

Arbeiten mit vRealize Log Insight-Agenten

5. September 2017
vRealize Log Insight 4.5

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

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

Auf der VMware-Website finden Sie auch die aktuellen Produkt-Updates.

Falls Sie Anmerkungen zu dieser Dokumentation haben, senden Sie Ihre Kommentare und Vorschläge an:

docfeedback@vmware.com

Copyright © 2014–2017 VMware, Inc. Alle Rechte vorbehalten. [Informationen zu Copyright und Marken.](#)

-

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

VMware Global, Inc.
Zweigniederlassung Deutschland
Freisinger Str. 3
85716 Unterschleißheim/Lohhof
Germany
Tel.: +49 (0) 89 3706 17000
Fax: +49 (0) 89 3706 17333
www.vmware.com/de

Inhalt

Informationen zu „Arbeiten mit vRealize Log Insight-Agenten“	5
1 Übersicht über vRealize Log Insight -Agenten	7
2 Installieren oder Aktualisieren von vRealize Log Insight -Agenten	11
Herunterladen der Agent-Installationsdateien	12
Installieren von Windows-Agenten	13
Installieren oder Aktualisieren des vRealize Log Insight -Windows-Agenten mit dem Installations-Assistenten	13
Installieren oder Aktualisieren des vRealize Log Insight -Windows-Agenten von der Befehlszeile	13
Bereitstellen des Log Insight Windows Agent auf mehreren Computern	15
Installieren oder Aktualisieren des RPM-Pakets für den Linux-Agent für vRealize Log Insight	18
Installieren oder Aktualisieren des DEB-Pakets des Linux-Agent für vRealize Log Insight	19
Installieren des Log Insight Linux Agent -Binärpakets	22
Optionen für vRealize Log Insight-Agenten	23
Automatische Aktualisierung von vRealize Log Insight-Agenten	25
Deaktivieren oder Aktivieren der automatischen Aktualisierung für einzelne Agenten	25
3 Konfigurieren eines vRealize Log Insight -Agenten	27
Konfigurieren des Log Insight Windows Agent nach der Installation	28
Standardkonfiguration des Log Insight Windows Agent	28
Festlegen des vRealize Log Insight -Zielservers	30
Erfassen von Ereignissen aus Windows-Ereigniskanälen	33
Erfassen von Ereignissen aus einer Protokolldatei	37
Weiterleiten von Ereignissen an den Log Insight Windows Agent	40
Konfigurieren des Log Insight Linux Agent	41
Standardkonfiguration des vRealize Log Insight Linux Agent	41
Festlegen des vRealize Log Insight -Zielservers	43
Erfassen von Ereignissen aus einer Protokolldatei	45
Zentrale Konfiguration von vRealize Log Insight -Agenten	49
Exemplarische Konfigurationszusammenführung	49
Verwenden von allgemeinen Werten für die Konfiguration von Agenten	51
Analysieren von Protokollen	52
Konfigurieren von Protokoll-Parsern	53
4 Deinstallieren von vRealize Log Insight -Agenten	77
Deinstallieren von Log Insight Windows Agent	77
Deinstallieren des Log Insight Linux Agent-RPM-Pakets	77
Deinstallieren des Log Insight Linux Agent-DEB-Pakets	78
Deinstallieren des Log Insight Linux Agent-BIN-Pakets	78

5 Fehlerbehebung für vRealize Log Insight -Agenten	81
Erstellen eines Support-Pakets für den Log Insight Windows Agent	81
Erstellen eines Support-Pakets für den Log Insight Linux Agent	82
Festlegen der Protokolldateiebene in den Log Insight Agents	82
Keine Anzeige von Log Insight Agents auf der Benutzeroberfläche für Administratoren	83
vRealize Log Insight Agents senden keine Ereignisse	84
Hinzufügen einer Ausnahmeregel für den Ausgang für den Log Insight Windows Agent	85
Zulassen von ausgehenden Verbindungen vom Log Insight Windows Agent in einer Windows-Firewall	86
Die Massenbereitstellung des Log Insight Windows Agent ist nicht erfolgreich	86
Ablehnen von selbstsignierten Zertifikaten durch Log Insight Agents	87
Der vRealize Log Insight -Server lehnt die Verbindung für nicht verschlüsselten Datenverkehr ab	88
 Index	 91

Informationen zu „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 sind für erfahrene Systemadministratoren bestimmt, die mit der Windows- oder Linux-VM-Technologie und Datencenteroperationen 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

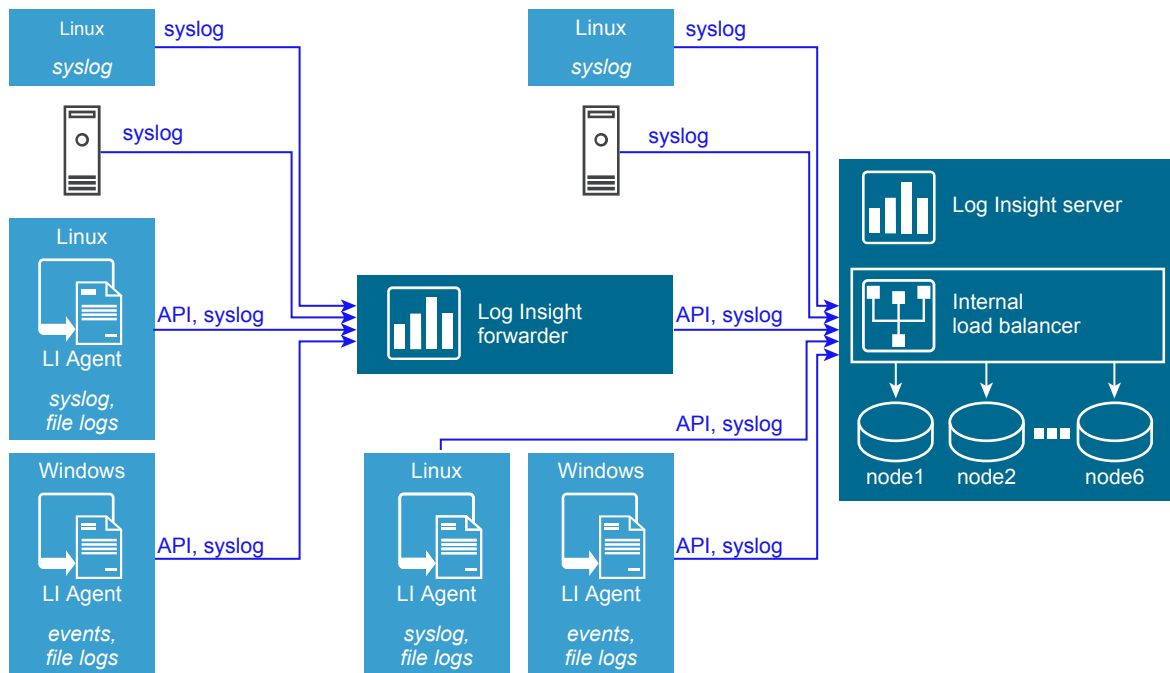
Ein vRealize Log Insight-Agent erfasst Ereignisse aus Protokolldateien und leitet sie an einen vRealize Log Insight-Server weiter.

