データコレクターリファレンス - サービス

Data Infrastructure Insights

NetApp
February 11, 2026

目次

データコレクターリファレンス - サービス

ノードデータ収集

Data Infrastructure Insights は、エージェントがインストールされているノードからメトリックを収集します。

インストール

1. **Observability > Collectors** から、オペレーティング システム/プラットフォームを選択します。統合データ コレクター (Kubernetes、Docker、Apache など) をインストールすると、ノード データ収集も構成されることに注意してください。
2. 指示に従ってエージェントを構成します。手順は、データ収集に使用しているオペレーティング システムまたはプラットフォームの種類によって異なります。

オブジェクトとカウンター

次のオブジェクトとそのカウンターがノード メトリックとして収集されます。

オブジェクト	識別子:	属性:	データポイント:
ノードファイルシステム	ノードUUIDデバイスパス タイプ	ノードIP ノード名 ノードOSモード	空きiノード 空きiノード 使用済みiノード合計 使用済み合計 使用済み合計
ノードディスク	ノードUUIDディスク	ノードIP ノード名 ノードOS	IO時間 進行中の合計IOPS 読み取りバイト数（秒あたり） 読み取り時間 合計読み取り数（秒あたり） 加重IO時間 合計書き込みバイト数（秒あたり） 書き込み時間 合計書き込み数（秒あたり） 現在のディスクキューの長さ 書き込み時間 読み取り時間 IO時間
ノードCPU	ノードUUID CPU	ノードIP ノード名 ノードOS	システムCPU使用率 ユーザーCPU使用率 アイドルCPU使用率 プロセッサCPU使用率 割り込みCPU使用率 DPC CPU使用率

オブジェクト	識別子:	属性:	データポイント:
ノード	ノードUUID	ノードIP ノード名 ノードOS	カーネルのブート時間、カーネルのコンテキストスイッチ（1秒あたり）カーネルの使用可能なエントロピー、カーネル割り込み（1秒あたり）フォークされたカーネルプロセス（1秒あたり）メモリ、アクティブメモリ、使用可能メモリ、合計メモリ、使用可能メモリ、バッファメモリ、キャッシュメモリ、コミット制限、コミット済みメモリ、ダーティメモリ、空きメモリ上限、空きメモリ上限、合計メモリ、ヒュージページサイズ、メモリ、ヒュージページ、空きメモリ、ヒュージページ、合計メモリ下限、空きメモリ下限、合計メモリ、マップメモリ、ページテーブル、メモリ、共有メモリ、スラブメモリ、スワップ、キャッシュメモリ、スワップ空きメモリ、スワップ合計メモリ、使用済みメモリ合計、使用済みメモリ合計、メモリ、vmallocチャンクメモリ、vmalloc使用済みメモリ合計、vmallocメモリ、使用済みメモリ、メモリ、ワイヤードメモリ、ライトバック、合計メモリ、ライトバック一時メモリ、メモリキャッシュフォールト、メモリ需要、ゼロフォールト、メモリページフォールト、メモリページ、メモリ、非ページメモリ、ページメモリ、キャッシュコアメモリ、スタンバイキャッシュ、通常メモリ、スタンバイキャッシュ、予約メモリ、遷移フォールト、プロセス、ブロックされたプロセス、デッドプロセス、アイドルプロセス、ページングプロセス、実行中のプロセススリープ中のプロセ

オブジェクト	識別子:	属性:	データポイント:
ノードネットワーク	ネットワークインターフェースノードUUID	ノード名 ノードIP ノードOS	受信バイト数 送信バイト数 送信パケット数 破棄された送信パケット数 エラー数 受信パケット数 破棄された受信パケット数 エラー数 受信パケット数 送信パケット数

セットアップ

セットアップとトラブルシューティングに関する情報は、"エージェントの設定"ページ。

ActiveMQ データコレクター

Data Infrastructure Insights は、このデータ コレクターを使用して ActiveMQ からメトリックを収集します。

インストール

1. **Observability > Collectors** から、**+ Data Collector** をクリックします。 ActiveMQを選択します。

   Telegraf エージェントがインストールされているオペレーティング システムまたはプラットフォームを選択します。

2. 収集用のエージェントをまだインストールしていない場合、または別のオペレーティングシステムまたはプラットフォーム用のエージェントをインストールする場合は、[手順を表示] をクリックして展開します。"エージェントのインストール"説明書。
3. このデータ コレクターで使用するエージェント アクセス キーを選択します。 **+** エージェント アクセス キー ボタンをクリックすると、新しいエージェント アクセス キーを追加できます。ベスト プラクティス: データ コレクターを OS/プラットフォーム別などにグループ化する場合にのみ、異なるエージェント アクセス キーを使用します。
4. 構成手順に従ってデータ コレクターを構成します。手順は、データ収集に使用しているオペレーティング システムまたはプラットフォームの種類によって異なります。

[ActiveMQの設定]

セットアップ

詳細は以下をご覧ください。 "ActiveMQドキュメント"

オブジェクトとカウンター

次のオブジェクトとそのカウンターが収集されます。

オブジェクト	識別子:	属性:	データポイント:
ActiveMQ キュー	名前空間キューポートサーバー	ノード名 ノードIP ノードUUID	コンシューマー数 デキュー数 エンキュー数 キューサイズ
ActiveMQ サブスクライバー	クライアントID 接続ID ポート サーバー 名前空間	アクティブ宛先ノード名 ノードIPノードUUIDノードOSセレクタサブスクリプション	デキュー数、ディスパッチ数、ディスパッチキューサイズ、エンキュー数、保留キューサイズ
ActiveMQトピック	トピックポートサーバー 名前空間	ノード名 ノードIP ノードUUID ノードOS	コンシューマー数 デキュー数 エンキュー数 サイズ

トラブルシューティング

追加情報は以下からご覧いただけます。"サポート"ページ。

Apache データコレクター

このデータ コレクターを使用すると、テナント上の Apache サーバーからデータを収集できます。

前提条件

- Apache HTTP Serverがセットアップされ、適切に実行されている必要があります。
- エージェントホスト/VMに対するsudoまたは管理者権限が必要です
- 通常、Apache *mod_status* モジュールは、Apache サーバーの '/server-status?auto' の場所にページを公開するように構成されます。利用可能なすべてのフィールドを収集するには、*ExtendedStatus* オプションを有効にする必要があります。サーバーの設定方法については、Apache モジュールのドキュメントを参照してください。 https://httpd.apache.org/docs/2.4/mod/mod_status.html#enable

インストール

1. **Observability > Collectors** から、**+ Data Collector** をクリックします。 Apacheを選択します。

   Telegraf エージェントがインストールされているオペレーティング システムまたはプラットフォームを選択します。

2. 収集用のエージェントをまだインストールしていない場合、または別のオペレーティングシステムまたはプラットフォーム用のエージェントをインストールする場合は、[手順を表示] をクリックして展開します。"エージェントのインストール"説明書。
3. このデータ コレクターで使用するエージェント アクセス キーを選択します。 **+** エージェント アクセス キー ボタンをクリックすると、新しいエージェント アクセス キーを追加できます。ベスト プラクティス: データ コレクターを OS/プラットフォーム別などにグループ化する場合にのみ、異なるエージェント アクセス キーを使用します。
4. 構成手順に従ってデータ コレクターを構成します。手順は、データ収集に使用しているオペレーティング システムまたはプラットフォームの種類によって異なります。

[Apacheの設定]

セットアップ

Apache の HTTP サーバー用の Telegraf プラグインは、'mod_status' モジュールが有効になっていることを前提としています。これを有効にすると、Apache の HTTP サーバーは、ブラウザーで表示したり、Apache の HTTP サーバーのすべての構成のステータスを抽出するためにスクレイピングしたりできる HTML エンドポイントを公開します。

互換性:

構成は、Apache の HTTP サーバー バージョン 2.4.38 に対して開発されました。

mod_status を有効にする:

「mod_status」モジュールを有効にして公開するには、次の 2 つの手順が必要です。

- 有効化モジュール
- モジュールから統計を公開する

有効化モジュール:

モジュールの読み込みは、「/usr/local/apache/conf/httpd.conf」の下の設定ファイルによって制御されます。設定ファイルを編集し、次の行のコメントを解除します。

```
LoadModule status_module modules/mod_status.so
Include conf/extra/httpd-info.conf
```

モジュールから統計を公開する:

'mod_status' の公開は、'/usr/local/apache2/conf/extra/httpd-info.conf' の下の設定ファイルによって制御されます。構成ファイルに次の内容が含まれていることを確認してください (少なくとも、他のディレクティブは含まれている必要があります)。

```
# Allow server status reports generated by mod_status,
# with the URL of http://servername/server-status
<Location /server-status>
    SetHandler server-status
</Location>

#
# ExtendedStatus controls whether Apache will generate "full" status
# information (ExtendedStatus On) or just basic information
(ExtendedStatus
# Off) when the "server-status" handler is called. The default is Off.
#
ExtendedStatus On
```

「mod_status」モジュールの詳細な手順については、"Apacheドキュメント"

オブジェクトとカウンター

次のオブジェクトとそのカウンターが収集されます。

オブジェクト	識別子:	属性:	データポイント:
アパッチ	名前空間サーバー	ノードIP ノード名 ポート 親サーバー構成生成 親サーバーMPM生成 サーバー稼働停止中	ビジーワーカー リクエストあたりのバイト数 1秒あたりのバイト数 CPU 子プロセス システム CPU 子プロセス ユーザープロセス CPU 負荷 システム CPU ユーザー 非同期接続 非同期接続のクローズ 維持 非同期接続 書き込み接続 リクエストあたりの合計継続時間 アイドルワーカー 平均負荷 (過去 1 分) 平均負荷 (過去 15 分) 平均負荷 (過去 5 分) プロセス 1 秒あたりのリクエスト数 合計アクセス数 合計継続時間 合計キロバイト数 スコアボード クローズ中 スコアボード DNS ルックアップ スコアボード 終了中 スコアボード アイドル クリーンアップ スコアボード 維持 スコアボード ログ記録中 スコアボード オープン中 スコアボード 読み取り中 スコアボード 送信中 スコアボード 開始中 スコアボード 待機中

トラブルシューティング

追加情報は以下からご覧いただけます。"サポート"ページ。

領事データコレクター

Data Infrastructure Insights は、このデータ コレクターを使用して Consul からメトリックを収集します。

インストール

1. **Observability > Collectors** から、**+ Data Collector** をクリックします。領事を選択してください。

収集用のエージェントを設定していない場合は、"エージェントをインストールする"テナントに。

エージェントがすでに構成されている場合は、適切なオペレーティング システムまたはプラットフォームを選択し、[続行] をクリックします。

2. Consul 構成画面の指示に従って、データ コレクターを構成します。手順は、データ収集に使用しているオペレーティング システムまたはプラットフォームの種類によって異なります。

セットアップ

詳細は以下をご覧ください。"領事文書" 。

領事館のオブジェクトとカウンター

次のオブジェクトとそのカウンターが収集されます。

オブジェクト	識別子:	属性:	データポイント:
領事	名前空間チェックIDサービスノード	ノードIP ノードOS ノードUUID ノード名 サービス名 チェック名 サービスID ステータス	クリティカルパス警告

トラブルシューティング

追加情報は以下からご覧いただけます。"サポート"ページ。

Couchbase データコレクター

Data Infrastructure Insights は、このデータ コレクターを使用して Couchbase からメトリックを収集します。

インストール

1. **Observability > Collectors** から、**+ Data Collector** をクリックします。 Couchbaseを選択します。

   Telegraf エージェントがインストールされているオペレーティング システムまたはプラットフォームを選択します。

2. 収集用のエージェントをまだインストールしていない場合、または別のオペレーティングシステムまたはプラットフォーム用のエージェントをインストールする場合は、[手順を表示] をクリックして展開します。"エージェントのインストール"説明書。
3. このデータ コレクターで使用するエージェント アクセス キーを選択します。 **+** エージェント アクセス キー ボタンをクリックすると、新しいエージェント アクセス キーを追加できます。ベスト プラクティス: データ コレクターを OS/プラットフォーム別などにグループ化する場合にのみ、異なるエージェント アクセス キーを使用します。
4. 構成手順に従ってデータ コレクターを構成します。手順は、データ収集に使用しているオペレーティング システムまたはプラットフォームの種類によって異なります。

[Couchbaseの設定]

セットアップ

詳細は以下をご覧ください。"Couchbaseのドキュメント" 。

オブジェクトとカウンター

次のオブジェクトとそのカウンターが収集されます。

オブジェクト	識別子:	属性:	データポイント:
カウチベースノード	名前空間 クラスタ Couchbase ノード ホスト名	ノード名 ノードIP	メモリ空き容量合計メモリ
カウチベースバケット	名前空間バケットクラスター	ノード名 ノードIP	使用データ データフェッチ 使用ディスク アイテム数 使用メモリ 1秒あたりの操作数 使用クォータ

トラブルシューティング

追加情報は以下からご覧いただけます。"サポート"ページ。

CouchDB データコレクター

Data Infrastructure Insights は、このデータ コレクターを使用して CouchDB からメトリックを収集します。

インストール

1. **Observability > Collectors** から、**+ Data Collector** をクリックします。 CouchDB を選択します。

   Telegraf エージェントがインストールされているオペレーティング システムまたはプラットフォームを選択します。

2. 収集用のエージェントをまだインストールしていない場合、または別のオペレーティングシステムまたはプラットフォーム用のエージェントをインストールする場合は、[手順を表示] をクリックして展開します。"エージェントのインストール"説明書。
3. このデータ コレクターで使用するエージェント アクセス キーを選択します。 **+** エージェント アクセス キー ボタンをクリックすると、新しいエージェント アクセス キーを追加できます。ベスト プラクティス: データ コレクターを OS/プラットフォーム別などにグループ化する場合にのみ、異なるエージェント アクセス キーを使用します。
4. 構成手順に従ってデータ コレクターを構成します。手順は、データ収集に使用しているオペレーティング システムまたはプラットフォームの種類によって異なります。

[CouchDBの設定]

セットアップ

詳細は以下をご覧ください。"CouchDBのドキュメント" 。

オブジェクトとカウンター

次のオブジェクトとそのカウンターが収集されます。

オブジェクト	識別子:	属性:	データポイント:
カウチDB	名前空間サーバー	ノード名 ノードIP	認証キャッシュヒット 認証キャッシュミス データベース読み取り データベース書き込み データベースオープン OS ファイル最大リクエスト時間 最小リクエスト時間 HTTP リクエストメソッド コピー HTTP リクエストメソッド 削除 HTTP リクエストメソッド 取得 HTTP リクエストメソッド ヘッド HTTP リクエストメソッド ポスト HTTP リクエストメソッド プット ステータスコード 200 ステータスコード 201 ステータスコード 202 ステータスコード 301 ステータスコード 304 ステータスコード 400 ステータスコード 401 ステータスコード 403 ステータスコード 404 ステータスコード 405 ステータスコード 409 ステータスコード 412 ステータスコード 500

トラブルシューティング

追加情報は以下からご覧いただけます。"サポート"ページ。

Docker データコレクター

Data Infrastructure Insights は、このデータ コレクターを使用して Docker からメトリックを収集します。

インストール

1. **Observability > Collectors** から、**+ Data Collector** をクリックします。 Dockerを選択します。

収集用のエージェントを設定していない場合は、"エージェントをインストールする"テナントに。

エージェントがすでに構成されている場合は、適切なオペレーティング システムまたはプラットフォームを選択し、[続行] をクリックします。

2. Docker 構成画面の指示に従って、データ コレクターを構成します。手順は、データ収集に使用しているオペレーティング システムまたはプラットフォームの種類によって異なります。

[Dockerの設定]

セットアップ

Docker 用の Telegraf 入力プラグインは、指定された UNIX ソケットまたは TCP エンドポイントを通じてメトリックを収集します。

互換性

構成は Docker バージョン 1.12.6 に対して開発されました。

セットアップ

UNIXソケット経由で**Docker**にアクセスする

Telegraf エージェントがベアメタル上で実行されている場合は、次のコマンドを実行して、telegraf Unix ユーザーを docker Unix グループに追加します。

```
 sudo usermod -aG docker telegraf
```

`Telegraf` エージェントが `Kubernetes` ポッド内で実行されている場合は、ソケットをボリュームとしてポッドにマッピングし、そのボリュームを `/var/run/docker.sock` にマウントすることで、`Docker Unix` ソケットを公開します。たとえば、`PodSpec` に以下を追加します。

```
volumes:
...
- name: docker-sock
hostPath:
path: /var/run/docker.sock
type: File
```

次に、コンテナーに次の内容を追加します。

```
volumeMounts:
...
- name: docker-sock
mountPath: /var/run/docker.sock
```

Kubernetes プラットフォーム用に提供されているData Infrastructure Insightsインストーラーがこのマッピングを自動的に処理することに注意してください。

TCPエンドポイント経由で**Docker**にアクセスする

デフォルトでは、Docker は暗号化されていないアクセスにポート 2375 を使用し、暗号化されたアクセスにポート 2376 を使用します。

オブジェクトとカウンター

次のオブジェクトとそのカウンターが収集されます。

オブジェクト	識別子:	属性:	データポイント:
Dockerエンジン	名前空間 Docker エンジン	ノード名 ノードIP ノードUUID ノードOS Kubernetes クラスター Docker バージョン ユニット	メモリ コンテナ 一時停止中のコンテナ 実行中のコンテナ 停止中のコンテナ CPU Go ルーチン イメージ リスナー 使用中のイベント ファイル記述子 使用可能なデータ 使用済みデータ合計 使用可能なメタデータ 使用可能なメタデータ 使用済みメタデータ合計 プール ブロックサイズ

オブジェクト	識別子:	属性:	データポイント:
Dockerコンテナ	名前空間 コンテナ名 Docker エンジン	Kubernetes コンテナ ハッシュ、Kubernetes コンテナ ポート、Kubernetes コンテナの再起動回数、Kubernetes コンテナの終了メッセージ パス、Kubernetes コンテナの終了メッセージ ポリシー、Kubernetes ポッドの終了猶予期間、コンテナ イメージ、コンテナの状態、コンテナのバージョン、ノード名、Kubernetes コンテナのログ パス、Kubernetes コンテナ名、Kubernetes Docker タイプ、Kubernetes ポッド名、Kubernetes ポッド名前空間、Kubernetes ポッド UID、Kubernetes サンドボックス ID、ノード IP、ノード UUID、Docker バージョン、Kubernetes IO 構成の確認、Kubernetes IO 構成ソース、OpenShift IO SCC、Kubernetes の説明、Kubernetes 表示名、OpenShift タグ、Kompose サービス、ポッド テンプレート ハッシュ、コントローラー リビジョン ハッシュ、ポッド テンプレートの生成、ライセンス スキーマのビルド日、スキーマ ライセンス スキーマ名、スキーマ URL、スキーマ VCS URL、スキーマ ベンダー、スキーマ バージョン、スキーマ スキーマ バージョン、メンテナー、顧客ポッド、Kubernetes StatefulSet、ポッド名、テナント、Web コンソール、アーキテクチャ、権限のあるソース URL、ビルド日、RH ビルド ホスト、RH コンポーネント ディストリビューション、スコープ、インストール、リリース、実行の概要、アンインストー	メモリ アクティブ匿名メモリ アクティブファイルメモリ キャッシュメモリ階層制限 メモリ 非アクティブ匿名メモリ 非アクティブファイルメモリ制限 メモリ マップファイルメモリ 最大使用量 メモリ ページフォールトメモリ ページメジャーフォールトメモリ ページインメモリ ページアウトメモリ 常駐セットサイズ メモリ常駐セットサイズ 巨大メモリ合計 アクティブ匿名メモリ合計 アクティブファイルメモリ合計 キャッシュメモリ合計 非アクティブ匿名メモリ合計 非アクティブファイルメモリ合計 マップファイルメモリ合計 ページフォールトメモリ合計 ページメジャーフォールトメモリ合計 ページインメモリ合計 ページアウトメモリ合計 常駐セットサイズ メモリ合計 常駐セットサイズ巨大メモリ合計 排除不可能メモリ 排除不可能メモリ使用量 メモリ使用量の割合 終了コード OOM 強制終了 PID 開始時 失敗ストリーク

