

Arbeiten mit vRealize Log Insight- Agenten

24. Mai 2022

vRealize Log Insight 8.0

Die aktuellste technische Dokumentation finden Sie auf der VMware-Website unter:

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

VMware, Inc.
3401 Hillview Ave.
Palo Alto, CA 94304
www.vmware.com

VMware Global, Inc.
Zweigniederlassung Deutschland
Willy-Brandt-Platz 2
81829 München
Germany
Tel.: +49 (0) 89 3706 17 000
Fax: +49 (0) 89 3706 17 333
www.vmware.com/de

Copyright © 2022 VMware, Inc. Alle Rechte vorbehalten. [Urheberrechts- und Markenhinweise](#).

Inhalt

Arbeiten mit vRealize Log Insight-Agenten	5
1 Übersicht über vRealize Log Insight-Agenten	6
2 Typen von Protokollrotationsschemata	9
3 Installieren oder Aktualisieren von vRealize Log Insight-Agenten	10
Herunterladen der Agent-Installationsdateien	11
Installieren von Windows-Agenten	12
Installieren oder Aktualisieren des vRealize Log Insight-Windows-Agenten mit dem Installations-Assistenten	12
Installieren oder Aktualisieren des vRealize Log Insight-Windows-Agenten von der Befehlszeile	13
Bereitstellen des Log Insight Windows Agent auf mehreren Computern	16
Installieren oder Aktualisieren des RPM-Pakets für den Linux-Agent für vRealize Log Insight	20
Installieren oder Aktualisieren des DEB-Pakets des Linux-Agent für vRealize Log Insight	21
Anpassen Ihrer Agenteninstallation für Debian Linux	22
Installieren des Log Insight Linux Agent-Binärpakets	25
Befehlszeilenoptionen für die vRealize Log Insight-Agenten-Installation auf Linux	27
Automatische Aktualisierung von vRealize Log Insight-Agenten	29
Deaktivieren oder Aktivieren der automatischen Aktualisierung für einzelne Agenten	29
4 Konfigurieren von vRealize Log Insight-Agenten	31
Konfigurieren des Log Insight Windows Agent	32
Standardkonfiguration des Log Insight Windows Agent	32
Erfassen von Ereignissen aus Windows-Ereigniskanälen	35
Erfassen von Ereignissen aus einer Protokolldatei	40
Weiterleiten von Ereignissen an den Log Insight Windows Agent	46
Konfigurieren des Log Insight Linux Agent	46
Standardkonfiguration des vRealize Log Insight Linux Agent	47
Erfassen von Ereignissen aus einer Protokolldatei	49
Filtern von Ereignissen von vRealize Log Insight-Agenten	58
Weiterleiten von Informationen von einem vRealize Log Insight-Agenten	59
Festlegen des vRealize Log Insight-Zielservers	60
Festlegen eines Agent-Ziels	64
Zentrale Konfiguration von vRealize Log Insight-Agenten	67
Exemplarische Konfigurationszusammenführung	68
Verwenden von allgemeinen Werten für die Konfiguration von Agenten	70

Analysieren von Protokollen	72
Konfigurieren von Protokoll-Parsern	72

5 Deinstallieren von vRealize Log Insight-Agenten 103

Deinstallieren von Log Insight Windows Agent	103
Deinstallieren des Log Insight Linux Agent-RPM-Pakets	103
Deinstallieren des Log Insight Linux Agent-DEB-Pakets	104
Deinstallieren des Log Insight Linux Agent-BIN-Pakets	104
Manuelles Deinstallieren des Log Insight Linux Agent-BIN-Pakets	105

6 Fehlerbehebung für vRealize Log Insight-Agenten 107

Erstellen eines Support-Pakets für den Log Insight Windows Agent	107
Erstellen eines Support-Pakets für den Log Insight Linux Agent	108
Festlegen der Protokolldateiebene in den Log Insight Agents	108
Keine Anzeige von Log Insight Agents auf der Benutzeroberfläche für Administratoren	109
vRealize Log Insight Agents senden keine Ereignisse	110
Hinzufügen einer Ausnahmeregel für den Ausgang für den Log Insight Windows Agent	112
Zulassen von ausgehenden Verbindungen vom Log Insight Windows Agent in einer Windows-Firewall	113
Die Massenbereitstellung des Log Insight Windows Agent ist nicht erfolgreich	114
Ablehnen von selbstsignierten Zertifikaten durch Log Insight Agents	114
Der vRealize Log Insight-Server lehnt die Verbindung für nicht verschlüsselten Datenverkehr ab	115

Arbeiten mit vRealize Log Insight-Agenten

In *Arbeiten mit vRealize Log Insight-Agenten* wird beschrieben, wie Sie vRealize™ Log Insight™-Agenten für Windows und Linux installieren und konfigurieren. Es enthält auch Tipps zur Fehlerbehebung.

Diese Informationen sind für Personen bestimmt, die Log Insight Agents installieren, konfigurieren oder Fehler beheben möchten. Die Informationen wurden für erfahrene Windows- oder Linux-Systemadministratoren verfasst, die mit der Technologie virtueller Maschinen sowie mit Datencenter-Vorgängen vertraut sind.

Informationen zum Erstellen von Konfigurationsklassen für Agenten beim vRealize Log Insight-Server finden Sie unter *Verwalten von vRealize Log Insight*.

Übersicht über vRealize Log Insight-Agenten

1

Ein vRealize Log Insight-Agent erfasst Ereignisse aus Protokolldateien und leitet diese an einen vRealize Log Insight-Server oder an beliebige Drittanbieter-Syslog-Ziele weiter.

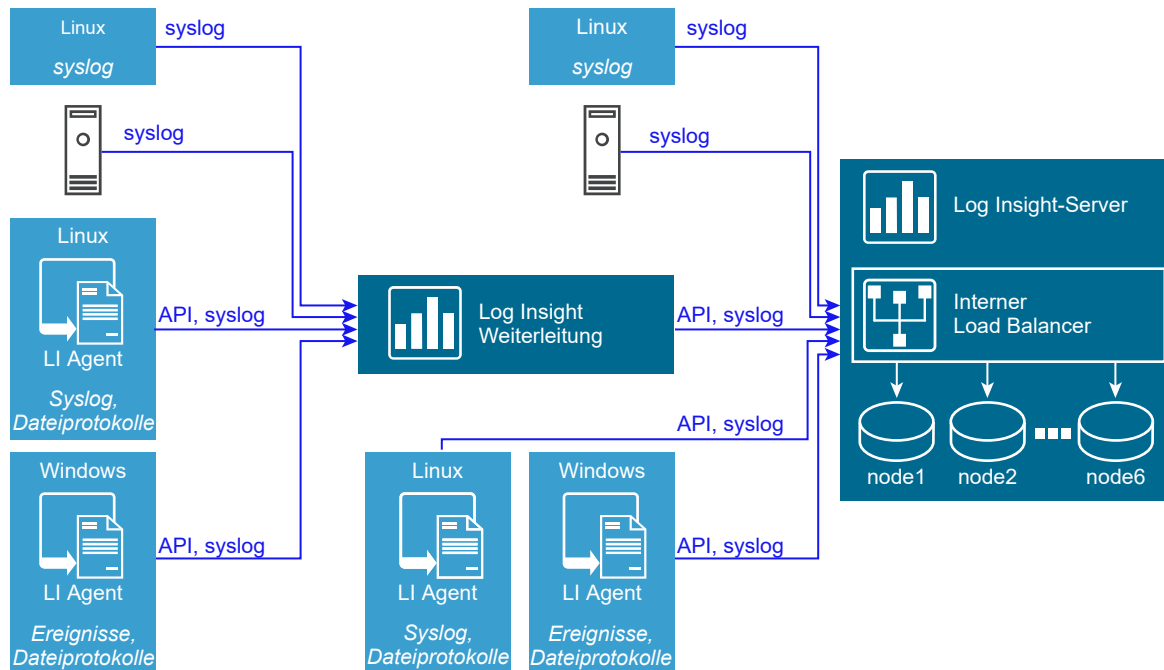
Agenten unterstützen Syslog und die vRealize Log Insight-Ingestion-API (cfapi-Protokoll) und können mit der Linux- oder Windows-Plattform verwendet werden. Sie konfigurieren Agenten über die Web-Benutzeroberfläche mit der liagent.ini-Datei auf dem Server wie auf dem Client oder als Teil der Installation.

Agenten verfügen über die folgenden Funktionen:

- Einzel- oder Gruppenbereitstellung
- Automatisches Upgrade
- Analysieren von Protokollmeldungen und zur Extraktion strukturierter Daten. Sie können Parser sowohl für FileLog- wie für WinLog-Collectors konfigurieren.
- Unterstützung mehrzeiliger Meldungen
- Native Unterstützung für mehrere Protokollrotationsschemata
- Eine umfangreiche Ingestion-API mit clientseitiger Komprimierung, Verschlüsselung und der Möglichkeit, Metadaten zu Ereignissen hinzuzufügen.

Der vRealize Log Insight-Server unterstützt eine zentrale Konfigurationsverwaltung sowie die Erstellung und Verwaltung von Agentengruppen.

Die folgende Abbildung zeigt die Elemente einer Agent-Bereitstellungskonfiguration.



Eine vRealize Log Insight-Weiterleitung ist eine dedizierte Instanz eines vRealize Log Insight-Servers, deren Hauptaufgabe die Weiterleitung von Ereignissen an ein Remoteziel ist. In der Regel wird eine Serverinstanz, die als Weiterleitung verwendet wird, nicht für die Abfrage genutzt. Die Weiterleitung verwendet einen internen Lastausgleichsdienst und ist ansonsten wie ein vRealize Log Insight-Server aufgebaut.

Die Agenten erstellen stattdessen gesonderte Vorgangsprotokolle. Für Windows sind diese Protokolle im Verzeichnis `C:\ProgramData\VMware\Log Insight Agent\logs` gespeichert. Für Linux lautet der Pfad für das Vorgangsprotokoll `/var/log/loginsight-agent/liagent_*.log`. Protokolldateien werden durch Rotation ausgetauscht, wenn ein Agent neu gestartet wird oder wenn die Datei die Größe von 10 MB erreicht. Für die Rotation gilt ein kombinierter Grenzwert von 50 MB für Dateien. Agent-Protokolle können nicht mit dem vRealize Log Insight-Agenten selbst erfasst werden.

Agenten werden für eine Protokollerfassung in Echtzeit verwendet. Für den Import bisheriger Protokollsammlungen einschließlich Support-Paketen verwenden Sie vRealize Log Insight Importer.

Für Windows- und Linux-Betriebssysteme werden keine separaten Installations-Downloads zur Verfügung gestellt.

Der Agent wird auf Windows-Systemen als Windows-Dienst ausgeführt und startet sofort nach der Installation. Der Agent überwacht Anwendungsprotokolldateien und Windows-Ereigniskanäle, aus denen zugehörige Windows-Systemereignisse erfasst werden können. Erfasste Ereignisse werden an vRealize Log Insight-Server oder Drittanbieter-Syslog-Ziele weitergeleitet.

Der Agent wird auf Linux-Systemen als Daemon ausgeführt und startet sofort nach der Installation. Der vRealize Log Insight-Linux-Agent erfasst Ereignisse aus Protokolldateien auf Linux-Systemen und leitet sie an vRealize Log Insight-Server oder Syslog-Ziele weiter. Es sind Debian, Red Hat und binäre Linux-Installationspakete verfügbar.

Von vRealize Log Insight-Agenten unterstützte Protokollrotationsschemata

2

vRealize Log Insight-Agenten unterstützen mehrere Protokollrotationsschemata.

Mit der Protokollrotation wird sichergestellt, dass die Protokolldateien eine bestimmte Größe nicht überschreiten. Es sind mehrere Protokollrotationsschemata für unterschiedliche Anwendungsfälle verfügbar. vRealize Log Insight unterstützt die im Folgenden aufgeführten Schemata.

Tabelle 2-1. Von vRealize Log Insight-Agenten unterstützte Protokollrotationsschemata

Protokollrotationsschema	Beschreibung
<code>create-new</code>	Wenn eine bestimmte Größe oder ein Zeitlimit erreicht ist, werden neue Protokolldateien erstellt. Protokolle werden dann nicht mehr in die aktuelle Protokolldatei geschrieben. Die Protokollausgabe erfolgt stattdessen in eine neu erstellte Datei. Es wird keine vorhandene Datei umbenannt oder auf andere Weise geändert.
<code>rename-recreate</code>	Ein externes Dienstprogramm wie <code>logrotate</code> benennt die Protokolldatei um, wenn eine Größe oder ein Zeitlimit erreicht ist. Im Protokollierungsvorgang wird dann eine Protokolldatei mit dem vorherigen Namen erstellt.
<code>copy-truncate</code>	Ein externes Dienstprogramm wie <code>logrotate</code> kopiert die Protokolldatei, wenn eine Größe oder ein Zeitlimit erreicht ist. Der Protokollprozess benennt die kopierte Datei um und kürzt die Originaldatei so, dass ihre Größe 0 wird. Im Protokollierungsvorgang können dann weiterhin Protokolle in die ursprüngliche Datei geschrieben werden.

Installieren oder Aktualisieren von vRealize Log Insight-Agenten

3

Sie können vRealize Log Insight-Agenten auf Windows- oder Linux-Maschinen installieren oder upgraden, inklusive jener mit Protokollverwaltungssystemen von Drittanbietern.

Agenten erfassen Ereignisse und leiten diese an den vRealize Log Insight-Server weiter. Bei der Installation können Sie Parameter für den Server, für den Port und für die Protokolleinstellungen festlegen oder die Standardeinstellungen übernehmen.

Beim Upgrade von Agenten gehen Sie vor wie bei der Installation. Alternativ können Sie ein automatisches Upgrade festlegen. Beim automatischen Upgrade werden Upgrades von Agenten bei der Bereitstellung einer neuen Version von vRealize Log Insight durchgeführt. Weitere Informationen finden Sie unter [Automatische Aktualisierung von vRealize Log Insight-Agenten](#). Für Linux-Binärpakete ist kein Upgrade verfügbar.

Hardwareunterstützung

Für die Installation und Ausführung eines vRealize Log Insight-Agenten muss Ihre Hardware die Minimalwerte für Hosts/Computer unterstützen, die die x86- und x86_64-Architektur sowie die Anweisungssätze MMX, SSE, SSE2 und SSE3 unterstützen.

Plattformunterstützung

Betriebssystem	Prozessorarchitektur
Windows 7, Windows 8, Windows 8.1 und 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 und Windows Server 2019	x86_64
RHEL 5, RHEL 6 und RHEL 7	x86_64, x86_32
SuSE Enterprise Linux (SLES) 11 SP3 und SLES 12 SP1	x86_64
Ubuntu 14.04 LTS, Ubuntu 16.04 LTS und Ubuntu 18.04	x86_64
VMware Photon, Version 1 Revision 2, Version 2 und Version 3	x86_64

Linux-Hinweise

Wenn Sie eine Standardinstallation des Log Insight-Linux-Agenten für einen Benutzer ohne Root-Berechtigung für die Verwendung implementieren, kann die Standardkonfiguration Probleme bei der Datenerfassung verursachen. Der Agent protokolliert keine Warnung, dass das Abonnement des Kanals nicht erfolgreich ist, und Dateien in der Sammlung haben keine Leseberechtigungen. Die Meldung `Inaccessible log file ... will try later` (Kein Zugriff auf Protokolldatei möglich ... Zugriff wird später erneut versucht) wird dem Protokoll mehrfach hinzugefügt. Sie können die Standardkonfiguration, die das Problem verursacht, auskommentieren oder die Benutzerberechtigungen ändern.

Wenn Sie ein rpm- oder DEB-Paket zum Installieren von Linux-Agenten verwenden, wird das init.d-Skript mit dem Namen `liagentd` als Teil der Paketinstallation installiert. Das bin-Paket fügt das Skript hinzu, registriert es aber nicht. Sie können das Skript manuell registrieren.

Sie können durch Ausführung des Befehls „(`/sbin/`) `service liagentd status`“ überprüfen, ob die Installation erfolgreich war.

Dieses Kapitel enthält die folgenden Themen:

- [Herunterladen der Agent-Installationsdateien](#)
- [Installieren von Windows-Agenten](#)
- [Installieren oder Aktualisieren des RPM-Pakets für den Linux-Agent für vRealize Log Insight](#)
- [Installieren oder Aktualisieren des DEB-Pakets des Linux-Agent für vRealize Log Insight](#)
- [Anpassen Ihrer Agenteninstallation für Debian Linux](#)
- [Installieren des Log Insight Linux Agent-Binärpakets](#)
- [Befehlszeilenoptionen für die vRealize Log Insight-Agenten-Installation auf Linux](#)
- [Automatische Aktualisierung von vRealize Log Insight-Agenten](#)

Herunterladen der Agent-Installationsdateien

Der erste Schritt zum Einrichten eines vRealize Log Insight-Agenten ist das Herunterladen eines Agenteninstallationspakets für Ihre Plattform.

Alle von der Agent-Seite des vRealize Log Insight-Servers heruntergeladenen Pakete enthalten den Zielhostnamen, der dem Paketnamen angehängt ist. Der Hostname des Servers wird während einer Erstinstallation auf die MSI-, RPM- und DEB-Agenten angewendet. Wenn in der Konfigurationsdatei ein Hostname vorhanden ist oder wenn Sie das Paket über den Parameter für den Hostnamen ausführen, wird der Hostname des eingebetteten Servers ignoriert.

Verfahren

- 1 Navigieren Sie zur Seite **Verwaltung** der vRealize Log Insight-Web-Benutzeroberfläche.
- 2 Klicken Sie im Abschnitt „Verwaltung“ auf **Agents**.

- 3 Blättern Sie zum unteren Rand des Bildschirms und klicken Sie auf **Log Insight-Agent herunterladen**.
- 4 Laden Sie das Installationspaket herunter, indem Sie es im Popup-Menü auswählen und auf **Speichern** klicken.

Option	Beschreibung
Windows MSI	Installationspaket für eine Windows-Plattform (32 Bit/64 Bit)
Linux RPM	Installationspaket für eine Linux Red Hat-, openSUSE- (32 Bit/64 Bit) oder VMware Photon Plattform
Linux DEB	Installationspaket für eine Linux Debian-Plattform (32 Bit/64 Bit)
Linux BIN	Selbstinstallierendes Paket für Linux (32 Bit/64 Bit). Ein Paketverwaltungssystem ist nicht erforderlich.

Nächste Schritte

Verwenden Sie die heruntergeladenen Dateien, um den vRealize Log Insight Agent bereitzustellen.

Installieren von Windows-Agenten

Sie können einen Agenten auf einem einzelnen Computer mithilfe des Installations-Assistenten oder über die Befehlszeile installieren. Sie haben auch die Möglichkeit, mehrere Instanzen eines Agenten mithilfe eines Skripts bereitzustellen.

Aktualisieren von Windows-Agenten

Sie können für einen Windows-Agenten ein Upgrade mithilfe einer Upgrade-Datei durchführen. Sie gehen dabei vor wie bei der Installation. Alternativ haben Sie die Möglichkeit, mithilfe der Funktion des automatischen Upgrades Ihre Agenten im Hintergrund zu aktualisieren.

Installieren oder Aktualisieren des vRealize Log Insight-Windows-Agenten mit dem Installations-Assistenten

Sie können mit dem Installations-Assistenten einen Windows-Agenten auf einem einzelnen Computer installieren oder ein Upgrade für diesen durchführen.

Voraussetzungen

- Vergewissern Sie sich, dass Sie über eine Kopie der `.msi`-Datei des Windows-Agenten für vRealize Log Insight verfügen. Weitere Informationen hierzu finden Sie unter [Herunterladen der Agent-Installationsdateien](#).
- Stellen Sie sicher, dass Sie über Berechtigungen zum Ausführen von Installationen und zum Starten von Diensten auf dem Windows-Computer verfügen.

Verfahren

- 1 Melden Sie sich bei dem Windows-Computer an, auf dem Sie den vRealize Log Insight-Windows-Agenten installieren möchten.
- 2 Wechseln Sie zu dem Verzeichnis, in dem die `.msi`-Datei des Windows-Agenten für vRealize Log Insight gespeichert ist.
- 3 Doppelklicken Sie auf die `.msi`-Datei des vRealize Log Insight Windows-Agenten, akzeptieren Sie die Bedingungen des Lizenzvertrags und klicken Sie auf **Weiter**.
- 4 Geben Sie die IP-Adresse oder den Hostnamen des vRealize Log Insight-Servers ein und klicken Sie auf **Installieren**.

Der Assistent installiert oder aktualisiert den vRealize Log Insight-Windows-Agenten als einen automatischen Windows-Dienst unter dem lokalen Systemdienstkonto.

- 5 Klicken Sie auf **Beenden**.

Nächste Schritte

Konfigurieren Sie den Windows-Agenten für vRealize Log Insight, indem Sie die Datei `liagent.ini` bearbeiten. Siehe [Konfigurieren des Log Insight Windows Agent](#).

Installieren oder Aktualisieren des vRealize Log Insight-Windows-Agenten von der Befehlszeile

Sie können den Windows-Agenten von der Befehlszeile aus installieren oder aktualisieren.

Sie können die Standardeinstellung verwenden oder ein Dienstkonto festlegen und mit Befehlszeilenparametern die Server-, Port- und Protokollinformationen angeben. Informationen zu MSI-Befehlszeilenoptionen finden Sie auf der Website der MSDN-Bibliothek (Microsoft Developer Network), wenn Sie nach MSI-Befehlszeilenoptionen suchen.

Voraussetzungen

- Vergewissern Sie sich, dass Sie über eine Kopie der `.msi`-Datei des Windows-Agenten für vRealize Log Insight verfügen. Weitere Informationen hierzu finden Sie unter [Herunterladen der Agent-Installationsdateien](#).
- Stellen Sie sicher, dass Sie über Berechtigungen zum Ausführen von Installationen und zum Starten von Diensten auf dem Windows-Computer verfügen.
- Wenn Sie die Option für die unbeaufsichtigte Installation `/quiet` oder `/qn` verwenden, stellen Sie sicher, dass Sie die Installation als Administrator ausführen. Wenn Sie kein Administrator sind und eine Installation im Hintergrund ausführen, fordert die Installation keine Administratorrechte an und schlägt fehl. Verwenden Sie die Protokollierungsoptionen und Parameter `/l{xv}* file_name` für Diagnosezwecke.

Verfahren

- 1 Melden Sie sich bei der Windows-Maschine an, auf der der vRealize Log Insight-Windows-Agent installiert oder aktualisiert werden soll.
- 2 Öffnen Sie ein Fenster mit der **Eingabeaufforderung**.
- 3 Wechseln Sie zu dem Verzeichnis, in dem die `.msi`-Datei des Windows-Agenten für vRealize Log Insight gespeichert ist.
- 4 Installieren oder aktualisieren Sie mit Standardwerten. Verwenden Sie dazu einen Befehl im folgenden Format. Ersetzen Sie `version-build_number` mit Ihrer Versions- und Build-Nummer.

Mit der Option `/quiet` wird der Befehl im Hintergrund ausgeführt. Mit der Option `/lxv` wird eine Protokolldatei im aktuellen Verzeichnis erstellt.

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

- 5 (Optional) Geben Sie ein Benutzerdienstkonto an, unter dem der Windows-Agentdienst für vRealize Log Insight ausgeführt wird.

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

Hinweis Das im Parameter `SERVICEACCOUNT` angegebene Konto muss über die Berechtigung **Anmelden als Dienst** verfügen und vollständigen Schreibzugriff auf das Verzeichnis `%ProgramData%\VMware\Log Insight Agent` haben. Wenn das angegebene Konto nicht vorhanden ist, wird es erstellt. Der Benutzername darf nicht länger als 20 Zeichen sein. Wenn Sie keinen `SERVICEACCOUNT`-Parameter angeben, wird der Windows-Agentdienst für vRealize Log Insight unter dem `LocalSystem`-Dienstkonto installiert bzw. aktualisiert.

- 6 (Optional) Sie können für die im Folgenden dargestellten Befehlszeilenoptionen nach Bedarf Werte festlegen.

Option	Beschreibung
<code>SERVERHOST=Hostname</code>	IP-Adresse oder Hostname der virtuellen vRealize Log Insight-Appliance.
<code>SERVERPROTO=Protokoll</code>	Protokoll, mit dem der Agent Ereignisse an den vRealize Log Insight-Server sendet. Die möglichen Werte lauten <code>cfapi</code> und <code>syslog</code> . Der Standardwert ist <code>cfapi</code> .

Option	Beschreibung
SERVERPORT= <i>Portnummer</i>	<p>Kommunikationsport, den der Agent zum Senden von Ereignissen an den vRealize Log Insight-Server oder an Server von Drittanbietern verwendet. Standardmäßig verwendet der Agent den richtigen Port basierend auf den Optionen, die für SSL und das Protokoll festgelegt sind. Die Standardwerte für den Port finden Sie in der unten angegebenen Liste. Sie müssen die Port-Option nur angeben, wenn sie sich von diesen Standardeinstellungen unterscheidet.</p> <ul style="list-style-type: none"> ■ cfapi mit aktiviertem SSL: 9543 ■ cfapi mit deaktiviertem SSL: 9000 ■ Syslog mit aktiviertem SSL: 6514 ■ Syslog mit deaktiviertem SSL: 514
SERVICEACCOUNT= <i>Kontoname</i>	<p>Benutzerdienstkonto, unter dem der Log Insight Windows Agent-Dienst ausgeführt wird.</p> <hr/> <p>Hinweis Das im Parameter <code>SERVICEACCOUNT</code> angegebene Konto muss über die Berechtigung Anmelden als Dienst und über Schreibzugriff für das Verzeichnis <code>%ProgramData%\VMware\Log Insight Agent</code> verfügen, damit das Installationsprogramm ordnungsgemäß ausgeführt wird. Wenn Sie keinen <code>SERVICEACCOUNT</code>-Parameter angeben, wird der Windows-Agentdienst für vRealize Log Insight unter dem LocalSystem-Dienstkonto installiert.</p>
SERVICEPASSWORD= <i>Kennwort</i>	Kennwort für das Benutzerdienstkonto.
AUTOUPDATE={yes no}	Aktivieren oder deaktivieren Sie die automatische Aktualisierung für den Agenten. Sie müssen auch die automatische Aktualisierung vom vRealize Log Insight-Server aktivieren, um die automatische Aktualisierung vollständig zu aktivieren. Die Standardeinstellung ist „yes“ (Ja).
LIAGENT_SSL={yes no}	Aktivieren Sie die sichere Verbindung. Wenn SSL aktiviert ist, verwendet der Agent ein TLS 1.2-Protokoll für die Kommunikation mit dem Server. Die Standardeinstellung ist „yes“ (Ja).

Ergebnisse

Der Befehl installiert bzw. aktualisiert den vRealize Log Insight Windows-Agenten als Windows-Dienst. Der Windows-Agentdienst für vRealize Log Insight startet, wenn Sie die Windows-Maschine starten.

Nächste Schritte

Stellen Sie sicher, dass die von Ihnen festgelegten Befehlszeilenparameter in der Datei `liagent.ini` ordnungsgemäß angewendet werden. Siehe [Konfigurieren des Log Insight Windows Agent](#).

Bereitstellen des Log Insight Windows Agent auf mehreren Computern

Die Massenbereitstellung des Log Insight Windows Agent ist auf Zielcomputern in der Windows-Domäne möglich.

