



## **Astra**データストアを監視

### Astra Data Store

NetApp  
June 15, 2022

# 目次

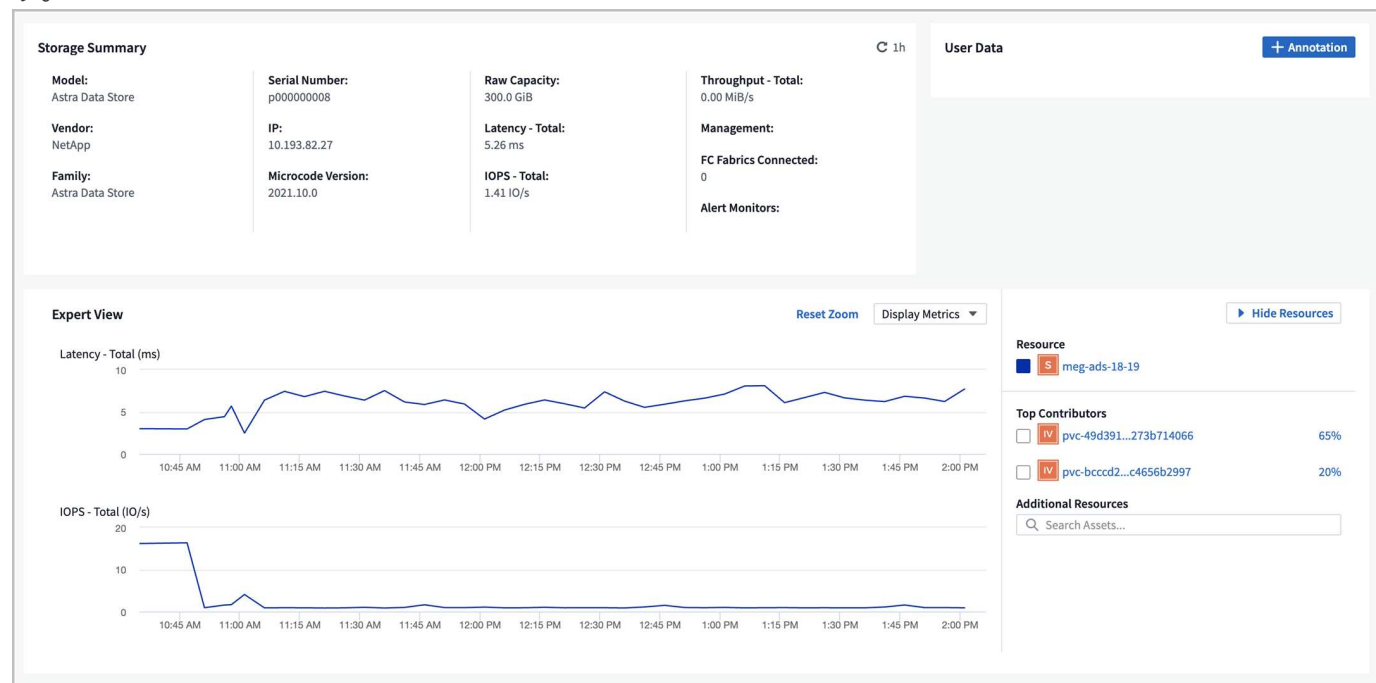
Astraデータストアを監視 .....	1
Cloud Insights で指標を監視 .....	1
Prometheus と Grafana で指標を監視します .....	12
イベントログを設定して監視する .....	14

# Astraデータストアを監視

## Cloud Insights で指標を監視

Cloud Insights を使用して、Astraデータストアの指標を監視できます。

Cloud Insights に表示されるAstraデータストアの指標の例を次に示します。



を使用して、Astraデータストアで生成された指標のリストを表示することもできます [\[Open Metrics API help\]](#)。

次のタスクを実行できます。

- [\[Complete Cloud Insights connection prerequisite tasks\]](#)
- [\[Acquisition Unit storage\]](#)
- [\[Download and run the installation script\]](#)
- [\[Edit the Cloud Insights connection\]](#)
- [\[Disconnect from Cloud Insights\]](#)