オブジェクト	識別子:	属性:	データポイント:
DockerコンテナブロックIO	名前空間 コンテナ名 デバイス Docker エンジン	Kubernetes コンテナ ハッシュ、Kubernetes コンテナ ポート、Kubernetes コンテナの再起動回数、Kubernetes コンテナの終了メッセージ パス、Kubernetes コンテナの終了メッセージ ポリシー、Kubernetes ポッドの終了猶予期間、コンテナ イメージ、コンテナの状態、コンテナのバージョン、ノード名、Kubernetes コンテナのログ パス、Kubernetes コンテナ名、Kubernetes Docker タイプ、Kubernetes ポッド名、Kubernetes ポッド名前空間、Kubernetes ポッド UID、Kubernetes サンドボックス ID、ノード IP、ノード UUID、Docker バージョン、Kubernetes 構成情報、Kubernetes 構成ソース、OpenShift SCC、Kubernetes の説明、Kubernetes 表示名、OpenShift タグ、スキーマ、スキーマ バージョン、ポッド テンプレート ハッシュ、コントローラー リビジョン ハッシュ、ポッド テンプレート生成、Kompose サービス スキーマ、ビルド日、スキーマ ライセンス、スキーマ名、スキーマ ベンダー、顧客ポッド、Kubernetes StatefulSet ポッド名、テナント Web コンソール ビルド日、ライセンス ベンダー、アーキテクチャ、信頼できるソース URL、RH ビルド ホスト、RH コンポーネント ディストリビューション スコープ、インストール、メンテナー、リリース、実行、概要、アンインストール、VCS 参照、VCS タイプ バージョ	IO サービス バイト数 (再帰非同期)、IO サービス バイト数 (再帰読み取り)、IO サービス バイト数 (再帰同期)、IO サービス バイト数 (再帰合計)、IO サービス バイト数 (再帰書き込み)、IO サービス数 (再帰非同期)、IO サービス数 (再帰読み取り)、IO サービス数 (再帰同期)、IO サービス数 (再帰合計)、IO サービス数 (再帰書き込み)

オブジェクト	識別子:	属性:	データポイント:
Dockerコンテナネットワーク	名前空間 コンテナ名 ネットワーク Docker エンジン	コンテナイメージ コンテナステータス コンテナバージョン ノード名 ノードIP ノードUUID ノードOS K8s クラスタ Dockerバージョン コンテナID	RXドロップ RXバイト RXエラー RXパケット TXドロップ TXバイト TXエラー TXパケット

オブジェクト	識別子:	属性:	データポイント:
DockerコンテナCPU	名前空間 コンテナ名 CPU Dockerエンジン	Kubernetes コンテナ ハッシュ、Kubernetes コンテナ ポート、Kubernetes コンテナの再起動回数、Kubernetes コンテナの終了メッセージ パス、Kubernetes コンテナの終了メッセージ ポリシー、Kubernetes ポッドの終了猶予期間、Kubernetes 構成の確認、Kubernetes 構成ソース、OpenShift SCC コンテナ イメージ、コンテナ ステータス、コンテナ バージョン、ノード名、Kubernetes コンテナ ログ パス、Kubernetes コンテナ名、Kubernetes Docker タイプ、Kubernetes ポッド名、Kubernetes ポッド名前空間、Kubernetes ポッド UID、Kubernetes サンドボックス ID、ノード IP、ノード UUID、ノード OS、Kubernetes クラスター Docker バージョン、Kubernetes の説明、Kubernetes 表示名、OpenShift タグ、スキーマ バージョン、ポッド テンプレート ハッシュ、コントローラー リビジョン ハッシュ、ポッド テンプレート生成、Kompose サービス スキーマ ビルド日、スキーマ ライセンス スキーマ名、スキーマ ベンダー、顧客ポッド、Kubernetes StatefulSet ポッド名、テナント Web コンソール ビルド日、ライセンス ベンダー、アーキテクチャ、権限のあるソース URL、RH ビルド ホスト、RH コンポーネント ディストリビューション スコープ、インストール、メンテナー、リリース、実行、サマリー、アンインストール、VCS 参照、VCS タイプ バージョ	スロットル期間 スロットル スロットル期間 スロットル スロットル時間 カーネルモードでの使用量 ユーザーモードでの使用量 使用量の割合 システム使用量 合計

トラブルシューティング

問題：	これを試してください:
構成ページの指示に従った後、 Data Infrastructure Insightsに Docker メトリックが表示されません。	Telegraf エージェント ログをチェックして、次のエラーが報告されているかどうかを確認します: E! プラグイン [inputs.docker] のエラー: Docker デーモン ソケットへの接続中に権限が拒否されました。その場合は、上記のように、Telegraf エージェントに Docker Unix ソケットへのアクセスを提供するために必要な手順を実行してください。

追加情報は以下からご覧いただけます。"サポート"ページ。

Elasticsearch データコレクター

Data Infrastructure Insights は、このデータ コレクターを使用して Elasticsearch からメトリックを収集します。

1. **Observability > Collectors** から、**+ Data Collector** をクリックします。 Elasticsearch を選択します。

   Telegraf エージェントがインストールされているオペレーティング システムまたはプラットフォームを選択します。

2. 収集用のエージェントをまだインストールしていない場合、または別のオペレーティングシステムまたはプラットフォーム用のエージェントをインストールする場合は、[手順を表示] をクリックして展開します。"エージェントのインストール"説明書。
3. このデータ コレクターで使用するエージェント アクセス キーを選択します。 **+** エージェント アクセス キー ボタンをクリックすると、新しいエージェント アクセス キーを追加できます。ベスト プラクティス: データ コレクターを OS/プラットフォーム別などにグループ化する場合にのみ、異なるエージェント アクセス キーを使用します。
4. 構成手順に従ってデータ コレクターを構成します。手順は、データ収集に使用しているオペレーティング システムまたはプラットフォームの種類によって異なります。

[Elasticsearchの設定]

セットアップ

詳細は以下をご覧ください。"Elasticsearch ドキュメント" 。

オブジェクトとカウンター

次のオブジェクトとそのカウンターが収集されます。

オブジェクト	識別子:	属性:
Elasticsearch クラスター	名前空間クラスター	ノードIP ノード名 クラスタステータス

オブジェクト	識別子:	属性:
Elasticsearchノード	名前空間 クラスター ES ノード ID ES ノード IP ES ノード	ゾーンID

トラブルシューティング

追加情報は以下からご覧いただけます。"サポート"ページ。

Flinkデータコレクター

Data Infrastructure Insights は、このデータ コレクターを使用して Flink からメトリックを収集します。

インストール

1. **Observability > Collectors** から、**+ Data Collector** をクリックします。 Flinkを選択します。

   Telegraf エージェントがインストールされているオペレーティング システムまたはプラットフォームを選択します。

2. 収集用のエージェントをまだインストールしていない場合、または別のオペレーティングシステムまたはプラットフォーム用のエージェントをインストールする場合は、[手順を表示] をクリックして展開します。"エージェントのインストール"説明書。
3. このデータ コレクターで使用するエージェント アクセス キーを選択します。 **+** エージェント アクセス キー ボタンをクリックすると、新しいエージェント アクセス キーを追加できます。ベスト プラクティス: データ コレクターを OS/プラットフォーム別などにグループ化する場合にのみ、異なるエージェント アクセス キーを使用します。
4. 構成手順に従ってデータ コレクターを構成します。手順は、データ収集に使用しているオペレーティング システムまたはプラットフォームの種類によって異なります。

[Flinkの設定]

セットアップ

完全な Flink デプロイメントには、次のコンポーネントが含まれます。

JobManager: Flink プライマリ システム。一連の TaskManager を調整します。高可用性設定では、システムには複数の JobManager が存在します。 TaskManager: ここで Flink オペレーターが実行されます。 Flink プラグインは、Telegraf の Jolokia プラグインに基づいています。すべての Flink コンポーネントから情報を収集する必要があるため、すべてのコンポーネントで Jolokia を介して JMX を構成して公開する必要があります。

互換性

構成は Flink バージョン 1.7.0 に対して開発されました。

セットアップ

ジョロキアエージェントジャー

すべての個別コンポーネントについて、Jolokia エージェント jar ファイルのバージョンをダウンロードする必要があります。テスト対象のバージョンは"Jolokia エージェント 1.6.0"。

以下の手順では、ダウンロードした jar ファイル (jolokia-jvm-1.6.0-agent.jar) が '/opt/flink/lib/' の場所に配置されていることを前提としています。

ジョブマネージャー

JobManager が Jolokia API を公開するように設定するには、ノードに次の環境変数を設定し、JobManager を再起動します。

```
 export FLINK_ENV_JAVA_OPTS="-javaagent:/opt/flink/lib/jolokia-jvm-1.6.0-agent.jar=port=8778,host=0.0.0.0"
```

`Jolokia (8778)` には別のポートを選択できます。 `Jolokia` をロックするための内部 `IP` がある場合は、「すべてをキャッチ」する `0.0.0.0` を独自の `IP` に置き換えることができます。この `IP` は `Telegraf` プラグインからアクセスできる必要があることに注意してください。

タスクマネージャー

Jolokia API を公開するように TaskManager を構成するには、ノードで次の環境変数を設定し、TaskManager を再起動します。

```
 export FLINK_ENV_JAVA_OPTS="-javaagent:/opt/flink/lib/jolokia-jvm-1.6.0-agent.jar=port=8778,host=0.0.0.0"
```

`Jolokia (8778)` には別のポートを選択できます。 `Jolokia` をロックするための内部 `IP` がある場合は、「すべてをキャッチ」する `0.0.0.0` を独自の `IP` に置き換えることができます。この `IP` は `Telegraf` プラグインからアクセスできる必要があることに注意してください。

オブジェクトとカウンター

次のオブジェクトとそのカウンターが収集されます。

オブジェクト	識別子:	属性:	データポイント:
Flinkタスクマネージャー	クラスター名前空間サーバー	ノード名 タスクマネージャーID ノードIP	ネットワーク使用可能メモリセグメント ネットワーク合計メモリセグメント ガベージコレクション PS マークスイープ数 ガベージコレクション PS マークスイープ時間 ガベージコレクション PS スカベンジ数 ガベージコレクション PS スカベンジ時間 コミット済みヒープメモリ ヒープメモリ初期化ヒープメモリ 最大使用ヒープメモリ スレッド数 デーモンスレッド数 ピークスレッド数 開始済みスレッド数合計
フリンクジョブ	クラスタ名前空間サーバジョブID	ノード名 ジョブ名 ノードIP 最終チェックポイント 外部パス 再起動時間	ダウンタイム 完全再起動 最終チェックポイント アライメント バッファリング 最終チェックポイントの持続時間 最終チェックポイントのサイズ 完了したチェックポイントの数 失敗したチェックポイントの数 進行中のチェックポイントの数 チェックポイントの数 稼働時間
Flinkジョブマネージャー	クラスター名前空間サーバー	ノード名 ノードIP	ガベージ コレクション PS MarkSweep 数 ガベージ コレクション PS MarkSweep 時間 ガベージ コレクション PS Scavenge 数 ガベージ コレクション PS Scavenge 時間 コミットされたヒープ メモリ ヒープ メモリの初期値 ヒープ メモリの最大使用ヒープ メモリ 登録されたタスク マネージャーの数 実行中のジョブの数 使用可能なタスク スロットの数 合計スレッド数 デーモン スレッド数 ピーク スレッド数 合計開始スレッド数

オブジェクト	識別子:	属性:	データポイント:
フリンクタスク	クラスター名前空間ジョブIDタスクID	サーバーノード名 ジョブ名 サブタスクインデックス タスク試行ID タスク試行番号 タスク名 タスクマネージャID ノードIP 現在の入力ウォーターマーク	プール内バッファー使用量、キュー内バッファー長、プール外バッファー使用量、キュー内バッファー長、ローカル内バッファー数、ローカル内バッファー数 (秒あたり)、ローカル内バッファー数 (秒あたり)、速度、リモート内バッファー数、リモート内バッファー数 (秒あたり)、リモート内バッファー数 (秒あたり)、速度、出力バッファー数、秒あたり出力バッファー数、秒あたり出力バッファー数、速度、ローカル内バイト数、ローカル内バイト数 (秒あたり)、速度、リモート内バイト数、リモート内バイト数 (秒あたり)、速度、出力バイト数、秒あたり出力バイト数、秒あたり出力バイト数、秒あたり出力バイト数、速度、入力レコード数、秒あたり入力レコード数、秒あたり入力レコード数、秒あたり出力レコード数、秒あたり出力レコード数、秒あたり出力レコード数

オブジェクト	識別子:	属性:	データポイント:
Flinkタスクオペレーター	クラスター名前空間ジョブIDオペレーターIDタスクID	サーバーノード名 ジョブ名 オペレータ名 サブタスクインデックス タスク試行ID タスク試行番号 タスク名 タスクマネージャID ノードIP	現在の入力ウォーターマーク 現在の出力ウォーターマーク 入力レコード数 1秒あたりの入力レコード数 1秒あたりの入力レコード数 レート 出力レコード数 1秒あたりの出力レコード数 1秒あたりの出力レコード数 レート 遅延レコード数 ドロップされた割り当て済みパーティション 消費バイト数 レート コミット待ち時間 平均コミット待ち時間 最大コミットレート 失敗したコミット数 成功したコミット数 接続クローズレート 接続数 接続作成レート数 フェッチ待ち時間 平均フェッチ待ち時間 最大フェッチレート フェッチサイズ 平均フェッチサイズ 最大フェッチスロットル時間 平均フェッチスロットル時間 最大ハートビートレート 受信バイトレート IO比率 IO時間平均（ナノ秒） IO待機比率 IO待機時間平均（ナノ秒） 参加レート 参加時間平均前回ハートビート前 ネットワークIOレート 送信バイトレート 消費レコード数 レコード遅延 リクエストあたりの最大レコード数 平均リクエストレート リクエストサイズ 平均リクエストサイズ 最大レスポンスレート 選択レート 同期レート 同期時間 平均ハートビートレスポンス時間最大参加時間 最大同期時間 最大

トラブルシューティング

追加情報は以下からご覧いただけます。"サポート"ページ。

Hadoop データコレクター

Data Infrastructure Insights は、このデータ コレクターを使用して Hadoop からメトリックを収集します。

インストール

1. **Observability > Collectors** から、**+ Data Collector** をクリックします。 Hadoop を選択します。

   Telegraf エージェントがインストールされているオペレーティング システムまたはプラットフォームを選択します。

2. 収集用のエージェントをまだインストールしていない場合、または別のオペレーティングシステムまたはプラットフォーム用のエージェントをインストールする場合は、[手順を表示] をクリックして展開します。"エージェントのインストール"説明書。
3. このデータ コレクターで使用するエージェント アクセス キーを選択します。 **+** エージェント アクセス キー ボタンをクリックすると、新しいエージェント アクセス キーを追加できます。ベスト プラクティス: データ コレクターを OS/プラットフォーム別などにグループ化する場合にのみ、異なるエージェント アクセス キーを使用します。
4. 構成手順に従ってデータ コレクターを構成します。手順は、データ収集に使用しているオペレーティング システムまたはプラットフォームの種類によって異なります。

[Hadoopの設定] [Hadoopの設定]

セットアップ

完全な Hadoop の展開には、次のコンポーネントが含まれます。

- NameNode: Hadoop 分散ファイル システム (HDFS) のプライマリ システム。一連の DataNode を調整します。
- セカンダリ NameNode: メイン NameNode のウォーム フェイルオーバー。 Hadoop では、NameNode への昇格は自動的には行われません。セカンダリ NameNode は、必要に応じて昇格できるように NameNode から情報を収集します。
- DataNode: データの実際の所有者。
- ResourceManager: コンピューティング プライマリ システム (Yarn)。一連の NodeManager を調整します。
- NodeManager: コンピューティングのリソース。アプリケーションを実行する実際の場所。
- JobHistoryServer: すべてのジョブ履歴関連のリクエストを処理する役割を担います。

Hadoop プラグインは、Telegraf の Jolokia プラグインに基づいています。すべての Hadoop コンポーネントから情報を収集する必要があるため、すべてのコンポーネントで Jolokia 経由で JMX を構成して公開する必要があります。

互換性

構成は Hadoop バージョン 2.9.2 に対して開発されました。

セットアップ

ジョロキアエージェントジャー

すべての個別コンポーネントについて、Jolokia エージェント jar ファイルのバージョンをダウンロードする必要があります。テスト対象のバージョンは"Jolokia エージェント 1.6.0"。

以下の手順では、ダウンロードした jar ファイル (jolokia-jvm-1.6.0-agent.jar) が '/opt/hadoop/lib/' の場所に配置されていることを前提としています。

ネームノード

NameNode を設定して Jolokia API を公開するには、<HADOOP_HOME>/etc/hadoop/hadoop-env.sh で次のように設定します。

```
export HADOOP_NAMENODE_OPTS="$HADOOP_NAMENODE_OPTS
-javaagent:/opt/hadoop/lib/jolokia-jvm-1.6.0
-agent.jar=port=7800,host=0.0.0.0 -Dcom.sun.management.jmxremote
-Dcom.sun.management.jmxremote.port=8000
-Dcom.sun.management.jmxremote.ssl=false
-Dcom.sun.management.jmxremote.password.file=$HADOOP_HOME/conf/jmxremote.p
assword"
You can choose a different port for JMX (8000 above) and Jolokia (7800).
If you have an internal IP to lock Jolokia onto you can replace the "catch
all" 0.0.0.0 by your own IP. Notice this IP needs to be accessible from
the telegraf plugin. You can use the option '-
Dcom.sun.management.jmxremote.authenticate=false' if you don't want to
authenticate. Use at your own risk.
```

セカンダリネームノード

セカンダリ NameNode が Jolokia API を公開するように設定するには、<HADOOP_HOME>/etc/hadoop/hadoop-env.sh で次のように設定します。

```
export HADOOP_SECONDARYNAMENODE_OPTS="$HADOOP_SECONDARYNAMENODE_OPTS
-javaagent:/opt/hadoop/lib/jolokia-jvm-1.6.0
-agent.jar=port=7802,host=0.0.0.0 -Dcom.sun.management.jmxremote
-Dcom.sun.management.jmxremote.port=8002
-Dcom.sun.management.jmxremote.ssl=false
-Dcom.sun.management.jmxremote.password.file=$HADOOP_HOME/conf/jmxremote.p
assword"
You can choose a different port for JMX (8002 above) and Jolokia (7802).
If you have an internal IP to lock Jolokia onto you can replace the "catch
all" 0.0.0.0 by your own IP. Notice this IP needs to be accessible from
the telegraf plugin. You can use the option '-
Dcom.sun.management.jmxremote.authenticate=false' if you don't want to
authenticate. Use at your own risk.
```

データノード

Jolokia API を公開するように DataNode を構成するには、<HADOOP_HOME>/etc/hadoop/hadoop-env.sh で次のように設定します。

```
export HADOOP_DATANODE_OPTS="$HADOOP_DATANODE_OPTS
-javaagent:/opt/hadoop/lib/jolokia-jvm-1.6.0
-agent.jar=port=7801,host=0.0.0.0 -Dcom.sun.management.jmxremote
-Dcom.sun.management.jmxremote.port=8001
-Dcom.sun.management.jmxremote.ssl=false
-Dcom.sun.management.jmxremote.password.file=$HADOOP_HOME/conf/jmxremote.p
assword"
You can choose a different port for JMX (8001 above) and Jolokia (7801).
If you have an internal IP to lock Jolokia onto you can replace the "catch
all" 0.0.0.0 by your own IP. Notice this IP needs to be accessible from
the telegraf plugin. You can use the option '-
Dcom.sun.management.jmxremote.authenticate=false' if you don't want to
authenticate. Use at your own risk.
```