Verfahren

1 Erstellen einer Umwandlungsdatei für die Bereitstellung mehrerer vRealize Log Insight-Windows-Agenten

Für die Bereitstellung mehrerer Agenten müssen Sie eine Umwandlungsdatei erstellen, die die Konfigurationsparameter für die Bereitstellung festlegt. Die .mst-Umwandlungsdatei wird auf die .msi-Datei angewendet, wenn Sie Agenten installieren. Zu den Parametern gehören der Zielservers für die Agenten sowie das Kommunikationsprotokoll, der Port und das Benutzerkonto für die Installation und den Start des Log Insight-Agent-Dienstes.

2 Bereitstellen von mehreren Instanzen des vRealize Log Insight Windows-Agenten

Sie können mehrere Instanzen des vRealize Log Insight Windows-Agenten auf Zielcomputern in einer Windows-Domäne bereitstellen.

Erstellen einer Umwandlungsdatei für die Bereitstellung mehrerer vRealize Log Insight-Windows-Agenten

Für die Bereitstellung mehrerer Agenten müssen Sie eine Umwandlungsdatei erstellen, die die Konfigurationsparameter für die Bereitstellung festlegt. Die .mst-Umwandlungsdatei wird auf die .msi-Datei angewendet, wenn Sie Agenten installieren. Zu den Parametern gehören der Zielservers für die Agenten sowie das Kommunikationsprotokoll, der Port und das Benutzerkonto für die Installation und den Start des Log Insight-Agent-Dienstes.

Zu den Parametern gehören der Zielservers für die Agenten sowie das Kommunikationsprotokoll, der Port und das Benutzerkonto für die Installation und den Start des Log Insight-Agent-Dienstes.

Voraussetzungen

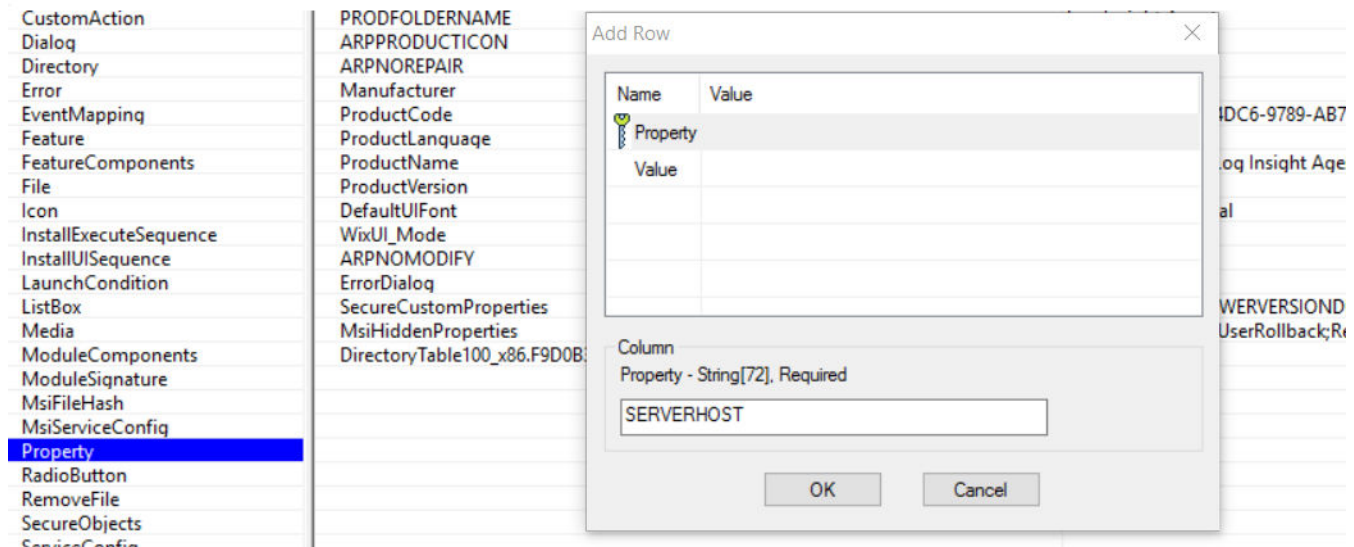
- Vergewissern Sie sich, dass Sie über eine Kopie der .msi-Datei des Windows-Agenten für vRealize Log Insight verfügen. Weitere Informationen hierzu finden Sie unter [Herunterladen der Agent-Installationsdateien](#).
- Laden Sie den Orca-Datenbankeditor herunter und installieren Sie ihn. Siehe <http://support.microsoft.com/kb/255905>.

Verfahren

- 1 Öffnen Sie die Datei .msi des vRealize Log Insight-Windows-Agenten im Orca-Editor und wählen Sie **Umwandeln > Neu umwandeln** aus.

- 2 Bearbeiten Sie die Tabelle „Eigenschaft“ und fügen Sie die nötigen Parameter und Werte für eine benutzerdefinierte Installation oder ein benutzerdefiniertes Upgrade hinzu.

Abbildung 3-1. Eigenschaftstabelle



- a Klicken Sie auf **Eigenschaft**.
- b Wählen Sie aus dem Dropdown-Menü **Tabelle** die Option **Zeile hinzufügen** aus.
- c Geben Sie einen Namen und einen Wert für die Eigenschaft in das Dialogfeld „Zeile hinzufügen“ ein.

Die Parameter sind in der folgenden Tabelle enthalten.

Parameter	Beschreibung
SERVERHOST	IP-Adresse oder Hostname der virtuellen vRealize Log Insight-Appliance. Die Standardeinstellung lautet loginsight .
SERVERPROTO	Protokoll, mit dem der Agent Ereignisse an den vRealize Log Insight-Server sendet. Die möglichen Werte lauten cfapi und syslog . Der Standardwert ist cfapi .
SERVERPORT	Kommunikationsport, den der Agent zum Senden von Ereignissen an den vRealize Log Insight-Server oder an Server von Drittanbietern verwendet. Standardmäßig verwendet der Agent den richtigen Port basierend auf den Optionen, die für SSL und das Protokoll festgelegt sind. Die Standardwerte für den Port finden Sie in der unten angegebenen Liste. Sie müssen die Port-Option nur angeben, wenn sie sich von diesen Standardeinstellungen unterscheidet. <ul style="list-style-type: none"> ■ cfapi mit aktiviertem SSL: 9543 ■ cfapi mit deaktiviertem SSL: 9000 ■ Syslog mit aktiviertem SSL: 6514 ■ Syslog mit deaktiviertem SSL: 514

Parameter	Beschreibung
SERVICEACCOUNT	Benutzerdienstkonto, unter dem der Log Insight Windows Agent-Dienst ausgeführt wird. Hinweis Das im Parameter <code>SERVICEACCOUNT</code> angegebene Konto muss über die Berechtigung Anmelden als Dienst und über Schreibzugriff für das Verzeichnis <code>%ProgramData%\VMware\Log Insight Agent</code> verfügen, damit das Installationsprogramm ordnungsgemäß ausgeführt wird. Wenn Sie keinen <code>SERVICEACCOUNT</code> -Parameter angeben, wird der Windows-Agentdienst für vRealize Log Insight unter dem LocalSystem-Dienstkonto installiert.
SERVICEPASSWORD	Kennwort für das Benutzerdienstkonto.
AUTOUPDATE	Aktivieren oder deaktivieren Sie die automatische Aktualisierung für den Agenten. Sie müssen auch die automatische Aktualisierung vom vRealize Log Insight-Server aktivieren, um die automatische Aktualisierung vollständig zu aktivieren. Die Standardeinstellung ist „yes“ (Ja).
LIAGENT-SSL	Aktivieren Sie die sichere Verbindung. Wenn SSL aktiviert ist, verwendet der Agent ein TLS 1.2-Protokoll für die Kommunikation mit dem Server. Die Standardeinstellung ist „yes“ (Ja).

- 3 Wählen Sie **Umwandeln > Umwandlung generieren** aus und speichern Sie die `.mst`-Umwandlungsdatei.

Nächste Schritte

Verwenden Sie die `.msi`-Datei und die `.mst`-Datei für die Bereitstellung des vRealize Log InsightWindows-Agenten.

Bereitstellen von mehreren Instanzen des vRealize Log Insight Windows-Agenten

Sie können mehrere Instanzen des vRealize Log Insight Windows-Agenten auf Zielcomputern in einer Windows-Domäne bereitstellen.

Weitere Informationen darüber, weshalb Sie den Clientcomputer zweimal neu starten müssen, finden Sie unter <http://support.microsoft.com/kb/305293>.

Voraussetzungen

- Stellen Sie sicher, dass Sie über ein Administratorkonto bzw. ein Konto mit Administratorrechten im Domänencontroller verfügen.
- Vergewissern Sie sich, dass Sie über eine Kopie der `.msi`-Datei des Windows-Agenten für vRealize Log Insight verfügen. Weitere Informationen hierzu finden Sie unter [Herunterladen der Agent-Installationsdateien](#).
- Machen Sie sich mit den Verfahren in <http://support.microsoft.com/kb/887405> und <http://support.microsoft.com/kb/816102> vertraut.

Verfahren

- 1 Melden Sie sich beim Domänencontroller als Administrator bzw. Benutzer mit Administratorrechten an.
- 2 Definieren Sie einen Bereitstellungspunkt und kopieren Sie die `-msi`-Datei für den vRealize Log Insight Windows-Agenten an den Bereitstellungspunkt.
- 3 Öffnen Sie die Gruppenrichtlinien-Managementkonsole und erstellen Sie ein Gruppenrichtlinienobjekt, um die `.msi`-Datei für den vRealize Log Insight Windows-Agenten bereitzustellen.
- 4 Bearbeiten Sie das Gruppenrichtlinienobjekt für die Softwarebereitstellung und weisen Sie ein Paket zu.
- 5 (Optional) Wenn Sie vor der Bereitstellung eine `.mst`-Datei erstellt haben, wählen Sie die `.mst`-Konfigurationsdatei auf der Registerkarte **Änderungen** in Fenster **GPO-Eigenschaften** aus, und verwenden Sie die erweiterte Methode, um ein Gruppenrichtlinienobjekt für die Bereitstellung des `.msi`-Pakets zu bearbeiten.
- 6 (Optional) Aktualisieren Sie den vRealize Log Insight Windows-Agenten.
 - a Kopieren Sie die `.msi`-Datei des Upgrades an den Verteilungspunkt.
 - b Klicken Sie auf die Registerkarte **Upgrade** im Fenster **Eigenschaften** des Gruppenrichtlinienobjekts.
 - c Fügen Sie die anfangs installierte Version der `.msi`-Datei in den Abschnitt „Pakete, die dieses Paket aktualisiert“ ein.
- 7 Stellen Sie den vRealize Log Insight Windows-Agenten für bestimmte Sicherheitsgruppen bereit, die die Domänenbenutzer umfassen.
- 8 Schließen Sie alle Fenster der Gruppenrichtlinien-Managementkonsole und des Gruppenrichtlinien-Management-Editors auf dem Domänencontroller, und starten Sie die Clientcomputer neu.

Wenn die Optimierung für schnelles Anmelden aktiviert ist, starten Sie die Clientcomputer zweimal neu.
- 9 Überprüfen Sie, ob der vRealize Log Insight Windows-Agent auf den Clientcomputern als lokaler Dienst installiert wurde.

Wenn Sie `SERVICEACCOUNT`- und `SERVICEPASSWORD`-Parameter für die Verwendung einer `.mst`-Datei zum Bereitstellen von mehreren Instanzen des vRealize Log Insight Windows-Agenten konfiguriert haben, vergewissern Sie sich, dass der vRealize Log Insight Windows-Agent auf den Clientcomputern unter dem angegebenen Benutzerkonto installiert ist.

Nächste Schritte

Für den Fall, dass die mehrfachen Instanzen des vRealize Log Insight Windows-Agenten nicht erfolgreich sind, finden Sie weitere Informationen unter [Die Massenbereitstellung des Log Insight Windows Agent ist nicht erfolgreich](#).

Installieren oder Aktualisieren des RPM-Pakets für den Linux-Agent für vRealize Log Insight

Sie können den Linux-Agenten für vRealize Log Insight als Root- oder Nicht-Root-Benutzer installieren oder aktualisieren und Sie können Konfigurationsparameter bei der Installation festlegen. Nach der Installation können Sie die installierte Version überprüfen.

Voraussetzungen

- Weitere Informationen zu den Standardeinstellungen der Installation und deren Änderung finden Sie unter [Befehlszeilenoptionen für die vRealize Log Insight-Agenten-Installation auf Linux](#).
- Melden Sie sich als **Root**-Benutzer an oder verwenden Sie `sudo`, um Konsolenbefehle auszuführen.
- Der Linux-Agent von vRealize Log Insight benötigt für den Betrieb Zugriff auf Syslog und die Netzwerkdienste. Installieren Sie den Linux-Agenten für vRealize Log Insight und führen Sie ihn auf den Ebenen 3 und 5 aus. Wenn der Linux-Agent für vRealize Log Insight auf anderen Ausführungsebenen betrieben werden soll, konfigurieren Sie das System entsprechend.

Verfahren

- 1 Sie können einen Agenten von der Konsole aus installieren oder aktualisieren.

- Um den vRealize Log Insight-Linux-Agenten mit den standardmäßigen Konfigurationseinstellungen zu installieren, öffnen Sie eine Konsole und führen Sie den folgenden Befehl aus:

```
rpm -i VMware-Log-Insight-Agent-<Versions-und-Build-Nummer>.rpm
```

- Um für den Agenten ein Upgrade durchzuführen, ohne die aktuellen Konfigurationseinstellungen zu ändern, öffnen Sie eine Konsole und führen Sie den folgenden Befehl aus:

```
rpm -Uvh VMware-Log-Insight-Agent-<Versions-und-Build-Nummer>.rpm
```

- 2 (Optional) Sie können die standardmäßigen Konfigurationswerte für die Installation oder die aktuellen Konfigurationswerte bei einem Update überschreiben. Dazu geben Sie bei der Installation oder im Upgrade-Befehl die betreffende Option an.

```
sudo <OPTION=Wert> rpm -i <Versions-und-Build-Nummer>.rpm
```

- 3 (Optional) Überprüfen Sie die installierte Version, indem Sie den folgenden Befehl ausführen:

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

Beispiel: Beispiele für Linux-Agenteninstallation und -aktualisierung

- Der folgende Befehl installiert den vRealize Log Insight-Agenten für eine Linux-RPM-Verteilung. Er installiert den Agenten auf einem separaten Server, weist eine nicht standardmäßige Portnummer zu und erstellt einen vRealize Log Insight-Agentenbenutzer.

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

- Der folgende Befehl aktualisiert den Agenten mit der angegebenen rpm-Datei. Die aktuelle Agentenkonfiguration bleibt unverändert.

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

Installieren oder Aktualisieren des DEB-Pakets des Linux-Agent für vRealize Log Insight

Sie können das DEB-Paket (Debian) des Linux-Agent für vRealize Log Insight über die Befehlszeile oder über die Debconf-Datenbank installieren oder aktualisieren. Nach der Installation können Sie die installierte Version überprüfen.

Voraussetzungen

- Weitere Informationen zu den Standardeinstellungen der Installation und deren Änderung finden Sie unter [Befehlszeilenoptionen für die vRealize Log Insight-Agenten-Installation auf Linux](#).
- Melden Sie sich als **Root**-Benutzer an oder verwenden Sie `sudo`, um Konsolenbefehle auszuführen.
- Überprüfen Sie, ob der Linux-Agent für vRealize Log Insight Zugriff auf syslog- und Netzwerkdienste hat, damit er funktioniert. Standardmäßig wird der Linux-Agent für vRealize Log Insight auf den Ausführungsebenen 2, 3, 4 und 5 ausgeführt und auf den Ausführungsebenen 0, 1 und 6 angehalten.
- Weitere Informationen und Beispiele finden Sie unter [Anpassen Ihrer Agenteninstallation für Debian Linux](#).

Verfahren

- 1 Um den Linux-Agent für vRealize Log Insight zu installieren oder zu aktualisieren, öffnen Sie eine Konsole und führen Sie den Befehl `dpkg -i package_name` aus.

package_name besteht aus dem Präfix **vmware-log-insight-agent-** und aus der Versions-Build-Nummer Ihrer Download-Version.

Mit dem im Folgenden aufgeführten Befehl wird das Paket mit den Standardwerten installiert.

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

- 2 (Optional) Überprüfen Sie die installierte Version, indem Sie den folgenden Befehl ausführen:

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

Beispiel

- Konfigurieren Sie das Verbindungsprotokoll über die Befehlszeile.

Zum Überschreiben des Standardverbindungsprotokolls verwenden Sie die Variable `SERVERPROTO`, wie im folgenden Beispiel dargestellt:

```
sudo SERVERPROTO=syslog dpkg -i vmware-log-insight-agent-<Versions-und-Build-Nummer>_all.deb
```

Anpassen Ihrer Agenteninstallation für Debian Linux

Sie können die Installation mithilfe von Befehlsoptionen anpassen und damit die aktuellen Konfigurationswerte für die Installation ändern oder die Debconf-Datenbank konfigurieren.

Anpassung über die Befehlszeile

Für die Konfiguration Ihrer Installation über die Befehlszeile verwenden Sie einen Befehl im folgenden Format:

```
sudo <OPTION=Wert> dpkg -i vmware-log-insight-agent-<Versions-und-Build-Nummer>_all.deb
```

Eine vollständige Liste der Optionen finden Sie unter [Befehlszeilenoptionen für die vRealize Log Insight-Agenten-Installation auf Linux](#).

Die folgenden Beispiele zeigen einige typische Konfigurationen über die Befehlszeile.

- Geben Sie einen Ziel-vRealize Log Insight-Server an.

- Um den Zielsever bei der Installation festzulegen, führen Sie den Befehl `sudo` aus, wobei Sie „hostname“ durch die IP-Adresse oder den Hostnamen des vRealize Log Insight-Servers ersetzen, wie im folgenden Beispiel dargestellt:

```
sudo SERVERHOST=hostname dpkg -iv mware-log-insight-agent-<Versions-und-Build-Nummer>_all.deb
```

Sofern Sie nicht das `--force-confold`-Flag während der Installation aktiviert haben, müssen Sie bei jeder Aktualisierung auf eine neuere Version wählen, ob Sie die Konfigurationsdatei `liagent.ini` beibehalten oder ersetzen. Die folgenden Systemmeldungen werden angezeigt:

```
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] ?
```

Verwenden Sie zum Beibehalten der vorhandenen Konfiguration [Standard = N]. Die von der Befehlszeile übergebenen zusätzlichen Parameter werden nach wie vor angewandt.

- Konfigurieren Sie das Verbindungsprotokoll.

Zum Überschreiben des Standardverbindungsprotokolls verwenden Sie die Variable `SERVERPROTO`, wie im folgenden Beispiel dargestellt:

```
sudo SERVERPROTO=syslog dpkg -i vmware-log-insight-agent-<Versions-und-Build-Nummer>_all.deb
```

- Konfigurieren Sie den Client-Verbindungsport.

Zum Überschreiben des Standardverbindungsports legen Sie einen Wert für die Variable `SERVERPORT` im Installationsprogramm wie im folgenden Beispiel dargestellt fest:

```
sudo SERVERPORT=1234 dpkg -i vmware-log-insight-agent-<Versions-und-Build-Nummer>_all.deb
```

- Führen Sie den Agenten als Nicht-Root-Benutzer aus.

Um den Linux-Agent für vRealize Log Insight als **Nicht-Root**-Benutzer auszuführen, verwenden Sie den Befehl `sudo`.

```
sudo LIAGENTUSER=liagent dpkg -i vmware-log-insight-agent-<Versions-Build-Nummer>_all.deb
```

Wenn der angegebene Benutzer nicht existiert, erstellt der Linux-Agent für vRealize Log Insight das entsprechende Benutzerkonto während der Installation. Das erstellte Konto wird nach der Deinstallation nicht gelöscht. Wenn Sie den Linux-Agent mit dem Parameter `LIAGENTUSER=non_root_user` installieren und versuchen, ein Upgrade mit dem Parameter `LIAGENTUSER=non_root_user2` auszuführen, tritt ein Konflikt auf. Es werden Warnmeldungen angezeigt, da der Benutzer `non_root_user2` nicht über die Berechtigungen des Benutzers `non_root_user` verfügt.

Anpassungsoptionen des DEB-Pakets für die Debconf-Datenbank

Das Agent-DEB-Paket kann auch über die Debconf-Datenbank konfiguriert werden. Die folgende Tabelle zeigt unterstützte Debconf-Optionen und entsprechende vRealize Log Insight Agent DEB-Installer-Optionen:

Befehlszeilenoptionen	Debconf-Optionen	Beschreibung
<code>SERVERHOST=Hostname</code>	<code>vmware-log-insight-agent/serverhost</code>	IP-Adresse oder Hostname der virtuellen vRealize Log Insight-Appliance. Die Standardeinstellung lautet loginsight .
<code>SERVERPROTO={cfapi syslog}</code>	<code>vmware-log-insight-agent/serverproto</code>	Protokoll, mit dem der Agent Ereignisse an den vRealize Log Insight-Server sendet. Die möglichen Werte lauten <code>cfapi</code> und <code>syslog</code> . Der Standardwert ist <code>cfapi</code> .
<code>SERVERPORT=Portnummer</code>	<code>vmware-log-insight-agent/serverport</code>	Kommunikationsport, den der Agent zum Senden von Ereignissen an den vRealize Log Insight-Server oder an Server von Drittanbietern verwendet. Standardmäßig verwendet der Agent den richtigen Port basierend auf den Optionen, die für SSL und das Protokoll festgelegt sind. Die Standardwerte für den Port finden Sie in der unten angegebenen Liste. Sie müssen die Port-Option nur angeben, wenn sie sich von diesen Standardeinstellungen unterscheidet. <ul style="list-style-type: none"> ■ <code>cfapi</code> mit aktiviertem SSL: 9543 ■ <code>cfapi</code> mit deaktiviertem SSL: 9000 ■ <code>Syslog</code> mit aktiviertem SSL: 6514 ■ <code>Syslog</code> mit deaktiviertem SSL: 514
<code>LIAGENT_INITSYSTEM={init systemd}</code>	<code>log-insight-agent/init_system</code>	Während der Installation erkennt der Agent automatisch den Init-Systemtyp für die Maschine, auf der Sie den Agent installieren. Sie können dieses Verhalten außer Kraft setzen, indem Sie den Typ des Systemwerts mit dieser Option angeben. Es gibt zwei Arten von unterstützten Init-Systemen: <code>init</code> und <code>systemd</code> .

Befehlszeilenoptionen	Debconf-Optionen	Beschreibung
<code>LIAGENT_AUTOUPDATE={yes no}</code>	<code>vmware-log-insight-agent/auto_update</code>	<p>Aktivieren oder deaktivieren Sie die automatische Aktualisierung für den Agenten. Sie müssen auch die automatische Aktualisierung vom vRealize Log Insight-Server aktivieren, um die automatische Aktualisierung vollständig zu aktivieren. Die Standardeinstellung ist „yes“ (Ja).</p> <p>Die automatische Aktualisierung wird nicht für Linux-BIN-Pakete unterstützt.</p>
<code>LI_AGENT_RUNSERVICES</code>	<code>vmware-log-insight-agent/init_system</code>	<p>Sofort nach der Installation werden die Dienste <code>liagentd</code> (Agent) und <code>liupdaterd</code> (Updater) standardmäßig gestartet. Sie können deren Start verhindern, indem Sie den Parameter „LIAGENT_RUNSERVICES Debconf“ auf no (Nein) festlegen. Die Standardeinstellung ist „yes“ (Ja). Es werden nur die Werte yes und no akzeptiert. Die Werte 1 oder 0 werden nicht unterstützt.</p>
<code>LIAGENT_SSL</code>	<code>vmware-log-insight-agent/ssl</code>	C
<code>LIAGENTUSER=Benutzername-Konto</code>	<code>vmware-log-insight-agent/liagentuser</code>	<p>Gibt ein Konto an, unter dem der Agent ausgeführt wird. Wenn der Benutzer nicht existiert, erstellt sie das Installationsprogramm als normaler Benutzer. Wenn das angegebene Benutzerkonto nicht existiert, erstellt der Linux-Agent für vRealize Log Insight das entsprechende Benutzerkonto während der Installation. Das erstellte Konto wird nach der Deinstallation nicht gelöscht.</p> <p>Standardmäßig wird der Agent installiert, um als Root-Benutzer ausgeführt zu werden.</p> <p>Wenn Sie den Agent mit dem Parameter <code>LIAGENTUSER=non_root_user</code> installieren und versuchen, mit <code>LIAGENTUSER=non_root_user2</code> ein Upgrade auszuführen, kommt es zu einem Konflikt. Es werden dann entsprechende Warnungen angezeigt, da der Benutzer <code>non_root_user2</code> nicht über die Berechtigungen des Benutzers <code>non_root_user</code> verfügt.</p> <p>Der erstellte Benutzer wird während der Deinstallation nicht entfernt. Er kann manuell entfernt werden. Dieser Parameter ist nur für den Agent-Dienst bestimmt. Der Updater-Dienst wird immer als ein Root-Benutzer ausgeführt.</p>

Installieren des Log Insight Linux Agent-Binärpakets

Zum Installieren des Binärpakets muss die .bin-Datei in eine ausführbare Datei geändert und dann der Agent installiert werden.

Ein Upgrade des .bin-Pakets wird offiziell nicht unterstützt. Wenn Sie das .bin-Paket verwendet haben, um einen vorhandenen Log Insight Linux Agent zu installieren, legen Sie eine Sicherheitskopie der Datei `liagent.ini` im Verzeichnis `/var/lib/loginsight-agent` an, um die lokale Konfiguration beizubehalten. Nach dem Anlegen der Sicherheitskopie können Sie den Log Insight Linux Agent manuell deinstallieren. Weitere Informationen hierzu finden Sie unter [Manuelles Deinstallieren des Log Insight Linux Agent-BIN-Pakets](#).

Wenn Sie das .bin-Paket verwenden, um Linux-Agenten zu installieren, ist das Skript `init.d` mit dem Namen `liagentd` als Teil der Paketinstallation installiert, aber das Paket registriert das Skript nicht. Sie können das Skript manuell registrieren.

Sie können überprüfen, ob die Installation erfolgreich war, indem Sie den Befehl (`/sbin/service liagentd status`) ausführen.

Voraussetzungen

- Laden Sie das .bin-Paket für den Log Insight Linux Agent herunter und kopieren Sie es auf den Linux-Zielcomputer.
- Überprüfen Sie, dass der Log Insight Linux Agent Zugriff auf syslog- und Netzwerkdienste hat.
- Weitere Informationen über die Standardwerte der Konfiguration und wie diese bei der Installation geändert werden können, finden Sie unter [Befehlszeilenoptionen für die vRealize Log Insight-Agenten-Installation auf Linux](#).

Verfahren

- 1 Öffnen Sie eine Konsole und führen Sie den Befehl `chmod` aus, um die .bin-Datei in eine ausführbare Datei umzuwandeln.

Ersetzen Sie *filename-version* durch die entsprechende Version.

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

- 2 Führen Sie an der Eingabeaufforderung den Befehl `./filename-version.bin` aus, um den Agenten zu installieren.

Ersetzen Sie *filename-version* durch die entsprechende Version.

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

- 3 (Optional) Um den vRealize Log Insight-Zielserver während der Installation festzulegen, führen Sie den Befehl `sudo SERVERHOST=hostname ./filename-version.bin` aus.