## Cloud Insights 接続の前提条件となる作業を完了する

Cloud Insights を使用して Astra データストアに接続する前に、次の作業を完了する必要があります。

- ["Astra Data Store Monitoring Operator をインストールします"](#) これは、Astra Data Storeのインストール手順の一部です。
- ["kubecti-astras バイナリをインストールします"](#) これは、Astra Data Storeのインストール手順の一部です。
- ["Cloud Insights アカウントを作成します"](#)。

- 'awk、curl、grep' および 'jq' の各コマンドが使用可能であることを確認します

次の情報を収集します。

- \* Cloud Insights API アクセストークン \*。各カテゴリに対する読み取り / 書き込み権限：Acquisition Unit、Data Collection、Data Ingestion、Log Ingestionこれは、読み取り / 書き込み処理、Acquisition Unit のセットアップ、およびデータの取り込みプロセスのセットアップに使用されます。
- \* Kubernetes API サーバの IP アドレスとポート \*。これは、Astraデータストアクラスタを監視するために使用されます。
- \* Kubernetes API トークン \*。これは Kubernetes API を呼び出すために使用されます。
- \* 永続ボリューム構成 \*。永続ボリュームのプロビジョニング方法に関する情報。

## Acquisition Unit のストレージ

Acquisition Unit には、インストールファイル、設定データ、およびログを格納するための永続ボリュームが 3 つ必要です。Monitoring Operator は、デフォルトのストレージクラスを使用して、永続的ボリューム要求を作成します。インストーラ・スクリプトの実行時に '-s' オプションを使用して '別のストレージ・クラス名を指定できます

Kubernetes クラスタにストレージプロビジョニングツール（NetApp Trident など）がない場合は、インストーラスクリプトの実行時に -r オプションを使用してローカルファイルシステムパスを提供できます。-r オプションが設定されている場合 'インストーラ・スクリプトは' 指定されたディレクトリ内に 3 つの永続ボリュームを作成しますこのディレクトリには、150GB 以上の空きスペースが必要です。

## インストールスクリプトをダウンロードして実行します

Cloud Insights では、モニタリングオペレータによるアストラデータストアのモニタリングを有効にするための Bash スクリプトを提供しています。インストールスクリプトは、Astra Data Store コレクタと Fluent ビット エージェントを備えた Acquisition Unit をインストールします。

Cloud Insights テナントのドメイン名と選択した Cloud Insights API アクセストークンは、ダウンロード時にインストーラスクリプトに組み込まれます。

その後、次のように指標が送信されます。

- Cloud Insights Acquisition Unit から Cloud Insights データレイクに指標が送信されます。
- FLUENT ビットは、ログ取り込みサービスにログを送信します。

インストーラスクリプトのヘルプを表示します

インストーラスクリプトの完全なヘルプテキストを次に示します。

インストーラスクリプトのヘルプテキストを表示します。

```
./cloudinsights-ads-monitoring.sh -h
```

対応：

```

USAGE: cloudinsights-ads-monitoring.sh [OPTIONS]
Configure monitoring of Astra Data Store by Cloud Insights.
OPTIONS:
  -h                                Display this help message.
  -d ci_domain_name                 Cloud Insights tenant domain name.
  -i kubernetes_ip                 Kubernetes API server IP address.
  -k ci_api_key                     Cloud Insights API Access Token.
  -n namespace                      Namespace for monitoring components. (default:
netapp-monitoring)
  -p kubernetes_port               Kubernetes API server port. (default: 6443)
  -r root_pv_dir                   Create 3 Persistent Volumes in this directory
for the Acquisition Unit.
                                   Only specify this option if there is no Storage
Provisioner installed and the PVs do not already exist.
  -s storage_class                 Storage Class name for provisioning Acquisition
Unit PVs. If not specified, the default storage class will be used.
  -t kubernetes_token              Kubernetes API server token.