リソースマネージャー

ResourceManager が Jolokia API を公開するように設定するには、<HADOOP_HOME>/etc/hadoop/hadoop-env.sh で次のように設定します。

```
export YARN_RESOURCEMANAGER_OPTS="$YARN_RESOURCEMANAGER_OPTS
-javaagent:/opt/hadoop/lib/jolokia-jvm-1.6.0
-agent.jar=port=7803,host=0.0.0.0 -Dcom.sun.management.jmxremote
-Dcom.sun.management.jmxremote.port=8003
-Dcom.sun.management.jmxremote.ssl=false
-Dcom.sun.management.jmxremote.password.file=$HADOOP_HOME/conf/jmxremote.p
assword"
You can choose a different port for JMX (8003 above) and Jolokia (7803).
If you have an internal IP to lock Jolokia onto you can replace the "catch
all" 0.0.0.0 by your own IP. Notice this IP needs to be accessible from
the telegraf plugin. You can use the option '-
Dcom.sun.management.jmxremote.authenticate=false' if you don't want to
authenticate. Use at your own risk.
```

ノードマネージャー

Jolokia API を公開するように NodeManagers を構成するには、<HADOOP_HOME>/etc/hadoop/hadoop-env.sh で次のように設定します。

```
export YARN_NODEMANAGER_OPTS="$YARN_NODEMANAGER_OPTS
-javaagent:/opt/hadoop/lib/jolokia-jvm-1.6.0
-agent.jar=port=7804,host=0.0.0.0 -Dcom.sun.management.jmxremote
-Dcom.sun.management.jmxremote.port=8004
-Dcom.sun.management.jmxremote.ssl=false
-Dcom.sun.management.jmxremote.password.file=$HADOOP_HOME/conf/jmxremote.p
assword"
You can choose a different port for JMX (8004 above) and Jolokia (7804).
If you have an internal IP to lock Jolokia onto you can replace the "catch
all" 0.0.0.0 by your own IP. Notice this IP needs to be accessible from
the telegraf plugin. You can use the option '-
Dcom.sun.management.jmxremote.authenticate=false' if you don't want to
authenticate. Use at your own risk.
```

ジョブ履歴サーバー

JobHistoryServer が Jolokia API を公開するように設定するには、<HADOOP_HOME>/etc/hadoop/hadoop-env.sh で次のように設定します。

```
export HADOOP_JOB_HISTORYSERVER_OPTS="$HADOOP_JOB_HISTORYSERVER_OPTS
-javaagent:/opt/hadoop/lib/jolokia-jvm-1.6.0
-agent.jar=port=7805,host=0.0.0.0 -Dcom.sun.management.jmxremote
-Dcom.sun.management.jmxremote.port=8005
-Dcom.sun.management.jmxremote.password.file=$HADOOP_HOME/conf/jmxremote.p
assword"
You can choose a different port for JMX (8005 above) and Jolokia (7805).
If you have an internal IP to lock Jolokia onto you can replace the "catch
all" 0.0.0.0 by your own IP. Notice this IP needs to be accessible from
the telegraf plugin. You can use the option '-
Dcom.sun.management.jmxremote.authenticate=false' if you don't want to
authenticate. Use at your own risk.
```

オブジェクトとカウンター

次のオブジェクトとそのカウンターが収集されます。

オブジェクト	識別子:	属性:
Hadoop セカンダリネームノード	クラスター名前空間サーバー	ノード名 ノードIP コンパイル情報 バージョン
Hadoop ノードマネージャー	クラスター名前空間サーバー	ノード名 ノードIP
Hadoop リソースマネージャー	クラスター名前空間サーバー	ノード名 ノードIP
Hadoop データノード	クラスター名前空間サーバー	ノード名 ノードIP クラスタID バージョン
Hadoop ネームノード	クラスター名前空間サーバー	ノード名 ノードIP トランザクションID 最終書き込み 最終ロード以降の時間 編集 HA状態 ファイルシステム状態 ブロックプールID クラスタID コンパイル情報 個別バージョン数 バージョン
Hadoop ジョブ履歴サーバー	クラスター名前空間サーバー	ノード名 ノードIP

トラブルシューティング

追加情報は以下からご覧いただけます。"サポート"ページ。

HAProxy データコレクター

Data Infrastructure Insights は、このデータ コレクターを使用して HAProxy からメトリックを収集します。

インストール

1. **Observability > Collectors** から、**+ Data Collector** をクリックします。 HAProxyを選択します。

   Telegraf エージェントがインストールされているオペレーティング システムまたはプラットフォームを選択します。

2. 収集用のエージェントをまだインストールしていない場合、または別のオペレーティングシステムまたはプラットフォーム用のエージェントをインストールする場合は、[手順を表示] をクリックして展開します。"エージェントのインストール"説明書。
3. このデータ コレクターで使用するエージェント アクセス キーを選択します。 **+** エージェント アクセス キー ボタンをクリックすると、新しいエージェント アクセス キーを追加できます。ベスト プラクティス: データ コレクターを OS/プラットフォーム別などにグループ化する場合にのみ、異なるエージェント アクセス キーを使用します。
4. 構成手順に従ってデータ コレクターを構成します。手順は、データ収集に使用しているオペレーティング システムまたはプラットフォームの種類によって異なります。

[HAProxy設定]

セットアップ

Telegraf の HAProxy 用プラグインは、HAProxy Stats の有効化に依存しています。これは HAProxy に組み込まれた構成ですが、そのままでは有効になっていません。有効にすると、HAProxy はブラウザで表示したり、すべての HAProxy 構成のステータスを抽出するためにスクレイピングしたりできる HTML エンドポイントを公開します。

互換性:

構成は HAProxy バージョン 1.9.4 に対して開発されました。

セットアップ:

統計を有効にするには、haproxy 構成ファイルを編集し、独自のユーザー名/パスワードおよび/または haproxy URL を使用して、 'defaults' セクションの後に次の行を追加します。

```
stats enable
stats auth myuser:mypassword
stats uri  /haproxy?stats
```

以下は、統計が有効になっている簡略化された構成ファイルの例です。

```
global
  daemon
  maxconn 256

defaults
  mode http
  stats enable
  stats uri /haproxy?stats
  stats auth myuser:mypassword
  timeout connect 5000ms
  timeout client 50000ms
  timeout server 50000ms

frontend http-in
  bind *:80
  default_backend servers

frontend http-in9080
  bind *:9080
  default_backend servers_2

backend servers
  server server1 10.128.0.55:8080 check ssl verify none
  server server2 10.128.0.56:8080 check ssl verify none

backend servers_2
  server server3 10.128.0.57:8080 check ssl verify none
  server server4 10.128.0.58:8080 check ssl verify none
```

完全かつ最新の手順については、"HAProxyドキュメント" 。

オブジェクトとカウンター

次のオブジェクトとそのカウンターが収集されます。

オブジェクト	識別子:	属性:	データポイント:
HAProxy フロントエンド	名前空間アドレスプロキシ	ノードIP ノード名 プロキシID モード プロセスID セッション レート制限 サーバーID セッション制限 ステータス	受信バイト数 送信バイト数 キャッシュヒット数 キャッシュルックアップ数 圧縮バイト数 バイパス圧縮バイト数 受信バイト数 圧縮バイト数 出力バイト数 圧縮レスポンス数 接続率 接続率 最大接続数 合計リクエスト数 接続ルールにより拒否されたリクエスト数 セキュリティ上の懸念により拒否されたリクエスト数 セキュリティ上の懸念により拒否されたレスポンス数 セッションルールにより拒否されたリクエスト数 エラーレスポンス数 1xxレスポンス数 2xxレスポンス数 3xxレスポンス数 4xxレスポンス数 5xxレスポンス数 その他のリクエスト数 インターセプトされたリクエスト数 セッション率 セッション率 最大リクエスト数 リクエスト率 最大リクエスト数 合計セッション数 セッション数 最大セッション数 合計リクエスト数 書き換え

オブジェクト	識別子:	属性:	データポイント:
HAProxyサーバー	名前空間アドレスプロキシサーバー	ノードIP ノード名 チェック完了時間 チェック下降構成チェック ヘルス値チェック上昇構成チェック ステータス プロキシID 最終変更時間 最終セッション時間 モード プロセスID サーバーID ステータス 重み	アクティブサーバー バックアップサーバー 受信バイト数 送信バイト数 チェックダウン数 チェック失敗数 クライアントの中止 接続数 接続平均時間 ダウンタイム 合計拒否数 応答数 接続エラー数 応答エラー数 応答数 1xx 応答数 2xx 応答数 3xx 応答数 4xx 応答数 5xx 応答数 選択されたその他のサーバー 合計キュー数 現在のキュー数 最大キュー数 平均時間 1秒あたりのセッション数 1秒あたりの最大セッション数 接続再利用数 応答時間 平均セッション数 最大セッション数 サーバー転送中止数 セッション数 合計セッション数 合計時間 平均リクエスト数 再ディスパッチ数 リクエスト数 再試行数 リクエスト数 書き換え数

オブジェクト	識別子:	属性:	データポイント:
HAProxyバックエンド	名前空間アドレスプロキシ	ノードIP ノード名 プロキシID 最終変更時刻 最終セッション時刻 モード プロセスID サーバーID セッション制限 ステータス 重み	アクティブサーバー バックアップサーバー 受信バイト数 送信バイト数 キャッシュヒット数 キャッシュ検索数 チェックダウン数 クライアントによる圧縮中止数 圧縮バイパス数 圧縮受信バイト数 圧縮送信バイト数 圧縮レスポンス数 接続数 接続平均時間 ダウンタイム合計 セキュリティ上の懸念により拒否されたリクエスト数 セキュリティ上の懸念により拒否されたレスポンス数 接続エラー数 レスポンスエラー数 レスポンス数 1xxレスポンス数 2xxレスポンス数 3xxレスポンス数 4xxレスポンス数 5xxレスポンス数 選択されたその他のサーバー キュー合計数 現在のキュー数 最大キュー数 平均時間 1秒あたりのセッション数 1秒あたりのセッション数 最大リクエスト数 合計接続再利用数 レスポンス時間 平均セッション数 最大セッション数 サーバー転送中止数 セッション数 合計セッション数 合計時間 平均リクエスト数 再ディスパッチ数 リクエスト数 再試行数 リクエスト数 書き換え数

トラブルシューティング

追加情報は以下からご覧いただけます。"サポート"ページ。

JVMデータコレクター

Data Infrastructure Insights は、このデータ コレクターを使用して JVM からメトリックを収集します。

インストール

1. **Observability > Collectors** から、**+ Data Collector** をクリックします。 JVM を選択します。

Telegraf エージェントがインストールされているオペレーティング システムまたはプラットフォームを選択します。

2. 収集用のエージェントをまだインストールしていない場合、または別のオペレーティングシステムまたはプラットフォーム用のエージェントをインストールする場合は、[手順を表示] をクリックして展開します。"エージェントのインストール"説明書。
3. このデータ コレクターで使用するエージェント アクセス キーを選択します。 + エージェント アクセス キー ボタンをクリックすると、新しいエージェント アクセス キーを追加できます。ベスト プラクティス: データ コレクターを OS/プラットフォーム別などにグループ化する場合にのみ、異なるエージェント アクセス キーを使用します。
4. 構成手順に従ってデータ コレクターを構成します。手順は、データ収集に使用しているオペレーティング システムまたはプラットフォームの種類によって異なります。

[JVM構成]

セットアップ

情報は以下をご覧ください"JVMのドキュメント"。

オブジェクトとカウンター

次のオブジェクトとそのカウンターが収集されます。

オブジェクト	識別子:	属性:	データポイント:
JVM	ネームスペースJVM	OS アーキテクチャ OS 名 OS バージョン ランタイム仕様 ランタイム仕様 ベンダー ランタイム仕様 バージョン 稼働時間 ランタイム VM 名 ランタイム VM ベンダー ランタイム VM バージョン ノード名 ノード IP	ロードされたクラス ロードされたクラスの合計 クラスがアンロードされたメモリ ヒープ コミットされたメモリ ヒープ初期メモリ 使用済みヒープ最大メモリ 使用済みメモリ 非ヒープ コミットされたメモリ 非ヒープ初期メモリ 非ヒープ最大メモリ 非ヒープ 使用済みメモリ ファイナライズ保留中のメモリ オブジェクト 使用可能な OS プロセッサ OS コミット済み仮想メモリ サイズ OS 空き物理メモリ サイズ OS 空きスワップ領域サイズ OS 最大ファイル記述子数 OS オープン ファイル記述子数 OS プロセッサ CPU 負荷 OS プロセッサ CPU 時間 OS システム CPU 負荷 OS システム平均負荷 OS 合計物理メモリ サイズ OS 合計スワップ領域サイズ スレッド デーモン数 スレッド ピーク数 スレッド数 開始されたスレッド数 ガベージ コレクタ コピー コレクション数 ガベージ コレクタ コピー コレクション時間 ガベージ コレクタ マークスイープ コレクション数 ガベージ コレクタ マークスイープ コレクション時間 ガベージ コレクタ G1 Old Generation コレクション数 ガベージ コレクタ G1 Old Generation コレクション時間 ガベージ コレクタ G1 Young Generation コレクション数 ガベージ コレクタ G1 Young Generation コレクション時間 ガベージ コレクタ同時マークスイープ コレクション数 ガベージ コレクター同時マークスイープ コレクション時間 ガベージ コレクター並列コレクション数 ガベージ コレクター並列コレクション時

トラブルシューティング

追加情報は以下からご覧いただけます。"サポート"ページ。

Kafka データコレクター

Data Infrastructure Insights は、このデータ コレクターを使用して Kafka からメトリックを収集します。

インストール

1. **Observability > Collectors** から、**+ Data Collector** をクリックします。 Kafka を選択します。

   Telegraf エージェントがインストールされているオペレーティング システムまたはプラットフォームを選択します。

2. 収集用のエージェントをまだインストールしていない場合、または別のオペレーティングシステムまたはプラットフォーム用のエージェントをインストールする場合は、[手順を表示] をクリックして展開します。"エージェントのインストール"説明書。
3. このデータ コレクターで使用するエージェント アクセス キーを選択します。 **+** エージェント アクセス キー ボタンをクリックすると、新しいエージェント アクセス キーを追加できます。ベスト プラクティス: データ コレクターを OS/プラットフォーム別などにグループ化する場合にのみ、異なるエージェント アクセス キーを使用します。
4. 構成手順に従ってデータ コレクターを構成します。手順は、データ収集に使用しているオペレーティング システムまたはプラットフォームの種類によって異なります。

[Kafkaの設定]

セットアップ

Kafka プラグインは、Telegraf の Jolokia プラグインに基づいています。すべての Kafka ブローカーから情報を収集する必要があるため、すべてのコンポーネントで Jolokia 経由で JMX を構成して公開する必要があります。

互換性

構成は Kafka バージョン 0.11.0.2 に対して開発されました。

セットアップ

以下のすべての手順では、kafka のインストール場所が '/opt/kafka' であることを前提としています。以下の手順をインストール場所に合わせて調整できます。

ジョロキアエージェントジャー

Jolokiaエージェントjarファイルのバージョン"ダウンロード済み"。テスト対象のバージョンは Jolokia エージェント 1.6.0 です。

以下の手順では、ダウンロードした jar ファイル (jolokia-jvm-1.6.0-agent.jar) が '/opt/kafka/libs/' の場所に配置されていることを前提としています。

Jolokia API を公開するように Kafka Brokers を構成するには、<KAFKA_HOME>/bin/kafka-server-start.sh の 'kafka-run-class.sh' 呼び出しの直前に以下を追加します。

```
export JMX_PORT=9999
export RMI_HOSTNAME=`hostname -I`
export KAFKA_JMX_OPTS="-javaagent:/opt/kafka/libs/jolokia-jvm-1.6.0-
agent.jar=port=8778,host=0.0.0.0
-Dcom.sun.management.jmxremote.password.file=/opt/kafka/config/jmxremote.p
assword -Dcom.sun.management.jmxremote.ssl=false
-Djava.rmi.server.hostname=$RMI_HOSTNAME
-Dcom.sun.management.jmxremote.rmi.port=$JMX_PORT"
```

上記の例では、「hostname -I」を使用して「RMI_HOSTNAME」環境変数を設定していることに注意してください。複数の IP を持つマシンでは、RMI 接続に必要な IP を収集するためにこれを調整する必要があります。

JMX (上記 9999) と Jolokia (8778) には異なるポートを選択できます。 Jolokia をロックするための内部 IP がある場合は、「すべてをキャッチ」する 0.0.0.0 を独自の IP に置き換えることができます。この IP は Telegraf プラグインからアクセスできる必要があることに注意してください。認証したくない場合は、オプション「-Dcom.sun.management.jmxremote.authenticate=false」を使用できます。自己責任でご使用ください。

オブジェクトとカウンター

次のオブジェクトとそのカウンターが収集されます。

オブジェクト	識別子:	属性:
Kafkaブローカー	クラスター名前空間ブローカー	ノード名 ノードIP

トラブルシューティング

追加情報は以下からご覧いただけます。"サポート"ページ。

Kibana データコレクター

Data Infrastructure Insights は、このデータ コレクターを使用して Kibana からメトリックを収集します。

インストール

1. **Observability > Collectors** から、**+ Data Collector** をクリックします。 Kibanaを選択します。

   Telegraf エージェントがインストールされているオペレーティング システムまたはプラットフォームを選択します。

2. 収集用のエージェントをまだインストールしていない場合、または別のオペレーティングシステムまたはプラットフォーム用のエージェントをインストールする場合は、[手順を表示] をクリックして展開します。"エージェントのインストール"説明書。
3. このデータ コレクターで使用するエージェント アクセス キーを選択します。 + エージェント アクセス キー ボタンをクリックすると、新しいエージェント アクセス キーを追加できます。ベスト プラクティス: データ コレクターを OS/プラットフォーム別などにグループ化する場合にのみ、異なるエージェント アクセス キーを使用します。
4. 構成手順に従ってデータ コレクターを構成します。手順は、データ収集に使用しているオペレーティング システムまたはプラットフォームの種類によって異なります。

[Kibanaの設定]

セットアップ

詳細は以下をご覧ください。"Kibanaのドキュメント" 。

オブジェクトとカウンター

次のオブジェクトとそのカウンターが収集されます。

オブジェクト	識別子:	属性:	データポイント:
キバナ	名前空間アドレス	ノードIP ノード名 バージョン ステータス	同時接続数 ヒープ最大使用数 1秒あたりのリクエスト数 応答時間平均応答時間最大稼働時間