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.

Sie können für Agenten die folgenden Funktionen verwenden:

- Einzel- oder Gruppenbereitstellung
- Manuelle oder automatische Upgrades
- Zentrale Konfigurationsverwaltung, einschließlich Unterstützung für die lokale und globale Konfiguration von Agenten bei der Installation oder über Konfigurationsdateien, die Web-Benutzeroberfläche oder die API
- Agentengruppen, die eine allgemeine Konfiguration gemeinsam nutzen
- Verwendung mit vRealize Log Insight-Ereignisweiterleitungen. Ereignisweiterleitungen funktionieren ähnlich wie Syslog-Aggregatoren und sind identisch mit den primären vRealize Log Insight-Cluster-Knoten.

Die folgende Abbildung zeigt die Elemente einer Agent-Bereitstellungskonfiguration.



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. Das Erfassen von Agentenprotokollen mithilfe des vRealize Log Insight-Agenten selbst wird nicht unterstützt.

Für die Windows- und Linux-Betriebssysteme werden keine separaten Agenten zur Verfügung gestellt.

Windows-Agenten

Der vRealize Log Insight Windows-Agent erfasst Ereignisse aus Windows-Ereigniskanälen und -Protokolldateien und leitet diese an den vRealize Log Insight-Server weiter. Ein Windows-Ereigniskanal ist ein Pool zur Erfassung verwandter Ereignisse in einem Windows-System. Anwendungen können auch Protokolldaten in einfachen Textdateien im Dateisystem speichern. Der vRealize Log Insight-Agent überwacht Ereigniskanäle und Verzeichnisse und erfasst Ereignisse aus Anwendungsprotokolldateien und leitet diese weiter.

Der vRealize Log Insight-Windows-Agent weist eine Beschränkung von 64 KB pro Anforderung an den vRealize Log Insight-Server auf.

Der vRealize Log Insight-Windows-Agent wird als Windows-Dienst ausgeführt und startet sofort nach der Installation. Während und nach der Installation können Sie die folgenden Optionen für den vRealize Log Insight-Windows-Agenten konfigurieren:

- Der vRealize Log Insight-Zielserver, an den der vRealize Log Insight-Windows-Agent Ereignisse weiterleitet
- Das Kommunikationsprotokoll und der Port, die der Agent verwendet.
- Das Hinzufügen oder Entfernen von Windows-Ereigniskanälen.
- Die Auswahl von Windows-Verzeichnissen für die Überwachung und das Hinzufügen von einfachen Protokolldateien für die Erfassung.

Linux-Agenten

Der vRealize Log Insight-Linux-Agent erfasst Ereignisse aus Protokolldateien auf Linux-Computern und leitet sie an den vRealize Log Insight-Server weiter.

Der Log Insight-Linux-Agent wird als Daemon ausgeführt und startet sofort nach der Installation. Nach der Installation können Sie die folgenden Optionen konfigurieren:

- Den vRealize Log Insight-Zielserver, an den der Agent Ereignisse weiterleitet.
- Die Verzeichnisse, die der Agent überwacht.

Installieren oder Aktualisieren von vRealize Log Insight -Agenten

2

Sie können vRealize Log Insight-Agenten auf Windows- oder Linux-Maschinen installieren, 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](#)“, auf Seite 25. 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 und Windows Server 2016	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 und 16.04 LTS	x86_64
VMware Photon, Version 1, Revision 2	x86_64

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 beim Kanal nicht erfolgreich war, 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 behandelt die folgenden Themen:

- [„Herunterladen der Agent-Installationsdateien“](#), auf Seite 12
- [„Installieren von Windows-Agenten“](#), auf Seite 13
- [„Installieren oder Aktualisieren des RPM-Pakets für den Linux-Agent für vRealize Log Insight“](#), auf Seite 18
- [„Installieren oder Aktualisieren des DEB-Pakets des Linux-Agent für vRealize Log Insight“](#), auf Seite 19
- [„Installieren des Log Insight Linux Agent-Binärpakets“](#), auf Seite 22
- [„Optionen für vRealize Log Insight-Agenten“](#), auf Seite 23
- [„Automatische Aktualisierung von vRealize Log Insight-Agenten“](#), auf Seite 25

Herunterladen der Agent-Installationsdateien

Wählen Sie eine Agenteninstallation für Ihre Plattform aus und laden Sie diese.

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 bereits ein Hostname vorhanden ist oder wenn Sie das Paket über den Parameter für den Hostnamen ausführen, wird der Hostname des eingebetteten Servers ignoriert.

Vorgehensweise

- 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.

Weiter

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“](#), auf Seite 12.
- Stellen Sie sicher, dass Sie über Berechtigungen zum Ausführen von Installationen und zum Starten von Diensten auf dem Windows-Computer verfügen.

Vorgehensweise

- 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**.

Weiter

Konfigurieren Sie den Windows-Agenten für vRealize Log Insight, indem Sie die Datei `liagent.ini` bearbeiten. Siehe [„Konfigurieren des Log Insight Windows Agent nach der Installation“](#), auf Seite 28.

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](#)“, auf Seite 12.
- 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 `/lxv* file_name` für Diagnosezwecke.

Vorgehensweise

- 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 Führen Sie den im Folgenden dargestellten Befehl zur Installation oder Aktualisierung mit Standardwerten aus. 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 SERVICE-
PASSWORD=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 20 Zeichen nicht überschreiten. 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
SERVERHOST=Hostname	IP-Adresse oder Hostname der virtuellen vRealize Log Insight-Appliance.
SERVERPROTO=protocol=Protokoll	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=Portnummer	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.
SERVICEACCOUNT=Kontoname	Benutzerdienstkonto, unter dem der Log Insight Windows Agent-Dienst ausgeführt wird. HINWEIS Das im Parameter SERVICEACCOUNT angegebene Konto muss über die Berechtigung Anmelden als Dienst und über Schreibzugriff für das Verzeichnis %ProgramData%\VMware\Log Insight Agent verfügen, damit das Installationsprogramm ordnungsgemäß ausgeführt wird. Wenn Sie keinen SERVICEACCOUNT-Parameter angeben, wird der Windows-Agentdienst für vRealize Log Insight unter dem LocalSystem-Dienstkonto installiert.
SERVICEPASSWORD=Kennwort	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).

