安裝 Astra Connector 以管理叢集

Astra Control Service

NetApp
September 23, 2024

This PDF was generated from https://docs.netapp.com/zh-tw/astra-control-service/get-started/install-astra-connector-previous.html on September 23, 2024. Always check docs.netapp.com for the latest.

目錄

安裝 Astra Connector 以管理叢集 1
安裝舊版 Astra Connector 1
（技術預覽）安裝宣告性 Kubernetes Astra Connector 4

安裝 Astra Connector 以管理叢集

Astra Connector 是一種軟體、位於託管叢集上、可促進託管叢集與 Astra Control 之間的通訊。對於使用 Astra Control Service 管理的叢集、有兩個可用版本的 Astra Connector ：

- * 舊版 Astra Connector* ： "安裝舊版 Astra Connector" 如果您計畫使用非 Kubernetes 原生工作流程來管理叢集、請在叢集上進行。
- [技術預覽] * 聲明性 Kubernetes Astra Connector* ： "安裝 Astra Connector for 叢集、以宣告式 Kubernetes 工作流程進行管理" 如果您打算使用宣告式 Kubernetes 工作流程來管理叢集、請在叢集上執行。在叢集上安裝 Astra Connector 之後、叢集會自動新增至 Astra Control 。

> 聲明性 Kubernetes Astra Connector 僅可作為 Astra Control Early Adopter Program （ EAP ）的一部分使用。請聯絡您的 NetApp 銷售代表、以取得加入 EAP 的相關資訊。

安裝舊版 Astra Connector

Astra Control Service 使用舊版 Astra Connector 、在 Astra Control Service 和使用非 Kubernetes 原生工作流程管理的私有叢集之間進行通訊。您需要在想要使用非 Kubernetes 原生工作流程管理的私有叢集上安裝 Astra Connector 。

舊版 Astra Connector 支援下列類型的私有叢集、這些叢集是以非 Kubernetes 原生工作流程管理：

- Amazon Elastic Kubernetes Service （ EKS ）
- Azure Kubernetes服務（KS）
- Google Kubernetes Engine（GKE）
- AWS 上的 Red Hat OpenShift 服務（ ROSA ）
- ROSA 搭配 AWS Private Link
- 內部部署 Red Hat OpenShift Container Platform

關於這項工作

- 當您執行這些步驟時、請針對您要使用 Astra Control Service 管理的私有叢集執行這些命令。
- 如果您使用的是堡壘主機、請從堡壘主機的命令列發出這些命令。

開始之前

- 您需要存取想要使用 Astra Control Service 管理的私有叢集。
- 您需要 Kubernetes 管理員權限、才能在叢集上安裝 Astra Connector 運算子。

步驟

1. 使用非 Kubernetes 原生工作流程、在您要管理的私有叢集上安裝先前的 Astra Connector 運算子。當您執行此命令時、命名空間 `astra-connector-operator` 已建立並套用組態至命名空間：

```
kubectl apply -f https://github.com/NetApp/astra-connector-
operator/releases/download/23.07.0-
202310251519/astraconnector_operator.yaml
```

2. 確認操作員已安裝就緒：

```
kubectl get all -n astra-connector-operator
```

3. 從 Astra Control 取得 API 權杖。請參閱 "Astra Automation文件" 以取得相關指示。
4. 建立 Astra 連接器命名空間：

```
kubectl create ns astra-connector
```

5. 建立 Astra Connector CR 檔案並命名 `astra-connector-cr.yaml`。更新括弧 <> 中的值以符合 Astra Control 環境和叢集組態：
   - * <ASTRA_CONTROL_SERVICE_URL> * ： Astra 控制服務的網路 UI URL 。例如：