トラブルシューティング

追加情報は以下からご覧いただけます。"サポート"ページ。

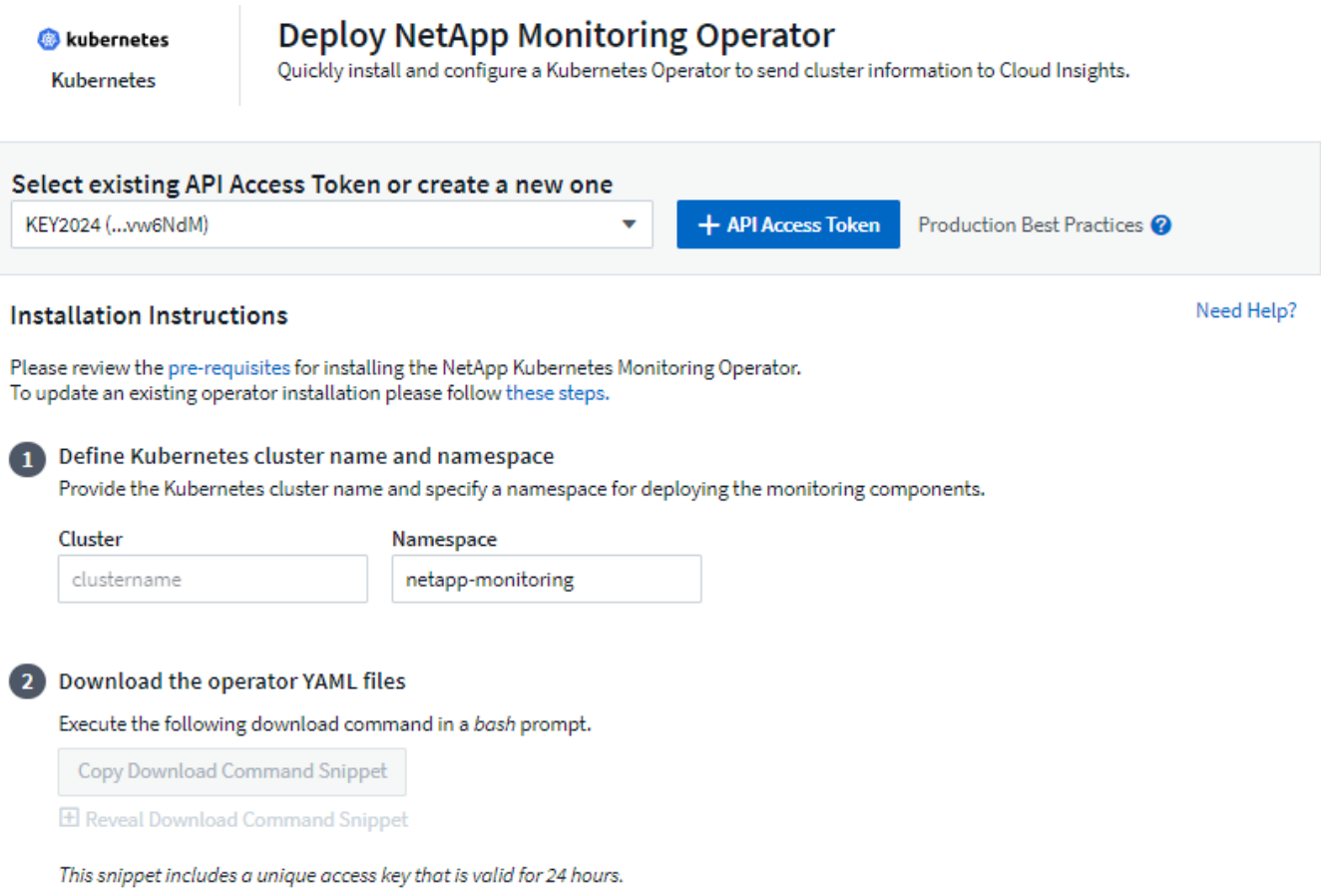
Kubernetes モニタリング オペレーターのインストールと構成

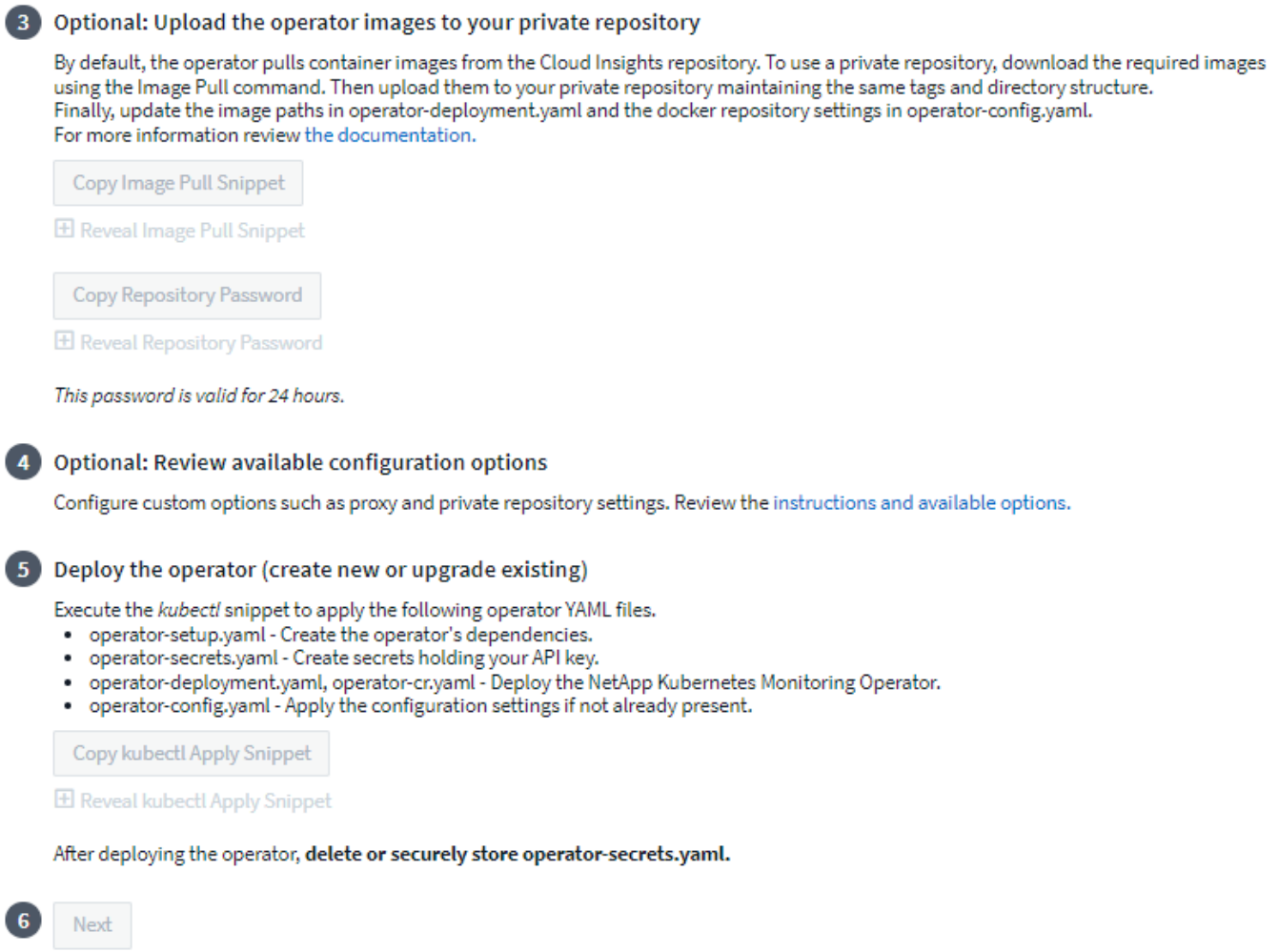
Data Infrastructure Insights は、 Kubernetes コレクション用の **Kubernetes Monitoring Operator** を提供します。新しいオペレーターをデプロイするには、**Kubernetes > Collectors > +Kubernetes Collector** に移動します。

Kubernetesモニタリングオペレーターをインストールする前に

参照"前提条件"Kubernetes Monitoring Operator をインストールまたはアップグレードする前に、ドキュメントを参照してください。

Kubernetes モニタリング オペレーターのインストール


kubernetes
Kubernetes
Deploy NetApp Monitoring Operator
Quickly install and configure a Kubernetes Operator to send cluster information to Cloud Insights.
Select existing API Access Token or create a new one
KEY2024 (...vw6NdM)
+ API Access Token
Production Best Practices
Installation Instructions
Need Help?
Please review the pre-requisites for installing the NetApp Kubernetes Monitoring Operator.
To update an existing operator installation please follow these steps.
1 Define Kubernetes cluster name and namespace
Provide the Kubernetes cluster name and specify a namespace for deploying the monitoring components.
Cluster
clustername
Namespace
netapp-monitoring
2 Download the operator YAML files
Execute the following download command in a bash prompt.
Copy Download Command Snippet
Reveal Download Command Snippet
This snippet includes a unique access key that is valid for 24 hours.



Kubernetes に **Kubernetes Monitoring Operator** エージェントをインストールする手順:

1. 一意のクラスター名と名前空間を入力します。もしあなたがアップグレード以前の Kubernetes Operator からの場合は、同じクラスター名と名前空間を使用します。
2. これらを入力すると、ダウンロード コマンド スニペットをクリップボードにコピーできます。
3. スニペットを *bash* ウィンドウに貼り付けて実行します。 Operator インストール ファイルがダウンロードされます。スニペットには一意のキーがあり、24 時間有効であることに注意してください。
4. カスタム リポジトリまたはプライベート リポジトリがある場合は、オプションの Image Pull スニペットをコピーし、*bash* シェルに貼り付けて実行します。イメージをプルしたら、それをプライベート リポジトリにコピーします。必ず同じタグとフォルダー構造を維持してください。 *operator-deployment.yaml* 内のパスと、*operator-config.yaml* 内の docker リポジトリ設定を更新します。
5. 必要に応じて、プロキシやプライベート リポジトリ設定などの利用可能な構成オプションを確認します。詳細については、"設定オプション" 。
6. 準備ができたら、kubectl Apply スニペットをコピーし、ダウンロードして実行して、Operator をデプロイします。
7. インストールは自動的に進行します。完了したら、[次へ] ボタンをクリックします。
8. インストールが完了したら、[次へ] ボタンをクリックします。 *operator-secrets.yaml* ファイルも必ず削除するか、安全に保存してください。

カスタムリポジトリをお持ちの場合は、以下をお読みください。カスタム/プライベートDockerリポジトリを

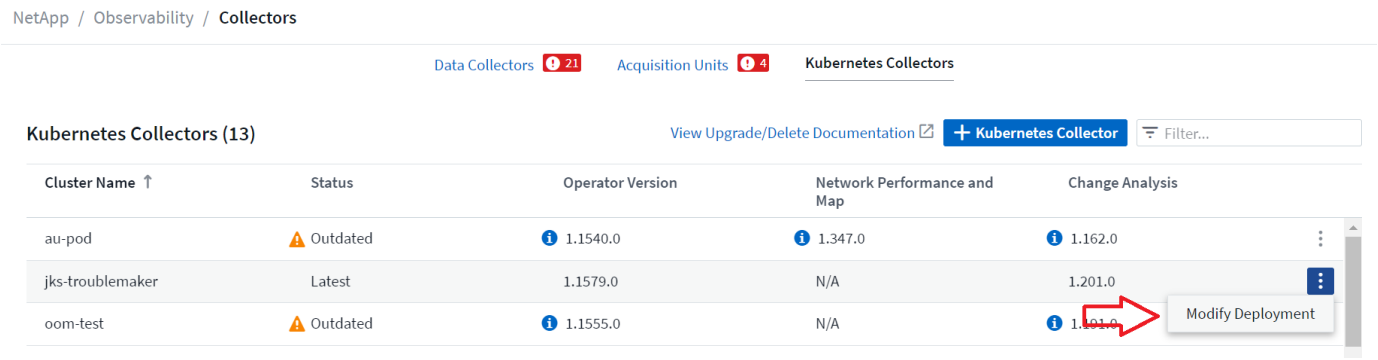
使用する 。

Kubernetes 監視コンポーネント

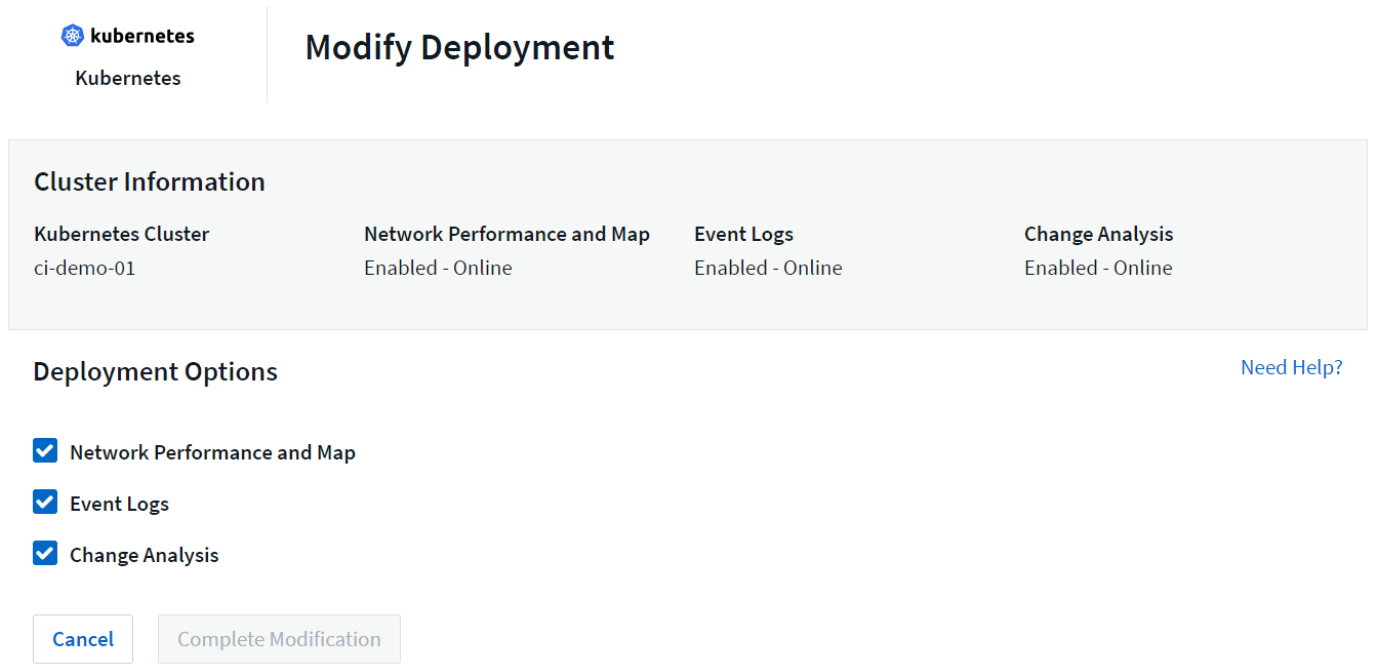
Data Infrastructure Insights Kubernetes モニタリングは、次の 4 つのモニタリング コンポーネントで構成されています。

- クラスターメトリック
- ネットワークパフォーマンスとマップ（オプション）
- イベントログ（オプション）
- 変更分析（オプション）

上記のオプション コンポーネントは、各 Kubernetes コレクターに対してデフォルトで有効になっています。特定のコレクターに対してコンポーネントが必要ないと判断した場合は、**Kubernetes > Collectors** に移動し、画面の右側にあるコレクターの「3 つのドット」メニューから *Modify Deployment* を選択して無効にすることができます。



画面には各コンポーネントの現在の状態が表示され、必要に応じてそのコレクターのコンポーネントを無効または有効にすることができます。



最新のKubernetesモニタリングオペレーターへのアップグレード

DII プッシュボタンアップグレード

DII Kubernetes Collectors ページから Kubernetes Monitoring Operator をアップグレードできます。アップグレードするクラスターの横にあるメニューをクリックし、「アップグレード」を選択します。オペレーターはイメージ署名を検証し、現在のインストールのスナップショットを実行して、アップグレードを実行します。数分以内に、オペレーターのステータスが「アップグレード進行中」から「最新」へと進行していくのが確認できます。エラーが発生した場合は、詳細を表示するにはエラー ステータスを選択し、以下のプッシュ ボタン アップグレードのトラブルシューティング表を参照してください。

プライベートリポジトリによるプッシュボタンアップグレード

オペレーターがプライベート リポジトリを使用するように構成されている場合は、オペレーターの実行に必要なすべてのイメージとその署名がリポジトリで使用可能であることを確認してください。アップグレード プロセス中にイメージが不足しているためにエラーが発生した場合は、イメージをリポジトリに追加して、アップグレードを再試行してください。リポジトリにイメージ署名をアップロードするには、次のようにcosign ツールを使用してください。3 オプション: オペレータイメージをプライベートリポジトリにアップロード > イメージプルスニペットで指定されたすべてのイメージの署名をアップロードしてください。

```
cosign copy example.com/src:v1 example.com/dest:v1
#Example
cosign copy <DII container registry>/netapp-monitoring:<image version>
<private repository>/netapp-monitoring:<image version>
```

以前実行していたバージョンにロールバックする

プッシュボタン アップグレード機能を使用してアップグレードし、アップグレード後 7 日以内に現在のバージョンのオペレーターに問題が発生した場合は、アップグレード プロセス中に作成されたスナップショットを使用して、以前実行していたバージョンにダウングレードできます。ロールバックするクラスターの横にあるメニューをクリックし、[ロールバック] を選択します。

手動アップグレード

既存の Operator に *AgentConfiguration* が存在するかどうかを判断します（名前空間がデフォルトの *netapp-monitoring* でない場合は、適切な名前空間に置き換えます）：

```
 kubectl -n netapp-monitoring get agentconfiguration netapp-ci-monitoring-
configuration
_AgentConfiguration_ が存在する場合：
```

- インストール既存のオペレータよりも最新のオペレータを優先します。
  - 必ず最新のコンテナイメージを取得するカスタム リポジトリを使用している場合。

AgentConfiguration が存在しない場合：

- Data Infrastructure Insightsによって認識されるクラスター名をメモします (名前空間がデフォルトの netapp-monitoring でない場合は、適切な名前空間に置き換えてください)。

```
 kubectl -n netapp-monitoring get agent -o
jsonpath='{.items[0].spec.cluster-name}'
* 既存の Operator のバックアップを作成します（名前空間がデフォルトの netapp-
monitoring でない場合は、適切な名前空間に置き換えます）。
```

```
 kubectl -n netapp-monitoring get agent -o yaml > agent_backup.yaml
* <<to-remove-the-kubernetes-monitoring-operator,アンインストール
>>既存のオペレータ。
* <<installing-the-kubernetes-monitoring-operator,インストール
>>最新のオペレーター。
```

- 同じクラスター名を使用します。
- 最新の Operator YAML ファイルをダウンロードした後、デプロイする前に、*agent_backup.yaml* にあるカスタマイズをダウンロードした *operator-config.yaml* に移植します。
- 必ず最新のコンテナイメージを取得するカスタム リポジトリを使用している場合。

Kubernetes モニタリング オペレーターの停止と起動

Kubernetes モニタリング オペレーターを停止するには:

```
 kubectl -n netapp-monitoring scale deploy monitoring-operator
--replicas=0
Kubernetes モニタリング オペレーターを起動するには：
```

```
kubectl -n netapp-monitoring scale deploy monitoring-operator --replicas=1
```

アンインストール

Kubernetesモニタリングオペレーターを削除するには

Kubernetes モニタリング オペレーターのデフォルトの名前空間は「netapp-monitoring」であることに注意してください。独自の名前空間を設定している場合は、これらのコマンドと後続のすべてのコマンドおよびファイルでその名前空間を置き換えます。

監視オペレーターの新しいバージョンは、次のコマンドでアンインストールできます。

```
kubectl -n <NAMESPACE> delete agent -l installed-by=nkmo-<NAMESPACE>
kubectl -n <NAMESPACE> delete
clusterrole,clusterrolebinding,crd,svc,deploy,role,rolebinding,secret,sa
-l installed-by=nkmo-<NAMESPACE>
```

監視オペレーターが専用のネームスペースにデプロイされている場合は、ネームスペースを削除します。

```
 kubectl delete ns <NAMESPACE>
```

注：
最初のコマンドで「リソースが見つかりません」と返された場合は、次の手順に従って、監視オペレーターの古いバージョンをアンインストールしてください。

以下の各コマンドを順番に実行します。現在のインストールによっては、これらのコマンドの一部が「オブジェクトが見つかりません」というメッセージを返す場合があります。これらのメッセージは無視しても問題ありません。

```
kubectl -n <NAMESPACE> delete agent agent-monitoring-netapp
kubectl delete crd agents.monitoring.netapp.com
kubectl -n <NAMESPACE> delete role agent-leader-election-role
kubectl delete clusterrole agent-manager-role agent-proxy-role agent-
metrics-reader <NAMESPACE>-agent-manager-role <NAMESPACE>-agent-proxy-role
<NAMESPACE>-cluster-role-privileged
kubectl delete clusterrolebinding agent-manager-rolebinding agent-proxy-
rolebinding agent-cluster-admin-rolebinding <NAMESPACE>-agent-manager-
rolebinding <NAMESPACE>-agent-proxy-rolebinding <NAMESPACE>-cluster-role-
binding-privileged
kubectl delete <NAMESPACE>-psp-nkmo
kubectl delete ns <NAMESPACE>
```