Die Befehlszeilenoptionen entsprechen `hostname`, `proto` und `port` im Abschnitt `[server]` der Datei `li-agent.ini`.

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.

Weiter

Stellen Sie sicher, dass die von Ihnen festgelegten Befehlszeilenparameter in der Datei `liagent.ini` ordnungsgemäß angewendet werden. Siehe [„Konfigurieren des Log Insight Windows Agent nach der Installation“](#), auf Seite 28.

Bereitstellen des Log Insight Windows Agent auf mehreren Computern

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

- 1 [Erstellen einer Umwandlungsdatei für die Bereitstellung mehrerer vRealize Log Insight-Windows-Agenten](#) auf Seite 16

Zum Festlegen von Installationsparametern, die während der Bereitstellung verwendet werden, erstellen Sie eine `.mst`-Umwandlungsdatei. Sie können den vRealize Log Insight Windows-Agenten so konfigurieren, dass Ereignisse an einen vRealize Log Insight-Server gesendet werden. Darüber hinaus können Sie das Kommunikationsprotokoll, den Port und das Benutzerkonto für die Installation und den Start des Log Insight Agent-Diensts festlegen.

- 2 [Bereitstellen von mehreren Instanzen des vRealize Log Insight Windows-Agenten](#) auf Seite 17

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

Zum Festlegen von Installationsparametern, die während der Bereitstellung verwendet werden, erstellen Sie eine .mst-Umwandlungsdatei. Sie können den vRealize Log Insight Windows-Agenten so konfigurieren, dass Ereignisse an einen vRealize Log Insight-Server gesendet werden. Darüber hinaus können Sie das Kommunikationsprotokoll, den Port und das Benutzerkonto für die Installation und den Start des Log Insight Agent-Diensts festlegen.

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](#)“, auf Seite 12.
- Laden Sie den Orca-Datenbankeditor herunter und installieren Sie ihn. Siehe <http://support.microsoft.com/kb/255905>.

Vorgehensweise

- 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.
 - a Klicken Sie auf **Eigenschaft**, um die Eigenschaftstabelle zu öffnen.
 - 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 gültigen 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.
SERVICEACCOUNT	Benutzerdienstkonto, unter dem der Log Insight Windows Agent-Dienst ausgeführt wird. HINWEIS Das im Parameter SERVICEACCOUNT angegebene Konto muss über die Berechtigung Anmelden als Dienst und über Schreibzugriff für das Verzeichnis %ProgramData%\VMware\Log Insight Agent verfügen, damit das Installationsprogramm ordnungsgemäß ausgeführt wird. Wenn Sie keinen SERVICEACCOUNT-Parameter angeben, wird der Windows-Agentdienst für vRealize Log Insight unter dem LocalSystem-Dienstkonto installiert.
SERVICEPASSWORD	Kennwort für das Benutzerdienstkonto.

Parameter	Beschreibung
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** und speichern Sie die .mst-Datei.

Weiter

Verwenden Sie die .msi-Datei und die .mst-Datei für die Bereitstellung des vRealize Log Insight Windows-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](#)“, auf Seite 12.
- Machen Sie sich mit den Verfahren in <http://support.microsoft.com/kb/887405> und <http://support.microsoft.com/kb/816102> vertraut.

Vorgehensweise

- 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.

Weiter

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“](#), auf Seite 86.

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 [„Optionen für vRealize Log Insight-Agenten“](#), auf Seite 23.
- 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.

Vorgehensweise

- 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

Wenn Sie das DEB-Paket des Linux-Agent für vRealize Log Insight installieren oder aktualisieren, können Sie den Zielsystem während der Installation festlegen und die Konfigurationsdatei „liagent.ini“ beibehalten oder ersetzen. Nach der Installation können Sie die installierte Version überprüfen.

Das Agenten-DEB-Paket kann über die Befehlszeile installiert oder aktualisiert und mit Paketoptionen geändert oder über die debconf-Datenbank konfiguriert werden. Die folgende Tabelle zeigt die unterstützten Optionen.

Sie können den Agenten auf zwei Arten installieren. Sie können während der Installation zusätzliche Parameter angeben.

Voraussetzungen

- Weitere Informationen zu den Standardeinstellungen der Installation und deren Änderung finden Sie unter „[Optionen für vRealize Log Insight-Agenten](#)“, auf Seite 23.
- 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.

Vorgehensweise

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

Ersetzen Sie *package_name* mit dem vRealize Log Insight-Namen **vmware-log-insight-agent-** und der entsprechenden Versions-Build-Nummer der Downloadversion. Dadurch wird das Paket mit allen Standardwerten installiert.

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

- 2 (Optional) Überprüfen Sie die installierte Version, indem Sie den Befehl `dpkg -l | grep -i vmware-log-insight-agent` ausführen.

Beispiel: Anpassen der Konfiguration

Beispiel: Debconf-Datenbank

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.

Eine vollständige Liste der Optionen finden Sie hier: „[Optionen für vRealize Log Insight-Agenten](#)“, auf Seite 23.

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

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

Um den Zielsever während der Installation festzulegen, führen Sie den Befehl „sudo“ aus, wobei Sie den Hostnamen 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 möchten. 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-Agenten 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 und Warnmeldungen werden angezeigt, weil der Benutzer `non_root_user2` keine Berechtigungen des Benutzers `non_root_user` hat.

vRealize Log Insight

Zusammen mit den Optionen der Umgebung kann das Agent-DEB-Paket 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.
<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> .
<code>LIAGENT_AUTOUPDATE={ja nein}</code>	<code>vmware-log-insight-agent/auto_update</code>	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). Die automatische Aktualisierung wird nicht für Linux-BIN-Pakete unterstützt.
<code>LI_AGENT_RUNSERVICES</code>	<code>vmware-log-insight-agent/init_system</code>	Sofort nach der Installation werden die Dienste <code>liagentd</code> (Agent) und <code>liupdaterd</code> (Updater) standardmäßig gestartet. Sie können dies verhindern, indem Sie den Parameter „LIAGENT_RUNSERVICES Debconf“ auf Nein festlegen. Die Standardeinstellung ist „yes“ (Ja). Beachten Sie, dass nur die Werte Ja und Nein gültige Werte sind. 1 oder 0 sind nicht unterstützte Werte.

Befehlszeilenoptionen	Debconf-Optionen	Beschreibung
LIAGENT_SSL	vmware-log-insight-agent/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).
LIAGENTUSER= <i>Benutzername-Konto</i>	vmware-log-insight-agent/liagentuser	<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>Beachten Sie, dass beim Installieren des Linux-Agenten für vRealize Log Insight mit dem Parameter LIAGENTUSER=<i>non_root_user</i> und dem Versuch, ein Upgrade mit LIAGENTUSER=<i>non_root_user2</i> auszuführen, ein Konflikt auftritt und Warnmeldungen angezeigt werden, weil <i>non_root_user2</i> keine Berechtigungen für <i>non_root_user</i> hat.</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 `li-agent.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 [„Deinstallieren des Log Insight Linux Agent-BIN-Pakets“](#), auf Seite 78.