Ersetzen Sie *hostname* mit der IP-Adresse oder dem Hostnamen des vRealize Log Insight-Servers.

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

- 4 (Optional) Zum Überschreiben des Standardverbindungsprotokolls verwenden Sie die Variable `SERVERPROTO`, wie im folgenden Beispiel dargestellt:

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

- 5 (Optional) Zum Überschreiben des Standardverbindungsports legen Sie einen Wert für die Variable `SERVERPORT` im Installationsprogramm wie im folgenden Beispiel dargestellt fest:

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

- 6 (Optional) Um Log Insight Linux Agent als **Nicht-Root**-Benutzer auszuführen, verwenden Sie den Befehl `sudo`.

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

Wenn der angegebene Benutzer nicht existiert, erstellt der Log Insight Linux Agent das entsprechende Benutzerkonto während der Installation. Das erstellte Konto wird nach der Deinstallation nicht gelöscht. Wenn Sie den Log Insight Linux Agent mit dem Parameter `LIAGENTUSER=non_root_user` installieren und versuchen, ein Upgrade mit dem Parameter `LIAGENTUSER=non_root_user2` auszuführen, tritt ein Konflikt ein und Warnmeldungen werden angezeigt, weil der Benutzer `non_root_user2` keine Berechtigungen des Benutzers `non_root_user` hat.

Befehlszeilenoptionen für die vRealize Log Insight-Agenten-Installation auf Linux

Wenn Sie vRealize Log Insight-Agenten über die Befehlszeile installieren, können Sie mithilfe von Optionen Ihre Bereitstellung bei der Installation konfigurieren. Diese Optionen entsprechen den Einstellungen in der Datei `liagent.ini`.

Mit den im Folgenden aufgeführten Optionen können während der Installation vRealize Log Insight-Agenten konfiguriert werden, die auf Linux-Systemen ausgeführt werden.

Option	Beschreibung
<code>SERVERHOST=Hostname</code>	IP-Adresse oder Hostname der virtuellen vRealize Log Insight-Appliance. Die Standardeinstellung lautet loginsight .
<code>SERVERPROTO={cfapi syslog}</code>	Protokoll, mit dem der Agent Ereignisse an den vRealize Log Insight-Server sendet. Die möglichen Werte lauten <code>cfapi</code> und <code>syslog</code> . Der Standardwert ist <code>cfapi</code> .

Option	Beschreibung
<code>SERVERPORT=Portnummer</code>	<p>Kommunikationsport, den der Agent zum Senden von Ereignissen an den vRealize Log Insight-Server oder an Server von Drittanbietern verwendet. Standardmäßig verwendet der Agent den richtigen Port basierend auf den Optionen, die für SSL und das Protokoll festgelegt sind. Die Standardwerte für den Port finden Sie in der unten angegebenen Liste. Sie müssen die Port-Option nur angeben, wenn sie sich von diesen Standardeinstellungen unterscheidet.</p> <ul style="list-style-type: none"> ■ cfapi mit aktiviertem SSL: 9543 ■ cfapi mit deaktiviertem SSL: 9000 ■ Syslog mit aktiviertem SSL: 6514 ■ Syslog mit deaktiviertem SSL: 514
<code>LIAGENT_INITSYSTEM={init systemd}</code>	<p>Während der Installation erkennt der Agent automatisch den Init-Systemtyp für die Maschine, auf der Sie den Agent installieren. Sie können dieses Verhalten außer Kraft setzen, indem Sie den Typ des Systemwerts mit dieser Option angeben. Es gibt zwei Arten von unterstützten Init-Systemen: <code>init</code> und <code>systemd</code>.</p>
<code>LIAGENT_AUTOUPDATE={Ja Nein}</code>	<p>Aktivieren oder deaktivieren Sie die automatische Aktualisierung für den Agenten. Sie müssen auch die automatische Aktualisierung vom vRealize Log Insight-Server aktivieren, um die automatische Aktualisierung vollständig zu aktivieren. Die Standardeinstellung ist „yes“ (Ja).</p> <p>Die automatische Aktualisierung wird nicht für Linux-BIN-Pakete unterstützt.</p>

Option	Beschreibung
<code>LIAGENT_SSL={Ja Nein}</code>	Aktivieren Sie die sichere Verbindung. Wenn SSL aktiviert ist, verwendet der Agent ein TLS 1.2-Protokoll für die Kommunikation mit dem Server. Die Standardeinstellung ist „yes“ (Ja).
<code>LIAGENTUSER=Benutzername-Konto</code>	<p>Gibt ein Konto an, unter dem der Agent ausgeführt wird. Wenn der Benutzer nicht existiert, erstellt sie das Installationsprogramm als normaler Benutzer. Wenn das angegebene Benutzerkonto nicht existiert, erstellt der Linux-Agent für vRealize Log Insight das entsprechende Benutzerkonto während der Installation. Das erstellte Konto wird nach der Deinstallation nicht gelöscht.</p> <p>Standardmäßig wird der Agent installiert, um als Root-Benutzer ausgeführt zu werden.</p> <p>Wenn Sie mit dem Parameter <code>LIAGENTUSER=non_root_user</code> installieren und versuchen, mit <code>LIAGENTUSER=non_root_user2</code> ein Upgrade durchzuführen, kommt es zu einem Konflikt. Es werden dann entsprechende Warnungen angezeigt, da der Benutzer <code>non_root_user2</code> nicht über die Berechtigungen des Benutzers <code>non_root_user</code> verfügt.</p> <p>Der erstellte Benutzer wird während der Deinstallation nicht entfernt. Er kann manuell entfernt werden. Dieser Parameter ist nur für den Agent-Dienst bestimmt.</p> <p>Der Updater-Dienst wird immer als ein Root-Benutzer ausgeführt.</p>

Automatische Aktualisierung von vRealize Log Insight-Agenten

Die Funktion zur automatischen Aktualisierung von vRealize Log Insight-Agenten ermöglicht aktiven Agenten das Überprüfen, Herunterladen und automatische Installieren von Updates auf der Grundlage der Installationspakete für Agenten auf dem vRealize Log Insight-Server.

Sie können die automatische Aktualisierung vom Server für alle Agenten oder von den Clients für einzelne Agent-Instanzen aktivieren. Agenten müssen aktiviert und in der Version 4.3 oder höher vorhanden sein.

Die automatische Aktualisierung wird nicht für Linux-BIN-Pakete unterstützt.

Deaktivieren oder Aktivieren der automatischen Aktualisierung für einzelne Agenten

Sie können die automatische Aktualisierung für einzelne Agenten durch Bearbeitung der Client-seitigen Konfigurationsdatei für den jeweiligen Agenten aktivieren bzw. deaktivieren.

Standardmäßig ist die automatische Aktualisierung Client-seitig für einen Agenten aktiviert.

Voraussetzungen

Die Agenten müssen in der Version 4.3 oder höher vorhanden sein.

Verfahren

- 1 Öffnen Sie die lokale Datei `liagent.ini` in einem Editor.
- 2 Suchen Sie den Abschnitt `[update]`.

Dieser entspricht ungefähr dem nachfolgenden Beispiel.

```
[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 Um die automatische Aktualisierung zu deaktivieren, entfernen Sie die Kommentarfunktion `auto_update=yes` und ändern Sie sie in `auto_update=no`.

Hinweis Die automatische Aktualisierung für Agenten ist standardmäßig aktiviert. Daher lautet der Standardwert für `auto_update` „Ja“, auch mit Kommentar.

- 4 Speichern und schließen Sie die Datei `liagent.ini`.

Konfigurieren von vRealize Log Insight-Agenten

4

Nachdem Sie einen Agenten bereitgestellt haben, können Sie ihn konfigurieren, indem Sie Ereignisse an den ausgewählten vRealize Log Insight-Server senden, Kommunikationsprotokolle und andere Parameter angeben.

Befolgen Sie diese Anleitungen nach Bedarf, um Ihre Agenten entsprechend Ihren Bedürfnissen zu konfigurieren.

- **Konfigurieren des Log Insight Windows Agent**

Nachdem Sie Log Insight Windows Agent installiert haben, können Sie ihn konfigurieren. Konfigurieren Sie Log Insight Windows Agent in der Datei `liagent.ini` so, dass Ereignisse an vRealize Log Insight gesendet werden, legen Sie das Kommunikationsprotokoll und den entsprechenden Port fest, fügen Sie Windows-Ereigniskanäle hinzu und konfigurieren Sie die Flatfile-Protokollierung. Die Datei ist im Verzeichnis `% ProgramData %\VMware\Log Insight Agent` gespeichert.

- **Konfigurieren des Log Insight Linux Agent**

Nachdem Sie Log Insight Linux Agent installiert haben, können Sie ihn konfigurieren.

- **Filtern von Ereignissen von vRealize Log Insight-Agenten**

Sie können mit der Filteroption im Abschnitt `[server|<dest_id>]` Ihrer lokalen `liagent.ini`-Datei die Informationen angeben, die ein Agent zu einem Ziel sendet.

- **Weiterleiten von Informationen von einem vRealize Log Insight-Agenten**

Sie können von einem Agenten erfasste Ereignisse an bis zu drei Ziele weiterleiten. Zu diesen Zielen können auch vRealize Log Insight-Server oder eine vRealize Log Insight-Weiterleitung bzw. Drittanbieterlösungen zur Protokollverwaltung gehören.

- **Zentrale Konfiguration von vRealize Log Insight-Agenten**

Sie können mehrere vRealize Log Insight-Agenten konfigurieren.

- **Verwenden von allgemeinen Werten für die Konfiguration von Agenten**

Sie können die Standardwerte der Agent-Konfigurationsdatei mit allgemeinen Parameterwerten, die in jedem Agent-Konfigurationsabschnitt für Windows- und Linux-Agenten angewendet werden, überschreiben.

■ Analysieren von Protokollen

Agentenseitige Protokoll-Parser extrahieren strukturierte Daten aus unstrukturierten Protokollen, bevor an den vRealize Log Insight-Server geliefert wird. Mithilfe von Protokoll-Parsern kann vRealize Log Insight Protokolle analysieren, Informationen daraus extrahieren und diese Ergebnisse auf dem Server anzeigen. Protokoll-Parser können für vRealize Log Insight-Agenten sowohl für Windows als auch für Linux konfiguriert werden.

Konfigurieren des Log Insight Windows Agent

Nachdem Sie Log Insight Windows Agent installiert haben, können Sie ihn konfigurieren. Konfigurieren Sie Log Insight Windows Agent in der Datei `liagent.ini` so, dass Ereignisse an vRealize Log Insight gesendet werden, legen Sie das Kommunikationsprotokoll und den entsprechenden Port fest, fügen Sie Windows-Ereigniskanäle hinzu und konfigurieren Sie die Flatfile-Protokollierung. Die Datei ist im Verzeichnis `% ProgramData %\VMware\Log Insight Agent` gespeichert.

Standardkonfiguration des Log Insight Windows Agent

Nach der Installation enthält die `liagent.ini`-Datei vorkonfigurierte Standardeinstellungen für den Log Insight Windows Agent.

Log Insight Windows Agent – liagent.ini – Standardkonfiguration

Wenn Sie Nicht-ASCII-Namen und -Werte verwenden, speichern Sie die Konfiguration als UTF-8.

Wenn Sie die zentrale Konfiguration verwenden, besteht die abschließende Konfiguration aus dieser Datei und Einstellungen vom Server, die zusammen die Datei `liagent-effective.ini` bilden.

Es kann für Sie effizienter sein, die Einstellungen von der Agentenseite des Servers aus zu konfigurieren.

```
; 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

```

Parameter	Standardwert	Beschreibung
hostname	LOGINSIGHT	IP-Adresse oder Hostname der virtuellen vRealize Log Insight-Appliance. Die Standardeinstellung lautet loginsight .
central_config	yes	Aktivieren oder deaktivieren Sie die zentrale Konfiguration für diesen Agent. Wenn die zentrale Konfiguration deaktiviert ist, ignoriert der Agent die vom vRealize Log Insight-Server bereitgestellte Konfiguration. Zugelassene Werte sind <code>yes</code> , <code>no</code> , <code>1</code> oder <code>0</code> . Der Standardwert lautet <code>yes</code> .
proto	cfapi	Protokoll, mit dem der Agent Ereignisse an den vRealize Log Insight-Server sendet. Die möglichen Werte lauten <code>cfapi</code> und <code>syslog</code> . Der Standardwert ist <code>cfapi</code> .
port	9543, 9000, 6514 und 514	Kommunikationsport, den der Agent zum Senden von Ereignissen an den vRealize Log Insight-Server oder an Server von Drittanbietern verwendet. Standardmäßig verwendet der Agent den richtigen Port basierend auf den Optionen, die für SSL und das Protokoll festgelegt sind. Die Standardwerte für den Port finden Sie in der unten angegebenen Liste. Sie müssen die Port-Option nur angeben, wenn sie sich von diesen Standardeinstellungen unterscheidet. <ul style="list-style-type: none"> ■ cfapi mit aktiviertem SSL: 9543 ■ cfapi mit deaktiviertem SSL: 9000 ■ Syslog mit aktiviertem SSL: 6514 ■ Syslog mit deaktiviertem SSL: 514
ssl	yes	Aktiviert oder deaktiviert SSL. Der Standardwert lautet „ja“. Wenn <code>ssl</code> aktiviert ist und Sie keinen Wert für den Port festlegen, wird der Port automatisch auf „9543“ gesetzt.

Parameter	Standardwert	Beschreibung
<code>max_disk_buffer</code>	200	Die maximale Festplattengröße in MB, die der Log Insight Windows Agent zum Puffern von Ereignissen und seinen eigenen Protokollen verwendet. Wenn der angegebene <code>max_disk_buffer</code> erreicht ist, beginnt der Agent, neue eingehende Ereignisse zu löschen.
<code>debug_level</code>	0	Legt die Protokolldateiebene fest. Weitere Informationen hierzu finden Sie unter Festlegen der Protokolldateiebene in den Log Insight Agents .
<code>channel</code>	Anwendung, Sicherheit, System	Die Windows-Ereigniskanäle „Anwendung“, „Sicherheit“ und „System“ sind standardmäßig kommentiert. Der Log Insight Windows Agent erfasst keine Protokolldaten von diesen Kanälen. Weitere Informationen hierzu finden Sie unter Erfassen von Ereignissen aus Windows-Ereigniskanälen .
<code>raw_syslog</code>	no	Ermöglicht dem Agent bei Agents, die das Syslog-Protokoll verwenden, nicht formatierte Syslog-Ereignisse zu erfassen und zu senden. Die Standardeinstellung lautet „Nein“, d. h. die erfassten Ereignisse werden mit benutzerspezifischen Syslog-Attributen umgewandelt. Aktivieren Sie diese Option, um Ereignisse ohne Syslog-Umwandlungen zu erfassen. Gültige Werte sind „Ja“ oder 1 und „Nein“ oder 0.

Erfassen von Ereignissen aus Windows-Ereigniskanälen

Sie können einen Windows-Ereigniskanal zur Log Insight Windows Agent-Konfiguration hinzufügen. Der Log Insight Windows Agent erfasst die Ereignisse und sendet sie an den vRealize Log Insight-Server.

Feldnamen sind eingeschränkt. Die folgenden Namen sind reserviert und können nicht als Feldnamen verwendet werden.

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

■ text

Voraussetzungen

Melden Sie sich bei dem Windows-Computer an, auf dem Sie den vRealize Log Insight Windows-Agent installiert haben, und starten Sie den Dienst-Manager, um zu überprüfen, ob der vRealize Log Insight-Agent-Dienst installiert ist.

Verfahren

- 1 Navigieren Sie zum Programmdatenordner des Windows-Agenten für vRealize Log Insight.

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

- 2 Öffnen Sie die Datei `liagent.ini` in einem beliebigen Texteditor.

- 3 Fügen Sie die folgenden Parameter hinzu und legen Sie die Werte für Ihre Umgebung fest.

Parameter	Beschreibung
<code>[winlog section_name]</code>	Ein eindeutiger Name für den Konfigurationsabschnitt.
<code>channel</code>	Der vollständige Name des Ereigniskanals, wie in der Ereignisanzeige der integrierten Windows-Anwendung, angezeigt. Um den richtigen Kanalnamen zu kopieren, klicken Sie in der Ereignisanzeige mit der rechten Maustaste auf einen Kanal, wählen Sie Eigenschaften aus und kopieren Sie die Inhalte des Felds Vollständiger Name .
<code>enabled</code>	Ein optionaler Parameter zum Aktivieren oder Deaktivieren des Konfigurationsabschnitts. Mögliche Werte sind „ja“ oder „nein“ (die Groß- und Kleinschreibung wird nicht beachtet). Der Standardwert lautet „ja“.
<code>tags</code>	<p>Optionaler Parameter zum Hinzufügen von benutzerdefinierten Tags zu den Feldern der erfassten Ereignisse. Definieren Sie Tags mit der JSON-Notation. Tagnamen können Buchstaben, Zahlen und Unterstriche enthalten. Ein Tagname darf nur mit einem Buchstaben oder einem Unterstrich beginnen und darf nicht mehr als 64 Zeichen enthalten. Bei Tagnamen wird die Groß- und Kleinschreibung nicht berücksichtigt. Wenn Sie zum Beispiel <code>tags={"tag_name1" : "tag value 1", "Tag_Name1" : "tag value 2" }</code> verwenden, wird <code>Tag_Name1</code> als Duplikat ignoriert. Sie können „event_type“ und „timestamp“ nicht als Tagnamen verwenden. Alle Duplikate innerhalb derselben Deklaration werden ignoriert.</p> <p>Wenn das Ziel ein Syslog-Server ist, können Tags das Feld „APP-NAME“ überschreiben. Beispiel: <code>tags={"appname": "VROPS"}</code>.</p>

Parameter	Beschreibung
whitelist, blacklist	Optionale Parameter zum expliziten Einbeziehen oder Ausschließen von Protokollereignissen. Hinweis Die Option <code>blacklist</code> gilt nur für Felder. Sie kann nicht verwendet werden, um Text zu blockieren.
exclude_fields	(Optional) Ein Parameter zum Ausschließen einzelner Felder aus der Erfassung. Sie können mehrere Werte als eine durch Semikolons getrennte Liste eingeben. Beispiel: <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 Speichern und schließen Sie die Datei `liagent.ini`.

Beispiel: Konfigurationen

Beachten Sie die folgenden `[winlog|`-Konfigurationsbeispiele.

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

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

Einrichten der Filterung für Windows-Ereigniskanäle

Sie können Filter für Windows-Ereigniskanäle einrichten, um Protokollereignisse explizit aufzunehmen bzw. auszuschließen.

Zur Auswertung eines Filterausdrucks verwenden Sie die Parameter `whitelist` und `blacklist`. Der Filterausdruck ist ein boolescher Ausdruck, der aus Ereignisfeldern und -operatoren besteht.

Hinweis Die Option `blacklist` gilt nur für Felder. Sie kann nicht verwendet werden, um Text zu blockieren.

- Der `whitelist`-Parameter erfasst nur Protokollereignisse, für welche die Auswertung des Filterausdrucks ungleich null ist. Wenn Sie diesen Parameter auslassen, ist der Wert eine implizierte 1.
- Der `blacklist`-Parameter schließt Protokollereignisse aus, für welche die Auswertung des Filterausdrucks ungleich null ist. Der Standardwert lautet 0.

Eine umfassende Liste der Windows-Ereignisfelder und -Operanden finden Sie unter [Ereignisfelder und Operanden](#).

Voraussetzungen

Melden Sie sich bei dem Windows-Computer an, auf dem Sie den vRealize Log Insight Windows-Agent installiert haben, und starten Sie den Dienst-Manager, um zu überprüfen, ob der vRealize Log Insight-Agent-Dienst installiert ist.

Verfahren

- 1 Navigieren Sie zum Programmdatenordner des Windows-Agenten für vRealize Log Insight.

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

- 2 Öffnen Sie die Datei `liagent.ini` in einem beliebigen Texteditor.

- 3 Fügen Sie den Parameter `whitelist` oder `blacklist` im Abschnitt `[winlog|]` hinzu.

Beispiel:

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

- 4 Erstellen Sie einen Filterausdruck aus Windows-Ereignisfeldern und -Operanden.

Beispiel:

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

- 5 Speichern und schließen Sie die Datei `liagent.ini`.

Beispiel: Filterkonfigurationen

Sie können den Agent beispielsweise so konfigurieren, dass nur Fehlerereignisse erfasst werden.

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

Sie können den Agent beispielsweise so konfigurieren, dass nur VMware Network-Ereignisse aus dem Anwendungskanal erfasst werden.

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

Sie können den Agent beispielsweise so konfigurieren, dass mit Ausnahme bestimmter Ereignisse alle Ereignisse aus dem Sicherheitskanal erfasst werden.

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

Ereignisfelder und Operanden

Erstellen Sie Filterausdrücke mithilfe der Windows-Ereignisfelder und -Operanden.

Operatoren für Filterausdrücke

Operator	Beschreibung
==, !=	gleich und ungleich. Sowohl mit Zahlen- als auch mit Zeichenfolgenfeldern verwenden.
>=, >, <, <=	größer oder gleich, größer als, kleiner als, kleiner oder gleich. Nur mit Zahlenfeldern verwenden.
&, , ^, ~	Bitweise AND, OR, XOR und ergänzende Operanden. Nur mit Zahlenfeldern verwenden.
AND, OR	Logisches AND und OR. Für den Aufbau komplexer Ausdrücke durch Kombination von Einzelausdrücken.
nicht	Unär logischer NOT-Operand. Für die Umkehr des Werts eines Ausdrucks.
()	Verwenden Sie Klammern in einem logischen Ausdruck, um die Auswertungsreihenfolge zu ändern.

Windows-Ereignisfelder

Sie können die folgenden Windows-Ereignisfelder in einem Filterausdruck verwenden:

Feldname	Feldtyp
Hostname	String
Text	String
ProviderName	String
EventSourceName	String
EventID	numeric
EventRecordID	numeric
Kanal	String
UserID	String
Level	numeric Sie können die folgenden vordefinierten Konstanten verwenden. <ul style="list-style-type: none"> ■ WINLOG_LEVEL_SUCCESS = 0 ■ WINLOG_LEVEL_CRITICAL = 1 ■ WINLOG_LEVEL_ERROR = 2 ■ WINLOG_LEVEL_WARNING = 3 ■ WINLOG_LEVEL_INFO = 4 ■ WINLOG_LEVEL_VERBOSE = 5
Aufgabe	numeric

Feldname	Feldtyp
OpCode	numeric
Schlüsselwörter	numeric Sie können die folgenden vordefinierten Bitmasken verwenden: <ul style="list-style-type: none"> ■ WINLOG_KEYWORD_RESPONSETIME = 0x0001000000000000; ■ WINLOG_KEYWORD_WDICONTEXT = 0x0002000000000000; ■ WINLOG_KEYWORD_WDIDIAGNOSTIC = 0x0004000000000000; ■ WINLOG_KEYWORD_SQM = 0x0008000000000000; ■ WINLOG_KEYWORD_AUDITFAILURE = 0x0010000000000000; ■ WINLOG_KEYWORD_AUDITSUCCESS = 0x0020000000000000; ■ WINLOG_KEYWORD_CORRELATIONHINT = 0x0040000000000000; ■ WINLOG_KEYWORD_CLASSIC = 0x0080000000000000;

Beispiele

Erfassen aller kritischen Ereignisse sowie Fehler- und Warnereignisse

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

Erfassen nur von Auditfehler-Ereignissen über den Sicherheitskanal

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

Erfassen von Ereignissen aus einer Protokolldatei

Der Windows-Agent für vRealize Log Insight kann so konfiguriert werden, dass er Ereignisse aus einer oder mehreren Protokolldateien erfasst.

Feldnamen sind eingeschränkt. Die folgenden Namen sind reserviert und können nicht als Feldnamen verwendet werden.

- event_type
- hostname
- source
- text

Sie können bis zu drei Ziele für Agent-Informationen verwenden und die Informationen vor dem Senden filtern. Weitere Informationen hierzu finden Sie unter [Weiterleiten von Informationen von einem vRealize Log Insight-Agenten](#).

Hinweis Die Überwachung einer großen Anzahl an Dateien, z. B. von tausend Dateien oder mehr, führt zu einer höheren Ressourcennutzung des vRealize Log Insight-Agents. Dies wirkt sich auf die Gesamtleistung der Hostmaschine aus. Um dies zu verhindern, konfigurieren Sie den Agent so, dass er nur die erforderlichen Dateien mithilfe von Mustern und Globs überwacht, oder archivieren Sie die alten Protokolldateien. Wenn die Überwachung einer großen Anzahl an Dateien erforderlich ist, können Sie es auch in Erwägung ziehen, die Hostparameter wie CPU und RAM zu erhöhen.

Hinweis Ein Agent kann Ereignisse aus verschlüsselten Ordnern erfassen. Der Agent kann nur dann Daten aus einem verschlüsselten Ordner erfassen, wenn er von dem Benutzer ausgeführt wird, der den Ordner verschlüsselt hat.

Voraussetzungen

Melden Sie sich bei dem Windows-Computer an, auf dem Sie den vRealize Log Insight Windows-Agent installiert haben, und starten Sie den Dienst-Manager, um zu überprüfen, ob der vRealize Log Insight-Agent-Dienst installiert ist.

Verfahren

- 1 Navigieren Sie zum Programmdatenordner des Windows-Agenten für vRealize Log Insight.

`%ProgramData%\VMware\Log Insight Agent`

- 2 Öffnen Sie die Datei `liagent.ini` in einem beliebigen Texteditor.

- 3 Suchen Sie den Abschnitt [server|<dest_id>] der Datei. Fügen Sie Konfigurationsparameter hinzu und legen Sie die Werte für Ihre Umgebung fest.

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

Parameter	Beschreibung
[filelog section_name]	Ein eindeutiger Name für den Konfigurationsabschnitt.
directory=full-path-to-log-file	<p>Der vollständige Pfad zum Protokolldateiverzeichnis. Glob-Muster werden unterstützt. Beispielkonfigurationen:</p> <ul style="list-style-type: none"> ■ Um eine Sammlung aus allen Unterverzeichnissen von D:\Logs\new_test_logs durchzuführen, verwenden Sie <code>directory=D:\Logs\new_test_logs*</code>. ■ Wenn Ihre Unterverzeichnisse über eigene Unterverzeichnisse verfügen, verwenden Sie die folgende Konfiguration, um alle Unterverzeichnisse <code>directory=D:\Logs\new_test_logs**</code> zu überwachen <p>Hinweis Um die Anzahl der Dateien und Ordner zu begrenzen und einen hohen Ressourcenverbrauch zu vermeiden, können Sie kein Verzeichnis-Glob für die Verzeichnisse der ersten oder der zweiten Ebene definieren, wie z. B. <code>directory=c:/tmp/*</code> oder <code>directory=c:\Logs*</code>. Der Verzeichnispfad muss mindestens zwei Ebenen aufweisen.</p> <p>Sie haben auch die Möglichkeit, einen Pfad zu einem nicht vorhandenen Verzeichnis zu definieren. Der Agent erfasst dann die Protokolldateien in diesem Verzeichnis, wenn das Verzeichnis und die Dateien erstellt wurden. Sie können dasselbe Verzeichnis unter einem oder mehreren verschiedenen Konfigurationsbereichen definieren, um Protokolle von derselben Datei mehrmals zu erfassen. Damit können verschiedene Tags und Filter auf dieselbe Quelle von Ereignissen angewendet werden.</p> <p>Hinweis Wenn Sie identische Konfigurationen für diese Bereiche verwenden, werden duplizierte Ereignisse auf der Serverseite beobachtet.</p>

Parameter	Beschreibung
<code>include= file_name; ...</code>	<p>(Optional) Der Name eines Dateinamens oder einer Dateimaske (Glob-Muster) für die Datenerfassung. Sie können Werte als eine durch Semikolon getrennte Liste eingeben. Der Standardwert lautet <code>*</code>. Das heißt, dass alle Dateien berücksichtigt werden. Bei diesem Parameter muss die Groß-/Kleinschreibung beachtet werden.</p> <p>Mit einer Dateimaske (Glob-Muster) können Dateien gruppiert werden, die der gleichen Benennungskonvention folgen. Dies ist auch auf der Ebene einzelner Zeichen in Dateinamen möglich. Beispielsweise können Dateinamen mit Leerzeichen wie z. B. <code>vRealize Ops Datei Analytics.log</code> und <code>vRealize Ops Collector.log</code> mit <code>vRealize?Ops?Analytics*.log</code> oder <code>vRealize*.log</code> angegeben werden. Dateimasken bieten die Möglichkeit, Dateinamen festzulegen, die für die Agentenkonfiguration auf Linux- und Windows-Hosts zulässig sind. <code>.zip</code>- und <code>.gz</code>-Dateien werden standardmäßig nicht erfasst.</p> <p>Wichtig Verwenden Sie bei der Erfassung einer rotierten Protokolldatei die Parameter <code>include</code> und <code>exclude</code>, um ein Glob-Muster anzugeben, das der primären und der rotierten Datei entspricht. Wenn das Glob-Muster nur der primären Protokolldatei entspricht, verpassen die vRealize Log Insight-Agenten möglicherweise Ereignisse während der Rotation. Die vRealize Log Insight-Agenten legen die richtige Reihenfolge der rotierten Dateien automatisch fest und senden Ereignisse in der richtigen Reihenfolge an den vRealize Log Insight-Server. Wenn es sich beispielsweise bei Ihrer primären Protokolldatei um <code>myapp.log</code> und bei den rotierten Protokollen um <code>myapp.log.1</code>, <code>myapp.log.2</code> usw. handelt, können Sie das folgende <code>include</code>-Muster verwenden:</p> <pre>include= myapp.log;myapp.log.*</pre>
<code>exclude= regular_expression</code>	<p>(Optional) Ein Dateiname oder eine Dateimaske (Glob-Muster) zum Ausschließen aus der Erfassung. Sie können Werte als eine durch Semikolon getrennte Liste eingeben. Der Standardwert ist leer. Das heißt, dass keine Datei ausgeschlossen wird.</p>
<code>event_marker= regular_expression</code>	<p>(Optional) Ein regulärer Ausdruck, der den Start eines Ereignisses in der Protokolldatei angibt. Wird dieser Ausdruck weggelassen, wird standardmäßig ein Zeilenumbruch ausgeführt. Die von Ihnen eingegebenen Ausdrücke müssen die Perl-Syntax für reguläre Ausdrücke verwenden.</p> <p>Hinweis Symbole, wie zum Beispiel Anführungszeichen (<code>"</code> <code>"</code>), werden nicht als Wrapper für reguläre Ausdrücke behandelt. Sie werden als Teil des Musters behandelt.</p> <p>Da der vRealize Log Insight Agent für die Erfassung in Echtzeit optimiert ist, können mit einer internen Verzögerung geschriebene Protokoll-Teilnachrichten in mehrere Ereignisse aufgeteilt werden. Wenn das Anhängen an Protokolldateien für mehr als 200 ms ohne einen beobachteten <code>event_marker</code> angehalten wird, wird das Teilereignis als abgeschlossen, analysiert und zugestellt betrachtet. Diese Timing-Logik ist nicht konfigurierbar und hat Vorrang vor der <code>event_marker</code>-Einstellung. Protokolldatei-Appender sollten alle vollständigen Ereignisse löschen.</p>
<code>enabled=yes no</code>	<p>(Optional) Ein Parameter zum Aktivieren oder Deaktivieren des Konfigurationsabschnitts. Die möglichen Werte lauten <code>yes</code> oder <code>no</code>. Der Standardwert lautet <code>yes</code>.</p>

Parameter	Beschreibung
charset=char-encoding-type	<p>(Optional) Die Zeichenkodierung der Protokolldateien, die der Agent überwacht. Mögliche Werte:</p> <ul style="list-style-type: none"> ■ UTF-8 ■ UTF-16LE ■ UTF-16BE <p>Der Standardwert lautet UTF-8.</p>
tags={"tag-name": "tag-value", ...}	<p>(Optional) Ein Parameter zum Hinzufügen von benutzerdefinierten Tags zu den Feldern der erfassten Ereignisse. Definieren Sie Tags mit der JSON-Notation. Tagnamen können Buchstaben, Zahlen und Unterstriche enthalten. Ein Tagname darf nur mit einem Buchstaben oder einem Unterstrich beginnen und darf nicht mehr als 64 Zeichen enthalten. Bei Tagnamen wird die Groß- und Kleinschreibung nicht berücksichtigt. Wenn Sie zum Beispiel tags={"tag_name1": "tag value 1", "Tag_Name1": "tag value 2"} verwenden, wird Tag_Name1 als Duplikat ignoriert. Sie können „event_type“ und „timestamp“ nicht als Tagnamen verwenden. Alle Duplikate innerhalb derselben Deklaration werden ignoriert.</p> <p>Wenn das Ziel ein Syslog-Server ist, können Tags das Feld „APP-NAME“ überschreiben. Beispiel: tags={"appname": "VROPS"}.</p>
exclude_fields	<p>(Optional) Ein Parameter zum Ausschließen einzelner Felder aus der Erfassung. Sie können mehrere Werte in Form einer durch Semikolons oder Kommas getrennten Liste eingeben. Beispiel:</p> <ul style="list-style-type: none"> ■ exclude_fields=hostname; filepath ■ exclude_fields=type; size ■ exclude_fields=type, size
raw_syslog=Yes No	<p>Für Agenten, die das Syslog-Protokoll verwenden, ermöglicht diese Option dem Agenten, nicht formatierte Syslog-Ereignisse zu erfassen und zu senden. Die Standardeinstellung ist „Nein“, d. h., die erfassten Ereignisse werden mit benutzerspezifischen Syslog-Attributen umgewandelt. Aktivieren Sie diese Option, um Ereignisse ohne Syslog-Umwandlungen zu erfassen.</p>

Beispiel: Konfigurationen