セキュリティ コンテキスト制約が以前に作成されている場合:

```
kubectl delete scc telegraf-hostaccess
```

Kube-state-metricsについて

NetApp Kubernetes Monitoring Operator は、他のインスタンスとの競合を避けるために独自の kube-state-metrics をインストールします。

Kube-State-Metricsの詳細については、以下を参照してください。"このページ" 。

オペレーターの設定/カスタマイズ

これらのセクションには、オペレーター構成のカスタマイズ、プロキシの操作、カスタムまたはプライベート Docker リポジトリの使用、OpenShift の操作に関する情報が含まれています。

設定オプション

最も頻繁に変更される設定は、*AgentConfiguration* カスタム リソースで構成できます。オペレーターをデプロイする前に、*operator-config.yaml* ファイルを編集してこのリソースを編集できます。このファイルには、コメントアウトされた設定の例が含まれています。リストを見る"利用可能な設定"オペレーターの最新バージ

ョン。

オペレーターをデプロイした後、次のコマンドを使用してこのリソースを編集することもできます。

```
kubectl -n netapp-monitoring edit AgentConfiguration
```

デプロイされたオペレーターのバージョンが _AgentConfiguration_ をサポートしているかどうかを確認するには、次のコマンドを実行します：

```
kubectl get crd agentconfigurations.monitoring.netapp.com
```

「サーバーからのエラー (NotFound)」というメッセージが表示された場合は、AgentConfiguration を使用する前にオペレーターをアップグレードする必要があります。

プロキシサポートの設定

Kubernetes モニタリング オペレーターをインストールするために、テナント上でプロキシを使用できる場所は 2 つあります。これらは同じプロキシ システムである場合もあれば、別のプロキシ システムである場合もあります。

- インストール コード スニペットの実行中（「curl」を使用）に、スニペットが実行されるシステムをData Infrastructure Insights環境に接続するために必要なプロキシ
- ターゲット Kubernetes クラスターがData Infrastructure Insights環境と通信するために必要なプロキシ

これらのいずれかまたは両方にプロキシを使用する場合、Kubernetes Operating Monitor をインストールするには、まずプロキシがData Infrastructure Insights環境との良好な通信を許可するように構成されていることを確認する必要があります。プロキシがあり、Operator をインストールするサーバー/VM からData Infrastructure Insightsにアクセスできる場合は、プロキシは適切に構成されている可能性があります。

Kubernetes オペレーティング モニターのインストールに使用するプロキシについては、Operator をインストールする前に、*http_proxy/https_proxy* 環境変数を設定します。一部のプロキシ環境では、*no_proxy environment* 変数も設定する必要がある場合があります。

変数を設定するには、Kubernetes モニタリング オペレーターをインストールする前に、システムで次の手順を実行します。

1. 現在のユーザーの *https_proxy* および/または *http_proxy* 環境変数を設定します。
   a. セットアップするプロキシに認証 (ユーザー名/パスワード) がない場合は、次のコマンドを実行します。

      ```
      export https_proxy=<proxy_server>:<proxy_port>
      ```

      . . セットアップするプロキシに認証 （ユーザー名/パスワード）がある場合は、次のコマンドを実行します。

```
export
http_proxy=<proxy_username>:<proxy_password>@<proxy_server>:<proxy_po
rt>
```

Kubernetes クラスターがData Infrastructure Insights環境と通信するために使用するプロキシについては、これらの手順をすべて読んだ後、Kubernetes Monitoring Operator をインストールしてください。

Kubernetes Monitoring Operator をデプロイする前に、*operator-config.yaml* の *AgentConfiguration* のプロキシセクションを設定します。

```
agent:
  ...
  proxy:
    server: <server for proxy>
    port: <port for proxy>
    username: <username for proxy>
    password: <password for proxy>

    # In the noproxy section, enter a comma-separated list of
    # IP addresses and/or resolvable hostnames that should bypass
    # the proxy
    noproxy: <comma separated list>

    isTelegrafProxyEnabled: true
    isFluentbitProxyEnabled: <true or false> # true if Events Log enabled
    isCollectorsProxyEnabled: <true or false> # true if Network
Performance and Map enabled
    isAuProxyEnabled: <true or false> # true if AU enabled
  ...
...
```

カスタムまたはプライベート**Docker**リポジトリの使用

デフォルトでは、Kubernetes Monitoring Operator はData Infrastructure Insightsリポジトリからコンテナ イメージをプルします。監視のターゲットとして Kubernetes クラスターが使用されており、そのクラスターがカスタムまたはプライベート Docker リポジトリまたはコンテナー レジストリからのみコンテナー イメージをプルするように構成されている場合は、Kubernetes 監視オペレーターに必要なコンテナーへのアクセスを構成する必要があります。

NetApp Monitoring Operator インストール タイルから「イメージ プル スニペット」を実行します。このコマンドは、 Data Infrastructure Insightsリポジトリにログインし、オペレーターのすべてのイメージ依存関係をプルし、 Data Infrastructure Insightsリポジトリからログアウトします。プロンプトが表示されたら、提供されたリポジトリの一時パスワードを入力します。このコマンドは、オプション機能を含む、オペレータが使用するすべてのイメージをダウンロードします。これらの画像がどの機能に使用されているかについては、以下を参照してください。

コアオペレーター機能とKubernetesモニタリング

- netapp 監視
- ci-kube-rbac-プロキシ
- ci-ksm
- ci-telegraf
- ディストロレスルートユーザー

イベントログ

- ci-fluent-bit
- ci-kubernetes-イベントエクスポーター

ネットワークパフォーマンスとマップ

- ci-net-オブザーバー

企業ポリシーに従って、オペレーターの Docker イメージをプライベート/ローカル/エンタープライズ Docker リポジトリにプッシュします。リポジトリ内のこれらのイメージへのイメージ タグとディレクトリ パスが、Data Infrastructure Insightsリポジトリのものと一致していることを確認します。

operator-deployment.yaml の monitoring-operator デプロイメントを編集し、すべてのイメージ参照を変更してプライベート Docker リポジトリを使用します。

```
image: <docker repo of the enterprise/corp docker repo>/ci-kube-rbac-
proxy:<ci-kube-rbac-proxy version>
image: <docker repo of the enterprise/corp docker repo>/netapp-
monitoring:<version>
```

新しい docker リポジトリの場所を反映するように、*operator-config.yaml* の *AgentConfiguration* を編集します。プライベートリポジトリ用の新しい imagePullSecret を作成します。詳細については、*https://kubernetes.io/docs/tasks/configure-pod-container/pull-image-private-registry/* を参照してください。

```
agent:
  ...
  # An optional docker registry where you want docker images to be pulled
from as compared to CI's docker registry
  # Please see documentation link here:
xref:{relative_path}task_config_telegraf_agent_k8s.html#using-a-custom-or-
private-docker-repository
  dockerRepo: your.docker.repo/long/path/to/test
  # Optional: A docker image pull secret that maybe needed for your
private docker registry
  dockerImagePullSecret: docker-secret-name
```

長期パスワード用の **API** アクセストークン

一部の環境（プロキシリポジトリなど）では、Data Infrastructure Insights docker リポジトリの長期パスワードが必要です。インストール時に UI で提供されるパスワードは 24 時間のみ有効です。それを使用する代わりに、API アクセストークンを docker リポジトリパスワードとして使用できます。このパスワードは、API アクセストークンが有効である限り有効です。この特定の目的のために新しい API アクセストークンを生成することも、既存のものを使用することもできます。

"こちらをお読みください"新しいAPIアクセストークンを作成する手順については、こちらをご覧ください。

ダウンロードした *operator-secrets.yaml* ファイルから既存の API アクセス トークンを抽出するには、ユーザーは以下を実行できます：

```
grep '\.dockerconfigjson' operator-secrets.yaml |sed 's/.*\.dockerconfigjson:
//g' |base64 -d |jq
```

実行中のオペレーターインストールから既存のAPIアクセストークンを抽出するには、ユーザーは以下を実行できます：

```
kubectl -n netapp-monitoring get secret netapp-ci-docker -o
jsonpath='{.data.\.dockerconfigjson}' |base64 -d |jq
```

OpenShift の手順

OpenShift 4.6 以降で実行している場合は、*operator-config.yaml* の *AgentConfiguration* を編集して *runPrivileged* 設定を有効にする必要があります：

```
# Set runPrivileged to true SELinux is enabled on your kubernetes nodes
runPrivileged: true
```

Openshift は、一部の Kubernetes コンポーネントへのアクセスをブロックする可能性のある追加のセキュリティ レベルを実装する場合があります。

寛容と汚点

netapp-ci-telegraf-ds、*netapp-ci-fluent-bit-ds*、および *netapp-ci-net-observer-l4-ds* DaemonSets は、すべてのノードでデータを正しく収集するために、クラスター内のすべてのノードでポッドをスケジュールする必要があります。オペレーターは、いくつかのよく知られた 汚染 を許容するように設定されています。ノードにカスタムテイントを設定して、ポッドがすべてのノードで実行されないようにしている場合は、それらのテイントに対して*許容*を作成できます。"_AgentConfiguration_内" 。クラスター内のすべてのノードにカスタム テイントを適用した場合は、オペレーター ポッドをスケジュールして実行できるように、オペレーター デプロイメントに必要な許容値も追加する必要があります。

Kubernetesについて詳しく知る"汚名と寛容"。

戻る"* NetApp Kubernetes Monitoring Operator のインストール* ページ"

秘密についてのメモ

Kubernetes モニタリング オペレーターがクラスター全体でシークレットを表示する権限を削除するには、インストール前に *operator-setup.yaml* ファイルから次のリソースを削除します。

```
 ClusterRole/netapp-ci<namespace>-agent-secret
 ClusterRoleBinding/netapp-ci<namespace>-agent-secret
```

アップグレードの場合は、クラスターからリソースも削除します。

```
 kubectl delete ClusterRole/netapp-ci-<namespace>-agent-secret-clusterrole
 kubectl delete ClusterRoleBinding/netapp-ci-<namespace>-agent-secret-
clusterrolebinding
```

変更分析が有効になっている場合は、*AgentConfiguration* または *operator-config.yaml* を変更して、変更管理セクションのコメントを解除し、変更管理セクションの下に *kindsToIgnoreFromWatch: '"secrets"'* を含めます。この行における一重引用符と二重引用符の存在と位置に注意してください。

```
change-management:
  ...
  # # A comma separated list of kinds to ignore from watching from the
default set of kinds watched by the collector
  # # Each kind will have to be prefixed by its apigroup
  # # Example: '"networking.k8s.io.networkpolicies,batch.jobs",
"authorization.k8s.io.subjectaccessreviews"'
  kindsToIgnoreFromWatch: '"secrets"'
  ...
```

Kubernetes モニタリング オペレーター イメージ署名の検証

オペレータのイメージとそれが展開するすべての関連イメージは、 NetAppによって署名されています。インストール前に cosign ツールを使用してイメージを手動で検証したり、Kubernetes アドミッション コントローラーを構成したりすることができます。詳細については、"Kubernetesドキュメント" 。

イメージ署名の検証に使用される公開鍵は、モニタリング オペレーターのインストール タイルの「オプション: オペレーター イメージをプライベート リポジトリにアップロード > イメージ署名公開鍵」で入手できます。

イメージ署名を手動で検証するには、次の手順を実行します。

1. 画像プルスニペットをコピーして実行する
2. リポジトリパスワードをコピーしてプロンプトが表示されたら入力します
3. イメージ署名公開鍵（例では dii-image-signing.pub）を保存します。
4. cosign を使用してイメージを検証します。共同署名の使用例を参照してください

```
$ cosign verify --key dii-image-signing.pub --insecure-ignore-sct
--insecure-ignore-tlog <repository>/<image>:<tag>
Verification for <repository>/<image>:<tag> --
The following checks were performed on each of these signatures:
  - The cosign claims were validated
  - The signatures were verified against the specified public key
[{"critical":{"identity":{"docker-
reference":"<repository>/<image>"},"image":{"docker-manifest-
digest":"sha256:<hash>"},"type":"cosign container image
signature"},"optional":null}]
```

トラブルシューティング

Kubernetes モニタリング オペレーターの設定中に問題が発生した場合に試すことは次のとおりです。

問題：	これを試してください:
Kubernetes 永続ボリュームと対応するバックエンド ストレージ デバイス間のハイパーリンク/接続が表示されません。私の Kubernetes 永続ボリュームは、ストレージ サーバーのホスト名を使用して構成されています。	手順に従って既存の Telegraf エージェントをアンインストールし、最新の Telegraf エージェントを再インストールします。 Telegraf バージョン 2.0 以降を使用している必要があり、Kubernetes クラスター ストレージがData Infrastructure Insightsによってアクティブに監視されている必要があります。

問題：	これを試してください:
ログには次のようなメッセージが表示されます: E0901 15:21:39.962145 1 reflector.go:178] k8s.io/kube-state-metrics/internal/store/builder.go:352: *v1.MutatingWebhookConfiguration の一覧を取得できませんでした: サーバーは要求されたリソースを見つけることができませんでした E0901 15:21:43.168161 1 reflector.go:178] k8s.io/kube-state-metrics/internal/store/builder.go:352: *v1.Lease の一覧を取得できませんでした: サーバーは要求されたリソースを見つけることができませんでした (get leases.coordination.k8s.io) など。	これらのメッセージは、Kubernetes バージョン 1.20 未満で kube-state-metrics バージョン 2.0.0 以上を実行している場合に表示されることがあります。 Kubernetes のバージョンを取得するには: *kubectl version* kube-state-metrics のバージョンを取得するには: *kubectl get deploy/kube-state-metrics -o jsonpath='{..image}'* これらのメッセージが表示されないようにするには、ユーザーは kube-state-metrics デプロイメントを変更して、次のリースを無効にすることができます: *mutatingwebhookconfigurations validatingwebhookconfigurations volumeattachments resources* 具体的には、次の CLI 引数を使用できます: resources=certificatesigningrequests,configmaps,cronjobs,daemonsets, deployments,endpoints,horizontalpodautoscalers,ingresses,jobs,limitranges, namespaces,networkpolicies,nodes,persistentvolumeclaims,persistentvolumes, poddisruptionbudgets,pods,replicasets,replicationcontrollers,resourcequotas, secrets、services 、statefulsets、storageclasses デフォルトのリソースリストは次のとおりです: "certificatesigningrequests、configmaps、cronjobs、daemonsets、deployments、endpoints、horizontalpodautoscalers、ingresses、jobs、leases、limitranges 、mutatingwebhookconfigurations、namespaces、networkpolicies、nodes、persistentvolumeclaims、persistentvolumes、poddisruptionbudgets、pods、replicasets、replicationcontrollers、resourcequotas、secrets 、services、statefulsets、storageclasses、validatingwebhookconfigurations、volumeattachments"
Telegraf から次のようなエラー メッセージが表示されますが、Telegraf は起動して実行されます: Oct 11 14:23:41 ip-172-31-39-47 systemd[1]: Started The plugin-driven server agent for reporting metrics into InfluxDB. 10月11日 14:23:41 ip-172-31-39-47 telegraf[1827]: time="2021-10-11T14:23:41Z" level=error msg="キャッシュディレクトリの作成に失敗しました。 /etc/telegraf/.cache/snowflake、err: mkdir /etc/telegraf/.ca che: 権限が拒否されました。無視されました\n" func="gosnowflake.(*defaultLogger).Errorf" file="log.go:120" Oct 11 14:23:41 ip-172-31-39-47 telegraf[1827]: time="2021-10-11T14:23:41Z" level=error msg="開けませんでした。無視されました。open /etc/telegraf/.cache/snowflake/ocsp_response_cache.json: そのようなファイルまたはディレクトリはありません\n" func="gosnowflake.(*defaultLogger).Errorf" file="log.go:120" Oct 11 14:23:41 ip-172-31-39-47 telegraf[1827]: 2021-10-11T14:23:41Z I! Telegraf 1.19.3 の起動	これは既知の問題です。参照"このGitHubの記事"詳細についてはこちらをご覧ください。 Telegraf が稼働している限り、ユーザーはこれらのエラー メッセージを無視できます。

問題：	これを試してください:
Kubernetes では、Telegraf ポッドが次のエラーを報告しています: 「mountstats 情報の処理中にエラーが発生しました: mountstats ファイルを開けませんでした: /hostfs/proc/1/mountstats、エラー: open /hostfs/proc/1/mountstats: 権限が拒否されました」	SELinux が有効になっていて強制されている場合、Telegraf ポッドが Kubernetes ノード上の /proc/1/mountstats ファイルにアクセスできない可能性があります。この制限を克服するには、エージェント構成を編集し、runPrivileged 設定を有効にします。詳細については、OpenShift の手順を参照してください。
Kubernetes では、Telegraf ReplicaSet ポッドが次のエラーを報告しています: [inputs.prometheus] プラグインのエラー: キーペア /etc/kubernetes/pki/etcd/server.crt をロードできませんでした:/etc/kubernetes/pki/etcd/server.key: open /etc/kubernetes/pki/etcd/server.crt: そのようなファイルまたはディレクトリはありません	Telegraf ReplicaSet ポッドは、マスターまたは etcd として指定されたノード上で実行されることを目的としています。 ReplicaSet ポッドがこれらのノードのいずれかで実行されていない場合は、これらのエラーが発生します。マスター/etcd ノードに taint があるかどうかを確認します。そうなる場合は、Telegraf ReplicaSet (telegraf-rs) に必要な許容範囲を追加します。たとえば、ReplicaSet を編集します… kubectl edit rs telegraf-rs …そして、適切な許容値を仕様に追加します。次に、ReplicaSet ポッドを再起動します。
PSP/PSA環境があります。これは監視オペレーターに影響しますか?	Kubernetes クラスターが Pod Security Policy (PSP) または Pod Security Admission (PSA) を適用した状態で実行されている場合は、最新の Kubernetes Monitoring Operator にアップグレードする必要があります。 PSP/PSA をサポートする現在の Operator にアップグレードするには、次の手順に従います。1. アンインストール以前の監視オペレーター: kubectl delete agent agent-monitoring-netapp -n netapp-monitoring kubectl delete ns netapp-monitoring kubectl delete crd agents.monitoring.netapp.com kubectl delete clusterrole agent-manager-role agent-proxy-role agent-metrics-reader kubectl delete clusterrolebinding agent-manager-rolebinding agent-proxy-rolebinding agent-cluster-admin-rolebinding 2. インストール監視オペレータの最新バージョン。
PSP/PSA を使用しているのですが、Operator を展開しようとして問題が発生しました。	1.次のコマンドを使用してエージェントを編集します: kubectl -n <name-space> edit agent 2. 「security-policy-enabled」を「false」としてマークします。これにより、ポッド セキュリティ ポリシーとポッド セキュリティ アドミッションが無効になり、オペレーターがデプロイできるようになります。次のコマンドを使用して確認します: kubectl get psp (Pod Security Policy が削除されたことが表示されます) kubectl get all -n <namespace>
grep -i psp (何も見つからないことが表示されます)	「ImagePullBackoff」エラーが発生
これらのエラーは、カスタムまたはプライベートの Docker リポジトリがあり、Kubernetes モニタリング オペレーターがそれを適切に認識するようにまだ構成していない場合に表示されることがあります。詳細はこちらカスタム/プライベート リポジトリの構成について。	モニタリング オペレーターのデプロイメントで問題が発生していますが、現在のドキュメントでは解決できません。