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.

Vorgehensweise

- 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.

Optionen für vRealize Log Insight-Agenten

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`.

Gemeinsame Optionen für einen Linux-Agent

Die im Folgenden aufgeführten Optionen lassen sich bei der Installation zur Konfiguration von vRealize Log Insight-Linux-Agenten verwenden.

Tabelle 2-1. Installationsoptionen für Linux-Agenten

Option	Beschreibung
SERVERHOST= <i>Hostname</i>	IP-Adresse oder Hostname der virtuellen vRealize Log Insight-Appliance. Die Standardeinstellung lautet loginsight .
SERVERPROTO={ cfapi syslog }	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= <i>Portnummer</i>	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.
LIAGENT_INITSYSTEM={ init systemd }	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: init und systemd .
LIAGENT_AUTOUPDATE={ Ja Nein }	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). Die automatische Aktualisierung wird nicht für Linux-BIN-Pakete unterstützt.
LIAGENT_SSL={ Ja Nein }	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).
LIAGENTUSER= <i>Benutzername-Konto</i>	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. Standardmäßig wird der Agent installiert, um als Root-Benutzer ausgeführt zu werden. Beachten Sie, dass beim Installieren des Linux-Agenten für vRealize Log Insight mit dem Parameter LIAGENTUSER= <i>non_root_user</i> und dem Versuch, ein Upgrade mit LIAGENTUSER= <i>non_root_user2</i> auszuführen, ein Konflikt auftritt und Warnmeldungen angezeigt werden, weil <i>non_root_user2</i> keine Berechtigungen für <i>non_root_user</i> hat. 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.

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.

Vorgehensweise

- 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 Kommentieren Sie `auto_update=yes` aus, um die automatische Aktualisierung zu aktivieren, oder ändern Sie den Wert von „`auto_update`“ in „`no`“, um die automatische Aktualisierung zu deaktivieren.
- 4 Speichern und schließen Sie die Datei `liagent.ini`.

Konfigurieren eines vRealize Log Insight -Agenten

3

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

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

- [Konfigurieren des Log Insight Windows Agent nach der Installation](#) auf Seite 28

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](#) auf Seite 41

Nachdem Sie Log Insight Linux Agent installiert haben, können Sie ihn konfigurieren. Bearbeiten Sie die Datei `liagent.ini`, um den Agenten so zu konfigurieren, 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. Die Datei `liagent.ini` ist im Verzeichnis `/var/lib/loginsight-agent/` gespeichert.

- [Zentrale Konfiguration von vRealize Log Insight-Agenten](#) auf Seite 49

Sie können mehrere vRealize Log Insight-Agenten (Windows oder Linux) konfigurieren.

- [Verwenden von allgemeinen Werten für die Konfiguration von Agenten](#) auf Seite 51

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](#) auf Seite 52

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 nach der Installation

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.

Für die endgültige Konfiguration wird diese Datei mit den serverseitigen Einstellungen verknüpft und in der Datei `liagent-effective.ini` gespeichert.

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

; 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
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 .
hostname	LOGINSIGHT	IP-Adresse oder Hostname der virtuellen vRealize Log Insight-Appliance. Die Standardeinstellung lautet loginsight .
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.

Parameter	Standardwert	Beschreibung
ssl	Ja	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.
max_disk_buffer	200	Die maximale Festplattengröße in MB, die der Log Insight Windows Agent zum Puffern von Ereignissen und seinen eigenen Protokollen verwendet. Wenn der angegebene max_disk_buffer erreicht ist, beginnt der Agent, neue eingehende Ereignisse zu löschen.
debug_level	0	Legt die Protokolldateiebene fest. Weitere Informationen hierzu finden Sie unter „Festlegen der Protokolldateiebene in den Log Insight Agents“ , auf Seite 82.
channel	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“ , auf Seite 33.
raw_syslog	Nein	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 benutzer-spezifischen Syslog-Attributen umgewandelt. Aktivieren Sie diese Option, um Ereignisse ohne Syslog-Umwandlungen zu erfassen. Gültige Werte sind „Ja“ oder 1 und „Nein“ oder 0.

Festlegen des vRealize Log Insight -Zielservers

Sie können den vRealize Log Insight-Zielservers festlegen oder ändern, der dem Windows-Agenten für vRealize Log Insight Ereignisse sendet, wenn Sie die Werte während der Installation nicht festgelegt haben. Sie können Ereignisse an ein oder mehrere Ziele senden.

Mehrere Ziel-Verbindungen werden über den [server|<dest_id>]-Abschnitt der Datei `li_agent.ini` definiert, auf dem <dest_id> eine eindeutige Verbindungs-ID pro Konfiguration ist. Sie können die gleichen Optionen für zusätzliche Ziele wie für den Abschnitt für den standardmäßigen [server]-Bereich verwenden. Zusätzliche Ziele sollten jedoch nicht für die automatische Aktualisierung konfiguriert werden, und die Zielservers können nicht für Agent-Konfigurationen verwendet werden. Sie können zwei zusätzliche Ziele angeben.

Wenn Sie zusätzliche [Server] Abschnitte definieren, müssen Sie einen Hostnamen angeben. Standardmäßig sendet der Agent alle erfassten Ereignisse an alle Ziele. Sie können Ereignisse filtern, um andere Ereignisse an unterschiedliche Ziele zu senden.

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.

Vorgehensweise

- 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 cfapi und syslog. Der Standardwert ist cfapi.
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: hostname = 2001:cdba::3257:9652 or hostname = [2001:cdba::3257:9652] Wenn der Host sowohl IPv4- als auch IPv6-Stacks unterstützt und ein Domänenname als Hostname angegeben ist, dann verwendet der Agent den IP-Stack abhängig von der IP-Adresse, die von dem Namensauflöser zurückgegeben wird. Wenn der Auflöser sowohl IPv4- als auch IPv6-Adressen zurückgibt, dann versucht der Agent, sequenziell in der angegebenen Reihenfolge eine Verbindung zu beiden Adressen herzustellen.
port	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.
ssl	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.
reconnect	Das Zeitintervall in Minuten, in dem der erneute Verbindungsaufbau zum Server erzwungen wird. Der Standardwert lautet 30.