```
[filelog|vCenterMain]
directory=C:\ProgramData\VMware\VMware VirtualCenter\Logs
include=vpzd-*.log
exclude=vpzd-alert-*.log;vpzd-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]
```

Einrichten der Windows-Protokolldateikanal-Filterung

Sie können Filter für Windows-Protokolldateien einrichten, um Protokollereignisse explizit aufzunehmen bzw. auszuschließen.

Zur Auswertung eines Filterausdrucks verwenden Sie die Parameter `whitelist` und `blacklist`. Der Filterausdruck ist ein boolescher Ausdruck, der aus Ereignisfeldern und -operatoren besteht.

Hinweis Die Option `blacklist` gilt nur für Felder. Sie kann nicht verwendet werden, um Text zu blockieren.

- Der `whitelist`-Parameter erfasst nur Protokollereignisse, für welche die Auswertung des Filterausdrucks ungleich null ist. Wenn Sie diesen Parameter auslassen, ist der Wert eine implizierte 1.
- Der `blacklist`-Parameter schließt Protokollereignisse aus, für welche die Auswertung des Filterausdrucks ungleich null ist. Der Standardwert lautet 0.

Eine umfassende Liste der Windows-Ereignisfelder und -Operanden finden Sie unter [Ereignisfelder und Operanden](#).

Voraussetzungen

Melden Sie sich bei dem Windows-Computer an, auf dem Sie den vRealize Log Insight Windows-Agent installiert haben, und starten Sie den Dienst-Manager, um zu überprüfen, ob der vRealize Log Insight-Agent-Dienst installiert ist.

Verfahren

- 1 Navigieren Sie zum Programmdatenordner des Windows-Agenten für vRealize Log Insight.
`%ProgramData%\VMware\Log Insight Agent`
- 2 Öffnen Sie die Datei `liagent.ini` in einem beliebigen Texteditor.
- 3 Fügen Sie den Parameter `whitelist` oder `blacklist` im Abschnitt `[filelog|]` hinzu.

Beispiel:

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

- 4 Erstellen Sie einen Filterausdruck aus Windows-Ereignisfeldern und -Operanden.

Beispiel:

```
whitelist = myServer
```

5 Speichern und schließen Sie die Datei `liagent.ini`.

Beispiel: Filterkonfigurationen

Sie können den Agent so konfigurieren, dass nur Apache-Protokolle mit folgendem „server_name“ erfasst werden:

```
[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"
```

Weiterleiten von Ereignissen an den Log Insight Windows Agent

Sie können Ereignisse von Windows-Systemen an ein System weiterleiten, auf dem der Log Insight Windows Agent ausgeführt wird.

Sie können die Windows-Ereignisweiterleitung nutzen, um Ereignisse von mehreren Windows-Computern an einen Computer zu senden, auf dem Log Insight Windows Agent installiert ist. Dann können Sie den Log Insight Windows Agent so konfigurieren, dass alle weitergeleiteten Ereignisse erfasst und an einen vRealize Log Insight-Server gesendet werden.

Machen Sie sich mit der Windows-Ereignisweiterleitung vertraut. Siehe <http://technet.microsoft.com/en-us/library/cc748890.aspx> und [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).

Voraussetzungen

Weitere Informationen hierzu finden Sie unter [Erfassen von Ereignissen aus Windows-Ereigniskanälen](#).

Verfahren

- 1 Fügen Sie der Log Insight Windows Agent-Konfiguration eine neue Seite hinzu, um Ereignisse aus dem Windows-Ereigniskanal, der weitergeleitete Ereignisse empfängt, zu erfassen.

Der Standard-Kanalname lautet ForwardedEvents.

- 2 Richten Sie die Windows-Ereignisweiterleitung ein.

Nächste Schritte

Navigieren Sie zur vRealize Log Insight-Web-Benutzeroberfläche und stellen Sie sicher, dass weitergeleitete Ereignisse eintreffen.

Konfigurieren des Log Insight Linux Agent

Nachdem Sie Log Insight Linux Agent installiert haben, können Sie ihn konfigurieren.

Sie können die [zentrale Agent-Konfiguration](#) verwenden, um den Agent so einzurichten, dass Ereignisse an einen vRealize Log Insight-Server gesendet werden. Legen Sie das Kommunikationsprotokoll und den Kommunikationsport fest und konfigurieren Sie eine Flat-Datei-Protokollsammlung. Informationen zum Speicherort der Datei `liagent.ini` finden Sie unter [Festlegen der Protokolldateiebene in den Log Insight Agents](#).

Standardkonfiguration des vRealize Log Insight Linux Agent

Nach der Installation enthält die `liagent.ini`-Datei vorkonfigurierte Standardeinstellungen für den Log Insight Windows Agent.

vRealize Log Insight Linux Agent – liagent.ini – Standardkonfiguration

Wenn Sie Nicht-ASCII-Namen und -Werte verwenden, speichern Sie die Konfiguration als UTF-8.

Wenn Sie die zentrale Konfiguration verwenden, besteht die abschließende Konfiguration aus dieser Datei und Einstellungen vom Server, die zusammen die Datei „liagent-effective.ini“ bilden.

Es kann für Sie effizienter sein, die Einstellungen von der Agentenseite des Servers aus zu konfigurieren.

```
[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.?.

```

Parameter	Standardwert	Beschreibung
hostname	LOGINSIGHT	IP-Adresse oder Hostname der virtuellen vRealize Log Insight-Appliance. Die Standardeinstellung lautet loginsight .
central_config	yes	Aktivieren oder deaktivieren Sie die zentrale Konfiguration für diesen Agent. Wenn die zentrale Konfiguration deaktiviert ist, ignoriert der Agent die vom vRealize Log Insight-Server bereitgestellte Konfiguration. Zugelassene Werte sind yes , no , 1 oder 0 . Der Standardwert lautet yes .
proto	cfapi	Protokoll, mit dem der Agent Ereignisse an den vRealize Log Insight-Server sendet. Die möglichen Werte lauten cfapi und syslog . Der Standardwert ist cfapi .
port	9543, 9000, 6514 und 514	Kommunikationsport, den der Agent zum Senden von Ereignissen an den vRealize Log Insight-Server verwendet. Die Standardwerte sind 9543 für cfapi mit aktiviertem SSL, 9000 für cfapi mit deaktiviertem SSL, 6514 für Syslog mit aktiviertem SSL und 514 für syslog mit deaktiviertem SSL.
ssl	yes	Aktiviert oder deaktiviert SSL. Der Standardwert lautet „ja“. Wenn ssl aktiviert ist und Sie keinen Wert für den Port festlegen, wird der Port automatisch auf „9543“ gesetzt.

Parameter	Standardwert	Beschreibung
<code>max_disk_buffer</code>	200	Die maximale Festplattengröße in MB, die der Log Insight Windows Agent zum Puffern von Ereignissen und seinen eigenen Protokollen verwendet. Wenn der angegebene <code>max_disk_buffer</code> erreicht ist, beginnt der Agent, neue eingehende Ereignisse zu löschen.
<code>debug_level</code>	0	Legt die Protokolldateiebene fest. Weitere Informationen hierzu finden Sie unter Festlegen der Protokolldateiebene in den Log Insight Agents .

Erfassen von Ereignissen aus einer Protokolldatei

Der Linux-Agent für vRealize Log Insight kann so konfiguriert werden, dass er Ereignisse aus einer oder mehreren Protokolldateien erfasst.

Standardmäßig erfasst der vRealize Log Insight-Linux-Agent ausgeblendete Dateien, die von Anwendungen oder Editoren erstellt werden. Die Namen der ausgeblendeten Dateien beginnen mit einem Punkt. Sie können verhindern, dass der vRealize Log Insight-Linux-Agent ausgeblendete Dateien erfasst, indem Sie den Ausschlussparameter `exclude=.` hinzufügen.

Feldnamen sind eingeschränkt. Die folgenden Namen sind reserviert und können nicht als Feldnamen verwendet werden.

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

Sie haben die Möglichkeit, bis zu drei Ziele für Agent-Informationen festzulegen und die Informationen vor dem Senden zu filtern. Siehe [Weiterleiten von Informationen von einem vRealize Log Insight-Agenten](#).

Hinweis Die Überwachung einer großen Anzahl an Dateien, z. B. von tausend Dateien oder mehr, führt zu einer höheren Ressourcennutzung des vRealize Log Insight-Agents. Dies wirkt sich auf die Gesamtleistung der Hostmaschine aus. Um dies zu verhindern, konfigurieren Sie den Agent so, dass er nur die erforderlichen Dateien mithilfe von Mustern und Globs überwacht, oder archivieren Sie die alten Protokolldateien. Wenn die Überwachung einer großen Anzahl an Dateien erforderlich ist, können Sie es auch in Erwägung ziehen, die Hostparameter wie CPU und RAM zu erhöhen.

Voraussetzungen

- Melden Sie sich als **Root**-Benutzer an oder verwenden Sie `sudo`, um Konsolenbefehle auszuführen.
- Stellen Sie sicher, dass der vRealize Log Insight-Linux-Agent installiert und ausgeführt ist. Melden Sie sich bei dem Linux-System an, auf dem Sie den vRealize Log Insight-Linux-Agent installiert haben. Öffnen Sie eine Konsole und führen Sie `pgrep liagent` aus.

Verfahren

- 1 Öffnen Sie die Datei `/var/lib/loginsight-agent/liagent.ini` in einem beliebigen Texteditor.
- 2 Suchen Sie den Abschnitt `[server|<dest_id>]` der Datei. Fügen Sie Konfigurationsparameter hinzu und legen Sie die Werte für Ihre Umgebung fest.

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

Parameter	Beschreibung
<code>[filelog section_name]</code>	Ein eindeutiger Name für den Konfigurationsabschnitt.
<code>directory=full-path-to-log-file</code>	<p>Der vollständige Pfad zum Protokolldateiverzeichnis. Glob-Muster werden unterstützt. Beispielkonfigurationen:</p> <ul style="list-style-type: none"> ■ Um eine Sammlung aus allen Unterverzeichnissen von <code>D:\Logs\new_test_logs</code> durchzuführen, verwenden Sie <code>directory=D:\Logs\new_test_logs*</code>. ■ Wenn Ihre Unterverzeichnisse über eigene Unterverzeichnisse verfügen, verwenden Sie die folgende Konfiguration, um alle Unterverzeichnisse <code>directory=D:\Logs\new_test_logs**</code> zu überwachen <p>Hinweis Um die Anzahl der Dateien und Ordner zu begrenzen und einen hohen Ressourcenverbrauch zu vermeiden, können Sie kein Verzeichnis-Glob für die Verzeichnisse der ersten oder der zweiten Ebene definieren, wie z. B: <code>directory=c:/tmp/*</code> oder <code>directory=c:\Logs*</code>. Der Verzeichnispfad muss mindestens zwei Ebenen aufweisen.</p> <p>Sie haben auch die Möglichkeit, einen Pfad zu einem nicht vorhandenen Verzeichnis zu definieren. Der Agent erfasst dann die Protokolldateien in diesem Verzeichnis, wenn das Verzeichnis und die Dateien erstellt wurden. Sie können dasselbe Verzeichnis unter einem oder mehreren verschiedenen Konfigurationsbereichen definieren, um Protokolle von derselben Datei mehrmals zu erfassen. Damit können verschiedene Tags und Filter auf dieselbe Quelle von Ereignissen angewendet werden.</p> <p>Hinweis Wenn Sie identische Konfigurationen für diese Bereiche verwenden, werden duplizierte Ereignisse auf der Serverseite beobachtet.</p>

Parameter	Beschreibung
<code>include= file_name; ...</code>	<p>(Optional) Der Name eines Dateinamens oder einer Dateimaske (Glob-Muster) für die Datenerfassung. Sie können Werte als eine durch Semikolon getrennte Liste eingeben. Der Standardwert lautet <code>*</code>. Das heißt, dass alle Dateien berücksichtigt werden. Bei diesem Parameter muss die Groß-/Kleinschreibung beachtet werden.</p> <p>Mit einer Dateimaske (Glob-Muster) können Dateien gruppiert werden, die der gleichen Benennungskonvention folgen. Dies ist auch auf der Ebene einzelner Zeichen in Dateinamen möglich. Beispielsweise können Dateinamen mit Leerzeichen wie z. B. <code>vRealize Ops Datei Analytics.log</code> und <code>vRealize Ops Collector.log</code> mit <code>vRealize?Ops?Analytics*.log</code> oder <code>vRealize*.log</code> angegeben werden. Dateimasken bieten die Möglichkeit, Dateinamen festzulegen, die für die Agentenkonfiguration auf Linux- und Windows-Hosts zulässig sind. <code>.zip</code>- und <code>.gz</code>-Dateien werden standardmäßig nicht erfasst.</p> <p>Wichtig Verwenden Sie bei der Erfassung einer rotierten Protokolldatei die Parameter <code>include</code> und <code>exclude</code>, um ein Glob-Muster anzugeben, das der primären und der rotierten Datei entspricht. Wenn das Glob-Muster nur der primären Protokolldatei entspricht, verpassen die vRealize Log Insight-Agenten möglicherweise Ereignisse während der Rotation. Die vRealize Log Insight-Agenten legen die richtige Reihenfolge der rotierten Dateien automatisch fest und senden Ereignisse in der richtigen Reihenfolge an den vRealize Log Insight-Server. Wenn es sich beispielsweise bei Ihrer primären Protokolldatei um <code>myapp.log</code> und bei den rotierten Protokollen um <code>myapp.log.1</code>, <code>myapp.log.2</code> usw. handelt, können Sie das folgende <code>include</code>-Muster verwenden:</p> <pre>include= myapp.log;myapp.log.*</pre>
<code>exclude= regular_expression</code>	<p>(Optional) Ein Dateiname oder eine Dateimaske (Glob-Muster) zum Ausschließen aus der Erfassung. Sie können Werte als eine durch Semikolon getrennte Liste eingeben. Der Standardwert ist leer. Das heißt, dass keine Datei ausgeschlossen wird.</p>
<code>event_marker= regular_expression</code>	<p>(Optional) Ein regulärer Ausdruck, der den Start eines Ereignisses in der Protokolldatei angibt. Wird dieser Ausdruck weggelassen, wird standardmäßig ein Zeilenumbruch ausgeführt. Die von Ihnen eingegebenen Ausdrücke müssen die Perl-Syntax für reguläre Ausdrücke verwenden.</p> <p>Hinweis Symbole, wie zum Beispiel Anführungszeichen (<code>"</code> <code>"</code>), werden nicht als Wrapper für reguläre Ausdrücke behandelt. Sie werden als Teil des Musters behandelt.</p> <p>Da der vRealize Log Insight Agent für die Erfassung in Echtzeit optimiert ist, können mit einer internen Verzögerung geschriebene Protokoll-Teilnachrichten in mehrere Ereignisse aufgeteilt werden. Wenn das Anhängen an Protokolldateien für mehr als 200 ms ohne einen beobachteten <code>event_marker</code> angehalten wird, wird das Teilereignis als abgeschlossen, analysiert und zugestellt betrachtet. Diese Timing-Logik ist nicht konfigurierbar und hat Vorrang vor der <code>event_marker</code>-Einstellung. Protokolldatei-Appender sollten alle vollständigen Ereignisse löschen.</p>
<code>enabled=yes no</code>	<p>(Optional) Ein Parameter zum Aktivieren oder Deaktivieren des Konfigurationsabschnitts. Die möglichen Werte lauten <code>yes</code> oder <code>no</code>. Der Standardwert lautet <code>yes</code>.</p>

Parameter	Beschreibung
charset=char-encoding-type	<p>(Optional) Die Zeichenkodierung der Protokolldateien, die der Agent überwacht. Mögliche Werte:</p> <ul style="list-style-type: none"> ■ UTF-8 ■ UTF-16LE ■ UTF-16BE <p>Der Standardwert lautet UTF-8.</p>
tags={"tag-name": "tag-value", ...}	<p>(Optional) Ein Parameter zum Hinzufügen von benutzerdefinierten Tags zu den Feldern der erfassten Ereignisse. Definieren Sie Tags mit der JSON-Notation. Tagnamen können Buchstaben, Zahlen und Unterstriche enthalten. Ein Tagname darf nur mit einem Buchstaben oder einem Unterstrich beginnen und darf nicht mehr als 64 Zeichen enthalten. Bei Tagnamen wird die Groß- und Kleinschreibung nicht berücksichtigt. Wenn Sie zum Beispiel tags={"tag_name1": "tag value 1", "Tag_Name1": "tag value 2"} verwenden, wird Tag_Name1 als Duplikat ignoriert. Sie können „event_type“ und „timestamp“ nicht als Tagnamen verwenden. Alle Duplikate innerhalb derselben Deklaration werden ignoriert.</p> <p>Wenn das Ziel ein Syslog-Server ist, können Tags das Feld „APP-NAME“ überschreiben. Beispiel: tags={"appname": "VROPS"}.</p>
exclude_fields	<p>(Optional) Ein Parameter zum Ausschließen einzelner Felder aus der Erfassung. Sie können mehrere Werte in Form einer durch Semikolons oder Kommas getrennten Liste eingeben. Beispiel:</p> <ul style="list-style-type: none"> ■ exclude_fields=hostname; filepath ■ exclude_fields=type; size ■ exclude_fields=type, size
raw_syslog=Yes No	<p>Für Agenten, die das Syslog-Protokoll verwenden, ermöglicht diese Option dem Agenten, nicht formatierte Syslog-Ereignisse zu erfassen und zu senden. Die Standardeinstellung ist „Nein“, d. h., die erfassten Ereignisse werden mit benutzerspezifischen Syslog-Attributen umgewandelt. Aktivieren Sie diese Option, um Ereignisse ohne Syslog-Umwandlungen zu erfassen.</p>

3 Speichern und schließen Sie die Datei liagent.ini.

Beispiel: Konfigurationen

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

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

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

Filtern von Ereignissen

Sie können alle gesammelten Ereignisse auf dem vRealize Log Insight-Linux-Agenten basierend auf den jeweiligen Feldwerten filtern, um anzugeben, welche Protokollereignisse ausgewählt oder verworfen werden sollen. Sie können die Collector-Optionen `whitelist` und `blacklist` zum Definieren von Filtern verwenden.

Tipp Standardmäßig erfasst der Linux-Agent für vRealize Log Insight ausgeblendete Dateien, die von Dateien oder Editoren erstellt werden. Die Namen der ausgeblendeten Dateien beginnen mit einem Punkt. Sie können verhindern, dass der Linux-Agent für vRealize Log Insight ausgeblendete Dateien erfasst, indem Sie den Ausschlussparameter `exclude=.*` hinzufügen.

Für jedes Ereignis wertet der Collector die Filterausdrücke `whitelist` und `blacklist` aus. Wenn der `whitelist`-Ausdruck „true“ ergibt und der `blacklist`-Ausdruck „false“ ergibt oder nicht ausgewertet werden kann, wird das Ereignis zur weiteren Verarbeitung in die Warteschlange verschoben. In jedem anderen Fall löscht der Collector das Ereignis. Der Standardwert des `whitelist`-Ausdrucks ist „true“ und der Standardwert des `blacklist`-Ausdrucks ist „false“.

Tipp Der `Filelog`-Collector bietet weniger Felder zum Filtern. Um Felder zum Filtern zu erhalten, können Sie die Protokolle analysieren. Weitere Informationen finden Sie unter [Analysieren von Protokollen](#).

Ein `whitelist`- oder `blacklist`-Filter ist eine Gruppe von Variablen, Literalen und Operatoren, die einen einzelnen logischen oder Ganzzahlwert ergibt. Sie verwenden die Ereignisfelder als Variablen und doppelte Zeichenfolgen und Ziffern als Literale. Informationen zu den Operatoren, die Sie in einem Filterausdruck verwenden können, finden Sie unter [Ereignisfelder und Operanden](#).

Hinweis

- Wenn Sie eine Zahl mit einer Zeichenfolge vergleichen oder wenn der Vergleich numerische Zeichenfolgen beinhaltet, wird jede Zeichenfolge in eine Zahl konvertiert und der Vergleich wird numerisch durchgeführt. Beispiel:
 - Der Ausdruck `whitelist = 123.0 == "000123"` ergibt „true“.
 - Der Ausdruck `whitelist = "00987" == "987.00"` ergibt „true“.
 - Im Ausdruck `whitelist = response_size >= "12.12"` wird der Ausdruck numerisch ausgewertet, wenn das Feld `response_size` einen numerischen Wert aufweist. Wenn die Antwortgröße größer als 12,12 ist, lautet der Ausdruck „true“. Andernfalls ist er „false“.
 - Im Ausdruck `whitelist = "09123" < "234"` werden beide Zeichenfolgenliterals in numerische Werte konvertiert und der Ausdruck wird als „false“ ausgewertet.
- Wenn einer der Zeichenfolgenoperanden nicht in numerische Werte konvertiert werden kann, werden beide Operanden in eine Zeichenfolge konvertiert. Es wird ein einfacher lexikografischer Vergleich zwischen Groß- und Kleinschreibung durchgeführt. Beispiel:
 - Der Ausdruck `whitelist = "1234a" == "1234A"` ist ein Zeichenfolgenvergleich, der als „false“ ausgewertet wird.
 - Der Ausdruck `whitelist = 4 < "four"` konvertiert 4 in „4“ und ergibt „true“.
 - Im Ausdruck `whitelist = response_size > "thousand"` wird der Wert des Felds `response_size` in einen Zeichenfolgenwert konvertiert, der den Ausdruck auf „false“ auswertet.
- Wenn ein Filterausdruck einen Ganzzahlwert ergibt, wird er als „false“ behandelt, wenn er 0 ist, andernfalls als „true“.

Beispiel: Der Ausdruck `whitelist = some_integer & 1` ergibt „true“, wenn das Feld `some_integer` ein wenig bedeutendes Bit festgelegt hat, andernfalls „false“.

Eine umfassende Liste der Ereignisfelder und Operatoren finden Sie unter [Erfassen von Ereignissen aus einer Protokolldatei](#).

In diesem Beispiel erfassen Sie Apache-Zugriffsprotokolle aus der Datei `/var/log/httpd/access`. Einige Beispielprotokolle aus der Datei sind:

- `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

Voraussetzungen

- Melden Sie sich als **Root**-Benutzer an oder verwenden Sie `sudo`, um Konsolenbefehle auszuführen.
- Melden Sie sich bei dem Linux-System an, auf dem Sie den vRealize Log Insight-Linux-Agenten installiert haben. Öffnen Sie eine Konsole und führen Sie `pgrep liagent` aus, um sicherzustellen, dass der vRealize Log Insight-Linux-Agent installiert ist und ausgeführt wird.

Verfahren

- 1 Definieren Sie einen Parser für die Protokolle, wie im folgenden Ausschnitt dargestellt:

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

Der von Ihnen definierte Parser extrahiert die Felder `remote_host`, `remote_log_name`, `remote_auth_user`, `timestamp`, `request`, `status_code` und `response_size` für jedes Ereignis, das aus der Datei `/var/log/httpd/access` erfasst wurde. Sie können diese Felder verwenden, um Ereignisse zu filtern.

- 2 Öffnen Sie die Datei `/var/lib/loginsight-agent/liagent.ini` in einem beliebigen Texteditor.
- 3 Definieren Sie einen `Filelog`-Abschnitt in der Datei, um Protokolle zu erfassen und zu analysieren, wie im folgenden Ausschnitt dargestellt:

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

- 4 Filtern Sie Ereignisse gemäß Ihren Anforderungen.
 - Um Protokolle zu erfassen, bei denen der HTTP-Status 200 lautet, können Sie eine `whitelist` im Abschnitt `Filelog` definieren, wie im folgenden Ausschnitt dargestellt:

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

Der `whitelist`-Ausdruck wird nur für das erste und zweite Ereignis aus den Beispielprotokollen als „true“ ausgewertet und der Collector wählt diese Ereignisse aus.

Wenn das Feld `status_code` im Ereignis nicht vorhanden ist, da es im Protokoll nicht vorhanden ist oder nicht analysiert wird, kann der `whitelist`-Ausdruck nicht ausgewertet werden, d. h., er ergibt „false“ und Collector löscht das Ereignis.

- Um ein Ereignis zu löschen, an dem Sie nicht interessiert sind, können Sie die Option `blacklist` verwenden. Wenn Sie z. B. nicht an lokalem Datenverkehr interessiert sind, können Sie die lokale IP blockieren, wie im folgenden Ausschnitt dargestellt:

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

Der Collector wählt das zweite, dritte und vierte Ereignis aus den Beispielprotokollen aus.

- Um Ereignisse basierend auf mehr als einem Prädikat zu filtern, können Sie die Operatoren `or` und `and` verwenden. Sie können z. B. Ereignisse, die von einer lokalen IP generiert wurden, oder Ereignisse, die von Testbenutzern generiert wurden, von einem beliebigen nicht benötigten Host löschen, wie im folgenden Ausschnitt dargestellt:

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

Die Verwendung des `or`-Operators wertet den `blacklist`-Ausdruck auf „true“ aus, um ein unerwünschtes Ereignis zu überspringen. Der Ausdruck weist den Collector an, das Ereignis abzulegen, wenn der Wert für den `remote_host`-Feldwert „127.0.0.1“ lautet oder der `remote_auth_user`-Feldwert „test“ lautet.

Der Collector wählt das zweite und das dritte Ereignis aus den Beispielprotokollen aus.

- Um von Testbenutzern generierte Ereignisse aus einer lokalen IP-Adresse zu löschen, können Sie `and` im `blacklist`-Ausdruck verwenden, wie im folgenden Ausschnitt dargestellt:

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

Der Collector löscht das fünfte Ereignis aus den Beispielprotokollen.

- Sie können die Filter `whitelist` und `blacklist` zusammen verwenden. Wenn Sie beispielsweise Ereignisse benötigen, bei denen die Antwortgröße größer als 1024 Byte ist, Sie aber keine Ereignisse benötigen, die von einem lokalen Host stammen, können Sie den folgenden Ausschnitt verwenden:

```
[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"
```

Der Collector wählt das zweite Ereignis aus den Beispielprotokollen aus.

Erfassen von Ereignissen von `journald`

Ab Version vRealize Log Insight 4.6 können Agents jetzt Protokolle des `journald`-Systemdienstes für Protokolldaten in Linux-Distributionen lesen, die `systemd` ausführen. `journald` ist jetzt der gängige Standard für die Protokollierung auf `systemd`-basierten Linux-Plattformen. Der Konfigurationsabschnitt `journald` unterstützt die folgenden Optionen:

`journal_files`

Die zu überwachenden Journaldateien. Die folgenden Werte werden unterstützt:

Wert	Beschreibung
all	Öffnen und überwachen Sie alle verfügbaren Journaldateien.
local	Überwachen und lesen Sie nur die auf der lokalen Maschine generierten Journaldateien.
runtime	Überwachen und lesen Sie nur die flüchtigen Journaldateien, ohne die Dateien im persistenten Speicher.
system	Überwachen und lesen Sie nur die Systemdienste und Kernel-Journaldateien.
user	Überwachen und lesen Sie nur die Journaldateien des aktuellen Benutzers.

`fetch_fields`

Die Felder, die mit der Meldung aus den Journalprotokolleinträgen abgerufen werden. Der Wert für diese Option ist eine Liste der durch Kommas getrennten Feldnamen mit Groß- und Kleinschreibung. Die folgenden Werte werden unterstützt:

Wert	Beschreibung
pri_severity,pri_facility,syslog_identifizier	Standardwert für diese Option.
*	Alle Felder abrufen.
all	Keine Felder abrufen.

Filtern von Ereignissen von vRealize Log Insight-Agenten

Sie können mit der Filteroption im Abschnitt `[server|<dest_id>]` Ihrer lokalen `liagent.ini`-Datei die Informationen angeben, die ein Agent zu einem Ziel sendet.

Für die Option gilt folgendes Format:

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

Filtertyp	Beschreibung
Collectortyp	Eine Liste mit durch Kommas getrennten Einträgen, die die Collectortypen definiert. Unterstützte Werte sind <code>filelog</code> oder <code>winlog</code> . Wenn kein Wert angegeben ist, werden alle Collectortypen verwendet.
Collectorfilter	Legt den Namen eines Collectorabschnitts in einem Regex-Format fest. Beispiel: <code>vcops_.*</code> bezieht sich auf alle Collectorabschnitte, die mit „vcops_“ beginnen.
Ereignisfilter	Für Ereignisfeldfilter gilt die gleiche Syntax wie für eine Positivliste oder eine Negativliste in Collectorabschnitten. Ein Agent sendet nur Ereignisse, die den Ausdruck auf „Wahr“ oder auf einen Wert ungleich null auswerten. Ein leerer Ereignisfilter wird immer auf „Wahr“ ausgewertet. Um Ereignisfilter auf Ereignisse anwenden zu können, müssen Sie in den entsprechenden Collectorabschnitten einen Parser für die Feldextrahierung definieren. Wenn ein Ausdruck aufgrund fehlender Felder im erfassten Ereignis nicht ausgewertet werden kann, wird das Ereignis verworfen.

Wenn Sie mehrere Filterausdrücke angeben möchten, müssen Sie diese durch ein Komma wie im folgenden Beispiel trennen:

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

Wenn eine Nachricht mehr als einer Gruppe von Filterkriterien für ein Ziel entspricht, wird sie nur einmal gesendet.

Tabelle 4-1. Syntaxbeispiele

Filter	Bedeutung
<code>filter= {winlog;Microsoft.*;}</code>	Sendet Ereignisse aus winlog-Collectoren nur dann, wenn der Name des Ereignisses mit „Microsoft“ beginnt.
<code>filter= {winlog;Microsoft.*; eventid == 1023}</code>	Sendet Ereignisse aus winlog-Collectoren nur dann, wenn der Name des Ereignisses mit „Microsoft“ beginnt und die Ereignis-ID 1023 lautet.
<code>filter= {.*;}</code>	Standardfilterwert. Sendet alle Ereignisse aus allen Quellen.
<code>filter= {winlog;.*;}</code>	Sendet alle Ereignisse aus winlog-Abschnitten.

Tabelle 4-1. Syntaxbeispiele (Fortsetzung)

Filter	Bedeutung
filter= {filelog;syslog;facility<5}	Sendet Ereignisse aus dem Abschnitt [filelog syslog], wenn „facility“ kleiner als 5 ist. Die Abschnitte [filelog syslog] müssen über einen Parser verfügen, der das Feld „facility“ extrahiert. Andernfalls werden alle Ereignisse übersprungen.
filter= {;;}	Entspricht keinem Ereignis. Verwenden Sie diese Syntax, um die Ereignisweiterleitung zu deaktivieren.

Im folgenden Beispiel wird ein Filter für die Konfiguration des zweiten Ziels des vorherigen Beispiels hinzugefügt.

```
; 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; }
```

Das nächste Beispiel verwendet einen komplexeren Filterausdruck.

```
; 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"}
```

Wenn Sie mehrere Filterausdrücke angeben möchten, müssen Sie diese durch ein Komma wie im folgenden Beispiel trennen.

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

Weiterleiten von Informationen von einem vRealize Log Insight-Agenten

Sie können von einem Agenten erfasste Ereignisse an bis zu drei Ziele weiterleiten. Zu diesen Zielen können auch vRealize Log Insight-Server oder eine vRealize Log Insight-Weiterleitung bzw. Drittanbieterlösungen zur Protokollverwaltung gehören.

Beispielsweise können Sie Überwachungs- oder Systemprotokolle an einen Server für Ihr Sicherheitsteam, Anwendungsprotokolle an einen DevOps-Teamserver und Metrikprotokolle an ein IT-Management-System senden. Verwenden Sie Filter, um festzulegen, welche Informationen zu einem Ziel gesendet werden sollen. Sie können Informationen von einem einzelnen vRealize Log Insight-Agenten auf bis zu drei Ziele weiterleiten.

Die Agent-Konfiguration wird im Abschnitt `[server|<dest_id>]` Ihrer lokalen `liagent.ini`-Datei durchgeführt. Verwenden Sie für vRealize Log Insight-Server oder für die vRealize Log Insight-Weiterleitung das `cfapi`-Protokoll und für alle anderen Ziele `Syslog`.

Wenn Sie mehrere Ziele für einen Agenten angeben, verwendet das erste Ziel den standardmäßigen `loginsight`-Speicherort. Für weitere Ziele müssen Sie Informationen zum Speicherort angeben.

Das nächste Beispiel zeigt einen Abschnitt einer `liagent.ini`-Datei, in dem zwei Ziele festgelegt werden. Der Standardservername `loginsight` gilt standardmäßig für das erste Ziel und wird nicht festgelegt. Der zweite Abschnitt `[server|<dest_id>]` gibt ein Ziel an.

```
; 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
```

Informationen zum Erstellen von Filtern für Agenten finden Sie unter [Filtern von Ereignissen von vRealize Log Insight-Agenten](#).

Festlegen des vRealize Log Insight-Zielservers

Sie können den vRealize Log Insight-Zielserver für einen vRealize Log Insight-Agenten festlegen oder ändern, der unter Windows ausgeführt wird. Sie haben die Möglichkeit, Ereignisse an bis zu drei Ziele zu senden und die Ausgabe nach Zielen zu filtern.

Das Standardziel kann im Abschnitt `[server]` der Datei `liagent.ini` konfiguriert werden. Das Standardziel ist immer vorhanden, und der Hostname ist standardmäßig auf `loginsight` festgelegt. Um weitere Ziele hinzuzufügen, erstellen Sie einen Abschnitt `[server|<dest_id>]` für jedes Ziel. Sie müssen einen eindeutigen Hostnamen als Ziel-ID für jede zusätzliche Verbindung angeben. Sie können die gleichen Optionen für zusätzliche Ziele wie für den Abschnitt für den standardmäßigen `[server]`-Bereich verwenden. Konfigurieren Sie keine zusätzlichen Ziele für das automatische Upgrade bzw. verwenden Sie diese nicht für die Agent-Konfiguration. Sie können zwei zusätzliche Ziele angeben.

Standardmäßig sendet der Agent alle erfassten Ereignisse an alle Ziele. Sie können Ereignisse filtern, um mit der `file`-Option andere Ereignisse an unterschiedliche Ziele zu senden. Weitere Informationen finden Sie unter [Filtern von Ereignissen von vRealize Log Insight-Agenten](#).

Voraussetzungen

- Melden Sie sich bei dem Windows-Computer an, auf dem Sie den vRealize Log Insight Windows-Agent installiert haben, und starten Sie den Dienst-Manager, um zu überprüfen, ob der vRealize Log Insight-Agent-Dienst installiert ist.
- Wenn Sie über einen vRealize Log Insight-Cluster mit integriertem Lastausgleich verfügen, finden Sie unter [Aktivieren des integrierten Lastausgleichsdiensts](#) Informationen zu den spezifischen Anforderungen für benutzerdefinierte SSL-Zertifikate.

Verfahren

- 1 Navigieren Sie zum Programmdatenordner des Windows-Agenten für vRealize Log Insight.
`%ProgramData%\VMware\Log Insight Agent`
- 2 Öffnen Sie die Datei `liagent.ini` in einem beliebigen Texteditor.
- 3 Ändern Sie die folgenden Parameter und legen Sie die Werte für Ihre Umgebung fest.

Parameter	Beschreibung
proto	Protokoll, mit dem der Agent Ereignisse an den vRealize Log Insight-Server sendet. Die möglichen Werte lauten <code>cfapi</code> und <code>syslog</code> . Der Standardwert ist <code>cfapi</code> .
hostname	IP-Adresse oder Hostname der virtuellen vRealize Log Insight-Appliance. Sie können eine IPv4- oder eine IPv6-Adresse angeben. Eine IPv6-Adresse kann mit oder ohne eckige Klammern angegeben werden. Beispiel: <pre>hostname = 2001:cdba::3257:9652 or hostname = [2001:cdba::3257:9652]</pre> <p>Wenn der Host sowohl IPv4- als auch IPv6-Stacks unterstützt und ein Domänenname als Hostname angegeben ist, wählt der Agent den IP-Stack auf der Basis der IP-Adresse, die vom Namensauflöser zurückgegeben wird. Wenn der Auflöser sowohl IPv4- als auch IPv6-Adressen zurückgibt, versucht der Agent, sequenziell in der angegebenen Reihenfolge eine Verbindung zu beiden Adressen herzustellen.</p>
max_disk_buffer	Der maximale Festplattenspeicher in MB, den der Log Insight-Windows-Agent zum Puffern von Ereignissen verwenden kann, die für diesen speziellen Server erfasst werden. Die Option hat Vorrang vor dem Wert für <code>[storage].max_disk_buffer</code> für diesen Server. Der Standardwert beträgt 150 MB. Sie können für die Puffergröße einen Wert von 50 bis 8000 MB festlegen.

Parameter	Beschreibung
port	<p>Kommunikationsport, den der Agent zum Senden von Ereignissen an den vRealize Log Insight-Server oder an Server von Drittanbietern verwendet. Standardmäßig verwendet der Agent den richtigen Port basierend auf den Optionen, die für SSL und das Protokoll festgelegt sind. Die Standardwerte für den Port finden Sie in der unten angegebenen Liste. Sie müssen die Port-Option nur angeben, wenn sie sich von diesen Standardeinstellungen unterscheidet.</p> <ul style="list-style-type: none"> ■ cfapi mit aktiviertem SSL: 9543 ■ cfapi mit deaktiviertem SSL: 9000 ■ Syslog mit aktiviertem SSL: 6514 ■ Syslog mit deaktiviertem SSL: 514
ssl	<p>Aktiviert oder deaktiviert SSL. Der Standardwert lautet „ja“.</p> <p>Wenn für <code>ssl</code> der Wert „Ja“ ausgewählt ist, wird der Port auf 9543 festgelegt, solange Sie nichts anderes angeben.</p>
reconnect	<p>Die Zeit in Minuten, in der die erneute Verbindung zum Server erzwungen wird. Der Standardwert lautet 30.</p>
filter	<p>Gibt die Informationen an, die ein Agent an ein Ziel sendet. Diese Option erfordert drei Argumente:</p> <pre>{Collectortyp; Collectorfilter; Ereignisfilter}</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 Speichern und schließen Sie die Datei `liagent.ini`.

Beispiel

Im folgenden Konfigurationsbeispiel wird ein vRealize Log Insight-Zielserver festgelegt, der eine vertrauenswürdige Zertifizierungsstelle verwendet.

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

Das folgende Beispiel zeigt eine Konfiguration für mehrere Ziele, die das Filtern von Meldungen nach Zielen beinhaltet.

```
; 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
```

Nächste Schritte

Sie können weitere SSL-Optionen für den vRealize Log Insight-Agenten festlegen. Weitere Informationen finden Sie unter [Konfigurieren der SSL-Verbindung zwischen dem Server und den Log Insight-Agenten](#)

Festlegen eines Agent-Ziels

Sie können zum Senden von Ereignissen an den vRealize Log Insight-Linux-Agent bis zu drei Ziele angeben.

Mehrere Zielverbindungen werden über den Abschnitt `[server|<dest_id>]` der Datei `li-agent.ini` definiert, wobei `<dest_id>` eine eindeutige Verbindungs-ID pro Konfiguration darstellt. Sie können die gleichen Optionen für zusätzliche Ziele wie für den Abschnitt für den standardmäßigen `[server]`-Bereich verwenden. Konfigurieren Sie aber keine zusätzlichen Ziele für ein automatisches Upgrade bzw. verwenden Sie diese nicht für die Agent-Konfiguration. Sie können zwei zusätzliche Ziele angeben.

Für das erste definierte Ziel können Sie den standardmäßigen Serverwert `loginsight` verwenden. Wenn Sie zusätzliche Ziele definieren, müssen Sie in den Abschnitten `[server]` einen Hostnamen für nachfolgende Ziele angeben. Ohne Filterung sendet der Agent alle erfassten Ereignisse an alle Ziele. Dies ist die Standardeinstellung. Sie können allerdings Ereignisse filtern, um unterschiedliche Ereignisse an unterschiedliche Ziele zu senden.

Voraussetzungen

- Melden Sie sich als **Root**-Benutzer an oder verwenden Sie `sudo`, um Konsolenbefehle auszuführen.
- Melden Sie sich bei dem Linux-System an, auf dem Sie den vRealize Log Insight-Linux-Agenten installiert haben. Öffnen Sie eine Konsole und führen Sie `pgrep liagent` aus, um sicherzustellen, dass der vRealize Log Insight-Linux-Agent installiert ist und ausgeführt wird.
- Wenn Sie über einen vRealize Log Insight-Cluster mit integriertem Lastausgleich verfügen, finden Sie unter [Aktivieren des integrierten Lastausgleichsdiensts](#) Informationen zu den spezifischen Anforderungen für benutzerdefinierte SSL-Zertifikate.

Verfahren

- 1 Öffnen Sie die Datei `/var/lib/loginsight-agent/liagent.ini` in einem beliebigen Texteditor.

2 Ändern Sie die folgenden Parameter und legen Sie die Werte für Ihre Umgebung fest.

Parameter	Beschreibung
proto	<p>Protokoll, mit dem der Agent Ereignisse an den vRealize Log Insight-Server sendet. Die möglichen Werte lauten <code>cfapi</code> und <code>syslog</code>.</p> <p>Der Standardwert ist <code>cfapi</code>.</p>
hostname	<p>IP-Adresse oder Hostname der virtuellen vRealize Log Insight-Appliance. Sie können eine IPv4- oder eine IPv6-Adresse angeben. Eine IPv6-Adresse kann mit oder ohne eckige Klammern angegeben werden. Beispiel:</p> <pre>hostname = 2001:cdba::3257:9652 or hostname = [2001:cdba::3257:9652]</pre> <p>Wenn der Host sowohl IPv4- als auch IPv6-Stacks unterstützt und ein Domänenname als Hostname angegeben ist, verwendet der Agent den IP-Stack abhängig von der IP-Adresse, die vom Namensauflöser zurückgegeben wird. Wenn der Auflöser sowohl IPv4- als auch IPv6-Adressen zurückgibt, versucht der Agent, sequenziell in der angegebenen Reihenfolge eine Verbindung zu beiden Adressen herzustellen.</p>
max_disk_buffer	<p>Der maximale Festplattenspeicher in MB, den der Log Insight-Linux-Agent zum Puffern von Ereignissen verwenden kann, die für diesen speziellen Server erfasst werden. Die Option hat Vorrang vor dem Wert für <code>[storage].max_disk_buffer</code> für diesen Server.</p> <p>Der Standardwert beträgt 150 MB. Sie können für die Puffergröße einen Wert von 50 bis 8000 MB festlegen.</p>
port	<p>Kommunikationsport, den der Agent zum Senden von Ereignissen an den vRealize Log Insight-Server oder an Server von Drittanbietern verwendet. Standardmäßig verwendet der Agent den richtigen Port basierend auf den Optionen, die für SSL und das Protokoll festgelegt sind. Die Standardwerte für den Port finden Sie in der unten angegebenen Liste. Sie müssen die Port-Option nur angeben, wenn sie sich von diesen Standardeinstellungen unterscheidet.</p> <ul style="list-style-type: none"> ■ cfapi mit aktiviertem SSL: 9543 ■ cfapi mit deaktiviertem SSL: 9000 ■ Syslog mit aktiviertem SSL: 6514 ■ Syslog mit deaktiviertem SSL: 514
ssl	<p>Aktiviert oder deaktiviert SSL. Der Standardwert lautet „ja“.</p> <p>Wenn <code>ssl</code> aktiviert ist und Sie keinen Wert für den Port festlegen, wird der Port automatisch auf „9543“ gesetzt.</p>
reconnect	<p>Das Zeitintervall in Minuten, in dem der erneute Verbindungsaufbau zum Server erzwungen wird. Der Standardwert lautet 30.</p>

```
[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 Speichern und schließen Sie die Datei `liagent.ini`.

Beispiel

Im folgenden Konfigurationsbeispiel wird ein vRealize Log Insight-Zielserver festgelegt, der eine vertrauenswürdige Zertifizierungsstelle verwendet.

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

Das folgende Beispiel zeigt eine Konfiguration mit mehreren Zielen.

- Das erste (Standard) Ziel empfängt alle erfassten Ereignisse.

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

- Das zweite Ziel empfängt nur Syslog-Ereignisse über das einfache Syslog-Protokoll.

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

- Das dritte Ziel empfängt vRealize Operations Manager-Ereignisse aus, wenn die Ebene für das Feld „Fehler“ oder „Warnung“ lautet, und sie werden durch Abschnitte, deren Name mit „Vrops-“ beginnt, erfasst.

```
[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

```

Nächste Schritte

Sie können weitere SSL-Optionen für den Linux-Agenten für vRealize Log Insight festlegen. Weitere Informationen finden Sie unter [Konfigurieren der SSL-Verbindung zwischen dem Server und den Log Insight-Agenten](#)

Zentrale Konfiguration von vRealize Log Insight-Agenten

Sie können mehrere vRealize Log Insight-Agenten konfigurieren.

Jeder vRealize Log Insight-Agent weist eine lokale Konfiguration und eine serverseitige Konfiguration auf. Die lokale Konfiguration wird in der Datei `liagent.ini` auf der virtuellen oder physischen Maschine gespeichert, auf der der vRealize Log Insight-Agent installiert ist. Die serverseitige Konfiguration ist zugänglich und editierbar, z. B. in der Web-Benutzeroberfläche über **Verwaltung > Agenten**. Die Konfiguration jedes vRealize Log Insight-Agenten setzt sich aus Abschnitten und Schlüsseln zusammen. Schlüssel haben konfigurierbare Werte.

vRealize Log Insight-Agenten fragen den vRealize Log Insight-Server in regelmäßigen Abständen ab und empfangen die serverseitige Konfiguration. Die serverseitige und die lokale Konfiguration werden zusammengeführt und das Ergebnis stellt eine effektive Konfiguration dar. Jeder vRealize Log Insight-Agent verwendet die effektive Konfiguration als Betriebskonfiguration. Konfigurationen werden abschnittsweise und Schlüssel für Schlüssel zusammengeführt. Die Werte in der serverseitigen Konfiguration überschreiben die Werte in der lokalen Konfiguration. Die Zusammenführungsregeln lauten wie folgt:

- Wenn ein Abschnitt nur in der lokalen oder nur in der serverseitigen Konfiguration vorhanden ist, werden dieser Abschnitt und der entsprechende Inhalt Teil der effektiven Konfiguration.

- Wenn ein Abschnitt sowohl in der lokalen als auch in der serverseitigen Konfiguration vorhanden ist, werden die Schlüssel im Abschnitt gemäß der folgenden Regeln zusammengeführt:
 - Wenn ein Schlüssel nur in der lokalen oder nur in der serverseitigen Konfiguration vorhanden ist, werden der Schlüssel und die entsprechenden Werte Teil dieses Abschnitts in der effektiven Konfiguration.
 - Wenn ein Schlüssel sowohl in der lokalen als auch in der serverseitigen Konfiguration vorhanden ist, wird der Schlüssel Teil dieses Abschnitts in der effektiven Konfiguration und der Wert in der serverseitigen Konfiguration wird verwendet.

Ein vRealize Log Insight-Administrator kann eine zentrale Konfiguration auf alle vRealize Log Insight-Agenten anwenden. Sie können z. B. zur Seite **Verwaltung** navigieren und im Abschnitt **Management** auf **Agenten** klicken. Geben Sie die Konfigurationseinstellungen im Feld **Agent-Konfiguration** ein und klicken Sie auf **Konfiguration für alle Agenten speichern**. Die Konfiguration wird beim nächsten Abrufzyklus auf alle konfigurierbaren aktiven Agents angewendet.

Ein vRealize Log Insight-Benutzer mit Administratorrechten kann auch bestimmte Filter in Agentengruppen verwenden, z. B. nach Betriebssystem, Agentenversion, Hostname oder IP-Bereichen, und die Konfiguration auf bestimmte vRealize Log Insight-Agenten anwenden. Informationen zu Agentengruppen finden Sie unter *Arbeiten mit Agentengruppen*.

Hinweis

- Die zentrale Konfiguration kann nur auf vRealize Log Insight-Agenten angewendet werden, die das cfapi-Protokoll verwenden.
 - Ein vRealize Log Insight-Agent kann in keinem der folgenden Szenarien konfiguriert werden:
 - Der aktuelle vRealize Log Insight-Server ist kein primäres Ziel. Weitere Informationen über das Konfigurieren von mehreren Zielen finden Sie unter [Festlegen eines Agent-Ziels](#).
 - Der Parameter `central_config = no` wird in der Agentenkonfiguration verwendet. Informationen zur standardmäßigen Agentenkonfiguration für Windows finden Sie unter [Standardkonfiguration des Log Insight Windows Agent](#).
-

Exemplarische Konfigurationszusammenführung

Exemplarische Zusammenführung einer lokalen und einer serverseitigen Log Insight Windows Agent-Konfiguration.