問題：	これを試してください:
次のコマンドの出力をキャプチャまたはメモして、テクニカル サポート チームに連絡してください。 ``` kubectl -n netapp-monitoring get all kubectl -n netapp-monitoring describe all kubectl -n netapp-monitoring logs <monitoring-operator-pod> --all-containers=true kubectl -n netapp-monitoring logs <telegraf-pod> --all-containers=true ```	Operator 名前空間の net-observer (ワークロード マップ) ポッドは CrashLoopBackOff にあります
これらのポッドは、ネットワーク可観測性のワークロード マップ データ コレクターに対応します。以下を試してください: • いずれかのポッドのログをチェックして、最小カーネル バージョンを確認します。例: ---- {"ci-tenant-id":"your-tenant-id","collector-cluster":"your-k8s-cluster-name","environment":"prod","level":"error","msg":"検証に失敗しました。理由: カーネル バージョン 3.10.0 は、最小カーネル バージョン 4.18.0 より小さいです","time":"2022-11-09T08:23:08Z"} ---- • Net-observer ポッドでは、Linux カーネル バージョンが少なくとも 4.18.0 である必要があります。 「uname -r」コマンドを使用してカーネルバージョンを確認し、4.18.0以上であることを確認します。	ポッドはオペレーター名前空間（デフォルト：netapp-monitoring）で実行されていますが、ワークロードマップのUIやクエリのKubernetesメトリックにデータが表示されません。
K8S クラスターのノード上の時刻設定を確認します。正確な監査とデータ レポートを実現するために、ネットワーク タイム プロトコル (NTP) または簡易ネットワーク タイム プロトコル (SNTP) を使用してエージェント マシンの時刻を同期することを強くお勧めします。	オペレーター名前空間内の一部のネットオブザーバーポッドが保留状態になっています
Net-observer は DaemonSet であり、k8s クラスターの各ノードでポッドを実行します。 • 保留中の状態のポッドに注意し、CPU またはメモリのリソースの問題が発生しているかどうかを確認します。ノードで必要なメモリと CPU が使用可能であることを確認します。	Kubernetes モニタリング オペレーターをインストールした直後、ログに次の内容が表示されます: [inputs.prometheus] プラグインでエラーが発生しました: http://kube-state-metrics.<namespace>.svc.cluster.local:8080/metrics への HTTP リクエストの作成エラー: http://kube-state-metrics.<namespace>.svc.cluster.local:8080/metrics を取得: tcp をダイヤル: kube-state-metrics.<namespace>.svc.cluster.local を検索: そのようなホストはありません

問題：	これを試してください:
このメッセージは通常、新しいオペレータがインストールされ、*ksm* ポッドが起動する前に *telegraf-rs* ポッドが起動している場合にのみ表示されます。すべてのポッドが実行されると同時に、これらのメッセージは停止します。	クラスター内に存在する Kubernetes CronJobs に対して収集されているメトリックが表示されません。
Kubernetesのバージョンを確認してください（つまり `kubectl version`）。 v1.20.x 以下の場合、これは予想される制限です。 Kubernetes Monitoring Operator とともにデプロイされた kube-state-metrics リリースは、v1.CronJob のみをサポートします。 Kubernetes 1.20.x 以下では、CronJob リソースは v1beta.CronJob にあります。その結果、kube-state-metrics は CronJob リソースを見つけることができません。	オペレーターをインストールすると、telegraf-ds ポッドは CrashLoopBackOff 状態になり、ポッド ログに「su: 認証失敗」と表示されます。
AgentConfiguration の telegraf セクションを編集し、*dockerMetricCollectionEnabled* を false に設定します。詳細については、オペレーターの"設定オプション"を参照してください。… spec：… telegraf：…-name：docker run-mode：- DaemonSet substitutions：- key：DOCKER_UNIX_SOCK_PLACEHOLDER value：unix:///run/docker.sock ……	Telegraf ログに次のようなエラー メッセージが繰り返し表示されます: E! [エージェント] 出力への書き込みエラー: http: Post "https://<tenant_url>/rest/v1/lake/ingest/influxdb": コンテキストの期限が切れました (ヘッダーの待機中に Client.Timeout を超えました)
AgentConfiguration の telegraf セクションを編集し、*outputTimeout* を 10 秒に増やします。詳細については、オペレーターの"設定オプション"。	一部のイベント ログの *involvedobject* データが見つかりません。
必ず、"権限"上記のセクション。	netapp-ci-monitoring-operator-<pod> と monitoring-operator-<pod> という名前の 2 つの監視オペレータ ポッドが実行されているのはなぜですか?
2023年10月12日現在、 Data Infrastructure Insights は、ユーザーへのサービス向上のため、オペレーターをリファクタリングしました。これらの変更を完全に適用するには、古い演算子を削除するそして新しいものをインストールする。	Kubernetes イベントが予期せずData Infrastructure Insightsへのレポートを停止しました。
イベント エクスポーター ポッドの名前を取得します。 `kubectl -n netapp-monitoring get pods`	grep event-exporter

問題：	これを試してください:
awk '{print $1}'	sed 's/event-exporter./event-exporter/'` 「netapp-ci-event-exporter」または「event-exporter」のいずれかである必要があります。次に、監視エージェントを編集します `kubectl -n netapp-monitoring edit agent`、LOG_FILE の値を、前の手順で見つかった適切なイベント エクスポーター ポッド名を反映するように設定します。具体的には、LOG_FILE は「/var/log/containers/netapp-ci-event-exporter.log」または「/var/log/containers/event-exporter*.log」のいずれかに設定する必要があります。 …. fluent-bit: … - name: event-exporter-ci substitutions: - key: LOG_FILE values: - /var/log/containers/netapp-ci-event-exporter*.log … …. あるいは、uninstallそして再インストールエージェント。
Kubernetes モニタリング オペレーターによってデプロイされたポッドが、リソース不足のためにクラッシュしているのがわかります。	Kubernetesモニタリングオペレーターを参照してください"設定オプション"必要に応じて CPU および/またはメモリの制限を増やします。
イメージが欠落しているか、構成が無効であるため、netapp-ci-kube-state-metrics ポッドの起動または準備ができませんでした。現在、StatefulSet はスタックしており、構成の変更が netapp-ci-kube-state-metrics ポッドに適用されていません。	StatefulSetは"壊れた"州。構成の問題を修正したら、netapp-ci-kube-state-metrics ポッドをバウンスします。
netapp-ci-kube-state-metrics ポッドは、Kubernetes Operator のアップグレードを実行した後に起動に失敗し、ErrImagePull (イメージのプルに失敗します) をスローします。	ポッドを手動でリセットしてみてください。
ログ分析の Kubernetes クラスターで、「イベントは maxEventAgeSeconds より古いため破棄されました」というメッセージが表示されています。	Operator の *agentconfiguration* を変更し、*event-exporter-maxEventAgeSeconds* (つまり 60 秒)、*event-exporter-kubeQPS* (つまり 100)、および *event-exporter-kubeBurst* (つまり 500) を増やします。これらの設定オプションの詳細については、"設定オプション"ページ。

問題：	これを試してください:
Telegraf は、ロック可能なメモリが不足しているために警告を発したりクラッシュしたりします。	基盤となるオペレーティング システム/ノードで Telegraf のロック可能なメモリの制限を増やしてみてください。制限を増やすことができない場合は、NKMO エージェント構成を変更し、*unprotected* を *true* に設定します。これにより、Telegraf はロックされたメモリ ページを予約しないように指示されます。復号化された秘密がディスクにスワップアウトされる可能性があるため、セキュリティ上のリスクが生じる可能性がありますが、ロックされたメモリを予約できない環境での実行が可能になります。*unprotected* 設定オプションの詳細については、"設定オプション"ページ。
Telegraf から次のような警告メッセージが表示されます: *W! [inputs.diskio] "vdc"* のディスク名を収集できません*: /dev/vdc* の読み取りエラー*:* そのようなファイルまたはディレクトリはありません	Kubernetes Monitoring Operator の場合、これらの警告メッセージは無害であり、無視しても問題ありません。または、AgentConfiguration の telegraf セクションを編集し、*runDsPrivileged* を true に設定します。詳細については、"オペレータの設定オプション"を参照してください。

問題：	これを試してください:
Fluent-bit ポッドが次のエラーで失敗しています: [2024/10/16 14:16:23] [error] [/src/fluent-bit/plugins/in_tail/tail_fs_inotify.c:360 errno=24] 開いているファイルが多すぎます [2024/10/16 14:16:23] [error] 入力 tail.0 の初期化に失敗しました [2024/10/16 14:16:23] [error] [engine] 入力の初期化に失敗しました	クラスター内の *fsnotify* 設定を変更してみます。 ``` sudo sysctl fs.inotify.max_user_instances (take note of setting) sudo sysctl fs.inotify.max_user_instances=<something larger than current setting> sudo sysctl fs.inotify.max_user_watches (take note of setting) sudo sysctl fs.inotify.max_user_watches=<something larger than current setting> ``` Fluent-bit を再起動します。 注意: これらの設定をノードの再起動後も維持するには、*/etc/sysctl.conf* に次の行を追加する必要があります。 ``` fs.inotify.max_user_instances=<something larger than current setting> fs.inotify.max_user_watches=<something larger than current setting> ```
Telegraf DS ポッドは、TLS 証明書を検証できないために kubernetes 入力プラグインが HTTP リクエストを実行できないことに関連するエラーを報告しています。例: E! [inputs.kubernetes] プラグインのエラー: HTTPリクエストの送信中にエラーが発生しました"<a href="https://<kubelet_IP>:10250/stats/summary":" class="bare">https://<kubelet_IP>:10250/stats/summary":得る"<a href="https://<kubelet_IP>:10250/stats/summary":" class="bare">https://<kubelet_IP>:10250/stats/summary":tls: 証明書の検証に失敗しました: x509: IP SANが含まれていないため、<kubelet_IP>の証明書を検証できません	これは、kubelet が自己署名証明書を使用している場合、および/または指定された証明書の証明書の *Subject Alternative Name* リストに <kubelet_IP> が含まれていない場合に発生します。これを解決するには、ユーザーは"エージェント構成"、*telegraf:insecureK8sSkipVerify* を *true* に設定します。これにより、Telegraf 入力プラグインが検証をスキップするように設定されます。あるいは、ユーザーはkubeletを次のように設定することができます。"サーバーTLSブートストラップ"これにより、「certificates.k8s.io」API からの証明書要求がトリガーされます。

問題：	これを試してください:
Fluent-bit ポッドで次のエラーが発生し、ポッドを起動できません：026/01/12 20:20:32] [error] [sqldb] error=unable to open database file [2026/01/12 20:20:32] [error] [input:tail:tail.0] db: could not create 'in_tail_files' table [2026/01/12 20:20:32] [error] [input:tail:tail.0] could not open/create database [2026/01/12 20:20:32] [error] failed initialize input tail.0 [2026/01/12 20:20:32] [error] [engine] input initialization failed	DBファイルが存在するホストディレクトリに適切な読み取り / 書き込み権限があることを確認してください。具体的には、ホストディレクトリは非ルートユーザーに読み取り / 書き込み権限を付与する必要があります。デフォルトのDBファイルの場所は、fluent-bit-dbFile *agentconfiguration* オプションで上書きされない限り、/var/log/ です。SELinuxが有効になっている場合は、fluent-bit-seLinuxOptionsType agentconfigurationオプションを 'spc_t' に設定してみてください。

追加情報は以下からご覧いただけます。"サポート"ページまたは"データコレクターサポートマトリックス"。

Memcached データコレクター

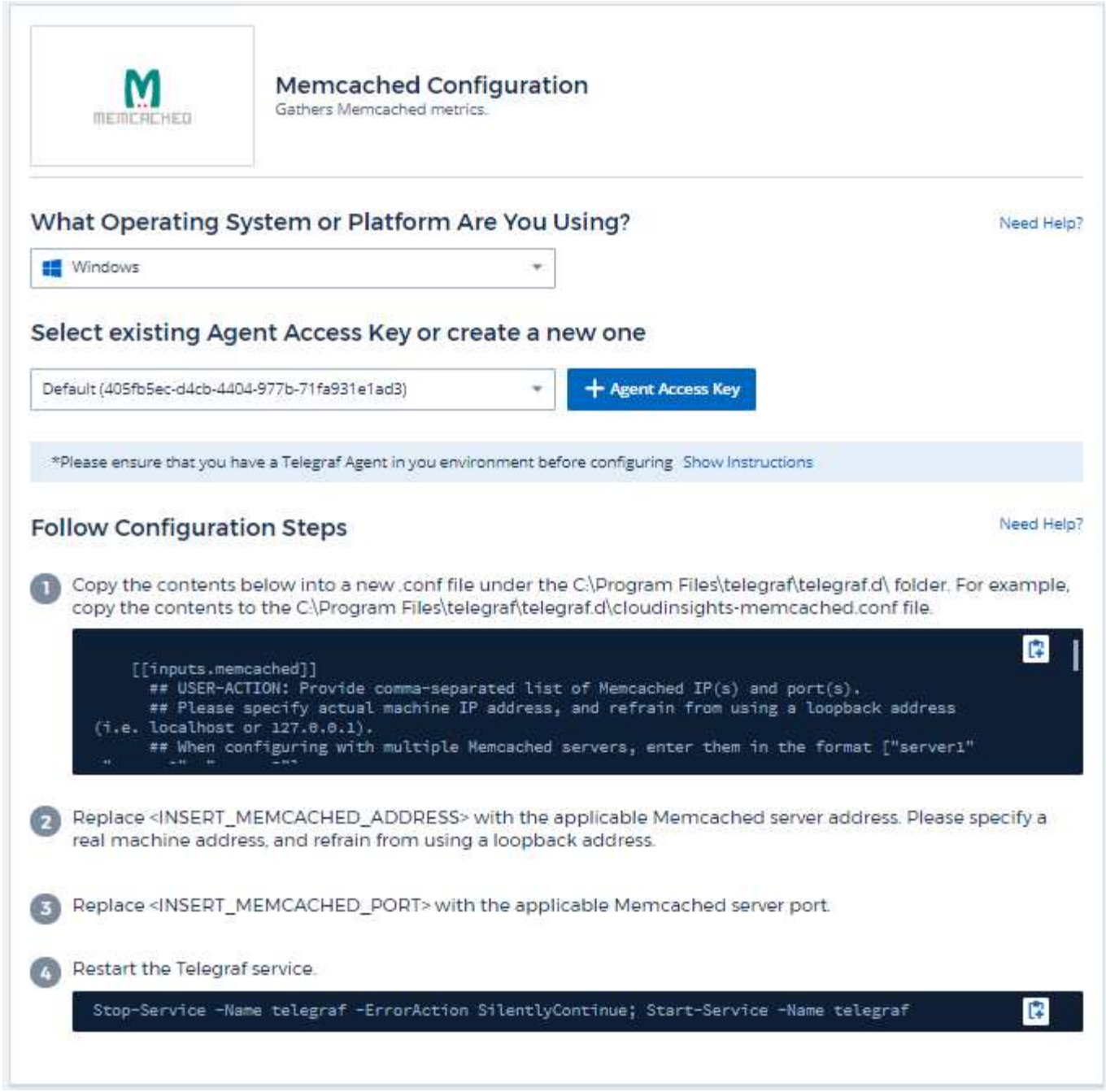
Data Infrastructure Insights は、このデータ コレクターを使用して Memcached からメトリックを収集します。

インストール

1. **Observability > Collectors** から、**+ Data Collector** をクリックします。 Memcached を選択します。

   Telegraf エージェントがインストールされているオペレーティング システムまたはプラットフォームを選択します。

2. 収集用のエージェントをまだインストールしていない場合、または別のオペレーティングシステムまたはプラットフォーム用のエージェントをインストールする場合は、[手順を表示] をクリックして展開します。"エージェントのインストール"説明書。
3. このデータ コレクターで使用するエージェント アクセス キーを選択します。 **+** エージェント アクセス キー ボタンをクリックすると、新しいエージェント アクセス キーを追加できます。ベスト プラクティス: データ コレクターを OS/プラットフォーム別などにグループ化する場合にのみ、異なるエージェント アクセス キーを使用します。
4. 構成手順に従ってデータ コレクターを構成します。手順は、データ収集に使用しているオペレーティング システムまたはプラットフォームの種類によって異なります。



セットアップ

詳細は以下をご覧ください。"Memcached ウィキ" 。

オブジェクトとカウンター

次のオブジェクトとそのカウンターが収集されます。

オブジェクト	識別子:	属性:	データポイント:
メムキャッシュ	名前空間サーバー	ノードIPノード名	受け入れる接続 処理された認証要求 失敗した認証 使用バイト数 読み取りバイト数 (1 秒あたり) 書き込みバイト数 (1 秒あたり) CAS 無効 CAS ヒット数 CAS ミス数 フラッシュ要求数 (1 秒あたり) 取得要求数 (1 秒あたり) セット要求数 (1 秒あたり) タッチ要求数 (1 秒あたり) 接続解放数 (1 秒あたり) 接続構造 オープン接続 現在保存されているアイテム 減少要求ヒット数 (1 秒あたり) 減少要求ミス数 (1 秒あたり) 削除要求ヒット数 (1 秒あたり) 削除要求ミス数 (1 秒あたり) 追い出されたアイテム 有効な追い出し 期限切れアイテム 取得ヒット数 (1 秒あたり) 取得ミス数 (1 秒あたり) 使用ハッシュバイト数 ハッシュが拡張中 ハッシュ電力レベル 増加要求ヒット数 (1 秒あたり) 増加要求ミス数 (1 秒あたり) サーバー最大バイト数 リッスン無効 再利用されたワーカースレッド数 オープン接続合計数 保存されたアイテム合計数 タッチヒット数 タッチミス数 サーバー稼働時間

トラブルシューティング

追加情報は以下からご覧いただけます。"サポート"ページ。

MongoDB データコレクター

Data Infrastructure Insights は、このデータ コレクターを使用して MongoDB からメトリックを収集します。

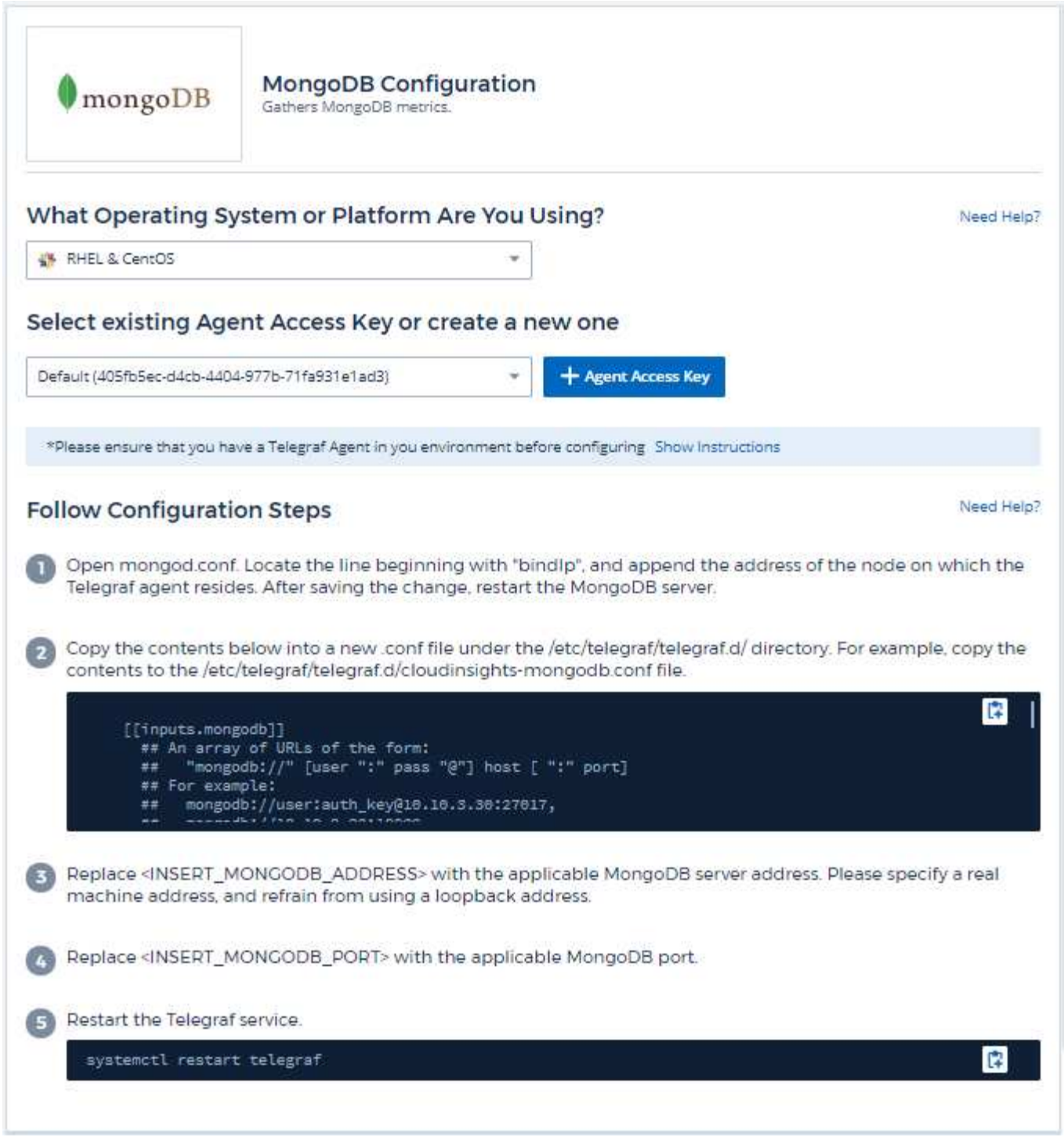
インストール

1. **Observability > Collectors** から、**+ Data Collector** をクリックします。 MongoDB を選択します。