     ```
     https://astra.netapp.io
     ```

   - * <ASTRA_CONTROL_SERVICE_API_TOKEN> * ：您在上述步驟中取得的 Astra 控制 API 權杖。
   - * <PRIVATE_AKS_CLUSTER_NAME> * ：（僅限 AKS 叢集） - 私有 Azure Kubernetes 服務叢集的叢集名稱。只有在您要新增私有 aks 叢集時、才取消註解並填入此行。
   - * <ASTRA_CONTROL_ACCOUNT_ID> * ：從 Astra Control 網路 UI 取得。選取頁面右上角的圖示、然後選取 * API 存取 * 。

```yaml
apiVersion: netapp.astraconnector.com/v1
kind: AstraConnector
metadata:
  name: astra-connector
  namespace: astra-connector
spec:
  natssync-client:
    cloud-bridge-url: <ASTRA_CONTROL_SERVICE_URL>
  imageRegistry:
    name: theotw
    secret: ""
  astra:
    token: <ASTRA_CONTROL_SERVICE_API_TOKEN>
    #clusterName: <PRIVATE_AKS_CLUSTER_NAME>
    accountId: <ASTRA_CONTROL_ACCOUNT_ID>
    acceptEULA: yes
```

6. 填入之後 `astra-connector-cr.yaml` 使用正確值的檔案、請套用 CR ：

```
kubectl apply -f astra-connector-cr.yaml
```

7. 確認 Astra Connector 已完全部署：

```
kubectl get all -n astra-connector
```

8. 確認叢集已向 Astra Control 註冊：

```
kubectl get astraconnector -n astra-connector
```

您應該會看到類似下列的輸出：

```
NAME                REGISTERED   ASTRACONNECTORID
STATUS
astra-connector     true         be475ae5-1511-4eaa-9b9e-712f09b0d065
Registered with Astra
```

記下 ASTRACONECTORID ；將叢集新增至 Astra Control 時、您將需要它。

接下來呢？

安裝 Astra Connector 之後、您就可以將私有叢集新增至 Astra Control Service 。

- "將私有提供者管理的叢集新增至 Astra Control Service"：請使用以下步驟來新增具有私有 IP 位址且由雲端供應商管理的叢集。您需要雲端供應商的服務主體帳戶、服務帳戶或使用者帳戶。
- "將私有自我管理叢集新增至 Astra Control Service"：請使用以下步驟來新增具有私有 IP 位址且由組織管理的叢集。您需要為想要新增的叢集建立一個 kubeconfig 檔案。

以取得更多資訊

- "新增叢集"

（技術預覽）安裝宣告性 Kubernetes Astra Connector

使用宣告式 Kubernetes 工作流程管理的叢集使用 Astra Connector 來啟用託管叢集與 Astra Control 之間的通訊。您需要在所有要使用宣告式 Kubernetes 工作流程管理的叢集上安裝 Astra Connector 。

您可以使用 Kubernetes 命令和自訂資源（ CR ）檔案來安裝宣告式 Kubernetes Astra Connector 。

關於這項工作

- 執行這些步驟時、請在您要使用 Astra Control 管理的叢集上執行這些命令。
- 如果您使用的是堡壘主機、請從堡壘主機的命令列發出這些命令。

開始之前

- 您需要存取想要使用 Astra Control 管理的叢集。
- 您需要 Kubernetes 管理員權限、才能在叢集上安裝 Astra Connector 運算子。

> 如果叢集已設定 Pod 安全許可強制（ Kubernetes 1.25 及更新叢集的預設值）、則您需要針對適當的命名空間啟用 PSA 限制。請參閱 "使用Astra Control為環境做好叢集管理準備" 以取得相關指示。

步驟

1. 使用宣告式 Kubernetes 工作流程、在您要管理的叢集上安裝 Astra Connector 運算子。當您執行此命令時、命名空間 `astra-connector-operator` 已建立並套用組態至命名空間：

```
kubectl apply -f https://github.com/NetApp/astra-connector-
operator/releases/download/24.02.0-
202403151353/astraconnector_operator.yaml
```