Lokale Konfiguration

Sie können die folgende lokale Log Insight Windows Agent-Konfiguration haben.

```
[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} - -
```

Serverseitige Konfiguration

Sie können die Seite **Verwaltung > Agents** der Web-Benutzeroberfläche zur Anwendung einer zentralen Konfiguration für alle Agents verwenden. Beispiel: Sie können Erfassungskanäle ausschließen und hinzufügen und die Standard-Neuverbindungseinstellung ändern.

```
[server]
reconnect=20

[winlog|Security]
channel=Security
enabled=no

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

Effektive Konfiguration

Die effektive Konfiguration ist das Ergebnis der Zusammenführung der lokalen mit der serverseitigen Konfiguration. Der Log Insight Windows Agent ist auf Folgendes konfiguriert:

- Neuverbindung mit dem vRealize Log Insight-Server alle 20 Minuten
- mit der Erfassung von Anwendungs- und Systemereigniskanälen fortfahren
- die Erfassung des Sicherheitsereigniskanals stoppen
- mit der Erfassung des Microsoft-Windows-DeviceSetupManager-Ereigniskanals/operativen Ereigniskanals beginnen
- die Erfassung von ApacheAccessLogs fortsetzen

```
[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} - -

```

Verwenden von allgemeinen Werten für die Konfiguration von Agenten

Sie können die Standardwerte der Agent-Konfigurationsdatei mit allgemeinen Parameterwerten, die in jedem Agent-Konfigurationsabschnitt für Windows- und Linux-Agenten angewendet werden, überschreiben.

Allgemeine Optionen

Optionen, die im Abschnitt `[common|global]` der Konfigurationsdatei `liagent.ini` angegeben sind, werden an alle Abschnitte weitergegeben. Optionen, die im Abschnitt `[common|filelog]` angegeben sind, werden an alle Filelog-Abschnitte weitergegeben und `[common|winlog]`-Optionen werden an alle Winlog-Abschnitte weitergegeben.

Sie können die Parameter `tags`, `include`, `exclude`, `event_marker`, `charset`, `exclude_fields` und `parser` in allgemeinen Abschnitten, wie in folgendem Beispiel gezeigt, definieren: Das Beispiel bezieht sich auf einen Windows-Agenten:

```

[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

```

...

In diesem Beispiel wird folgendes Verhalten aufgezeigt:

- Alle Protokolle der Filelog-Abschnitte enthalten die Tags `log_source_vm` und `collector_type` mit ihren entsprechenden Werten.
- Die Tags `test_tag` und `some_other_tag` sind aus allen gesendeten Protokollen ausgeschlossen.
- Der Parser `auto` wird auf alle erfassten Protokolle angewendet.
- Standardmäßig sind bei allen Filelog-Collectors *.trc-Dateien aus der Überwachung ausgeschlossen.

Optionen im Abschnitt `[common|global]` werden ebenfalls auf alle Winlog-Abschnitte angewendet.

Zusammenführen und Überschreiben von Kriterien

Wenn Optionen in mehr als einem Abschnitt definiert sind, werden ihre Werte zusammengeführt oder überschrieben und der Abschnitt mit einem geringeren Umfang erhält beim Zusammenführen/Überschreiben eine höhere Priorität. So wird ein Wert aus `[common|global]` mit einem Wert aus `[common|filelog]` zusammengeführt oder von diesem überschrieben, dieser wird wiederum mit einem Wert aus `[filelog|sample_section]` kombiniert oder von diesem überschrieben.

Das Zusammenführungs- und Überschreibeverhalten entspricht dabei folgenden Regeln:

- Optionen, deren Werte aus einer Liste von Werten bestehen (Tags „include“, „exclude“ und „exclude_fields“) werden mit Werten dieser Option aus einem Abschnitt mit einer höheren Priorität zusammengeführt. Im Fall von Tags überschreiben die Werte der Tags aus Abschnitten mit einer höheren Priorität, wie zuvor beschrieben, den Wert desselben Tags aus einem Abschnitt mit einer geringeren Priorität.
- Der Wert von Optionen, die einen einzigen Wert haben können („event_marker“, „charset“ und „parser“) wird von Werten dieser Option aus Abschnitten mit einer höheren Priorität überschrieben.

Dies bedeutet, dass der allgemeine Wert „charset= UTF-16LE“ aus `[common|global]` vom Wert „charset=UTF-8“ aus `[filelog|sample_section]` überschrieben wird.

So wird beispielsweise bei `tags={"app":"global-test"} in [common|filelog]` und `tags={"app":"local-test", "section":"flg_test_section"} in [filelog|flg_test_section]` der Wert `[common|filelog]` vom Wert des Tags „app“ aus dem Abschnitt `[filelog|flg_test_section]` überschrieben. Alle Protokolle, die über diesen Filelog-Abschnitt erfasst wurden, besitzen einen zusätzlichen Tag „app“ mit dem Wert „local-test“ und einen Tag „section“ mit dem Wert „flg_test_section“. Die Prioritätsreihenfolge für Winlog-Abschnitte ist identisch: `[winlog|...]`-Abschnitte haben die höchste Priorität und `[common|global]`-Abschnitte haben die geringste Priorität.

Wenn in allgemeinen Abschnitten ungültige Werte angegeben sind, werden diese im Allgemeinen übersprungen und nicht mit Werten vorheriger bzw. entsprechender Filelog-/Winlog-Abschnitte zusammengeführt. Bei ungültigen Werten in Tags oder `exclude_fields`-Optionen extrahiert der Agent so viele gültige Daten wie möglich und überspringt beim Antreffen ungültiger Daten den Rest der Datei. Alle Anomalien werden in der Agent-Protokolldatei berichtet. Konsultieren Sie beim Antreffen unerwarteten Verhaltens die Protokolldatei und korrigieren Sie alle durch den Agenten berichteten Fehler.

Wenn von Agenten ein ungültiger Wert für eine Option in einem Filelog- oder Winlog-Abschnitt erkannt wird, werden die Optionswerte aus diesem Abschnitt vom Agenten nicht mit Optionswerten aus allgemeinen Abschnitten zusammengeführt und dieser Abschnitt nicht aktiviert. Alle Fehler werden in einer Agent-Protokolldatei berichtet. Konsultieren Sie beim Antreffen unerwarteten Verhaltens die Protokolldatei und korrigieren Sie alle durch den Agenten berichteten Fehler.

Analysieren von Protokollen

Agentenseitige Protokoll-Parser extrahieren strukturierte Daten aus unstrukturierten Protokollen, bevor an den vRealize Log Insight-Server geliefert wird. Mithilfe von Protokoll-Parsern kann vRealize Log Insight Protokolle analysieren, Informationen daraus extrahieren und diese Ergebnisse auf dem Server anzeigen. Protokoll-Parser können für vRealize Log Insight-Agenten sowohl für Windows als auch für Linux konfiguriert werden.

Wenn das Syslog-Protokoll verwendet wird, stellen RFC5424 zufolge die von Parsern extrahierten Felder einen Teil von STRUCTURED-DATA dar.

Konfigurieren von Protokoll-Parsern

Sie können Parser für `FileLog`- und für `WinLog`-Collectors konfigurieren.

Voraussetzungen

Für den Linux-Agenten von vRealize Log Insight:

- Melden Sie sich als Root-Benutzer an oder verwenden Sie `sudo`, um Konsolenbefehle auszuführen.
- Melden Sie sich bei dem Linux-System an, auf dem Sie den Log Insight Linux Agent installiert haben. Öffnen Sie eine Konsole und führen Sie `pgrep liagent` aus, um sicherzustellen, dass der Log Insight Linux Agent installiert ist und ausgeführt wird.

Für den Windows-Agenten von vRealize Log Insight:

- Melden Sie sich bei dem Windows-Computer an, auf dem Sie den Log Insight Windows Agent installiert haben, und starten Sie den Dienst-Manager, um zu überprüfen, ob der vRealize Log Insight-Dienst installiert ist.

Verfahren

- 1 Navigieren Sie zu dem Ordner, der die Datei `liagent.ini` enthält.

Betriebssystem	Pfad
Linux	<code>/var/lib/loginsight-agent/</code>
Windows	<code>%ProgramData%\VMware\Log Insight Agent</code>

- 2 Öffnen Sie die Datei `liagent.ini` in einem beliebigen Texteditor.
- 3 Um einen bestimmten Parser zu konfigurieren, definieren Sie einen Parser-Abschnitt.
`[parser|myparser]`

Dabei ist `myparser` ein zufälliger Name des Parsers, der aus Protokollquellen stammen kann. Der Parser-Abschnitt sollte sich auf einen beliebigen integrierten (oder anderweitig definierten) Parser beziehen und die obligatorischen Optionen dieses Parsers sowie bei Bedarf nicht obligatorische Optionen konfigurieren.

So zeigt z. B. `base_parser=csv`, dass der `myparser`-Parser von dem integrierten Parser `csv` abgeleitet ist. Er erwartet, dass Eingabeprotokolle aus zwei durch Semikolon getrennten Feldern bestehen.

```
[parser|myparser]

base_parser=csv

fields=field_name1,field_name2

delimiter=";"
```

- 4 Nach dem Definieren von `myparser` verweisen Sie aus Protokollquellen `winlog` oder `filelog` darauf.

```
[filelog|some_csv_logs]

directory=D:\Logs

include=*.txt;*.txt.*

parser=myparser
```

Die aus `some_csv_logs`-Quellen erfassten Protokolle, z. B. dem Verzeichnis `D:\Logs`, werden von `myparser` analysiert, und die extrahierten Ereignisse werden auf dem Server als `field_name1` bzw. `field_name2` angezeigt.

Hinweis Die statischen Protokolle im Verzeichnis `D:\Logs` werden nicht vom Agenten in vRealize Log Insight gezogen. Neue, im Verzeichnis `D:\Logs` erstellte Dateien sind jedoch in vRealize Log Insight verfügbar.

- 5 Speichern und schließen Sie die Datei `liagent.ini`.

Häufige Optionen für Parser

Sie können häufige Optionen für alle Parser konfigurieren, die benannte Felder generieren.

Für Feldnamen reservierte Wörter

Feldnamen sind eingeschränkt. Die folgenden Namen sind reserviert und können nicht als Feldnamen verwendet werden.

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

Gängige Parser-Optionen

Die in der folgenden Tabelle aufgeführten Optionen können mit allen unterstützten Parseern verwendet werden.

Option	Beschreibung
<code>base_parser</code>	Der Name des Basis-Parsers, den dieser benutzerdefinierte Parser erweitert. Es kann sich um einen integrierten Parser-Namen oder einen anderen benutzerdefinierten Namen handeln. Dieser Konfigurationsschlüssel ist obligatorisch.
<code>field_decoder</code>	Als JSON-Zeichenfolgen angegebene verschachtelte Parser. Schlüssel sind die Namen des Felds, auf das der verschachtelte Parser angewandt wird, und der Wert ist der Name des Parsers, der für das betreffende Feld verwendet wird. Jeder verschachtelte Parser wird auf das entsprechende , vom Basis-Parser decodierte Feld angewandt. Felddecoder sind nützlich, wenn der Wert eines Felds ein komplexer Wert ist, z. B. ein Zeitstempel. Die Option field_decoder unterstützt ebenfalls weitere JSON-Objekte als Argumente, mit denen Sie Bedingungen für bestimmte Feldwerte verwenden können, die vor dem Anwenden der verschachtelten Analyse überprüft werden. Hinweis Weitere Informationen zur Nutzung und zu bedingten Konfigurationen finden Sie unter „Bedingte Konfigurationen“ für den Abschnitt mit der <code>field_decoder</code> -Option weiter unten.
<code>field_rename</code>	Benennt extrahierte Felder um. Verwenden Sie eine JSON-Zeichenfolge, bei der die Schlüssel die Originalnamen der Felder und die Werte die neuen, gewünschten Namen der Felder sind. Die Option <code>field_decoder</code> wird immer vor <code>field_rename</code> angewendet. Die Reihenfolge dieser Optionen in der INI-Datei ist unwichtig. Geben Sie aus Gründen der Deutlichkeit <code>field_decoder</code> zuerst an.

Option	Beschreibung
<code>next_parser</code>	<p>Name des nächsten auszuführenden Parsers. Ermöglicht die Ausführung von mehreren Parsern sequenziell für die gleiche Eingabe.</p> <hr/> <p>Hinweis Parser verarbeiten alle nachfolgenden Parser, die vom <code>next_parser</code>-Schlüsselwort definiert sind und können einen Feldwert ersetzen, der bereits von einem vorherigen Parser extrahiert wurde.</p> <hr/>
<code>exclude_fields</code>	<p>Eine Liste der mit Semikolons getrennten Feldnamen, die aus dem Ereignis entfernt werden sollen, bevor es an den Server geliefert wird. Die Feldnamen werden vor der Ereignisfilterung entfernt, damit das Feld, das Sie beim Analysieren ausgeschlossen haben, nicht in der Filterbedingung verwendet werden kann.</p> <hr/>
<code>debug</code>	<p>Ja- oder Nein-Option, welche Debuggen für einen bestimmten Parser aktiviert. Wenn Debuggen aktiviert ist, führt der Parser detaillierte Protokollierung der erhaltenen Eingabe, des durchgeführten Vorgangs und des erhaltenen Ergebnisses durch. Die Option gilt pro Abschnitt, also nur für den Parser, der für den betreffenden Abschnitt definiert ist.</p> <p>Der Debug-Standardwert für Parser ist <code>debug=no</code>.</p> <hr/>

Bedingte Konfigurationen für die `field_decoder`-Option

Für Protokolle mit dem gleichen allgemeinen Format, aber signifikanten Unterschieden in Bezug auf bestimmte Feldwerte, z. B. Protokolle mit den Schweregraden **info** und **error**, können Sie mit der bedingten verschachtelten Analyse die Anwendung unnötiger Analysen auf die entsprechenden Felder bereits analysierter Protokolle reduzieren.

Verwenden Sie beispielsweise diese Protokolle:

```
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
--
```

Sie können die folgende Konfiguration verwenden, um sie zu analysieren:

```
[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"}
```

Diese Konfiguration erzeugt die folgenden Ergebnisse:

```
timestamp=2019-03-29T11:00:54.858000 generator_host="host-FQDN" log_severity="error"
thread_id="2099230" common_info=Originator@6876 sub=Default opID=1983bdbbe-c1-800f
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.)"
```

Zudem werden die folgenden Felder für das **info**-Protokoll analysiert:

```
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 --"
```

Protokoll-Parser für das kommagetrennte Format (CSV-Parser)

Sie können Parser für das kommagetrennte Format (CSV) sowohl für `FileLog` als auch für `WinLog-Collectors` konfigurieren.

Die verfügbaren Optionen für den `csv`-Parser sind `fields` und `delimiter`.

Optionen für Parser für das kommagetrennte Format (CSV-Parser)

Beachten Sie die folgenden Informationen hinsichtlich der Struktur des `csv`-Parsers.

Option	Beschreibung
<code>fields</code>	<p>Die <code>fields</code>-Option gibt die Namen der Felder an, die im Protokoll vorhanden sind. Die Gesamtzahl der aufgelisteten Feldnamen muss der Gesamtzahl der kommagetrennten Felder in den Protokollen entsprechen.</p> <p>Die <code>fields</code>-Option ist für den CSV-Parser obligatorisch. Wenn sie nicht angegeben ist, wird nichts analysiert. Optional und abhängig vom Feldinhalt kann der Feldwert von doppelten Anführungszeichen eingeschlossen sein.</p> <p>Feldnamen müssen durch Kommas getrennt werden, beispielsweise</p> <pre>fields = field_name1, field_name2, field_name3, field_name4</pre> <p>Diese Definition basiert auf der Annahme, dass die Namen <code>field_name1</code>, <code>field_name2</code>, <code>field_name3</code> und <code>field_name4</code> sequenziell den extrahierten Feldern zugeordnet werden.</p> <p>Wenn manche Felder vom CSV-Parser ausgelassen werden müssen, können die Namen in der Liste ausgelassen werden. Beispiel:</p> <pre>fields = field_name1, , field_name3, field_name4</pre> <p>In diesem Fall extrahiert der Parser nur das erste, dritte und vierte Feld des Ereignisses und weist diesen dann die Namen <code>field_name1</code>, <code>field_name3</code> und <code>field_name4</code> zu.</p> <p>Wenn die <code>fields</code>-Option keine vollständige Liste der Felder in Ihren Protokollen angibt, gibt der Parser eine leere Liste zurück. Wenn die Protokolldatei z. B. <code>field1</code>, <code>field2</code>, <code>field3</code>, <code>field4</code> und <code>field5</code> enthält, aber nur <code>fields= field1,field2,field3</code> angegeben ist, gibt der Parser eine leere Feldliste zurück.</p> <p><code>fields=*</code> kann für einen CSV-Parser nicht verwendet werden, da der Parser eine leere Feldliste zurückgibt. Es muss eine vollständige Liste angegeben werden, es sei denn, bestimmte Felder müssen wie bereits beschrieben ausgelassen werden.</p>
<code>delimiter</code>	<p>Die <code>delimiter</code>-Option gibt das Trennzeichen an, das vom Parser verwendet werden soll. Der <code>csv</code>-Parser verwendet standardmäßig ein Komma als Trennzeichen. Sie können dieses jedoch in ein Semikolon, ein Leerzeichen oder ein anderes Sonderzeichen ändern. Das festgelegte Trennzeichen muss in doppelte Anführungszeichen eingeschlossen werden.</p> <p>Beispielsweise <code>delimiter=","</code> und <code>delimiter=";"</code>.</p> <p>Der <code>csv</code>-Parser unterstützt alle Zeichenfolgen als Trennzeichen, welche in Anführungszeichen eingeschlossen sind, z. B. <code>" "</code> oder <code>"asd"</code>. Die Trennzeichen der Feldwerte in den Protokollen müssen mit dem durch den Trennzeichen-Parameter definierten Muster exakt übereinstimmen, da der Parser andernfalls fehlschlägt.</p> <p>Sonderzeichen wie z. B. Leerzeichen oder Tabstopps können als Trennzeichen für den <code>csv</code>-Parser definiert werden, so lange das Espacezeichen dem Sonderzeichen für (<code>\</code>, <code>\s</code>, <code>\t</code>) vorangestellt wird. Beispielsweise <code>delimiter="\s"</code> oder <code>delimiter=" "</code>.</p> <p>Die Option <code>delimiter</code> ist optional.</p>

CSV-Parser-Konfiguration

Verwenden Sie die folgende Konfiguration, um Protokolle aus `winlog`- oder `filelog`-Quellen zu analysieren.

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

```
[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
```

Bei Verwendung dieser Konfiguration werden Protokolle, die aus der `some_csv_logs`-Quelle erfasst werden (z. B. aus dem Verzeichnis `directory=D:\Logs`) von `myparser` analysiert. Wenn die erfassten Protokolle drei durch Semikolon getrennte Werte enthalten, erhalten die analysierten Ereignisse sequenziell die Namen `field_name1`, `field_name2` und `field_name3`.

So analysieren Sie das folgende CSV-Protokoll:

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

Definieren Sie die CSV-Parser-Konfiguration:

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

Der CSV-Parser gibt folgende Felder zurück:

```
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.
```

Protokoll-Parser im Common Log Format (Apache)

Sie können den Apache-Parser für das Common Log Format (CLF) sowohl für `FileLog` als auch für `WinLog-Collectors` konfigurieren.

Parser im Common Log Format (Apache)

Der Standard-CLF-Parser definiert die folgende Reihenfolge und Namen der Felder.

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

Parser-Name: `clf`

Die spezifische Option für den CLF-Parser ist `format`.

format-Option

Die `format`-Option gibt das Format an, in dem Apache-Protokolle erstellt werden. Die Option ist nicht obligatorisch.

Wenn kein Format angegeben ist, wird das folgende Standardformat als Common Log Format verwendet.

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

Die CLF-Parser-Formatzeichenfolge akzeptiert keine Regex-Ausdrücke. Geben Sie z. B. statt des Ausdrucks „\s+“ ein Leerzeichen an.

Zum Analysieren anderer Protokollformate geben Sie das betreffende Format in der Agentenkonfiguration an. Analyisierte Felder werden auf der Serverseite mit den folgenden Namen angezeigt.

Hinweis In Fällen, in denen eine Variable erforderlich ist, werden die Felder ignoriert, wenn {VARNAME} nicht in der Konfiguration bereitgestellt ist.

Felder	Wert
'%a':	"remote_ip"
'%A':	"local_ip"
'%B', '%b':	"response_size"
'%C':	abhängig vom Namen der im Format angegebenen Variablen
'%c':	abhängig vom Namen der im Format angegebenen Variablen
'%D':	"request_time_mcs"
'%E':	"error_status"
'%e':	abhängig vom Namen der im Format angegebenen Variablen
'%F', '%f':	"file_name"
'%h':	"remote_host"
'%H':	"request_protocol"
'%i':	abhängig vom Namen der im Format angegebenen Variablen
'%k':	"keepalive_request_count"
'%l':	"remote_log_name"
'%L':	"request_log_id"
'%M':	"log_message" (Parser beendet nach dem Erreichen des Spezifizierers die Analyse des Eingabeprotokolls)
'%m':	"request_method"

Felder	Wert
'%n':	abhängig vom Namen der im Format angegebenen Variablen
'%o':	abhängig vom Namen der im Format angegebenen Variablen
'%p':	"server_port" Mit dem Spezifizierer %{format}p können weitere Formate verwendet werden. Unterstützte Format sind "canonical", "local" oder "remote". Wenn das Format "canonical" verwendet wird, bleibt der Feldname "server_port". Wenn das Format "local" verwendet wird, lautet der Feldname "local_server_port" und wenn das Format "remote" verwendet wird, lautet der Feldname "remote_server_port".
'%P':	"process_id" Mit dem Spezifizierer %{format}P können weitere Formate verwendet werden. Unterstützte Format sind pid, "tid" und "hextid". Wenn "pid" als ein Format verwendet wird, lautet der Feldname "process_id". Durch die Formate "tid" und "hextid" werden Felder mit dem Namen "thread_id" generiert.
'%q':	"query_string"
'%r':	"request"
'%R':	"response_handler"
'%s':	"status_code", durch das der endgültige Status der Anfrage generiert wird, wird ebenfalls unterstützt. Dies wird auf dem Server als "status_code" angezeigt.

Felder	Wert
'%t':	<p>"timestamp" dient bei der Erfassung als Ereigniszeitstempel und bindet den Zeitstempel-Parser ein. Die automatische Erkennung von Zeitstempel, Datums- und Zeitformat kann durch Angabe in geschweiften Klammern: <code>%{Y-%m-%d %H:%M:%S}t</code> überschrieben werden. Weitere Informationen finden Sie unter Timestamp-Parser.</p> <p>Das Format des Zeitstempels für den CLF-Parser kann mit "begin:"- oder "end:"-Präfixen beginnen. Wenn das Format mit <code>begin:</code> beginnt (Standard), wird die Zeit zu Beginn der Anforderungsverarbeitung erfasst. Wenn es mit <code>end:</code> beginnt, ist die Zeit diejenige, zu welcher der Protokolleintrag verfasst wird, beinahe am Ende der Anforderungsverarbeitung. Vom CLF-Parser werden beispielsweise Formate wie die folgenden unterstützt: <code>%h %l %u [%{begin:%d/%b/%Y %T}t.%{msec_frac}t] \"%r\" %>s %b</code></p> <p>Die folgenden Format-Token werden ebenfalls für die Formatspezifizierung des Zeitstempels des CLF-Parsers unterstützt:</p> <p>sec</p> <p>Anzahl der Sekunden seit der Epoche. Dies entspricht dem Spezifizierer <code>%s</code> des Zeitstempel-Parsers.</p> <p>msec</p> <p>Anzahl der Millisekunden seit der Epoche</p> <p>usec</p> <p>Anzahl der Mikrosekunden seit der Epoche</p> <p>msec_frac</p> <p>Millisekundenbruchteile (entspricht dem Spezifizierer <code>%f</code> des Zeitstempel-Parsers)</p> <p>usec</p> <p>Mikrosekundenbruchteile (entspricht dem Spezifizierer <code>%f</code> des Zeitstempel-Parsers)</p> <p>Um Protokolle zu analysieren, in denen Zeitstempel durch Format-Token wiedergegeben werden, können die folgenden Formate in der Konfiguration verwendet werden:</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>Diese Token können in derselben Formatzeichenfolge nicht miteinander oder mit der Formatierung des Zeitstempel-Parsers kombiniert werden. Sie können stattdessen mehrere <code>%{format}t</code>-Token verwenden. Um beispielsweise einen Zeitstempel mit Millisekunden zu verwenden, ohne den Spezifizierer <code>%f</code> des Zeitstempel-Parsers zu verwenden, kann der folgende kombinierte Zeitstempel verwendet werden: <code>%{%d/%b/%Y %T}t.%{msec_frac}t</code>.</p>
'%T':	"request_time_sec"
'%u':	"remote_auth_user"
'%U':	"requested_url"
'%v':	"server_name"
'%V':	"self_referential_server_name"

Felder	Wert
'%X':	"connection_status" ist abhängig vom Namen der im Format angegebenen Variablen
'%x':	abhängig vom Namen der im Format angegebenen Variablen
'%I':	"received_bytes"
'%O':	"sent_bytes"
'%S':	"transferred_size"

Um z. B. Protokolle mit dem CLF-Parser zu analysieren, die entweder aus winlog- oder aus filelog-Quellen erfasst werden, geben Sie die folgende Konfiguration an:

```
[filelog|clflows]
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
```

Bei Verwendung dieser Konfiguration werden Protokolle, die aus der clflows-Quelle erfasst werden (z. B. aus dem Verzeichnis directory=D:\Logs) von myclf analysiert. Der myclf-Parser analysiert nur die Protokolle, die mit dem in der Konfiguration beschriebenen Format generiert wurden.

Der Debug-Standardwert für Parser ist debug=no.

Analysieren von mit CLF generierten Protokollen

Um Protokolle zu analysieren, die mit CLF generiert wurden, müssen Sie das entsprechende Format in der Konfiguration definieren. Beispiel:

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

Felder, die nicht leer sind und die Bezeichner %{Referer}i und %{User_Agent}i verwenden, werden im vRealize Log Insight-Server mit den Namen referer bzw. user_agent angezeigt.

Integrieren des Timestamp-Parsers mit dem CLF-Parser

Sie können Apache-Protokolle mit einem benutzerdefinierten Zeitformat analysieren.

Greifen Sie wie folgt auf Protokolle zu, die ein benutzerdefiniertes Zeitformat haben.

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

Wenn keine benutzerdefinierte Zeit angegeben ist, versucht der CLF-Parser, das Zeitformat automatisch zu erschließen, indem er den automatischen Timestamp-Parser ausführt. Andernfalls wird das benutzerdefinierte Zeitformat verwendet.

Folgende benutzerdefinierte Zeitformate werden für Fehlerprotokolle unterstützt:

Benutzerdefiniertes Zeitformat	Beschreibung	Konfigurationsformat
%{u}t	Aktuelle Zeit einschließlich Mikrosekunden	format=[%{u}t] [%l] [pid %P] [client %a] %M
%{cu}t	Aktuelle Zeit im kompakten ISO 8601-Format, einschließlich Mikrosekunden	format=[%{cu}t] [%l] [pid %P] [client %a] %M

Eine vollständige Liste der unterstützten Zeitstempelbezeichner finden Sie unter [Timestamp-Parser](#).

Beispiel: Apache-Standard-Zugriffsprotokollkonfiguration für Windows

Beispiel: Apache-Standard-Fehlerprotokollkonfiguration für Windows

Dieses Beispiel zeigt, wie Sie die Zugriffsprotokollkonfiguration für Windows für Apache v2.4 formatieren können.