```

インストールスクリプトを実行します

1. Cloud Insights アカウントがない場合は作成します。
2. Cloud Insights にログインします。
3. Cloud Insights メニューから、**Admin>\*Data Collector\*** をクリックします。
4. 「\* + Data Collector \*」 をクリックして、新しいコレクタを追加します。



5. 「\* アストラデータストア \*」 タイルをクリックします。
6. 正しい Cloud Insights API アクセストークンを選択するか、新しいトークンを作成します。
7. 指示に従って、インストーラスクリプトをダウンロードし、権限を更新し、スクリプトを実行します。

このスクリプトには、Cloud Insights テナントの URL と選択した Cloud Insights API アクセストークンが含まれています。



Kubernetes

default\_ads\_api\_key1 (...d0gHof) ▼

+ API Access Token

Production Best Practices ?

[Need Help?](#)

- ## 2 Copy Installer Script

☐ Reveal Installer Script

```
#!/usr/bin/env bash
SCRIPT='basename $0'

CI_DOMAIN_NAME="f49uaky.gstabler-ads.cloudinsights-dev.netapp.com"
CI_API_KEY="eyJyJWQwQ0I0I50Tk5IiwidHlwIjoiaSIdUIiwiaWxnIjoiaSFMzODQifQ.eyJjcVhdG9yTG9naW4iOiJhZGpibGllImRmcc3B5YXl0YW1lIjoizGVmYXVsdF9hZHNfYXBpX2tleTEgK69uIGJlaGF5ZiBvZiBhZGpibGll"
```

- #### 4 Copy Permissions Command

 Reveal Permissions Command

```
chmod +x cloudinsights-ads-monitoring.sh
```

- 6 Copy Install Command

☐ Reveal Install Command

```
./cloudinsights-ads-monitoring.sh -i <KUBERNETES_IP> -t <KUBERNETES_TOKEN>
```

- ## 8 Complete Setup

インストールスクリプトが完了すると、「データソース」リストに「Astra Data Store」コネクタが表示されます。



エラーが原因でスクリプトが終了した場合は、エラーが解決してから再度実行できます。デフォルトの設定を使用しない環境では、Monitoring Operator 名前空間や Kubernetes API サーバポートなどの追加のパラメータがサポートされます。使い方とヘルプテキストを表示するには、cloudinsights-ads-monitoring.sh -h オプションを使用します。

設定が正常に完了すると、次のような出力が生成されます。

```
Configuring Cloud Insights monitoring for Astra Data Store . . .
Configuring monitoring namespace
...
Configuring output sink and Fluent Bit plugins
Configuring Acquisition Unit
...
Acquisition Unit has been installed successfully.
Configuring Astra Data Store data collector
Astra Data Store collector data '<CLUSTER_NAME>' created
Configuration done!
```

## エージェント **CR** の例

以下に、インストーラスクリプトの実行後の「Monitoring - NetApp」エージェントの CR の例を示します。

```

spec:
  au:
    isEnabled: true
    storageClassName: auto-sc
  cluster-name: meg-ads-21-22-29-30
  docker-repo: docker.repo.eng.netapp.com/global/astra
  fluent-bit:
    - name: ads-tail
      outputs:
        - sink: ADS_STDOUT
      substitutions:
        - key: TAG
          value: firetapems
        - key: LOG_FILE
          values:
            - /var/log/firetap/*/ems/ems
            - /var/log/firetap/ems/*/ems/ems
        - key: ADS_CLUSTER_NAME
          value: meg-ads-21-22-28-29-30
    - name: agent
    - name: ads-tail-ci
      outputs:
        - sink: CI
      substitutions:
        - key: TAG
          value: netapp.ads
        - key: LOG_FILE
          values:
            - /var/log/firetap/*/ems/ems
            - /var/log/firetap/ems/*/ems/ems
        - key: ADS_CLUSTER_NAME
          value: meg-ads-21-22-28-29-30
  output-sink:
    - api-key: abcd
      domain-name: bz19ngz.gst-adsdemo.ci-dev.netapp.com
      name: CI
  serviceAccount: sa-netapp-monitoring
  status:
    au-pod-status: UP
    au-uuid: eddeccc6-3aa3-4dd2-a98c-220085fae6a9