[server]

hostname=LOGINSIGHT

; Hostname or IP address of your Log Insight server / cluster load balancer. Default:

;hostname=LOGINSIGHT

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

;proto=cfapi

; Log Insight server port to connect to. Default ports for protocols (all TCP):

; syslog: 514; syslog with ssl: 6514; cfapi: 9000; cfapi with ssl: 9543. Default:

```
;port=9543
```

```
; SSL usage. Default:
```

```
;ssl=yes
```

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

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

```

Weiter

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

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 Feldnamen sind reserviert und können nicht als Feldnamen verwendet werden.

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.

Vorgehensweise

- 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
[winlog section_name]	Ein eindeutiger Name für den Konfigurationsabschnitt.
channel	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.
enabled	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“.

Parameter	Beschreibung
tags	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. Die Groß- und Kleinschreibung wird bei Tagnamen 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. Tags können das APP-NAME-Feld überschreiben, wenn das Ziel ein Syslog-Server ist. Beispiel: <code>tags={"appname":"VROPS"}</code> .
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 in die <code>blacklist</code> aufzunehmen.
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 in die `blacklist` aufzunehmen.

- `whitelist` erfasst nur Protokollereignisse, für welche die Auswertung des Filterausdrucks ungleich null ist. Wenn Sie `whitelist` auslassen, ist der Wert eine implizierte 1.
- `blacklist` schließt Protokollereignisse aus, bei denen 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](#)“, auf Seite 35.

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.

Vorgehensweise

- 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
<code>==, !=</code>	gleich und ungleich. Sowohl mit Zahlen- als auch mit Zeichenfolgenfeldern verwenden.
<code>>=, >, <, <=</code>	größer oder gleich, größer als, kleiner als, kleiner oder gleich. Nur mit Zahlenfeldern verwenden.
<code>&, , ^, ~</code>	Bitweise AND, OR, XOR und ergänzende Operanden. Nur mit Zahlenfeldern verwenden.

Operator	Beschreibung
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
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.

Erfassen aus verschlüsselten Ordnern

Ein Agent kann Daten 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.

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

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.

Vorgehensweise

- 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 Konfigurationsparameter hinzu und legen Sie die Werte für Ihre Umgebung fest.

Parameter	Beschreibung
[filelog section_name]	Ein eindeutiger Name für den Konfigurationsabschnitt.
directory	<p>Der vollständige Pfad zum Protokolldateiverzeichnis. Glob-Muster werden unterstützt.</p> <p>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 genau dieselben Konfigurationen für diese Bereiche verwenden, werden duplizierte Ereignisse auf der Serverseite beobachtet.</p>

Parameter	Beschreibung
include	<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 *. Das heißt, dass alle Dateien berücksichtigt werden. Bei dem Parameter muss die Groß-/Kleinschreibung beachtet werden.</p> <p>HINWEIS .zip- und .gz-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> <p><code>include= myapp.log;myapp.log.*</code></p>
exclude	<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>
event_marker	<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 (" "), 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>
enabled	<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>
charset	<p>(Optional) Die Zeichenkodierung der Protokolldateien, die der Agent überwacht. Die möglichen Werte lauten UTF-8, UTF-16LE und UTF-16BE. Der Standardwert lautet UTF-8.</p>
tags	<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. Die Groß- und Kleinschreibung wird bei Tagnamen 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>Tags können das APP-NAME-Feld überschreiben, wenn das Ziel ein Syslog-Server ist. Beispiel: <code>tags={"appname":"VROPS"}</code>.</p>

Parameter	Beschreibung
exclude_fields	(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: <ul style="list-style-type: none"> ■ exclude_fields=hostname; filepath ■ exclude_fields=type; size ■ exclude_fields=type, size
raw_syslog	Für Agenten, die das Syslog-Protokoll verwenden, ermöglicht der Agent, 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.

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

Beispiel: Konfigurationen

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

[filelog|ApacheAccessLogs]
enabled=yes
directory=C:\Program Files (x86)\Apache Software Foundation\Apache2.2\logs
include=*.log
exclude=*_old.log
tags={"Provider" : "Apache"}

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

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 in die `blacklist` aufzunehmen.

- `whitelist` erfasst nur Protokollereignisse, für welche die Auswertung des Filterausdrucks ungleich null ist. Wenn Sie `whitelist`, auslassen, ist der Wert eine implizierte 1.
- `blacklist` schließt Protokollereignisse aus, bei denen 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](#)“, auf Seite 35.

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.

Vorgehensweise

- 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“, auf Seite 33.

Vorgehensweise

- 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.

Weiter

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. Bearbeiten Sie die Datei `liagent.ini`, um den Agenten so zu konfigurieren, 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. Die Datei `liagent.ini` ist im Verzeichnis `/var/lib/loginsight-agent/` gespeichert.

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.

Für die endgültige Konfiguration wird diese Datei mit den serverseitigen Einstellungen zusammengeführt und in der Datei „`liagent-effective.ini`“ gespeichert.

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

; 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 per-
```

```

formance).
; 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	Wert	Beschreibung
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> .
hostname	LOGINSIGHT	IP-Adresse oder Hostname der virtuellen vRealize Log Insight-Appliance. Die Standardeinstellung lautet loginsight .
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 <code>cfapi</code> mit aktiviertem SSL, 9000 für <code>cfapi</code> mit deaktiviertem SSL, 6514 für <code>syslog</code> mit aktiviertem SSL und 514 für <code>syslog</code> mit deaktiviertem SSL.
ssl	Ja	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.
max_disk_buffer	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.
debug_level	0	Legt die Protokolldateiebene fest. Weitere Informationen hierzu finden Sie unter „Festlegen der Protokolldateiebene in den Log Insight Agents“ , auf Seite 82.

Festlegen des vRealize Log Insight -Zielservers

Sie können den vRealize Log Insight-Zielserver, an den der Linux-Agent für vRealize Log Insight Ereignisse sendet, festlegen oder ändern. Sie können Ereignisse an ein oder mehrere Ziele senden.

Mehrere Ziel-Verbindungen werden über den `[server|<dest_id>]`-Abschnitt der Datei `li_agent.ini` definiert, auf dem `<dest_id>` eine eindeutige Verbindungs-ID pro Konfiguration ist. Sie können die gleichen Optionen für zusätzliche Ziele wie für den Abschnitt für den standardmäßigen `[server]`-Bereich verwenden. Zusätzliche Ziele sollten jedoch nicht für die automatische Aktualisierung konfiguriert werden, und die Zielservers können nicht für Agent-Konfigurationen verwendet werden. Sie können zwei zusätzliche Ziele angeben.

Wenn Sie zusätzliche `[Server]` Abschnitte definieren, müssen Sie einen Hostnamen angeben. Standardmäßig sendet der Agent alle erfassten Ereignisse an alle Ziele. Sie können Ereignisse filtern, um andere 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.

Vorgehensweise

- 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	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</pre> <pre>or</pre> <pre>hostname = [2001:cdba::3257:9652]</pre> Wenn der Host sowohl IPv4- als auch IPv6-Stacks unterstützt und ein Domänenname als Hostname angegeben ist, dann verwendet der Agent den IP-Stack abhängig von der IP-Adresse, die von dem Namensauflöser zurückgegeben wird. Wenn der Auflöser sowohl IPv4- als auch IPv6-Adressen zurückgibt, dann versucht der Agent, sequenziell in der angegebenen Reihenfolge eine Verbindung zu beiden Adressen herzustellen.
port	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.

Parameter	Beschreibung
ssl	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.
reconnect	Das Zeitintervall in Minuten, in dem der erneute Verbindungsaufbau zum Server erzwungen wird. Der Standardwert lautet 30.

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

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

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

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

3 Speichern und schließen Sie die Datei liagent.ini.

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
```

Weiter

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](#)

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.