   Telegraf エージェントがインストールされているオペレーティング システムまたはプラットフォームを選

択します。

2. 収集用のエージェントをまだインストールしていない場合、または別のオペレーティングシステムまたはプラットフォーム用のエージェントをインストールする場合は、[手順を表示] をクリックして展開します。"エージェントのインストール"説明書。
3. このデータ コレクターで使用するエージェント アクセス キーを選択します。 + エージェント アクセス キー ボタンをクリックすると、新しいエージェント アクセス キーを追加できます。ベスト プラクティス: データ コレクターを OS/プラットフォーム別などにグループ化する場合にのみ、異なるエージェント アクセス キーを使用します。
4. 構成手順に従ってデータ コレクターを構成します。手順は、データ収集に使用しているオペレーティング システムまたはプラットフォームの種類によって異なります。



セットアップ

詳細は以下をご覧ください。"MongoDBのドキュメント" 。

オブジェクトとカウンター

次のオブジェクトとそのカウンターが収集されます。

オブジェクト	識別子:	属性:	データポイント:
MongoDB	名前空間ホスト名		
MongoDB データベース	名前空間 ホスト名 データベース名		

トラブルシューティング

情報は以下から入手できます。"サポート"ページ。

MySQL データコレクター

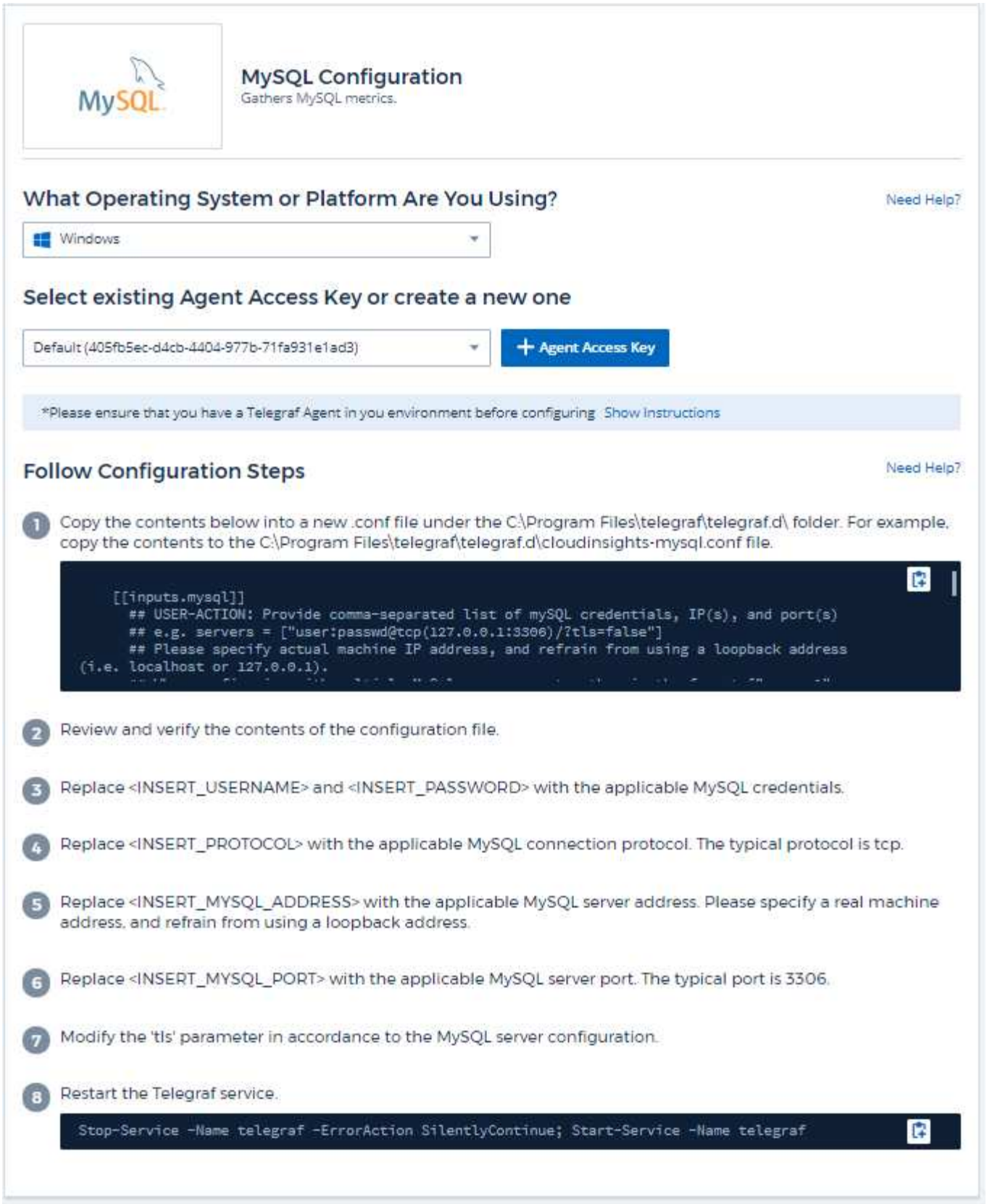
Data Infrastructure Insights は、このデータ コレクターを使用して MySQL からメトリックを収集します。

インストール

1. **Observability > Collectors** から、**+ Data Collector** をクリックします。 MySQLを選択します。

   Telegraf エージェントがインストールされているオペレーティング システムまたはプラットフォームを選択します。

2. 収集用のエージェントをまだインストールしていない場合、または別のオペレーティングシステムまたはプラットフォーム用のエージェントをインストールする場合は、[手順を表示] をクリックして展開します。"エージェントのインストール"説明書。
3. このデータ コレクターで使用するエージェント アクセス キーを選択します。 **+** エージェント アクセス キー ボタンをクリックすると、新しいエージェント アクセス キーを追加できます。ベスト プラクティス: データ コレクターを OS/プラットフォーム別などにグループ化する場合にのみ、異なるエージェント アクセス キーを使用します。
4. 構成手順に従ってデータ コレクターを構成します。手順は、データ収集に使用しているオペレーティング システムまたはプラットフォームの種類によって異なります。



セットアップ

詳細は以下をご覧ください。"MySQLドキュメント" 。

オブジェクトとカウンター

次のオブジェクトとそのカウンターが収集されます。

オブジェクト	識別子:	属性:	データポイント:
MySQL	名前空間 MySQL サーバー	ノードIPノード名	中止されたクライアント数（1秒あたり） 中止された接続数（1秒あたり） 受信バイト数（1秒あたり） 送信バイト数（1秒あたり） 管理コマンド数（1秒あたり） イベント変更コマンド 関数変更コマンド インスタンス変更コマンド プロシージャ変更コマンド サーバー変更コマンド テーブル変更コマンド テーブルスペース変更コマンド ユーザー変更コマンド 分析コマンド キーキャッシュへの割り当てコマンド 開始コマンド バイナリログコマンド プロシージャ呼び出しコマンド DB変更コマンド マスター変更コマンド レプリケーションフィルター変更コマンド チェックコマンド チェックサムコマンド コミットコマンド DB作成コマンド イベント作成コマンド 関数作成コマンド インデックス作成コマンド プロシージャ作成コマンド サーバー作成コマンド テーブル作成コマンド トリガー作成コマンド UDF作成コマンド ユーザー作成コマンド ビュー作成コマンド SQL接続エラーの割り当て解除 作成された一時ディスクテーブルの受け入れ 遅延エラー フラッシュコマンド ハンドラーコミット Innodbバッファプールバイトデータ フラッシュされていないキーブロック キー読み取り要求 キー書き込み要求 キー書き込み最大実行時間使用中の接続の最大数を超えました 開いているファイル パフォーマンス スキーマ アカウント 失われました 準備されたステートメントの数 Qcache 空きブロック クエリ 質問 フル結合の選択 フル範囲結合の選択 範

トラブルシューティング

追加情報は以下からご覧いただけます。"サポート"ページ。

Netstat データコレクター

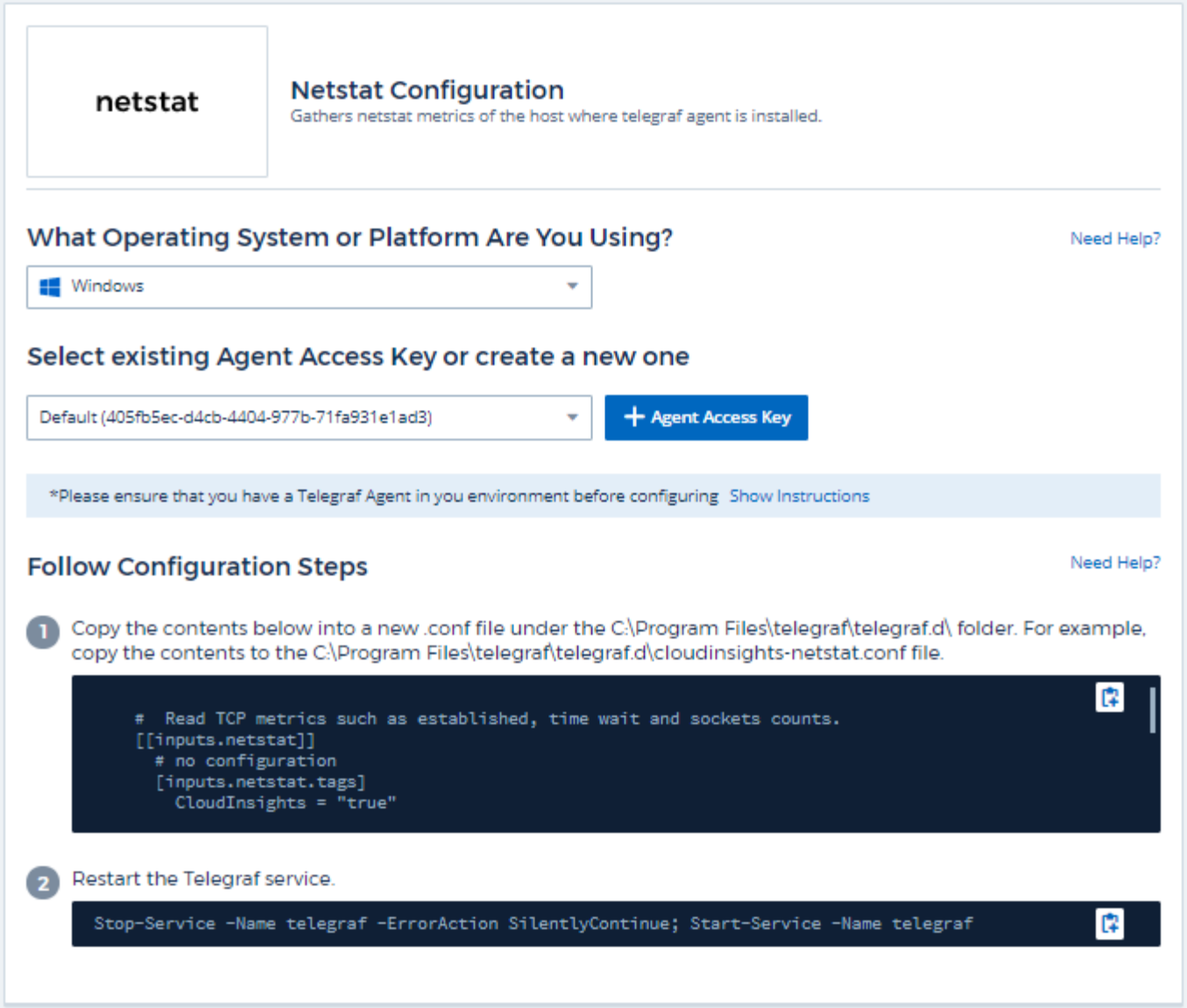
Data Infrastructure Insights は、このデータ コレクターを使用して Netstat メトリックを収集します。

インストール

1. **Observability > Collectors** から、**+ Data Collector** をクリックします。 Netstat を選択します。

   Telegraf エージェントがインストールされているオペレーティング システムまたはプラットフォームを選択します。

2. 収集用のエージェントをまだインストールしていない場合、または別のオペレーティングシステムまたはプラットフォーム用のエージェントをインストールする場合は、[手順を表示] をクリックして展開します。"エージェントのインストール"説明書。
3. このデータ コレクターで使用するエージェント アクセス キーを選択します。 **+** エージェント アクセス キー ボタンをクリックすると、新しいエージェント アクセス キーを追加できます。ベスト プラクティス: データ コレクターを OS/プラットフォーム別などにグループ化する場合にのみ、異なるエージェント アクセス キーを使用します。
4. 構成手順に従ってデータ コレクターを構成します。手順は、データ収集に使用しているオペレーティング システムまたはプラットフォームの種類によって異なります。



セットアップ

オブジェクトとカウンター

次のオブジェクトとそのカウンターが収集されます。

オブジェクト	識別子:	属性:	データポイント:
ネットスタット	ノードUUID	ノードIPノード名	

トラブルシューティング

追加情報は以下からご覧いただけます。"サポート"ページ。

Nginx データコレクター

Data Infrastructure Insights は、このデータ コレクターを使用して Nginx からメトリッ

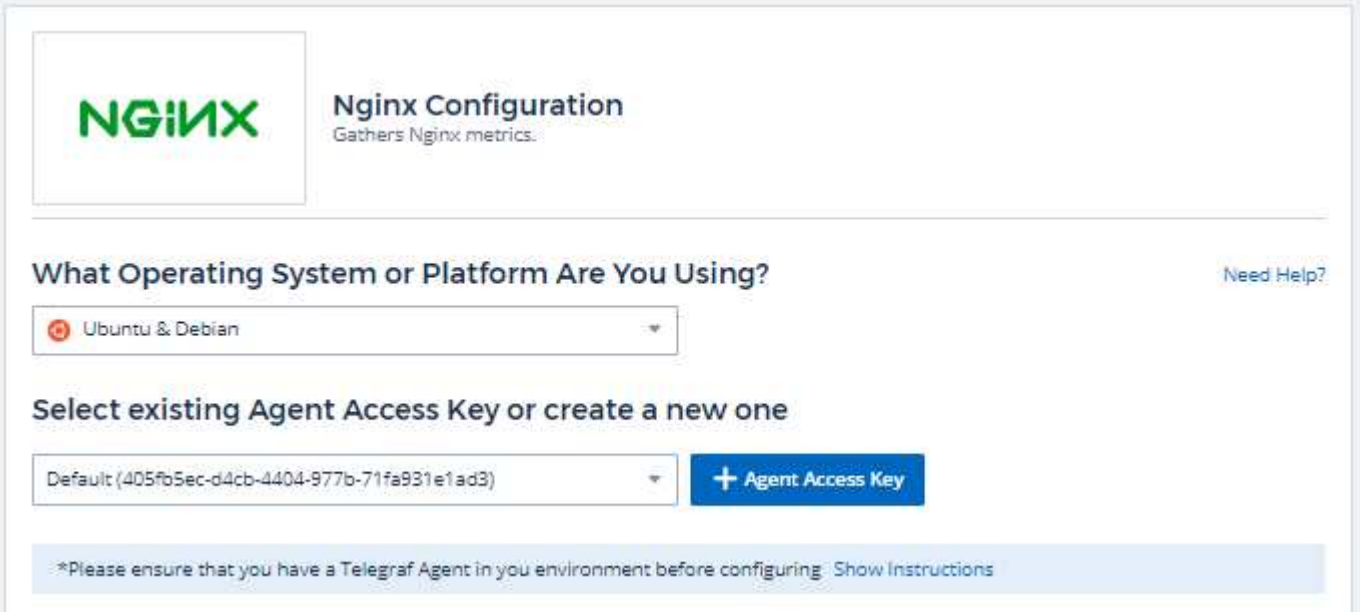
クを収集します。

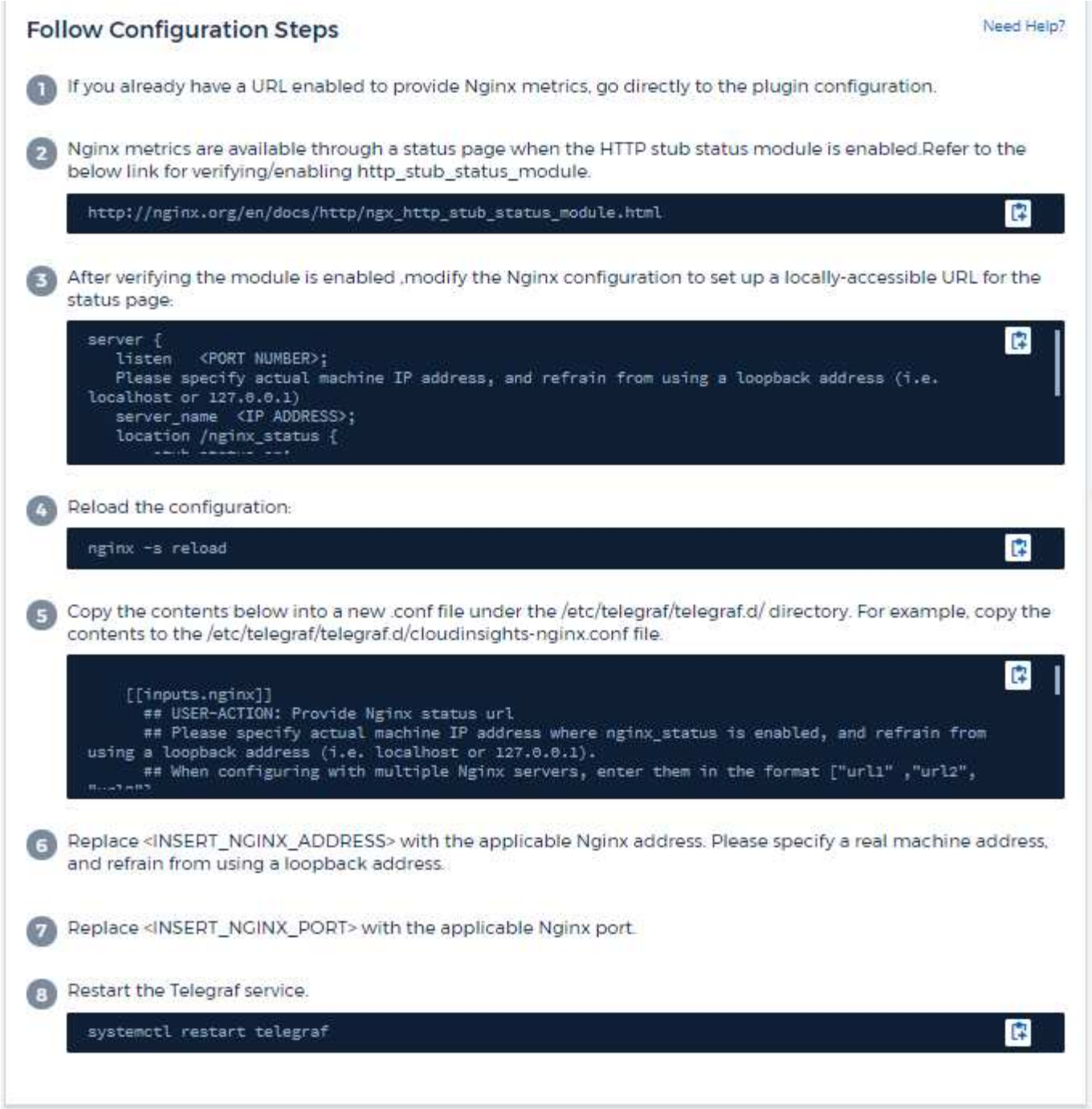
インストール

1. **Observability > Collectors** から、**+ Data Collector** をクリックします。 Nginx を選択します。