```

## Cloud Insights 接続を編集します

Kubernetes API トークンまたは Cloud Insights API アクセストークンはあとから編集できます。



- Kubernetes API トークンを更新する場合は、Cloud Insights UI から Astra データストアコレクタを編集する必要があります。
- テレメトリとログに使用される Cloud Insights API アクセストークンを更新する場合は、kubectl コマンドを使用して Monitoring Operator CR を編集する必要があります。

### Kubernetes API トークンを更新します

1. Cloud Insights にログインします。
2. **[Admin>]** > **[\* Data Collectors]** を選択して、**[Data Collectors]** ページにアクセスします。
3. Astra データストアクラスターのエントリを探します。
4. ページの右側にあるメニューをクリックし、「\* 編集 \*」を選択します。
5. Kubernetes API トークンフィールドを新しい値で更新します。
6. **[コレクタの保存 \*]** を選択します

### Cloud Insights API アクセストークンを更新します

1. Cloud Insights にログインします。
2. **[Admin>\*API Access\*]** を選択し、**[\*+API アクセストークン \*]** をクリックして、新しい Cloud Insights API アクセストークンを作成します。
3. エージェント CR を編集します。

```
kubectl --namespace netapp-monitoring edit agent agent-monitoring-netapp
```

4. 「output-sink」セクションを探し、「ci」という名前のエントリを見つけます。
5. ラベル「api-key」の場合は、現在の値を新しい Cloud Insights API アクセストークンに置き換えます。

セクションは次のようになります。

```
output-sink:
- api-key: <api key value>
  domain-name: <tenant url>
  name: CI
```

6. エディタウィンドウを保存して終了します。

モニタリングオペレータは、新しいCloud Insights APIアクセストークンを使用するようにFluentビットを更新します。

### Cloud Insights から切断します

Cloud Insights から切断するには、最初に Cloud Insights UI から Astra データストアコレクタを削除する必要があります。これが完了したら、モニタリングオペレータからAcquisition Unit、Telegraf（設定されている場合）、およびFluentビットの設定を削除できます。

## Astra Data Store コレクタを削除

1. Cloud Insights にログインします。
2. [Admin]> > [\* Data Collectors] を選択して、[Data Collectors] ページにアクセスします。
3. Astra データストアクラスターのエントリを探します。
4. 画面の右側のメニューを選択し、「\* 削除 \*」を選択します。
5. 確認ページで \* Delete \* をクリックします。

## Acquisition Unit、Telegraf（設定されている場合）、およびFluentビットを取り外します

1. エージェント CR を編集します。

```
kubectl --namespace netapp-monitoring edit agent agent-monitoring-netapp
```

2. 「au」セクションを探し、「IsEnabled」を「false」に設定します
3. 「FLUENT ビット」セクションを探し、「ADS テール CI」という名前のプラグインを削除します。プラグインがない場合は、「FLUENT - BIT」セクションを削除できます。
4. Telegrafが設定されている場合は、「テレグラム」セクションを探し、「ADSオープンメトリック」という名前のプラグインを削除します。プラグインがない場合は、「テレグラム」セクションを削除できます。
5. 「output-sink」セクションを探し、「ci」という名前のシンクを取り外します。
6. エディタウィンドウを保存して終了します。

モニタリングオペレータがTelegraf（設定されている場合）およびFluentビット設定を更新し、Acquisition Unitポッドを削除します。

7. ストレージプロビジョニング担当者ではなく Acquisition Unit PVS にローカルディレクトリを使用した場合は、PVS を削除します。

```
kubectl delete pv au-lib au-log au-pv
```

次に、Acquisition Unit を実行していたノードの実際のディレクトリを削除します。

8. Acquisition Unit ポッドが削除されたら、Cloud Insights から Acquisition Unit を削除できます。
  - a. Cloud Insights メニューで、**Admin>\*Data Collector\*** を選択します。
  - b. [\* Acquisition Units \*（Acquisition Unit \*）] タブをクリックします。
  - c. Acquisition Unit ポッドの横にあるメニューをクリックします。
  - d. 「\* 削除」を選択します。