HINWEIS 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.

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

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.

Vorgehensweise

- 1 Öffnen Sie die Datei `/var/lib/loginsight-agent/liagent.ini` in einem beliebigen Texteditor.
- 2 Fügen Sie Konfigurationsparameter hinzu und legen Sie die Werte für Ihre Umgebung fest.

Parameter	Beschreibung
[filelog section_name]	Ein eindeutiger Name für den Konfigurationsabschnitt.
directory	<p>Der vollständige Pfad zum Protokolldateiverzeichnis. Glob-Muster werden unterstützt.</p> <p>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 genau dieselben Konfigurationen für diese Bereiche verwenden, werden duplizierte Ereignisse auf der Serverseite beobachtet.</p>
include	<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 dem Parameter muss die Groß-/Kleinschreibung beachtet werden.</p> <p>HINWEIS <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> <p><code>include= myapp.log;myapp.log.*</code></p>
exclude	<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>
event_marker	<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>), 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>
enabled	<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>
charset	<p>(Optional) Die Zeichenkodierung der Protokolldateien, die der Agent überwacht. Die möglichen Werte lauten <code>UTF-8</code>, <code>UTF-16LE</code> und <code>UTF-16BE</code>. Der Standardwert lautet <code>UTF-8</code>.</p>

Parameter	Beschreibung
tags	<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. Die Groß- und Kleinschreibung wird bei Tagnamen 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>Tags können das APP-NAME-Feld überschreiben, wenn das Ziel ein Syslog-Server ist. Beispiel: <code>tags={"appname":"VROPS"}</code>.</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"> ■ <code>exclude_fields=hostname; filepath</code> ■ <code>exclude_fields=type; size</code> ■ <code>exclude_fields=type, size</code>
raw_syslog	<p>Für Agenten, die das Syslog-Protokoll verwenden, ermöglicht der Agent, 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>

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

- 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=*
```

Einrichten der Linux-Protokolldateikanal-Filterung

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

HINWEIS 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.

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 in die `blacklist` aufzunehmen.

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

Eine umfassende Liste der Linux-Ereignisfelder und -Operatoren finden Sie unter [„Erfassen von Ereignissen aus einer Protokolldatei“](#), auf Seite 45.

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.

Vorgehensweise

- 1 Öffnen Sie die Datei `/var/lib/loginsight-agent/liagent.ini` in einem beliebigen Texteditor.
- 2 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
```

- 3 Erstellen Sie einen Filterausdruck aus Linux-Ereignisfeldern und -Operatoren.

Beispiel:

```
whitelist = server_name
```

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

Beispiel: Filterkonfigurationen

Sie können den Agent so konfigurieren, dass nur Apache-Protokolle erfasst werden, bei denen der „server_name“ `sample.com` lautet und `remote_host` nicht `127.0.0.1` ist, z. B.

```
[filelog|apache]
directory=/var/log/httpd
include=access_log
parser=clf
whitelist = server_name == "sample.com"
blacklist = remote_host == "127.0.0.1"
```


Zentrale Konfiguration von vRealize Log Insight -Agenten

Sie können mehrere vRealize Log Insight-Agenten (Windows oder Linux) 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 Maschine gespeichert, auf der der vRealize Log Insight-Agent installiert ist. Die serverseitige Konfiguration ist zugänglich und editierbar, z. B. unter Windows 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.

Die 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. Unter Windows können Sie 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 verbundenen Agents angewendet.

HINWEIS Die zentrale Konfiguration kann nur auf vRealize Log Insight-Agenten angewendet werden, die das cfapi-Protokoll verwenden.

Weitere Informationen hierzu finden Sie unter [„Konfigurieren des Log Insight Windows Agent nach der Installation“](#), auf Seite 28.

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 Systemereigniskanaln fortfahren
- die Erfassung des Sicherheitsereigniskanalns stoppen
- mit der Erfassung des Microsoft-Windows-DeviceSetupManager-Ereigniskanalns/operativen Ereigniskanalns 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.

Vorgehensweise

- 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.

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

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

Häufige 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>	Verschachtelte Parser werden als JSON-Zeichenfolge angegeben, in denen die Schlüssel die Namen des Felds sind, 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. Felddecodierer sind nützlich, wenn der Wert eines Feldes ein komplexer Wert ist, z. B. ein Zeitstempel.
<code>field_rename</code>	Benennt extrahierte Felder um. Dies ist 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.

Häufige Option	Beschreibung
next_parser	<p>Name des nächsten auszuführenden Parsers. Ermöglicht die Ausführung von mehreren Parnern sequenziell für die gleiche Eingabe.</p> <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>
exclude_fields	<p>Eine Liste der mit Semikolons getrennten Feldnamen, die aus dem Ereignis entfernt werden sollen, bevor es an den Server geliefert wird. Dies wird vor der Ereignisfilterung angewandt, damit das Feld, das Sie beim Analysieren ausgeschlossen haben, nicht in der Filterbedingung verwendet werden kann.</p>
debug	<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>

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

Sie können Parser für das kommasetrennte 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 kommasetrennte Format (CSV-Parser)

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

Option	Beschreibung
fields	<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>
delimiter	<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 Escapezeichen 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=myparser