```
;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\"
```

Definieren Sie das Zugriffsprotokollformat:

1 Konfigurieren Sie Apache für das Zugriffsprotokollformat (httpd.conf):

```
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 Definieren Sie die CLF-Parser-Konfiguration:

```
;ACCESS LOG
;127.0.0.1 unknown - 21-May-2015:13:59:35 "GET /xampp/navi.php HTTP/1.1" 127.0.0.1 127.0.0.1
```

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

Der CLF-Parser gibt Folgendes zurück:

```
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
```

Dieses Beispiel zeigt, wie Sie für Windows die Fehlerprotokollkonfiguration für Apache v2.4 formatieren können.

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

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

;Parser to parse Apache ERROR logs
[parser|clfparsers_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=clfparsers_apache_error2

;Parser to parse Apache ERROR logs
[parser|clfparsers_apache_error2]
    debug=yes
```

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

Hinweis Die bereitgestellten Namen entsprechen dem kombinierten Protokollformat. Apache-Fehlerprotokolle werden auch anhand der obigen Formatierungsschlüssel anstelle des Apache-Fehlerprotokollformats beschrieben.

Definieren Sie das Fehlerprotokollformat:

1 Konfigurieren Sie Apache für das Fehlerprotokollformat (httpd.conf):

```
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 Definieren Sie die CLF-Parser-Konfiguration:

```
;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
```

Protokolleintrag:

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

Der CLF-Parser gibt die folgenden Felder für den Protokolleintrag zurück (bei Verwendung eines Parsers in einer +0400-Zeitzone):

```
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.
```

Protokolleintrag:

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

Der CLF-Parser gibt die folgenden Felder für den Protokolleintrag zurück (bei Verwendung eines Parsers in einer +0400-Zeitzone):

```
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
```

Schlüssel/Wert-Paar-Parser

Sie können den Parser für das Schlüssel/Wert-Paar (Key/Value Pair Parser, KVP) sowohl für FileLog- als auch für WinLog-Collectors konfigurieren.

Schlüssel/Wert-Paar-Parser (KVP-Parser)

Die `kvp`-Parser finden und extrahieren alle `key=value`-Übereinstimmungen in einem beliebigen Protokollnachrichtentext. Im folgenden Beispiel wird das `kvp`-Parser-Format dargestellt.

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

Beispielsweise kann das Schlüsselwertprotokoll folgendes Format aufweisen: `scope=local;`
`abstract=false;` `lazyInit=false;` `autowireMode=0;` `dependencyCheck=0;`

Für den `kvp`-Parser müssen Sie die Felder angeben, aus denen die Werte extrahiert werden sollen. Wenn z. B. die Definition `fields=name,lastname,country` in der Konfiguration vorhanden ist, werden nur die Werte mit den angegebenen Schlüsseln analysiert und an den Server gesandt.

Sowohl der Schlüssel als auch der Wert kann optional von doppelten Anführungszeichen (") eingeschlossen sein, um ein Leerzeichen oder andere Sonderzeichen zu definieren.

Wenn doppelte Anführungszeichen für den Schlüssel oder den Wert verwendet werden, kann der umgekehrte Schrägstrich (\) als Escape-Zeichen verwendet werden. Jedes auf einen umgekehrten Schrägstrich folgende Zeichen ist wörtlich definiert, einschließlich eines doppelten Anführungszeichens oder eines umgekehrten Schrägstrichs. Beispiel: " \ \"

Beachten Sie die folgenden Überlegungen.

- Wenn dem Schlüssel in einem Schlüssel/Wert-Paar kein Ist-gleich-Zeichen folgt und kein `VALUE` angegeben ist, wird die Option wie im Fall von Freitext übersprungen.
- Der Schlüssel darf nicht leer sein, der Wert darf jedoch leer sein.
- Ein Gleichheitszeichen, dem kein Wert folgt, wird als Freitext behandelt und übersprungen.
- Ein Wert kann eine von doppelten Anführungszeichen umgebene Zeichenfolge oder leer sein. Verwenden Sie einen umgekehrten Schrägstrich als Escape-Zeichen für Sonderzeichen, die zum Wert gehören.

Optionen für den KVP-Parser

Beachten Sie die folgenden Informationen hinsichtlich der Struktur des `kvp`-Parsers.

Option	Beschreibung
<code>fields</code>	<p>Die zu extrahierenden Informationen, als Dateneinheiten beschrieben. Beispielsweise <code>fields=name,lastname,country</code>.</p> <p>Wenn bestimmte Feldnamen durch die Option <code>fields</code> definiert sind, werden alle ungültigen Zeichen in einem Feldnamen, der aus einem Protokoll extrahiert wurde, durch einen Unterstrich ersetzt. Wenn z. B. die Option <code>fields</code> nach den Feldern „x-A“ und „a*(X+Y)“ sucht, extrahiert der Parser diese Felder aus den Protokollen und benennt die Felder in „x_a“ und „a__x_y“ um. Auf diese Weise können Felder mit beliebigen Zeichen im Namen extrahiert werden.</p> <p>Wenn die Option <code>fields</code> in der Form „*“ angegeben wird, d. h., wenn der <code>kvp</code>-Parser Feld/Wert-Paare automatisch erkennt, sucht der Parser nach Feldern, die nur alphanumerische und Unterstrichzeichen enthalten (unterstützt durch den LI-Server). Alle anderen ungültigen Zeichen werden verworfen und nicht in Unterstriche umgewandelt. Damit wird verhindert, dass der Parser nicht benötigte Informationen in statische Felder extrahiert.</p>
<code>delimiter</code>	<p>Optional.</p> <p>Standardtrennzeichen sind Leerzeichen, Tabstopp, Zeilenumbruchzeichen, Komma und Semikolon.</p> <p>Wenn in der Konfiguration keine Trennzeichen angegeben sind, verwendet der <code>kvp</code>-Parser Standardtrennzeichen zum Analysieren.</p> <p>Um Standardtrennzeichen in bestimmte Trennzeichen zu ändern, müssen Sie diese zwischen doppelten Anführungszeichen definieren. Beispiel: <code>delimiter = "#^ "</code>. Diese Definition bedeutet, dass jedes der in doppelten Anführungszeichen eingeschlossenen Zeichen als Trennzeichen verwendet wird. Beim <code>kvp</code>-Parser kann jedes Zeichen als Trennzeichen angesehen werden. Sie können die Standardtrennzeichen mit anderen Trennzeichen in die Definition einfügen.</p> <p>Die <code>delimiter = "#^ \t\r\n\s"</code>-Anweisung beispielsweise enthält den Tabstopp, Zeilenumbruchzeichen und das Leerzeichen als Trennzeichen. Wenn diese Zeichen verwendet werden, muss ihnen das Escape-Zeichen vorangestellt werden. Um z. B. das Leerzeichen als Trennzeichen zu definieren, geben Sie das Escape-Zeichen "\" vor dem Leerzeichen ein, wenn Sie dieses als Trennzeichen definieren. Beispiel: <code>delimiter="\s"</code>.</p>
<code>field_decoder</code>	<p>Verschachtelte Parser werden als JSON-Zeichenfolge angegeben, in der die Schlüssel die Namen des Felds sind, auf das der verschachtelte Parser angewendet wird. Der Wert ist der Name des Parsers, der für das betreffende Feld verwendet wird.</p> <p>Jeder verschachtelte Parser wird entsprechend der Decodierung durch den Basis-Parser auf das zutreffende Feld angewandt.</p> <p>Felddecodierer sind nützlich, wenn es sich beim Wert eines Schlüssel/Wert-Paars um einen komplexen Wert wie ein Zeitstempel oder eine kommagetrennte Liste handelt.</p>
<code>debug =</code>	<p>Optional. Der <code>debug =</code>-Wert kann <code>yes</code> oder <code>no</code> betragen. Der Debug-Standardwert für Parser ist <code>debug=no</code>.</p> <p>Wenn die Option auf <code>yes</code> festgelegt ist, können Sie detaillierte Protokolle der Parser-Verarbeitung unter <code>liagent_<Datum>.log</code> anzeigen.</p>

Zusätzliche Optionen für Schlüsselwerte

Schlüssel	Definition
KVP_MESSAGE = * (MESSAGE_ENTRY [WSPR])	Eine Liste mit Nachrichteneinträgen, optional getrennt durch Leerzeichen
MESSAGE_ENTRY = KVP / FREE_TEXT	Ein Eintrag ist ein Schlüssel/Wert-Paar oder nur ein Freitext
KVP = KEY ["=" VALUE]	Schlüssel/Wert-Paar. Wenn auf KEY kein Gleichheitszeichen und VALUE folgen, wird er wie Freitext übersprungen.
KEY = BARE_KEY / QUOTED_KEY	
FREE_TEXT = "="	Ein frei stehendes Gleichheitszeichen wird als Freitext betrachtet und übersprungen.
BARE_KEY = *1BARE_KEY_CHAR	Mindestens ein Zeichen
BARE_KEY_CHAR = %0x00-08 / %0x10-19 / %0x21-3C / %3E-%FF	Ein beliebiges Zeichen mit Ausnahme von Gleichheitszeichen, Leerzeichen oder Tabstopp
QUOTED_KEY = 0x22 *1 (QUOTED_STRING_CHAR / "\" CHAR) 0x22	Mindestens ein in doppelten Anführungszeichen eingeschlossenes Zeichen. Der umgekehrte Schrägstrich wird als Escape-Zeichen verwendet.
QUOTED_STRING_CHAR = %0x00-21 / %0x23-FF	Jedes Zeichen mit Ausnahme von doppelten Anführungszeichen
VALUE = BARE_VALUE / QUOTED_VALUE	
BARE_VALUE = *BARE_VALUE_CHAR	Null oder mehr Zeichen
BARE_VALUE_CHAR = %0x00-08 / %0x10-19 / %0x21-FF	Ein beliebiges Zeichen mit Ausnahme von Leerzeichen oder Tabstopp
QUOTED_VALUE = 0x22 * (QUOTED_STRING_CHAR / "\" CHAR) 0x22	Eine in doppelten Anführungszeichen eingeschlossene Zeichenfolge. Darf leer sein. Der umgekehrte Schrägstrich wird als Escape-Zeichen verwendet.

Beispiele für die Konfiguration von KVP-Parsern

Falls erforderlich, können Sie `fields=*` zum Analysieren aller Felder verwenden.

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

In diesem Beispiel wird dargestellt, wie der Felddecoder angegeben wird.

```
[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"}
```

So analysieren Sie das folgende KVP-Protokoll:

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

Definieren Sie die KVP-Parser-Konfiguration:

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

Der KVP-Parser gibt folgende Felder zurück:

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

Beispiel: Beispiele für einfache und komplexe KVP-Parser

Beispiel für einen einfachen KVP-Parser

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

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

Beispiel für einen komplexen KVP-Parser

```
[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"}
```

Timestamp-Parser

Der `timestamp`-Parser erzeugt keine Felder. Stattdessen wandelt er seine Eingabe aus einer Zeichenfolge in ein internes Zeitstempelformat um, das in Millisekunden seit dem Beginn der UNIX-Epoche, 1. Januar 1970 (Mitternacht UTC/GMT), angezeigt wird.

Die einzige unterstützte Konfigurationsoption ist `format`. Beispielsweise `format=%Y-%m-%d %H:%M:%S`.

Anders als der CLF-Parser kann der `timestamp`-Parser Zeit analysieren, wenn keine Trennzeichen zwischen den Zeitbezeichnern vorliegen, z. B. `%A%B%d%H%M%S%Y%z`.

Formatbezeichner, die vom `timestamp`-Parser verwendet werden, sind:

```
'%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
```

Zusätzliche Bezeichner werden vom `Timestamp`-Parser akzeptiert, aber deren Werte werden ignoriert und haben keine Auswirkung auf die analysierte Zeit.

```
'%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

```

Wenn kein `format`-Parameter definiert ist, analysiert der `Timestamp`-Parser die Zeitstempel anhand der Standardformate.

Automatischer Zeitstempel-Parser

Der automatische Zeitstempel-Parser wird aufgerufen, wenn für den Zeitstempel-Parser kein Format definiert ist oder der Parser ohne eine Zeitstempel-Parser-Definition mithilfe von `timestamp` im `field_decoder` direkt aufgerufen werden kann. Beispiel:

```

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

```

Beispiel: Ein Timestamp-Parser mit der Standardkonfiguration

Dieses Beispiel zeigt einen `timestamp`-Parser mit einer Standardkonfiguration.

```

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

```

Um einen `timestamp`-Parser mit anderen Parsern, z. B. dem `CSV`-Parser, zu integrieren, geben Sie die folgende Konfiguration an.

```

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

```

Wenn diese Konfiguration definiert ist, extrahiert der `mycsv`-Parser die Felder mit den Namen, die in der Konfiguration angegeben sind, und führt `tsp_parser` für den Inhalt des Felds `timestamp` aus. Wenn `tsp_parser` einen gültigen Zeitstempel abrufen kann, verwendet der Server diesen Zeitstempel für die Protokollnachricht.

Automatischer Protokoll-Parser

Der automatische Protokoll-Parser erkennt automatisch den Zeitstempel in den ersten 200 Zeichen einer Zeile. Das Format von automatisch erkannten Zeitstempeln ist das Gleiche wie für den `timestamp`-Parser.

Die automatischen Parser haben keine Optionen. Zusätzlich zur automatischen Erkennung des Zeitstempels wird der Schlüssel/Wert-Parser auf dem Protokolleintrag ausgeführt und erkennt automatisch alle vorhandenen Schlüssel/Wert-Paare in den Protokollen und extrahiert die Felder entsprechend. Beispiel:

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

Wie bei anderen Parsern können Sie eine getrennte Aktion für den automatischen Parser definieren.

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

base_parser=auto
debug=yes
```

Wenn Sie `debug` für den automatischen Parser aktiviert haben, werden zusätzliche Informationen zur Analyse gedruckt. Das sind beispielsweise Informationen zum Protokoll, auf dem der automatische Parser ausgeführt wurde, und dazu, welche Felder aus dem Protokoll extrahiert wurden.

Der Debug-Standardwert für Parser ist `debug=no`.

Syslog-Parser

Der Syslog-Parser unterstützt die Optionen „message_decoder“ und „extract_sd“ und erkennt automatisch zwei Formate: RFC-6587, RFC-5424 und RFC-3164.

Konfigurieren der Option „message_decoder“

Für den Syslog-Parser stehen alle gängigen Optionen sowie die Option `message_decoder` zur Verfügung. Standardmäßig werden nur die Felder `timestamp` und `appname` extrahiert. Aktivieren Sie die Option „message_decoder“ durch Festlegung von Konfigurationswerten in Ihrer Datei `liagent.ini` wie im Folgenden beispielhaft dargestellt:

```
[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=*
```

Beispiel: Analysieren mit der Option „message_decoder“

Das folgende Beispiel zeigt ein Ereignis und die Felder, die diesem Ereignis durch einen Syslog-Parser hinzugefügt wurden, der für die Verwendung der Option „message_decoder“ konfiguriert wurde:

- **Beispielereignis:**

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

- **Zurückgegeben durch einen Syslog-Parser, auf den die Option „message_decoder“ angewendet wird, um einen KVP-Parser auszuführen:**

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

Konfigurieren der Option „extract_sd“ für das Analysieren strukturierter Daten

Um strukturierte Daten zu analysieren, aktivieren Sie die Option „extract_sd“ durch Festlegung der entsprechenden Konfigurationswerte in Ihrer Datei `liagent.ini` wie im Folgenden beispielhaft dargestellt:

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

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

Beispiel: Analysieren mit der Option „extract_sd“

Das folgende Beispiel zeigt ein Ereignis und die Felder, die diesem Ereignis durch einen Syslog-Parser hinzugefügt wurden, der für die Verwendung der Option „extract_sd“ konfiguriert wurde:

- **Das Beispielereignis:** `<165>1 2017-01-24T09:17:15.719Z localhost evntslog - ID47 [exampleSDID@32473 iut="3" eventSource="Application" eventID="1011"] [examplePriority@32473 class="high"] Found entity IPSet, display name dummy ip set 1411`
- **Die folgenden Felder werden dem Ereignis vom Syslog-Parser hinzugefügt:**

```
timestamp=2017-01-24T09:17:15.719000
pri_facility=20
pri_severity=5
procid="-"
msgid="ID47"
iut="3"
```

```

eventsourc="Application"
eventid="1011"
class="high"
appname="evntslog"

```

Vom Parser extrahierte Felder

Der Parser extrahiert die folgenden Felder automatisch aus einem Ereignis:

RFC-Klassifizierung	pri_facility	pri_severity	timestamp	appname	procid	msgid
Nicht-RFC			X	X		
RFC 3164	X	X	X	X		
RFC 5424	X	X	X	X	X	X

Optionen des Syslog-Parsers

Die folgende Tabelle beschreibt die verfügbaren Syslog-Optionen.

Option	Beschreibung
message_decoder	Definiert einen zusätzlichen Parser, der zum Analysieren des Nachrichtentextes eines Ereignisses verwendet wird. Dies kann ein integrierter Parser sein, wie z. B. „auto“ oder ein beliebiger benutzerdefinierter Parser.
extract_sd	Analysiert strukturierte Daten. Es werden nur Ja- oder Nein-Werte für die Option „extract_sd“ unterstützt. Diese Option ist standardmäßig deaktiviert. Wenn die Option „extract_sd“ aktiviert ist, werden einfach alle Schlüssel/Wert-Paare aus den strukturierten Daten extrahiert.

Beispiel: Analysieren für den Standard RFC 5424

Die folgenden Beispiele beinhalten zwei Ereignisse, die von einer konfigurierten Syslog-Instanz analysiert werden, und zeigen die für den Collector verwendete Konfiguration, ein Beispielereignis und die Felder, die der Syslog-Parser dem Ereignis hinzufügt.

■ Konfiguration:

```

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

```

■ Ein in der überwachten Datei generiertes Ereignis:

```

<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

```

- Felder, die dem Ereignis vom Syslog-Parser hinzugefügt werden:

```
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
```

Beispiel: Analysieren für den Standard RFC 3164

Das folgende Beispiel zeigt die für den Collector verwendete Konfiguration, ein RFC 3164-Beispielereignis und die Felder, die Syslog dem Ereignis hinzufügt.

- Konfiguration:

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

- Ein in der überwachten Datei generiertes RFC 3164-Ereignis:

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

- Felder, die dem Ereignis vom Syslog-Parser hinzugefügt werden:

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

Parser für bezeichnete tabulatorgetrennte Werte (LTSV)

Das LTSV-Format („Labeled Tab-Separated Values = bezeichnete tabulatorgetrennte Werte) ist eine Variante der tabulatorgetrennten Werte (TSV).

Jeder Datensatz einer LTSV-Datei wird in einer Zeile dargestellt. Die Felder werden durch <TAB> getrennt und jedes Feld verfügt über eine Bezeichnung und einen Wert. Die Bezeichnung und der Wert werden durch : getrennt. Mithilfe des LTSV-Formats können Sie jede Zeile analysieren, indem Sie die Zeile durch <TAB> teilen (wie beim TSV-Format) und Felder mit eindeutigen Bezeichnungen in beliebiger Reihenfolge erweitern. Weitere Informationen über die LTSV-Definition und das LTSV-Format finden Sie unter <http://ltsv.org/>.

Beispiel: Konfiguration des LTSV-Parsers

Beispiel: Beispiel für LTSV-Protokoll

Die LTSV-Parser erfordert keine spezifischen Konfigurationsoptionen. Geben Sie zum Verwenden des LTSV-Parsers den integrierten `ltsv`-Parsernamen in der Konfiguration an.

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

Eine LTSV-Datei muss eine Byte-Sequenz aufweisen, die mit der LTSV-Produktion im ABNF-Format identisch ist.

```
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)
```

Bei der Beispiel-LTSV-Konfiguration sollten beim Analysieren des Protokolls die folgenden Felder zurückgegeben werden:

```
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)
```

Debug-Konfiguration

Zusätzliches Debugging steht auch für den LTSV-Parser zur Verfügung. Das LTSV-Debugging ist standardmäßig deaktiviert. Um das LTSV-Debugging zu aktivieren, geben Sie `debug=yes` ein.

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

Wenn das Debugging aktiviert ist, extrahiert der LTSV-Parser die Werte aller gültigen Bezeichnungen aus der Protokolldatei. Der LTSV-Parser erfordert, dass die Bezeichnungen nur aus alphanumerischen Zeichen, dem Unterstrich („_“), dem Punkt („.“) und dem Bindestrich („-“) bestehen. Falls im Protokoll mindestens ein ungültiger Bezeichnungsname vorhanden ist, schlägt die Analyse fehl. Auch wenn der Bezeichnungsname gültig ist, wird der Agent den Feldnamen überprüfen. Falls ungültige Namen vorhanden sind, sollte der Bezeichnungsname in einen gültigen Feldnamen geändert werden.

Konfigurieren des LTSV-Parsers vom Abschnitt `filelog` aus

Sie können den LTSV-Parser auch direkt vom Abschnitt `filelog` aus konfigurieren.

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

Regex-Parser

Der `regex`-Parser ermöglicht die Verwendung einiger regulärer Ausdrücke für erfasste Daten.

vRealize Log Insight-Agenten verwenden die C++ Boost-Bibliothek `Regex`, die in der Perl-Syntax geschrieben ist. Der `regex`-Parser kann definiert werden, indem ein Muster für reguläre Ausdrücke angegeben wird, welches benannte Erfassungsgruppen enthält. Beispiel: `(?<field_1>\d{4}) [-] (?<field_2>\d{4}) [-] (?<field_3>\d{4}) [-] (?<field_4>\d{4})`

Die in den Gruppen angegebenen Namen (z. B.: `field_1`, `field_2`, `field_3` und `field_4`) werden zu Namen der entsprechenden extrahierten Felder. Für Namen gelten die folgenden Anforderungen:

- Bei den im Muster für reguläre Ausdrücke angegebenen Namen muss es sich um gültige Feldnamen für vRealize Log Insight handeln.
- Die Namen dürfen nur alphanumerische Zeichen und den Unterstrich (`_`) enthalten.
- Der Name darf nicht mit einer Ziffer beginnen.

Wenn ungültige Namen angegeben werden, schlägt die Konfiguration fehl.

Optionen für den Regex-Parser

Die einzig erforderliche Option für den `regex`-Parser ist die `format`-Option.

Die `debug`-Option kann verwendet werden, wenn weitere Informationen zum Debuggen benötigt werden.

Konfiguration

Um einen `Regex`-Parser zu erstellen, verwenden Sie `regex` als `base_parser` (Basis-Parser) und geben Sie die Option `format` an.

Beispiel: Beispiele für Regex-Konfigurationen

Beispiel: Beispiele für das Analysieren von Apache-Protokollen

Das folgende Beispiel kann für die Analyse von 1234-5678-9123-4567 verwendet werden:

```
[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
```

Das Ergebnis lautet:

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

Um Apache-Protokolle mit dem `regex`-Parser zu analysieren, geben Sie das spezifische `regex`-Format für Apache-Protokolle an:

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

Das Ergebnis lautet:

```
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
```

Bei dem folgenden Code handelt es sich um ein weiteres Beispiel für die Analyse von Apache-Protokollen.

```
[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
```

Überlegungen zur Leistung

Der `regex`-Parser verbraucht mehr Ressourcen als andere Parser, wie z. B. der `CLF`-Parser. Wenn Sie Protokolle mit anderen Parsern analysieren können, ziehen Sie die Verwendung dieser Parser anstelle des `regex`-Parsers in Betracht, um eine höhere Leistung zu erzielen.

Wenn kein Parser angegeben wurde und Sie den `regex`-Parser verwenden, definieren Sie die Formate so eindeutig wie möglich. Im folgenden Beispiel wird eine Konfiguration dargestellt, mit der bessere Leistungsergebnisse erzielt werden. Bei diesem Beispiel werden Felder angegeben, die Ziffernwerte aufweisen.

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

JSON-Parser

Sie können die Konfiguration des JSON-Parser anpassen, um das JSON-Protokoll selektiv zu analysieren.

Sie können Parser für das kommagetrennte Format (CSV) sowohl für **FileLog**- als auch für **WinLog**-Collectors konfigurieren. Nur gültige JSON-Protokolle werden mit dem JSON-Parser des Log Insight-Agenten analysiert. Ungültige JSON-Protokoll-Parser geben leere Ergebnisse zurück.

Bei der standardmäßigen Konfiguration des JSON-Parser werden alle Felder aus dem JSON-Protokoll vom Log Insight-Agenten extrahiert. Wenn sich das JSON-Protokoll als komplexes JSON-Objekt darstellt, das auch JSON-Objekte enthalten kann, verwendet der Parser einen Unterstrich (`_`), um Namen von verschachtelten und höherrangigen JSON-Objekten zu verketteten. Dies erzeugt einen informativen Feldnamen für die entsprechenden Elemente. Wenn das JSON-Protokoll auch ein Array enthält, enthalten die Elementnamen des Elements den Array-Namen, gefolgt vom Index des Elements im Array.

Der JSON-Parser bietet auch eine bestimmte Option, die als **fields** bezeichnet wird.

JSON-Parser-Option „fields“

Sie können die Option **fields** verwenden, um anzugeben, welche Felder in der Konfiguration analysiert werden. Diese Option dient der selektiven Analyse des JSON-Protokolls.

Hinweis Für die selektive Analyse müssen Sie den Pfad zum gewünschten JSON-Element angeben. JSON-Objekte aus verschiedenen Ebenen müssen durch einen Punkt (`.`) getrennt werden.

Die folgende Liste enthält Beispielkonfigurationen, mit denen Sie das JSON-Protokoll nach Bedarf selektiv analysieren können.

- Um mehr als ein Element aus dem JSON-Protokoll zu analysieren, müssen die gewünschten Elemente als Parameter für die Option **fields** angegeben und durch Kommas getrennt werden. Siehe Beispiel unten:

```
{
  "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"
  }
}
```

- Schauen Sie sich das folgende Beispiel an, um nur die innersten JSON-Objekte, z. B. **timestamp**, **log_severity** und **log_message**, zu analysieren. In dieser Beispielkonfiguration werden die folgenden Feldergebnisse generiert: `operation_timestamp = "2018-11-22T15:28:58.094000"` und `operation_log_severity = "info"`

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

- Um das gesamte JSON-Objekt zu analysieren, schließen Sie den Pfad zu dem Objekt ein, gefolgt von einem Sternchen (*).

