

**참고** 에이전트와의 연결 문제가 있는 경우 에이전트의 디버그 수준을 1로 변경하여 확인할 세부 로그를 생성할 수 있습니다. 자세한 내용은 [Log Insight Agents의 로그 세부 정보 수준 정의](#) 항목을 참조하십시오.

### 원인

에이전트 로그에 표시된 메시지에 관련 원인이 포함되어 있습니다.

메시지	원인
Rejecting peer self-signed certificate. Public key doesn't match previously stored certificate's key.	<ul style="list-style-type: none"> <li>■ 이는 vRealize Log Insight 인증서를 교체하는 경우 발생할 수 있습니다.</li> <li>■ 클러스터 환경에서 사용하도록 설정된 HA가 vRealize Log Insight 노드에서 다른 자체 서명된 인증서로 구성된 경우에 이 문제가 발생할 수 있습니다.</li> </ul>
Rejecting peer self-signed certificate. Have a previously received certificate which is signed by trusted CA.	에이전트 측에 CA 서명된 인증서가 저장되어 있습니다.

## 해결책

- ◆ 대상 호스트 이름이 신뢰할 수 있는 vRealize Log Insight 인스턴스인지 확인한 후 vRealize Log Insight Agent cert 디렉토리에서 이전 인증서를 수동으로 삭제합니다.
  - Log Insight Windows Agent의 경우 C:\ProgramData\VMware\Log Insight Agent\cert로 이동합니다.
  - Log Insight Linux Agent의 경우 /var/lib/loginsight-agent/cert로 이동합니다.

**참고** 일부 플랫폼은 신뢰할 수 있는 인증서를 저장하는 데 비표준 경로를 사용할 수도 있습니다. Log Insight Agents에는 `ssl_ca_path=<fullpath>` 구성 매개 변수를 설정하여 신뢰할 수 있는 인증서 저장소의 경로를 구성하는 옵션이 있습니다. <fullpath>를 신뢰할 수 있는 루트 인증서 번들 파일의 경로로 바꿉니다. [Log Insight Agent SSL 매개 변수 구성](#)을 참조하십시오.

## vRealize Log Insight 서버가 암호화되지 않은 트래픽의 연결을 거부함

vRealize Log Insight 서버는 사용자가 암호화되지 않은 트래픽을 보내려고 하면 Log Insight Agents와의 연결을 거부합니다.

SSL 이외의 연결을 수락하도록 vRealize Log Insight 서버를 구성하거나, SSL cfapi 프로토콜 연결을 통해 데이터를 보내도록 Log Insight Agents를 구성할 수 있습니다.

### 문제

cfapi를 사용하여 암호화되지 않은 트래픽을 보내려고 하면 vRealize Log Insight 서버가 해당 연결을 거부합니다. 다음 오류 메시지 중 하나가 에이전트 로그에 표시됩니다. 403 사용 권한 없음 또는 403 SSL 연결만 허용됨

### 원인

vRealize Log Insight가 SSL 연결만 허용하도록 구성된 반면 Log Insight Agents는 SSL 이외의 연결을 사용하도록 구성되었습니다.

## 해결책

- 1 SSL 이외의 연결을 수락하도록 vRealize Log Insight 서버를 구성합니다.
  - a **관리** 탭으로 이동합니다.
  - b 구성 아래에서 **SSL**을 클릭합니다.
  - c API 서버 SSL 머리글 아래에서 **SSL 연결 필요**를 선택 취소합니다.
  - d **저장**을 클릭합니다.
- 2 SSL Cfapi 프로토콜 연결을 통해 데이터를 보내도록 vRealize Log Insight 에이전트를 구성합니다.
  - a liagent.ini 파일을 포함하는 폴더로 이동합니다.

운영 체제	경로
Linux	/var/lib/loginsight-agent/
Windows	%ProgramData%\VMware\Log Insight Agent

- b 텍스트 편집기에서 liagent.ini 파일을 엽니다.
- c liagent.ini 파일의 [server] 섹션에서 ssl 키의 값을 **yes**로 변경하고 프로토콜을 cfapi로 변경합니다.

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

- d liagent.ini 파일을 저장한 후 닫습니다.