モニタリングオペレータは、Telegraf（設定されている場合）およびFluentビットの設定を更新し、Acquisition Unitを削除します。

## Open Metrics API のヘルプを参照してください

Astraデータストアから指標を収集するために使用できるAPIのリストを次に示します。

- 「help」行は指標を表します。
- 「type」行は、メトリックがゲージかカウンタかを示します。

```
# HELP astrads_cluster_capacity_logical_percent Percentage cluster logical
capacity that is used (0-100)
# TYPE astrads_cluster_capacity_logical_percent gauge
# HELP astrads_cluster_capacity_max_logical Max Logical capacity of the
cluster in bytes
# TYPE astrads_cluster_capacity_max_logical gauge
# HELP astrads_cluster_capacity_max_physical The sum of the space in the
cluster in bytes for storing data after provisioning efficiencies, data
reduction algorithms and replication schemes are applied
# TYPE astrads_cluster_capacity_max_physical gauge
# HELP astrads_cluster_capacity_ops The IO operations capacity of the
cluster
# TYPE astrads_cluster_capacity_ops gauge
# HELP astrads_cluster_capacity_physical_percent The percentage of cluster
physical capacity that is used (0-100)
# TYPE astrads_cluster_capacity_physical_percent gauge
# HELP astrads_cluster_capacity_used_logical The sum of the bytes of data
in all volumes in the cluster before provisioning efficiencies, data
reduction algorithms and replication schemes are applied
# TYPE astrads_cluster_capacity_used_logical gauge
# HELP astrads_cluster_capacity_used_physical Used Physical capacity of a
cluster in bytes
# TYPE astrads_cluster_capacity_used_physical gauge
# HELP astrads_cluster_other_latency The sum of the accumulated latency in
seconds for other IO operations of all the volumes in a cluster. Divide by
astrads_cluster_other_ops to get the average latency per other operation
# TYPE astrads_cluster_other_latency counter
# HELP astrads_cluster_other_ops The sum of the other IO operations of all
the volumes in a cluster
# TYPE astrads_cluster_other_ops counter
# HELP astrads_cluster_read_latency The sum of the accumulated latency in
seconds of read IO operations of all the volumes in a cluster. Divide by
astrads_cluster_read_ops to get the average latency per read operation
# TYPE astrads_cluster_read_latency counter
# HELP astrads_cluster_read_ops The sum of the read IO operations of all
the volumes in a cluster
# TYPE astrads_cluster_read_ops counter
# HELP astrads_cluster_read_throughput The sum of the read throughput of
all the volumes in a cluster in bytes
```

```

# TYPE astrads_cluster_read_throughput counter
# HELP astrads_cluster_storage_efficiency Efficacy of data reduction
technologies. (logical used / physical used)
# TYPE astrads_cluster_storage_efficiency gauge
# HELP astrads_cluster_total_latency The sum of the accumulated latency in
seconds of all IO operations of all the volumes in a cluster. Divide by
astrads_cluster_total_ops to get average latency per operation
# TYPE astrads_cluster_total_latency counter
# HELP astrads_cluster_total_ops The sum of the IO operations of all the
volumes in a cluster
# TYPE astrads_cluster_total_ops counter
# HELP astrads_cluster_total_throughput The sum of the read and write
throughput of all the volumes in a cluster in bytes
# TYPE astrads_cluster_total_throughput counter
# HELP astrads_cluster_utilization_factor The ratio of the current cluster
IO operations based on recent IO sizes to the cluster iops capacity. (0.0
- 1.0)
# TYPE astrads_cluster_utilization_factor gauge
# HELP astrads_cluster_volume_used The sum of used capacity of all the
volumes in a cluster in bytes
# TYPE astrads_cluster_volume_used gauge
# HELP astrads_cluster_write_latency The sum of the accumulated latency in
seconds of write IO operations of all the volumes in a cluster. Divide by
astrads_cluster_write_ops to get the average latency per write operation
# TYPE astrads_cluster_write_latency counter
# HELP astrads_cluster_write_ops The sum of the write IO operations of all
the volumes in a cluster
# TYPE astrads_cluster_write_ops counter
# HELP astrads_cluster_write_throughput The sum of the write throughput of
all the volumes in a cluster in bytes
# TYPE astrads_cluster_write_throughput counter
# HELP astrads_disk_base_seconds Base for busy, pending and queued.
Seconds since collection began
# TYPE astrads_disk_base_seconds counter
# HELP astrads_disk_busy Seconds the disk was busy. 100 *
(astrads_disk_busy / astrads_disk_base_seconds) = percent busy (0-100)
# TYPE astrads_disk_busy counter
# HELP astrads_disk_capacity Raw Capacity of a disk in bytes
# TYPE astrads_disk_capacity gauge
# HELP astrads_disk_io_pending Summation of the count of pending io
operations for a disk times time. Divide by astrads_disk_base_seconds to
get the average pending operation count
# TYPE astrads_disk_io_pending counter
# HELP astrads_disk_io_queued Summation of the count of queued io
operations for a disk times time. Divide by astrads_disk_base_seconds to
get the average queued operations count