   Telegraf エージェントがインストールされているオペレーティング システムまたはプラットフォームを選択します。

2. 収集用のエージェントをまだインストールしていない場合、または別のオペレーティングシステムまたはプラットフォーム用のエージェントをインストールする場合は、[手順を表示] をクリックして展開します。"エージェントのインストール"説明書。
3. このデータ コレクターで使用するエージェント アクセス キーを選択します。 **+** エージェント アクセス キー ボタンをクリックすると、新しいエージェント アクセス キーを追加できます。ベスト プラクティス: データ コレクターを OS/プラットフォーム別などにグループ化する場合にのみ、異なるエージェント アクセス キーを使用します。
4. 構成手順に従ってデータ コレクターを構成します。手順は、データ収集に使用しているオペレーティング システムまたはプラットフォームの種類によって異なります。





セットアップ

Nginxのメトリック収集には、Nginx"http_stub_status_module"有効になります。

追加情報は以下をご覧ください。"Nginxのドキュメント" 。

オブジェクトとカウンター

次のオブジェクトとそのカウンターが収集されます。

オブジェクト	識別子:	属性:	データポイント:
Nginx	名前空間サーバー	ノードIP ノード名 ポート	アクティブに処理された読み取り要求を受け入れ、書き込みを待機

トラブルシューティング

追加情報は以下からご覧いただけます。"サポート"ページ。

PostgreSQL データコレクター

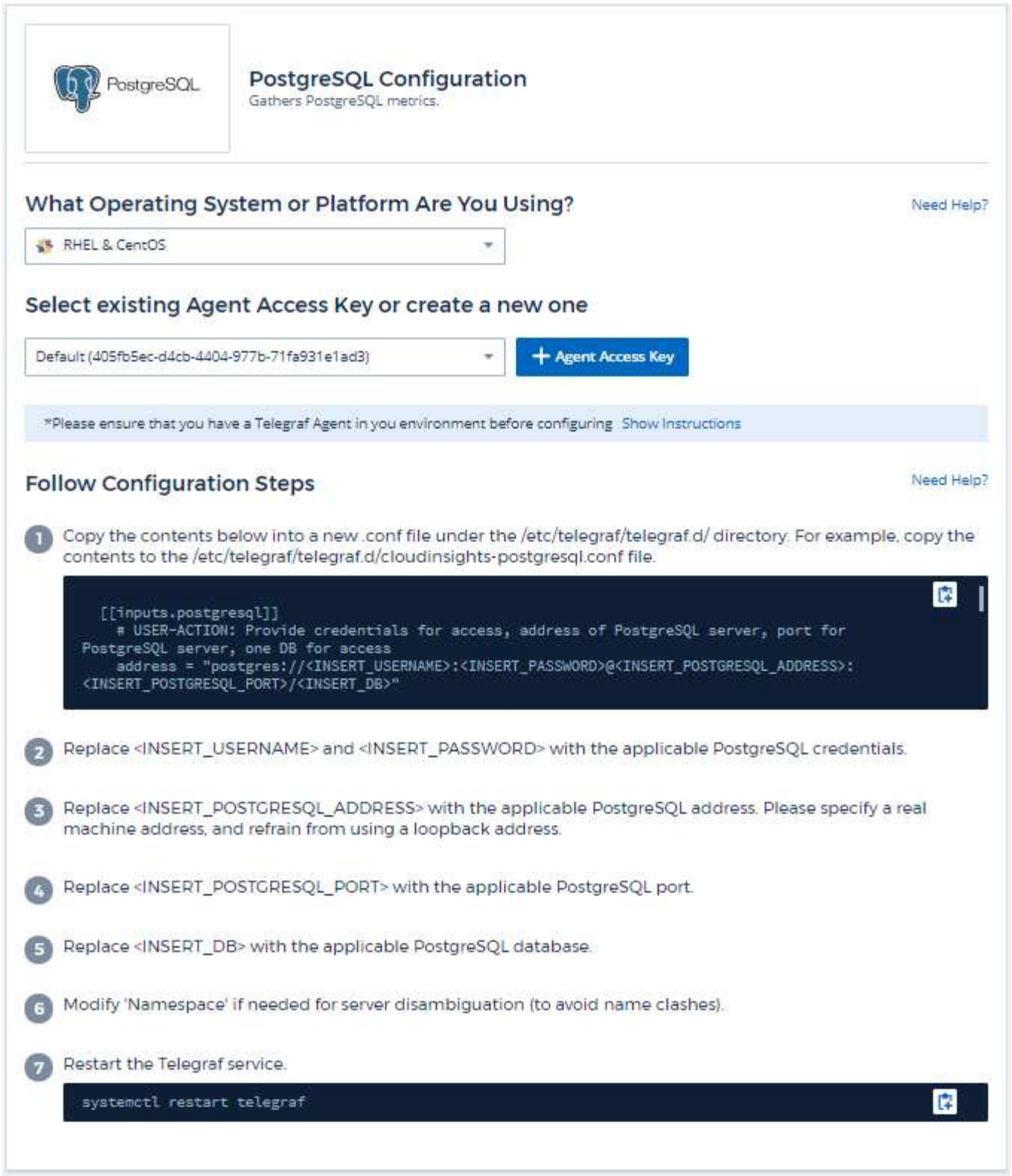
Data Infrastructure Insights は、このデータ コレクターを使用して PostgreSQL からメトリックを収集します。

インストール

1. **Observability > Collectors** から、**+ Data Collector** をクリックします。 PostgreSQLを選択します。

   Telegraf エージェントがインストールされているオペレーティング システムまたはプラットフォームを選択します。

2. 収集用のエージェントをまだインストールしていない場合、または別のオペレーティングシステムまたはプラットフォーム用のエージェントをインストールする場合は、[手順を表示] をクリックして展開します。"エージェントのインストール"説明書。
3. このデータ コレクターで使用するエージェント アクセス キーを選択します。 **+** エージェント アクセス キー ボタンをクリックすると、新しいエージェント アクセス キーを追加できます。ベスト プラクティス: データ コレクターを OS/プラットフォーム別などにグループ化する場合にのみ、異なるエージェント アクセス キーを使用します。
4. 構成手順に従ってデータ コレクターを構成します。手順は、データ収集に使用しているオペレーティング システムまたはプラットフォームの種類によって異なります。



セットアップ

詳細は以下をご覧ください。"PostgreSQLドキュメント" 。

オブジェクトとカウンター

次のオブジェクトとそのカウンターが収集されます。

オブジェクト	識別子:	属性:	データポイント:
PostgreSQLサーバー	名前空間データベースサーバー	ノード名 ノードIP	バッファ 割り当てられたバッファ バックエンドバッファ バックエンドファイル 同期バッファ チェックポイントバッファ クリーンチェックポイント 同期時間 チェックポイント書き込み時間 チェックポイント 要求 チェックポイント 時間指定 最大書き込みクリーン
PostgreSQLデータベース	名前空間データベースサーバー	データベースOID ノード名 ノードIP	ブロック読み取り時間、ブロック書き込み時間、ブロックヒット数、ブロック読み取り数、競合数、デッドロック数、クライアント数、一時ファイル数、バイト数、一時ファイル数、削除された行数、フェッチされた行数、挿入された行数、返された行数、更新されたトランザクション数、コミットされたトランザクション数、ロールバックされたトランザクション数

トラブルシューティング

追加情報は以下からご覧いただけます。"サポート"ページ。

Puppet エージェント データコレクター

Data Infrastructure Insights は、このデータ コレクターを使用して、Puppet Agent からメトリックを収集します。

インストール

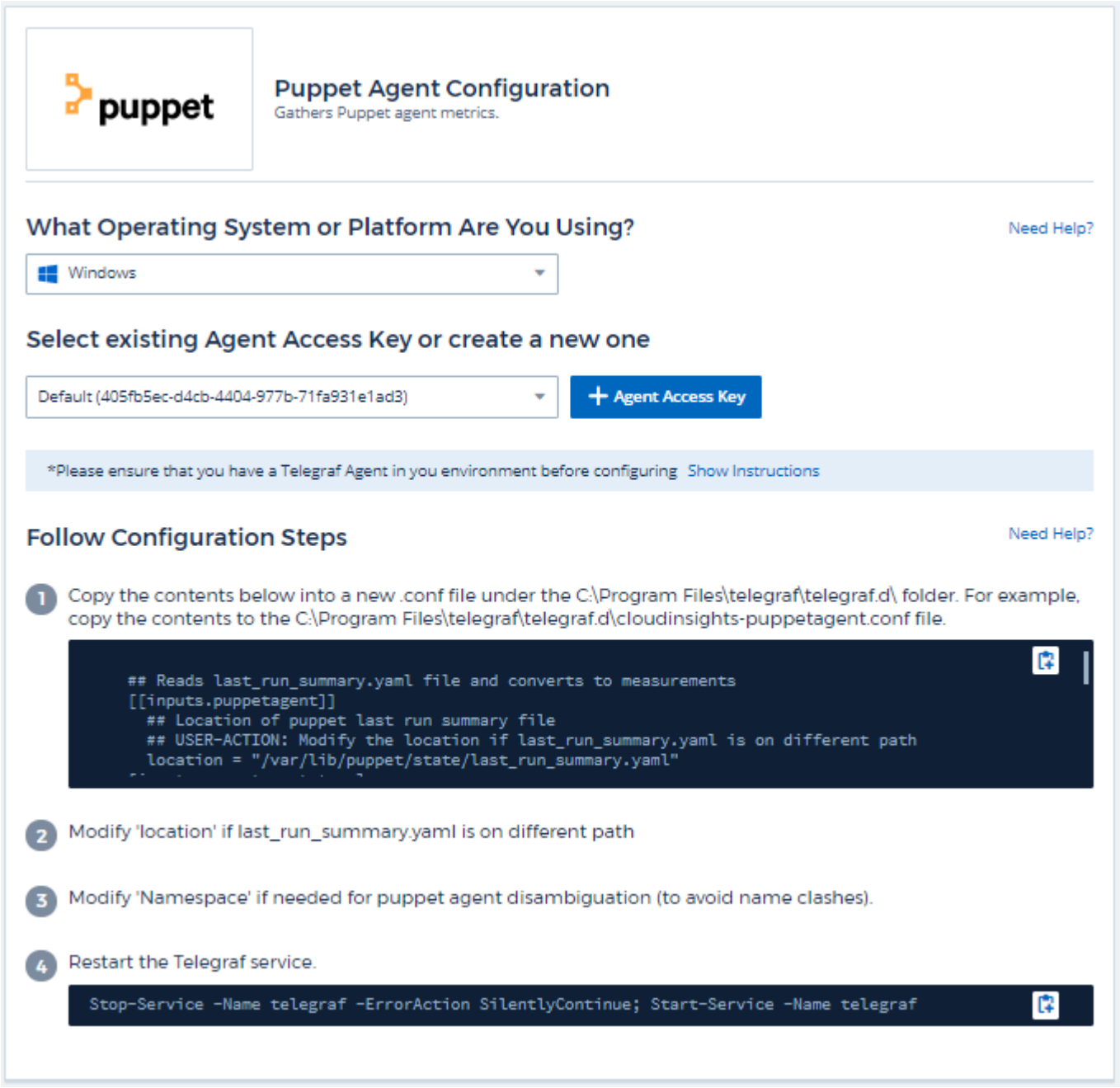
1. **Observability > Collectors** から、**+ Data Collector** をクリックします。パペットを選択します。

   Telegraf エージェントがインストールされているオペレーティング システムまたはプラットフォームを選択します。

2. 収集用のエージェントをまだインストールしていない場合、または別のオペレーティングシステムまたは

プラットフォーム用のエージェントをインストールする場合は、[手順を表示] をクリックして展開します。"エージェントのインストール"説明書。

3. このデータ コレクターで使用するエージェント アクセス キーを選択します。 + エージェント アクセス キー ボタンをクリックすると、新しいエージェント アクセス キーを追加できます。ベスト プラクティス: データ コレクターを OS/プラットフォーム別などにグループ化する場合にのみ、異なるエージェント アクセス キーを使用します。
4. 構成手順に従ってデータ コレクターを構成します。手順は、データ収集に使用しているオペレーティング システムまたはプラットフォームの種類によって異なります。



セットアップ

詳細は以下をご覧ください。 "Puppetのドキュメント"

オブジェクトとカウンター

次のオブジェクトとそのカウンターが収集されます。

オブジェクト	識別子:	属性:	データポイント:
操り人形エージェント	名前空間ノードUUID	ノード名 場所 ノードIP バージョン Configstring バージョン Puppet	変更合計イベント 失敗イベント 成功イベント 合計リソース 変更されたリソース 失敗したリソース 再起動に失敗したリソース 同期していないリソース 再起動されたリソース スケジュールされたリソース スキップされたリソース 合計時間 アンカー時間 Configretrieval時間 Cron時間 Exec時間 ファイル時間 ファイルバケット時間 最終実行時間 パッケージ時間 スケジュール時間 サービス時間 sshauthorizedkey時間 合計時間 ユーザー

トラブルシューティング

追加情報は以下からご覧いただけます。"サポート"ページ。

Redis データコレクター

Data Infrastructure Insights は、このデータ コレクターを使用して Redis からメトリックを収集します。 Redis は、データベース、キャッシュ、メッセージ ブローカーとして使用されるオープン ソースのインメモリ データ構造ストアであり、文字列、ハッシュ、リスト、セットなどのデータ構造をサポートしています。

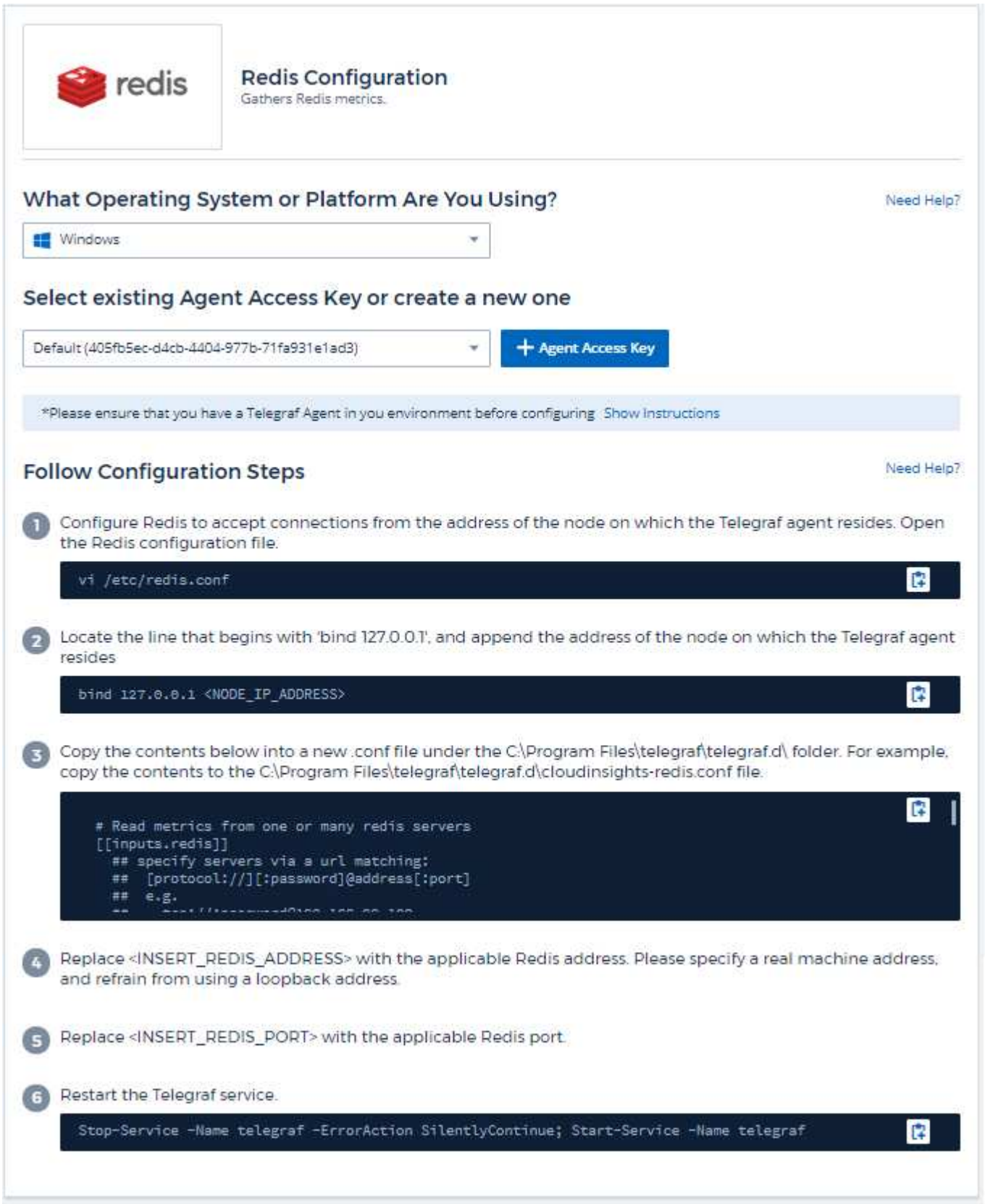
インストール

1. **Observability > Collectors** から、**+ Data Collector** をクリックします。 Redis を選択します。

   Telegraf エージェントがインストールされているオペレーティング システムまたはプラットフォームを選択します。

2. 収集用のエージェントをまだインストールしていない場合、または別のオペレーティングシステムまたはプラットフォーム用のエージェントをインストールする場合は、[手順を表示] をクリックして展開します。"エージェントのインストール"説明書。
3. このデータ コレクターで使用するエージェント アクセス キーを選択します。 **+** エージェント アクセス キー ボタンをクリックすると、新しいエージェント アクセス キーを追加できます。ベスト プラクティス: データ コレクターを OS/プラットフォーム別などにグループ化する場合にのみ、異なるエージェント アクセス キーを使用します。

4. 構成手順に従ってデータ コレクターを構成します。手順は、データ収集に使用しているオペレーティング システムまたはプラットフォームの種類によって異なります。



セットアップ

詳細は以下をご覧ください。"Redisのドキュメント" 。

オブジェクトとカウンター

次のオブジェクトとそのカウンターが収集されます。

オブジェクト	識別子:	属性:	データポイント:
レディス	名前空間サーバー		

トラブルシューティング

追加情報は以下からご覧いただけます。"サポート"ページ。

著作権に関する情報

商標に関する情報