```
{
  "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"
  }
}
```

- Um nur das Objekt **operation** zu analysieren, verwenden Sie die folgende Konfiguration:

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

- Wenn das JSON-Protokoll ein Array enthält und Sie nur bestimmte Elemente des Arrays analysieren möchten, verwenden Sie den Elementindex des Arrays in der Konfiguration, wie in dieser Beispielkonfiguration gezeigt:

```
{
  "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"
        }
    }
]
}

```

- Um nur die Elemente **key** und **size** aus demselben Protokoll zu analysieren, verwenden Sie die folgende Konfiguration, um die folgenden Felder zu erstellen:

```
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

```

- Um das Feld **key** für alle Array-Elemente zu analysieren, verwenden Sie die folgende Konfiguration:

```

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

```

- Um alle Felder zu analysieren, verwenden Sie die Option „fields“ mit einem Sternchen (*). Diese Konfiguration entspricht der standardmäßigen JSON-Parser-Konfiguration.

```

[parser|json_parser]
base_parser=json
fields=*

```

Deinstallieren von vRealize Log Insight-Agenten

5

Wenn Sie einen vRealize Log Insight-Agenten deinstallieren müssen, befolgen Sie die Anleitung, die dem installierten Agentenpaket entspricht.

Dieses Kapitel enthält die folgenden Themen:

- [Deinstallieren von Log Insight Windows Agent](#)
- [Deinstallieren des Log Insight Linux Agent-RPM-Pakets](#)
- [Deinstallieren des Log Insight Linux Agent-DEB-Pakets](#)
- [Deinstallieren des Log Insight Linux Agent-BIN-Pakets](#)
- [Manuelles Deinstallieren des Log Insight Linux Agent-BIN-Pakets](#)

Deinstallieren von Log Insight Windows Agent

Sie können Log Insight Windows Agent im Bildschirm „Programme und Funktionen“ der Windows-Systemsteuerung deinstallieren.

Voraussetzungen

Melden Sie sich bei dem Windows-Computer an, auf dem Sie den vRealize Log Insight Windows-Agent installiert haben, und starten Sie den Dienst-Manager, um zu überprüfen, ob der vRealize Log Insight-Agent-Dienst installiert ist.

Verfahren

- 1 Navigieren Sie zu **Systemsteuerung > Programme und Funktionen**.
- 2 Wählen Sie den Log Insight Windows Agent von VMware vRealize aus und klicken Sie auf **Deinstallieren**.

Ergebnisse

Das Deinstallationsprogramm beendet den VMware vRealize Log Insight Windows Agent-Dienst und entfernt die entsprechenden Dateien aus dem System.

Deinstallieren des Log Insight Linux Agent-RPM-Pakets

Das Log Insight Linux Agent-RPM-Paket kann deinstalliert werden.

Voraussetzungen

- Melden Sie sich als **Root**-Benutzer an oder verwenden Sie `sudo`, um Konsolenbefehle auszuführen.
- Melden Sie sich bei dem Linux-Computer an, auf dem Sie Log Insight Linux Agent installiert haben. Öffnen Sie eine Terminalkonsole und führen Sie `pgrep liagent` aus, um sicherzustellen, dass VMware Log Insight Linux Agent installiert ist und ausgeführt wird.

Verfahren

- ◆ Führen Sie den folgenden Befehl aus, wobei Sie *VERSION* und *BUILD_NUMBER* durch die Versions- und Buildnummer des installierten Agent ersetzen.

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

Ergebnisse

Das Deinstallationsprogramm stoppt den VMware Log Insight Linux Agent-Daemon und entfernt all seine Dateien, mit Ausnahme seiner eigenen Protokolle, vom System.

Deinstallieren des Log Insight Linux Agent-DEB-Pakets

Sie können das Log Insight Linux Agent-DEB-Paket deinstallieren.

Voraussetzungen

- Melden Sie sich als **Root**-Benutzer an oder verwenden Sie `sudo`, um Konsolenbefehle auszuführen.
- Melden Sie sich bei dem Linux-Computer an, auf dem Sie Log Insight Linux Agent installiert haben. Öffnen Sie eine Terminalkonsole und führen Sie `pgrep liagent` aus, um sicherzustellen, dass VMware Log Insight Linux Agent installiert ist und ausgeführt wird.

Verfahren

- ◆ Führen Sie den folgenden Befehl aus:

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

Ergebnisse

Das Deinstallationsprogramm stoppt den VMware Log Insight Linux Agent-Daemon und entfernt all seine Dateien, mit Ausnahme seiner eigenen Protokolle, vom System.

Deinstallieren des Log Insight Linux Agent-BIN-Pakets

Sie können das Log Insight Linux Agent-BIN-Paket mit einem vRealize Log Insight-Skript deinstallieren.

Voraussetzungen

- Melden Sie sich als **Root**-Benutzer an oder verwenden Sie `sudo`, um Konsolenbefehle auszuführen.
- Melden Sie sich auf dem Linux-Computer an, auf dem Sie den Log Insight Linux Agent installiert haben, öffnen Sie eine Terminalkonsole und führen Sie `pgrep liagent` aus, um sicherzustellen, dass der VMware vRealize Log Insight Linux Agent installiert ist und ausgeführt wird.

Verfahren

- 1 Geben Sie bei der Shell-Eingabeaufforderung folgenden Befehl ein, um das Skript zu starten.

```
loginsight-agent-uninstall
```

- 2 Sie können überprüfen, ob die Deinstallation erfolgreich abgeschlossen wurde, indem Sie prüfen, ob der zurückgegebene Fehlercode des folgenden Befehls 0 ist.

```
echo $?
```

Manuelles Deinstallieren des Log Insight Linux Agent-BIN-Pakets

Sie können das Log Insight Linux Agent-BIN-Paket manuell deinstallieren, wenn Sie das Deinstallationsskript nicht verwenden möchten.

Voraussetzungen

Manuelles Deinstallieren des Log Insight Linux Agent-BIN-Pakets

- Melden Sie sich als **Root**-Benutzer an oder verwenden Sie `sudo`, um Konsolenbefehle auszuführen.
- Melden Sie sich auf dem Linux-Computer an, auf dem Sie den Log Insight Linux Agent installiert haben, öffnen Sie eine Terminalkonsole und führen Sie `pgrep liagent` aus, um sicherzustellen, dass der VMware vRealize Log Insight Linux Agent installiert ist und ausgeführt wird.

Verfahren

- 1 Halten Sie den Log Insight Linux Agent-Daemon an, indem Sie den folgenden Befehl ausführen.

```
sudo service liagentd stop oder sudo /sbin/service liagentd stop für ältere Distributionen.
```

- 2 Manuelles Entfernen der Dateien des Log Insight Linux Agent

- `/usr/lib/loginsight-agent` – Verzeichnis für Binär- und Lizenzdateien des Daemons.

- `/usr/bin/loginsight-agent-support` – Wird verwendet, um das Support-Paket für den Log Insight Linux Agent zu generieren.
- `/var/lib/loginsight-agent` – Verzeichnis für Konfigurationsdateien und Datenbankspeicher.
- `/var/log/loginsight-agent` – Protokollverzeichnis des Log Insight Linux Agent.
- `/var/run/liagent/liagent.pid` – PID-Datei des Log Insight Linux Agent. Sollte sie nicht automatisch gelöscht werden, entfernen Sie sie manuell.
- `/etc/init.d/liagentd` – Skriptverzeichnis des Log Insight Linux Agent-Daemons.
- `/usr/lib/systemd/system/liagentd.service`

Fehlerbehebung für vRealize Log Insight-Agenten

6

Informationen zur Behebung bekannter Fehler können Ihnen bei der Diagnose und Behebung von Problemen bei der Bedienung von vRealize Log Insight-Agenten helfen.

Dieses Kapitel enthält die folgenden Themen:

- Erstellen eines Support-Pakets für den Log Insight Windows Agent
- Erstellen eines Support-Pakets für den Log Insight Linux Agent
- Festlegen der Protokolldateiebene in den Log Insight Agents
- Keine Anzeige von Log Insight Agents auf der Benutzeroberfläche für Administratoren
- vRealize Log Insight Agents senden keine Ereignisse
- Hinzufügen einer Ausnahmeregel für den Ausgang für den Log Insight Windows Agent
- Zulassen von ausgehenden Verbindungen vom Log Insight Windows Agent in einer Windows-Firewall
- Die Massenbereitstellung des Log Insight Windows Agent ist nicht erfolgreich
- Ablehnen von selbstsignierten Zertifikaten durch Log Insight Agents
- Der vRealize Log Insight-Server lehnt die Verbindung für nicht verschlüsselten Datenverkehr ab

Erstellen eines Support-Pakets für den Log Insight Windows Agent

Falls der Log Insight Windows Agent aufgrund eines Problems nicht wie erwartet funktioniert, können Sie eine Kopie der Protokoll- und Konfigurationsdateien an die VMware Support-Dienste senden.

Verfahren

- 1 Melden Sie sich auf dem Zielcomputer an, auf dem der Log Insight Windows Agent installiert wurde.
- 2 Klicken Sie auf die Windows-Schaltfläche **Start** und wählen Sie **VMware > Log Insight Agent - Support-Paket erfassen** aus.

- 3 (Optional) Wenn die Verknüpfung nicht verfügbar ist, navigieren Sie zum Installationsverzeichnis von Log Insight Windows Agent und doppelklicken Sie auf `loginsight-agent-support.exe`.

Hinweis Das Standardinstallationsverzeichnis lautet `C:\Programme (x86)\VMware\Log Insight Agent`.

Ergebnisse

Das Paket wird generiert und als ZIP-Datei in `Eigene Dokumente` gespeichert.

Nächste Schritte

Leiten Sie das Support-Paket wie verlangt an die VMware Support-Dienste weiter.

Erstellen eines Support-Pakets für den Log Insight Linux Agent

Falls der Log Insight Linux Agent aufgrund eines Problems nicht wie erwartet funktioniert, können Sie eine Kopie der Protokoll- und Konfigurationsdateien an die VMware Support-Dienste senden.

Verfahren

- 1 Melden Sie sich auf dem Zielcomputer an, auf dem der Log Insight Linux Agent installiert wurde.
- 2 Führen Sie den folgenden Befehl aus.

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

Ergebnisse

Das Paket wird erstellt und als ZIP-Datei im aktuellen Verzeichnis gespeichert.

Nächste Schritte

Leiten Sie das Support-Paket wie verlangt an die VMware Support-Dienste weiter.

Festlegen der Protokolldateiebene in den Log Insight Agents

Sie können die Konfigurationsdatei des vRealize Log Insight-Agenten bearbeiten, um die Protokollierungsebene zu ändern.

Voraussetzungen

Für den Log Insight Linux Agent:

- Melden Sie sich als **Root**-Benutzer an oder verwenden Sie `sudo`, um Konsolenbefehle auszuführen.

- Melden Sie sich bei dem Linux-System an, auf dem Sie Log Insight Linux Agent installiert haben. Öffnen Sie eine Konsole und führen Sie `pgrep liagent` aus, um sicherzustellen, dass VMware vRealize Log Insight Linux Agent installiert ist und ausgeführt wird.

Für den Log Insight Windows Agent:

- Melden Sie sich bei dem Windows-Computer an, auf dem Sie den vRealize Log Insight Windows-Agent installiert haben, und starten Sie den Dienst-Manager, um zu überprüfen, ob der vRealize Log Insight-Agent-Dienst installiert ist.

Verfahren

- 1 Navigieren Sie zu dem Ordner, der die Datei `liagent.ini` enthält.

Betriebssystem	Pfad
Linux	<code>/var/lib/loginsight-agent/</code>
Windows	<code>%ProgramData%\VMware\Log Insight Agent</code>

- 2 Öffnen Sie die Datei `liagent.ini` in einem beliebigen Texteditor.
- 3 Ändern Sie die Debug-Protokollierungsebene im Abschnitt `[logging]` der Datei `liagent.ini`.

Hinweis Je höher die Debug-Ebene, desto größer sind die Auswirkungen auf den vRealize Log Insight Agent. Der empfohlene Standardwert lautet 0. Debug-Ebene 1 bietet weitere Informationen und wird zur Behebung der meisten Fehler empfohlen. Debug-Ebene 2 bietet detaillierte Informationen. Verwenden Sie die Ebenen 1 und 2 nur, wenn Sie vom VMware-Support dazu aufgefordert werden.

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

- 4 Speichern und schließen Sie die Datei `liagent.ini`.

Ergebnisse

Die Debug-Protokollierungsebene wurde geändert.

Keine Anzeige von Log Insight Agents auf der Benutzeroberfläche für Administratoren

Auf der Seite „Agents“ der Benutzeroberfläche für Administratoren werden keine Log Insight Agents-Instanzen angezeigt.

Problem

Nach der Installation des Log Insight Agents werden die Log Insight Agents auf der Seite „Agents“ der Benutzeroberfläche für Administratoren nicht angezeigt.

Ursache

Die gängigsten Ursachen sind Netzwerkverbindungsprobleme oder die nicht ordnungsgemäße Konfiguration des Log Insight Agents in der Datei `liagent.ini`.

Lösung

- ◆ Vergewissern Sie sich, dass das Windows- oder Linux-System, auf dem die Log Insight Agents installiert sind, mit dem vRealize Log Insight-Server verbunden ist.
- ◆ Überprüfen Sie, ob der Log Insight Agents das cfapi-Protokoll verwendet.

Bei Verwendung des Syslog-Protokolls wird Log Insight Windows Agents nicht auf der Benutzeroberfläche angezeigt.

- ◆ Zeigen Sie die Inhalte der Log Insight Agents-Protokolldateien an, die sich in den folgenden Verzeichnissen befinden.

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

Suchen Sie nach Protokollmeldungen, die die Ausdrücke `Config transport error: Couldn't resolve host name` und `Resolver failed. No such host is known` enthalten.

- ◆ Vergewissern Sie sich, dass die Datei `liagent.ini` die richtige Konfiguration für den vRealize Log Insight-Zielserver enthält. Siehe [Festlegen des vRealize Log Insight-Zielservers](#) und [Festlegen eines Agent-Ziels](#).

vRealize Log Insight Agents senden keine Ereignisse

Eine falsche Konfiguration kann den vRealize Log Insight-Agents daran hindern, Ereignisse an den vRealize Log Insight-Server weiterzuleiten. Wenn ein Flatfile-Erfassungskanal nicht korrekt konfiguriert ist, sehen Sie möglicherweise Meldungen wie „Für den Kanal 'CHANNEL_NAME' wurden ungültige Einstellungen erhalten“. Der Kanal 'CHANNEL_NAME' bleibt inaktiv, bis er ordnungsgemäß konfiguriert ist.

Problem

Die Instanzen der vRealize Log Insight-Agents werden auf der Seite **Verwaltung > Agent** angezeigt. Auf der Seite „Interaktive Analyse“ werden hinsichtlich der Hostnamen der vRealize Log Insight-Agents jedoch keine Ereignisse angezeigt. Der Flatfile-Erfassungskanal ist nicht richtig konfiguriert.

Ursache

Eine falsche Konfiguration kann den vRealize Log Insight Agent daran hindern, Ereignisse an den vRealize Log Insight-Server weiterzuleiten.

Lösung

- ◆ Definieren Sie einen gültigen Erfassungskanal. Stellen Sie sicher, dass der Flatfile-Erfassungskanal richtig konfiguriert ist. Weitere Informationen hierzu finden Sie unter [Kapitel 4 Konfigurieren von vRealize Log Insight-Agenten](#).
- ◆ Versuchen Sie für den vRealize Log Insight-Windows-Agenten Folgendes.
 - Wenn Windows-Kanäle aktiviert sind, zeigen Sie den Inhalt der Protokolldateien des vRealize Log Insight-Windows-Agenten an, die sich im Verzeichnis `%ProgramData%\VMware\Log Insight Agent\log` befinden. Suchen Sie nach Protokollmeldungen in Bezug auf die Kanalkonfiguration, welche die Ausdrücke `Hat den Kanal CHANNEL_NAME abonniert` enthalten. Typischerweise verwendete Kanäle sind `Application`, `System` und `Security`.
 - Wenn ein Kanal nicht korrekt konfiguriert ist, sehen Sie Protokollmeldungen wie diese: Abonnieren der Ereignisse des Kanals `CHANNEL_NAME` nicht möglich. Fehlercode: 15007. Der angegebene Kanal wurde nicht gefunden. Überprüfen Sie die Kanalkonfiguration. Möglicherweise sehen Sie eine andere Fehlercodenummer als 15007.
 - Wenn ein Flatfile-Erfassungskanal nicht korrekt konfiguriert ist, sehen Sie möglicherweise Meldungen wie Für den Kanal '`CHANNEL_NAME`' wurden ungültige Einstellungen erhalten. Der Kanal '`CHANNEL_NAME`' bleibt inaktiv, bis er ordnungsgemäß konfiguriert ist
- ◆ Versuchen Sie für den vRealize Log Insight-Windows-Agenten und den vRealize Log Insight-Agenten Folgendes.
 - ◆ Wenn kein Flatfile-Erfassungskanal konfiguriert ist, sehen Sie möglicherweise Meldungen wie Bereich '`filelog`' wurde in der Konfiguration nicht gefunden. Der Flat-Dateiprotokoll-Collector bleibt inaktiv, bis er ordnungsgemäß konfiguriert wurde

Die Protokolldateien der vRealize Log Insight Agents befinden sich in den folgenden Verzeichnissen.

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

Nächste Schritte

Weitere Informationen über das Konfigurieren der vRealize Log Insight Agents finden Sie unter [Konfigurieren des Log Insight Windows Agent](#) und [Konfigurieren des Log Insight Linux Agent](#).

Hinzufügen einer Ausnahmeregel für den Ausgang für den Log Insight Windows Agent

Definieren Sie eine Ausnahme für die Aufhebung der Blockierung des Log Insight Windows Agent in der Windows-Firewall.

Dieses Verfahren gilt für Windows Server 2008 R2 und höher und für Windows 7 und höher.

Voraussetzungen

- Stellen Sie sicher, dass Sie über ein Administratorkonto oder ein Konto mit Administratorberechtigungen verfügen.

Verfahren

- 1 Wählen Sie **Start > Ausführen** aus.
- 2 Geben Sie `wf.msc` ein und klicken Sie auf **OK**.
- 3 Klicken Sie mit der rechten Maustaste im linken Fensterbereich auf **Ausgehende Regeln** und klicken Sie im linken Fensterbereich auf **Neue Regel**.
- 4 Wählen Sie **Benutzerdefiniert** aus und folgen Sie dem Assistenten, um die folgenden Optionen festzulegen.

Option	Beschreibung
Programm	liwinsvc.exe
Dienst	LogInsightAgentService
Protokolle und Ports	TCP 9000 für CFAPI und 514 für Syslog

- 5 Wählen Sie auf der Seite „Angaben der Profile, für die diese Regel angewendet wird“ den gewünschten Netzwerktyp aus.
 - Domäne
 - Öffentlich
 - Privat

Hinweis Sie können alle Netzwerktypen auswählen, um sicherzustellen, dass die Ausnahmeregel unabhängig vom Netzwerktyp aktiv ist.

Nächste Schritte

Navigieren Sie zum Log Insight Windows Agent-Protokollverzeichnis `%ProgramData%\VMware\Log Insight Agent\log` und öffnen Sie die aktuelle Protokolldatei. Wenn aktuelle Ereignisse die Meldungen `Config transport error: Couldn't resolve host name` und `Resolver failed. No such host is known` enthalten, starten Sie den Log Insight Windows Agent-Dienst und den Windows-Computer neu.

Hinweis Es kann bis zu 5 Minuten dauern, bis der Log Insight Windows Agent-Dienst erneut eine Verbindung zum Server herstellt.

Zulassen von ausgehenden Verbindungen vom Log Insight Windows Agent in einer Windows-Firewall

Konfigurieren Sie die Einstellungen der Windows-Firewall, um ausgehende Verbindungen des Log Insight Windows Agent zum vRealize Log Insight-Server zuzulassen.

Nach der Installation des Log Insight Windows Agent-Diensts schränkt die Windows-Domäne oder die lokale Firewall die Verbindung zum vRealize Log Insight-Zielserver möglicherweise ein.

Dieses Verfahren gilt für Windows Server 2008 R2 und höher und für Windows 7 und höher.

Voraussetzungen

- Stellen Sie sicher, dass Sie über ein Administratorkonto oder ein Konto mit Administratorberechtigungen verfügen.

Verfahren

- 1 Wählen Sie **Start > Ausführen** aus.
- 2 Geben Sie `wf.msc` ein und klicken Sie auf **OK**.
- 3 Klicken Sie im Fensterbereich „Aktionen“ auf **Eigenschaften**.
- 4 Wählen Sie auf der Registerkarte **Domänenprofil** die Option **Zulassen (Standard)** aus dem Dropdown-Menü **Ausgehende Verbindungen** aus.

Wenn der Computer nicht mit einer Domäne verbunden ist, können Sie je nach Netzwerktyp, mit dem der Computer verbunden ist, die Option **Privates Profil** oder **Öffentliches Profil** auswählen.

- 5 Klicken Sie auf **OK**.

Nächste Schritte

Definieren Sie eine Ausnahmeregel für die Aufhebung der Blockierung für den Log Insight Windows Agent in der Windows-Firewall. Weitere Informationen hierzu finden Sie unter [Hinzufügen einer Ausnahmeregel für den Ausgang für den Log Insight Windows Agent](#).

Die Massenbereitstellung des Log Insight Windows Agent ist nicht erfolgreich

Die Massenbereitstellung des Log Insight Windows Agent ist auf Zielcomputern nicht erfolgreich.

Problem

Nach der Durchführung einer Massenbereitstellung auf Windows-Domänencomputern unter Verwendung von Gruppenrichtlinienobjekten schlägt die Installation von Log Insight Windows Agent als lokaler Dienst fehl.

Ursache

Gruppenrichtlinieneinstellungen verhindern die ordnungsgemäße Installation des Log Insight Windows Agent.

Lösung

- 1 Bearbeiten Sie die Einstellungen für das Gruppenrichtlinienobjekt und stellen Sie den Log Insight Windows Agent erneut bereit.
 - a Klicken Sie mit der rechten Maustaste auf das Gruppenrichtlinienobjekt, klicken Sie auf **Bearbeiten** und navigieren Sie zu **Computerkonfiguration > Richtlinien > Administrative Vorlagen > System > Anmeldung**.
 - b Aktivieren Sie die Richtlinie **Beim Neustart des Computers und bei der Anmeldung immer auf das Netzwerk warten**.
 - c Navigieren Sie zu **Computerkonfiguration > Richtlinien > Administrative Vorlagen > System > Gruppenrichtlinie**.
 - d Aktivieren Sie die **Wartezeit für Richtlinienverarbeitung beim Systemstart** und legen Sie die **Wartezeit (in Sekunden)** auf 120 fest.
- 2 Führen Sie den Befehl `gpupdate /force /boot` auf den Zielcomputern aus.

Ablehnen von selbstsignierten Zertifikaten durch Log Insight Agents

Die Log Insight Agents lehnen ein selbstsigniertes Zertifikat ab.

Problem

Ein vRealize Log Insight-Agent lehnt selbstsignierte Zertifikate ab und kann keine Verbindung mit dem Server herstellen.

Hinweis Bei Verbindungsproblemen mit dem Agent können Sie detaillierte Protokolle für die Überprüfung generieren, indem Sie die Debug-Stufe für den Agent auf 1 ändern. Weitere Informationen finden Sie unter [Festlegen der Protokolldateiebene in den Log Insight Agents](#).

Ursache

Die Meldungen im Agent-Protokoll wurden aus bestimmten Gründen in das Protokoll eingetragen.

Meldung	Ursache
Selbstsigniertes Zertifikat wird abgelehnt. Der öffentliche Schlüssel stimmt nicht mit dem Schlüssel des vorher gespeicherten Zertifikats überein.	<ul style="list-style-type: none"> ■ Diese Meldung kann auftreten, wenn das vRealize Log Insight-Zertifikat ersetzt wird. ■ Diese Meldung kann auftreten, wenn die in einer Clusterumgebung aktivierte Hochverfügbarkeit mit selbstsignierten Zertifikaten auf vRealize Log Insight-Knoten konfiguriert ist.
Selbstsigniertes Zertifikat wird abgelehnt. Es ist bereits ein Zertifikat einer vertrauenswürdigen Zertifizierungsstelle vorhanden.	Auf dem Agent ist bereits ein Zertifikat einer vertrauenswürdigen Zertifizierungsstelle gespeichert.

Lösung

- ◆ Überprüfen Sie, ob es sich bei Ihrem Zielhostnamen um eine vertrauenswürdige vRealize Log Insight-Instanz handelt, und löschen Sie das frühere Zertifikat anschließend manuell aus dem Verzeichnis `cert` des vRealize Log Insight-Agent.
 - Log Insight Windows Agent: Navigieren Sie zu `C:\ProgramData\VMware\Log Insight Agent\cert`.
 - Log Insight Linux Agent: Navigieren Sie zu `/var/lib/loginsight-agent/cert`.

Hinweis Auf einigen Plattformen können die Pfade der Verzeichnisse mit den vertrauenswürdigen Zertifikaten anders ausfallen. In den Log Insight Agents kann der Pfad zu den vertrauenswürdigen Zertifikaten über den Konfigurationsparameter `ssl_ca_path=<fullpath>` festgelegt werden. Ersetzen Sie `<fullpath>` durch den Pfad zur Paketdatei der vertrauenswürdigen Stammzertifikate. Siehe [Konfigurieren der SSL-Parameter für die Log Insight-Agenten](#)

Der vRealize Log Insight-Server lehnt die Verbindung für nicht verschlüsselten Datenverkehr ab

Der vRealize Log Insight-Server weist die Verbindung mit dem Log Insight Agents ab, wenn Sie versuchen, nicht verschlüsselte Daten zu senden.

Sie können einen vRealize Log Insight-Server so konfigurieren, dass Nicht-SSL-Verbindungen angenommen werden, oder Sie können Log Insight Agents so konfigurieren, dass Daten über eine SSL `cfapi`-Protokollverbindung gesendet werden.


Problem

Wenn Sie versuchen, `cfapi` zum Senden von nicht verschlüsseltem Datenverkehr zu verwenden, lehnt der vRealize Log Insight-Server Ihre Verbindung ab. Im Agent-Protokoll ist eine dieser beiden Fehlermeldungen enthalten: 403 Verboten oder 403 Es sind nur SSL-Verbindungen zulässig.

Ursache

vRealize Log Insight ist so konfiguriert, dass nur SSL-Verbindungen angenommen werden, aber die Log Insight Agents sind für die Verwendung einer Nicht-SSL-Verbindung konfiguriert.

Lösung

- 1 Konfigurieren Sie Ihren vRealize Log Insight-Server so, dass Nicht-SSL-Verbindungen akzeptiert werden.
 - a Klicken Sie auf das Dropdown-Menüsymbol für die Konfiguration  und wählen Sie **Verwaltung** aus.
 - b Klicken Sie unter „Konfiguration“ auf **SSL**.
 - c Deaktivieren Sie unter „API-Server-SSL“ die Einstellung **SSL-Verbindung erforderlich**.
 - d Klicken Sie auf **Speichern**.
- 2 Konfigurieren Sie den vRealize Log Insight-Agent so, dass Daten über eine SSL `cfapi`-Protokollverbindung gesendet werden.
 - a Navigieren Sie zum Ordner mit der Datei `liagent.ini`.

Betriebssystem	Pfad
Linux	<code>/var/lib/loginsight-agent/</code>
Windows	<code>%ProgramData%\VMware\Log Insight Agent</code>

- b Öffnen Sie die Datei `liagent.ini` in einem beliebigen Texteditor.
- c Ändern Sie den Wert des Schlüssels `ssl` im [Server]-Bereich der Datei `liagent.ini` in „Ja“ und das Protokoll in `cfapi`.

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

- d Speichern und schließen Sie die Datei `liagent.ini`.