[parser|myparser]
base_parser = csv
fields = timestamp,field_name1, field_name2, field_name3
delimiter = ";"
field_decoder={"timestamp": "tsp_parser"}
[parser|tsp_parser]
; timestamp is a built-in parser
base_parser=timestamp
; "format" is an option of timestamp parser
format=%Y-%m-%d %H:%M:%S
```


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. Analysierte 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

Felder	Wert
'%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"
'%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 Formate 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 Formate 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: %[%Y-%m-%d %H:%M:%S]t überschrieben werden. Weitere Informationen finden Sie unter „Timestamp-Parser“, auf Seite 68.</p> <p>Das Format des Zeitstempels für den CLF-Parser kann mit "begin:"- oder "end:"-Präfixen beginnen. Wenn das Format mit begin: beginnt (Standard), wird die Zeit zu Beginn der Anforderungsverarbeitung erfasst. Wenn es mit end: 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: %h %l %u [%{begin:%d/%b/%Y %T}t.%{msec_frac}t] \"%r\" %s %b Die folgenden Format-Token werden ebenfalls für die Formatspezifizierung des Zeitstempels des CLF-Parsers unterstützt:</p> <p>sec Anzahl der Sekunden seit der Epoche. Dies entspricht dem Spezifizierer %s des Zeitstempel-Parsers.</p> <p>msec Anzahl der Millisekunden seit der Epoche</p> <p>usec Anzahl der Mikrosekunden seit der Epoche</p> <p>msec_frac Millisekundenbruchteile (entspricht dem Spezifizierer %f des Zeitstempel-Parsers)</p> <p>musec Mikrosekundenbruchteile (entspricht dem Spezifizierer %f 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 %{format}t-Token verwenden. Um beispielsweise einen Zeitstempel mit Millisekunden zu verwenden, ohne den Spezifizierer %f des Zeitstempel-Parsers zu verwenden, kann der folgende kombinierte Zeitstempel verwendet werden: %[%d/%b/%Y %T}t.%{msec_frac}t .</p>
'%T':	"request_time_sec"
'%u':	"remote_auth_user"
'%U':	"requested_url"
'%v':	"server_name"
'%V':	"self_referential_server_name"
'%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|clflogs]
directory=D:\Logs
include=*.txt
parser=myclf
```

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

Bei Verwendung dieser Konfiguration werden Protokolle, die aus der `cllogs`-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
<code>%{u}t</code>	Aktuelle Zeit einschließlich Mikrosekunden	<code>format=[%{u}t] [%l] [pid %P] [client %a] %M</code>
<code>%{cu}t</code>	Aktuelle Zeit im kompakten ISO 8601-Format, einschließlich Mikrosekunden	<code>format=[%{cu}t] [%l] [pid %P] [client %a] %M</code>

Eine vollständige Liste der unterstützten Zeitstempelbezeichner finden Sie unter „[Timestamp-Parser](#)“, auf Seite 68.

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|cllogs-access]
    directory=C:\xampp\apache\logs
    include=acc*
```

```

parser=clfparser_apache_access
enabled=yes

```

```

;Parser to parse Apache ACCESS logs
[parser|clfparser_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=clfparser_apache_access
    enabled=yes
; Parser to parse Apache ACCESS logs
[parser|clfparser_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 %t] %E: %M
;format=[%a %b %d %H:%M:%S%f %Y}t] [%m:%{severity}i] [pid %P] %E: %M

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

```

```
;Parser to parse Apache ERROR logs
[parser|clfparsed_apache_error]
  debug=yes
  base_parser=clf
  format=[%{a %b %d %H:%M:%S%f %Y}t] [%m:%{severity}i] [pid %P:tid %{thread_id}i] %E: %M
  next_parser=clfparsed_apache_error2

;Parser to parse Apache ERROR logs
```

```
[parser|clfparsed_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 \"%{Referer}i\" \"%{User-Agent}i\"" combined
```

- 2 Definieren Sie die CLF-Parser-Konfiguration:

```
;Parser to parse Apache ERROR logs
[parser|clfparsed_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=clfparsed_error2
```

```
;Parser to parse Apache ERROR logs
[parser|clfparsed_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 Protokoll-nachrichtentext. 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
fields	<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>
delimiter	<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>
field_decoder	<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>
debug =	<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
<code>KVP_MESSAGE = *(MESSAGE_ENTRY [WSPR])</code>	Eine Liste mit Nachrichteneinträgen, optional getrennt durch Leerzeichen
<code>MESSAGE_ENTRY = KVP / FREE_TEXT</code>	Ein Eintrag ist ein Schlüssel/Wert-Paar oder nur ein Freitext
<code>KVP = KEY ["=" VALUE]</code>	Schlüssel/Wert-Paar. Wenn auf KEY kein Gleichheitszeichen und VALUE folgen, wird er wie Freitext übersprungen.
<code>KEY = BARE_KEY / QUOTED_KEY</code>	
<code>FREE_TEXT = "="</code>	Ein frei stehendes Gleichheitszeichen wird als Freitext betrachtet und übersprungen.
<code>BARE_KEY = *1BARE_KEY_CHAR</code>	Mindestens ein Zeichen

Schlüssel	Definition
BARE_KEY_CHAR = %0x00-08 / %0x10-19 / %0x21-3C / %3E-%FF	Ein beliebiges Zeichen mit Ausnahme von Gleichheitszeichen, Leerzeichen oder Tabstopp
QUOTED_KEY = 0x22 *(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 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 [exampleS-DID@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"
eventsource="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 Beispielergebnis 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-Beispielergebnis 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*byte
field-value = *fbyte
```

```
TAB = %x09
```

```
NL = [%x0D] %x0A
```

```
byte = %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 Bezeichnungsnamen nur aus alphanumerischen Zeichen, dem Unterstrich („_“), dem Punkt („.“) und dem Bindestrich („-“) bestehen. Falls im Protokoll mindestens ein ungültiger Bezeichnungsnamen vorhanden ist, schlägt die Analyse fehl. Auch wenn der Bezeichnungsnamen gültig ist, wird der Agent den Feldnamen überprüfen. Falls ungültige Namen vorhanden sind, sollte der Bezeichnungsnamen 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.

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>.*)\[(?<log_time-
stamp>.*)\] "(?<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>.*)\[(?<log_time-
stamp>.*)\] "(?<request>.*(?<resource>.*)(?<protocol>.*))" (?<status_code>.*)(?<respon-
se_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>.*) \[(?<log_time-stamp>.*)\] "(?<request>.*)" (?<status_code>\d+) (?<response_size>\d+)
```

Deinstallieren von vRealize Log Insight -Agenten

4

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

Dieses Kapitel behandelt die folgenden Themen:

- „Deinstallieren von Log Insight Windows Agent“, auf Seite 77
- „Deinstallieren des Log Insight Linux Agent-RPM-Pakets“, auf Seite 77
- „Deinstallieren des Log Insight Linux Agent-DEB-Pakets“, auf Seite 78
- „Deinstallieren des Log Insight Linux Agent-BIN-Pakets“, auf Seite 78

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.

Vorgehensweise

- 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**.

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.

Vorgehensweise

- ◆ 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
```

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.

Vorgehensweise

- ◆ Führen Sie den folgenden Befehl aus:

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

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

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

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.