```

```

# TYPE astrads_disk_io_queued counter
# HELP astrads_disk_read_latency Total accumulated latency in seconds for
disk reads. Divide by astrads_disk_read_ops to get the average latency per
read operation
# TYPE astrads_disk_read_latency counter
# HELP astrads_disk_read_ops Total number of read operations for a disk
# TYPE astrads_disk_read_ops counter
# HELP astrads_disk_read_throughput Total bytes read from a disk
# TYPE astrads_disk_read_throughput counter
# HELP astrads_disk_write_latency Total accumulated latency in seconds for
disk writes. Divide by astrads_disk_write_ops to get the average latency
per write operation
# TYPE astrads_disk_write_latency counter
# HELP astrads_disk_write_ops Total number of write operations for a disk
# TYPE astrads_disk_write_ops counter
# HELP astrads_disk_write_throughput Total bytes written to a disk
# TYPE astrads_disk_write_throughput counter
# HELP astrads_value_scrape_duration Duration to scrape values
# TYPE astrads_value_scrape_duration gauge
# HELP astrads_volume_capacity_available The minimum of the available
capacity of a volume and the available capacity of the cluster in bytes
# TYPE astrads_volume_capacity_available gauge
# HELP astrads_volume_capacity_available_logical Logical available
capacity of a volume in bytes
# TYPE astrads_volume_capacity_available_logical gauge
# HELP astrads_volume_capacity_percent Percentage of volume capacity
available (0-100). (capacity available / provisioned) * 100
# TYPE astrads_volume_capacity_percent gauge
# HELP astrads_volume_capacity_provisioned Provisioned capacity of a
volume in bytes after setting aside the snapshot reserve. (size - snapshot
reserve = provisioned)
# TYPE astrads_volume_capacity_provisioned gauge
# HELP astrads_volume_capacity_size Total capacity of a volume in bytes
# TYPE astrads_volume_capacity_size gauge
# HELP astrads_volume_capacity_snapshot_reserve_percent Snapshot reserve
percentage of a volume (0-100)
# TYPE astrads_volume_capacity_snapshot_reserve_percent gauge
# HELP astrads_volume_capacity_snapshot_used The amount of volume snapshot
data that is not in the active file system in bytes
# TYPE astrads_volume_capacity_snapshot_used gauge
# HELP astrads_volume_capacity_used Used capacity of a volume in bytes.
This is bytes in the active filesystem unless snapshots are consuming more
than the snapshot reserve. (bytes in the active file system + MAX(0,
snapshot_used-(snapshot_reserve_percent/100*size))
# TYPE astrads_volume_capacity_used gauge
# HELP astrads_volume_other_latency Total accumulated latency in seconds