Vorgehensweise

- 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

5

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 behandelt die folgenden Themen:

- „Erstellen eines Support-Pakets für den Log Insight Windows Agent“, auf Seite 81
- „Erstellen eines Support-Pakets für den Log Insight Linux Agent“, auf Seite 82
- „Festlegen der Protokolldateiebene in den Log Insight Agents“, auf Seite 82
- „Keine Anzeige von Log Insight Agents auf der Benutzeroberfläche für Administratoren“, auf Seite 83
- „vRealize Log Insight Agents senden keine Ereignisse“, auf Seite 84
- „Hinzufügen einer Ausnahmeregel für den Ausgang für den Log Insight Windows Agent“, auf Seite 85
- „Zulassen von ausgehenden Verbindungen vom Log Insight Windows Agent in einer Windows-Firewall“, auf Seite 86
- „Die Massenbereitstellung des Log Insight Windows Agent ist nicht erfolgreich“, auf Seite 86
- „Ablehnen von selbstsignierten Zertifikaten durch Log Insight Agents“, auf Seite 87
- „Der vRealize Log Insight-Server lehnt die Verbindung für nicht verschlüsselten Datenverkehr ab“, auf Seite 88

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.

Vorgehensweise

- 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`.

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

Weiter

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.

Vorgehensweise

- 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
```

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

Weiter

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.

Vorgehensweise

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

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

- 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.

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](#)“, auf Seite 30 und „[Festlegen des vRealize Log Insight-Zielservers](#)“, auf Seite 43.

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 3, „Konfigurieren eines vRealize Log Insight-Agenten“](#), auf Seite 27.
- 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/

Weiter

Weitere Informationen über das Konfigurieren der vRealize Log Insight Agents finden Sie unter „[Konfigurieren des Log Insight Windows Agent nach der Installation](#)“, auf Seite 28 und „[Konfigurieren des Log Insight Linux Agent](#)“, auf Seite 41.

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.

Vorgehensweise

- 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.

Weiter

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.

Vorgehensweise

- 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**.

Weiter

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“](#), auf Seite 85.

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](#)“, auf Seite 82.

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 Clusternumgebung 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.

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

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.

Vorgehensweise

- 1 Konfigurieren Sie Ihren vRealize Log Insight-Server so, dass Nicht-SSL-Verbindungen akzeptiert werden.
 - a Klicken Sie auf das Symbol für das Konfigurations-Drop-down-Menü und wählen Sie **Administration** 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`.

Index

A

- Agent, mit Parametern installieren 13
- Agent deinstallieren 77
- Agent konfigurieren 16
- Agent wird nicht angezeigt 83
- Agent-Dateien herunterladen 12
- Agent-Konfiguration, Standardwerte überschreiben 51
- Agent-StandardEinstellungen 28, 41
- Agent, automatische Aktualisierung 25
- Agenten
 - Deinstallieren 77
 - Installieren 11
 - konfigurieren 27
- Agenten installieren 11
- Agenten konfigurieren 27
- Agenten-Installationsoptionen 23
- Agentenseitige Parser 52
- Agentseitige Parser, konfigurieren 53
- Aktualisieren des Windows-Agenten 13
- ausgehende Verbindung 86
- Ausnahme für ausgehende Regel 85
- Automatische Aktualisierung, Einzelne Agenten 25
- automatischer Parser 70

B

- Beispiel für Agent-Konfiguration 49
- Bereitstellung an mehrere Maschinen 15
- Bereitstellung von mehreren Agenten 17

C

- centralized configuration 49
- CLF-Parser, Timestamp-Parser integrieren 57
- CSV-Parser 55

D

- Debugebene des Protokolls 82
- Debugebene des Protokolls ändern 82
- Deinstallieren von Agenten 77

E

- Ebene der Protokolldetails 82
- effektive Agent-Konfiguration 49

- Ereignisse, Von Windows-Ereigniskanal erfassen 33

- Ereignisse von Protokolldatei erfassen 37

- Ereignisweiterleitung, Ereignisse an Windows-Agent für Log Insight weiterleiten 40

- Erfassung flacher Dateien 37

F

- Falsche Agent-Konfiguration 84

- Fehlerbehebung für Linux-Agent für Log Insight 81

- Fehlerbehebung für Log Insight-Agenten 81

- Fehlerbehebung für Windows-Agent für Log Insight 81

- Fehlerbehebungs-Agent 83, 86

- Firewallausnahme hinzufügen 85

- Firewallverbindung zulassen 86

G

- Glossar 5

- Gruppenrichtlinienobjekt 17

H

- Häufige Optionen für Parser 54

K

- Konfiguration

- Linux-Agent 41

- Windows-Agent 28

- Konfiguration des Fehlerbehebungs-Agenten 84

- Konfiguration des Linux-Agenten 41

- Konfigurationen zusammenführen 49

L

- Linux agent, uninstall bin package 78

- Linux-Agent

- Bin-Paket installieren 22

- Deb-Paket deinstallieren 78

- Ereignisse von Protokolldatei erfassen 45

- Erfassung flacher Dateien 45

- Installieren des Debian-Pakets 19

- RPM-Paket deinstallieren 77

- RPM-Paket installieren 18

- Zielserver festlegen 43

- Linux-Agent für Log Insight 87

- Linux-Ereigniskanal, Filter hinzufügen 47

Log Insight Windows Agent **87**
Log Insight-Agenten **88**
LTSV-Parser **73**

M

Massenbereitstellung **15**
Massenbereitstellung fehlgeschlagen **86**
mehrere Upgrades von Agenten **17**
Mit Standardkonfiguration installieren **13**
multiple agents configuration **49**

P

Parser
 automatisch **70**
 CLF (Apache) **57**
 Schlüssel/Wert **65**
 syslog **70**
 Zeitstempel **68**
Protokollrotation, Agenten **7**

R

Regex-Parser **74**

S

Schlüssel/Wert-Paar-Parser **65**
Selbstsigniertes Zertifikat **87**
Selbstsigniertes Zertifikat ablehnen **87**
Server lehnt die Verbindung ab **88**
SSL-Verbindung **88**
Standardkonfiguration **28, 41**
Support-Paket **82**
Support-Paket für Linux-Agent **82**
Support-Paket für Windows-Agent, Support-Paket **81**
syslog-Parser **70**

T

Timestamp-Parser, automatisch **70**

U

unverschlüsselter Datenverkehr **88**

W

Windows event channel
 Ereignisfelder und Operatoren **35**
 Filter hinzufügen **34**
Windows events channel, hinzufügen **33**
Windows-Agent **13**
Windows-Agent-Konfiguration **28**
Windows-Ereignisse weiterleiten **40**
Windows-Protokolldatei, Filterung **39**

Z

Zielgruppe **5**
Zielserver festlegen **30**