```

```
for operations on a volume that are neither read or write. Divide by
astrads_volume_other_ops to get the average latency per other operation
# TYPE astrads_volume_other_latency counter
# HELP astrads_volume_other_ops Total number of operations for a volume
that are neither read or write
# TYPE astrads_volume_other_ops counter
# HELP astrads_volume_read_latency Total accumulated read latency in
seconds for a volume. Divide by astrads_volume_read_ops to get the average
latency per read operation
# TYPE astrads_volume_read_latency counter
# HELP astrads_volume_read_ops Total number of read operations for a
volume
# TYPE astrads_volume_read_ops counter
# HELP astrads_volume_read_throughput Total read throughput for a volume
in bytes
# TYPE astrads_volume_read_throughput counter
# HELP astrads_volume_total_latency Total accumulated latency in seconds
for all operations on a volume. Divide by astrads_volume_total_ops to get
the average latency per operation
# TYPE astrads_volume_total_latency counter
# HELP astrads_volume_total_ops Total number of operations for a volume
# TYPE astrads_volume_total_ops counter
# HELP astrads_volume_total_throughput Total throughput for a volume in
bytes
# TYPE astrads_volume_total_throughput counter
# HELP astrads_volume_write_latency Total accumulated write latency in
seconds for volume. Divide by astrads_volume_write_ops to get the average
latency per write operation
# TYPE astrads_volume_write_latency counter
# HELP astrads_volume_write_ops Total number of write operations for a
volume
# TYPE astrads_volume_write_ops counter
# HELP astrads_volume_write_throughput Total write throughput for a volume
in bytes
# TYPE astrads_volume_write_throughput counter
```

## Prometheus と Grafana で指標を監視します

PrometheusとGrafanaを使用して、Astraデータストアの指標を監視できます。PrometheusでAstra Data Store Kubernetesクラスター指標エンドポイントから指標を収集するように設定でき、Grafanaを使用して指標データを表示できます。

### 必要なもの

- PrometheusパッケージとGrafanaパッケージをAstra Data Storeクラスター、またはAstra Data Storeクラスターと通信可能な別のクラスターでダウンロードしてインストールしておく必要があります。公式ドキュメントの指示に従って、各ツールをインストールします。

- ["Prometheus をインストールする"](#)
- ["Grafana をインストールします"](#)
- PrometheusとGrafanaがAstra Data Store Kubernetesクラスタと通信する必要があります。PrometheusとGrafanaがAstra Data Storeクラスタにインストールされていない場合は、Astra Data Storeクラスタで実行されている指標サービスと通信できることを確認する必要があります。

## Prometheus を設定する

Astraデータストアは、KubernetesクラスタのTCPポート9341で指標サービスを公開します。このサービスから指標を収集するには、Prometheus を設定する必要があります。

### 手順

1. Prometheus インストール用の「prometheus.yml」構成ファイルを編集します。
2. Astra Data Storeサービス名とそのポートを指すサービスターゲットを追加します。例：

```
scrape_configs:
static_configs:
- targets: ['astrads-metrics-service.astrads-system:9341']
```

3. Prometheus サービスを開始します。

## Grafana を設定します

Prometheus で収集された指標を表示するように Grafana を設定できます。

### 手順

1. Grafana インストールの「datasources.yaml」設定ファイルを編集します。
2. Prometheus をデータソースとして追加します。例：

```
apiVersion: 1

datasources:
- name: astradatastore-prometheus
  type: prometheus
  access: proxy
  url: http://localhost:9090
  jsonData:
    manageAlerts: false
```

3. Grafana サービスを開始します。
4. Grafana のマニュアルに記載されている手順に従ってください ["始めましょう"](#)。

## Grafana ダッシュボードテンプレートをインポートします

Astra Data Storeをインストールするためにダウンロードしたバンドルファイルには、Grafana内からインポートできるGrafanaダッシュボードテンプレートファイルが含まれています。このダッシュボードテンプレートを使用すると、Astraデータストアから取得できる指標の種類とその表示方法を確認できます。

手順

1. Astra Data Storeの.tar.gzバンドルを開きます。
2. 'マニフェスト'ディレクトリを開きます
3. 「 graafana \_cluster.json 」 ファイルと「 graafana \_volume.json 」 ファイルを展開します。
4. Grafana Web UI の使用、 ["でのダッシュボードテンプレートファイルを Grafana にインポートします"](#)。

## イベントログを設定して監視する

Event Management System （ EMS ；イベント管理システム）ログを監視するには、次の高度なタスクを実行します。

- [\[Configure monitoring in the Astra Data Store cluster custom resource \(CR\)\]](#)
- [\[Set up Cloud Insights\]](#)
- [\[Stream event logs to Elastic\]](#)。

## Astra Data Store クラスタカスタムリソース（CR）での監視の設定

Astra Data StoreクラスタCRでモニタリングオプションが設定されていない場合は、「astras」拡張機能を使用して設定できます。

入力するコマンド

```
kubectl astrads monitoring setup -n <NAMESPACE OF AGENT INSTALLED> -r  
<DOCKER REPO TO FIND FLUENT/TELEGRAF ETC IMAGES>
```

ここで、

- インストールされているエージェントのネームスペース： Monitoring Agent のネームスペースを入力します。この名前は、 Monitoring Operator のデフォルトの名前である NetApp CR になります。
- -r は 'Fluent イメージまたは Telegraf イメージが置かれている Docker レジストリをセットアップするためのオプションですデフォルトでは 'パスは docker.repo.eng.netapp.com/global/astra` に設定されていますこのパスは変更できます

## Cloud Insights をセットアップする

ログを表示するには Cloud Insights の設定は任意ですが、 Cloud Insights を使用してデータを表示すると便利です。を参照してください ["NetApp Cloud Insights のセットアップ方法"](#) Astraデータストアで使用。



## イベントログを **Elastic** にストリーミングする

EMS イベントやその他のポッドログを Elastic などのサードパーティのエンドポイントにストリーミングするには、「astrads」拡張機能を使用します。

入力するコマンド

```
kubect1 astrads monitoring --host <ELASTIC HOST NAME> --port <ELASTIC HOST PORT> es
```



Elastic ホスト名は IP アドレスでもかまいません。

## 著作権情報

Copyright © 2022 NetApp, Inc. All rights reserved. 米国で印刷されていますこのドキュメントは著作権によって保護されています。画像媒体、電子媒体、および写真複写、記録媒体などの機械媒体など、いかなる形式および方法による複製も禁止します。テープ媒体、または電子検索システムへの保管-著作権所有者の書面による事前承諾なし。

ネットアップの著作物から派生したソフトウェアは、次に示す使用許諾条項および免責条項の対象となります。

このソフトウェアは、ネットアップによって「現状のまま」提供されています。ネットアップは明示的、または商品性および特定目的に対する適合性の暗示的保証を含み、かつこれに限定されないいかなる暗示的な保証も行いません。ネットアップは、いかなる場合でも、間接的、偶発的、特別、懲罰的、またはまたは結果的損害（代替品または代替サービスの調達、使用の損失、データ、利益、またはこれらに限定されないものを含みますが、これらに限定されません。）ただし、契約、厳格責任、または本ソフトウェアの使用に起因する不法行為（過失やその他を含む）のいずれであっても、かかる損害の可能性について知らされていた場合でも、責任の理論に基づいて発生します。

ネットアップは、ここに記載されているすべての製品に対する変更を随時、予告なく行う権利を保有します。ネットアップによる明示的な書面による合意がある場合を除き、ここに記載されている製品の使用により生じる責任および義務に対して、ネットアップは責任を負いません。この製品の使用または購入は、ネットアップの特許権、商標権、またはその他の知的所有権に基づくライセンスの供与とはみなされません。

このマニュアルに記載されている製品は、1 つ以上の米国特許、その他の国の特許、および出願中の特許により特許、その他の国の特許、および出願中の特許。

権利の制限について：政府による使用、複製、開示は、DFARS 252.227-7103（1988 年 10 月）および FAR 52-227-19（1987 年 6 月）の Rights in Technical Data and Computer Software（技術データおよびコンピュータソフトウェアに関する諸権利）条項の（c）（1）（ii）項、に規定された制限が適用されます。

## 商標情報

NetApp、NetAppのロゴ、に記載されているマーク <http://www.netapp.com/TM> は、NetApp, Inc.の商標です。その他の会社名と製品名は、それを所有する各社の商標である場合があります。