2. 確認操作員已安裝就緒：

```
kubectl get all -n astra-connector-operator
```

3. 從 Astra Control 取得 API 權杖。請參閱 "Astra Automation文件" 以取得相關指示。
4. 使用權杖建立秘密。將 <API_TOKEN> 取代為您從 Astra Control 收到的權杖：

```
kubectl create secret generic astra-token \
--from-literal=apiToken=<API_TOKEN> \
-n astra-connector
```

5. 建立 Docker 秘密以用於拉出 Astra Connector 映像。以您環境的資訊取代括弧 <> 中的值：

> 您可以在 Astra 控制網路 UI 中找到 <ASTRA_CONTROL_ACCOUNT_ID> 。在 Web UI 中、選取頁面右上角的圖示、然後選取 * API 存取 * 。

```
kubectl create secret docker-registry regcred \
--docker-username=<ASTRA_CONTROL_ACCOUNT_ID> \
--docker-password=<API_TOKEN> \
-n astra-connector \
--docker-server=cr.astra.netapp.io
```

6. 建立 Astra Connector CR 檔案並命名 `astra-connector-cr.yaml`。更新括弧 <> 中的值以符合 Astra Control 環境和叢集組態：
   - <ASTRA_CONTROL_ACCOUNT_ID> ：在前一個步驟中從 Astra Control 網路 UI 取得。
   - <CLUSTER_NAME> ：應在 Astra Control 中指派此叢集的名稱。
   - <ASTRA_CONTROL_URL> ： Astra Control 的網路 UI URL 。例如：

     ```
     https://astra.control.url
     ```

```yaml
apiVersion: astra.netapp.io/v1
kind: AstraConnector
metadata:
  name: astra-connector
  namespace: astra-connector
spec:
  astra:
    accountId: <ASTRA_CONTROL_ACCOUNT_ID>
    clusterName: <CLUSTER_NAME>
    #Only set `skipTLSValidation` to `true` when using the default
self-signed
    #certificate in a proof-of-concept environment.
    skipTLSValidation: false #Should be set to false in production
environments
    tokenRef: astra-token
  natsSyncClient:
    cloudBridgeURL: <ASTRA_CONTROL_HOST_URL>
  imageRegistry:
    name: cr.astra.netapp.io
    secret: regcred
```

7. 填入之後 `astra-connector-cr.yaml` 使用正確值的檔案、請套用 CR ：

```
kubectl apply -n astra-connector -f astra-connector-cr.yaml
```

8. 確認 Astra Connector 已完全部署：

```
kubectl get all -n astra-connector
```

9. 確認叢集已向 Astra Control 註冊：

```
kubectl get astraconnectors.astra.netapp.io -A
```

您應該會看到類似下列的輸出：

```
NAMESPACE          NAME               REGISTERED   ASTRACONNECTORID
STATUS
astra-connector    astra-connector    true         00ac8-2cef-41ac-8777-
ed0583e   Registered with Astra
```

10. 驗證叢集是否出現在 Astra Control 網路 UI 的 * Clusters* 頁面上的受管理叢集清單中。

版權資訊

Copyright © 2024 NetApp, Inc. 版權所有。台灣印製。非經版權所有人事先書面同意，不得將本受版權保護文件的任何部分以任何形式或任何方法（圖形、電子或機械）重製，包括影印、錄影、錄音或儲存至電子檢索系統中。

由 NetApp 版權資料衍伸之軟體必須遵守下列授權和免責聲明：

此軟體以 NETAPP「原樣」提供，不含任何明示或暗示的擔保，包括但不限於有關適售性或特定目的適用性之擔保，特此聲明。於任何情況下，就任何已造成或基於任何理論上責任之直接性、間接性、附隨性、特殊性、懲罰性或衍生性損害（包括但不限於替代商品或服務之採購；使用、資料或利潤上的損失；或企業營運中斷），無論是在使用此軟體時以任何方式所產生的契約、嚴格責任或侵權行為（包括疏忽或其他）等方面，NetApp 概不負責，即使已被告知有前述損害存在之可能性亦然。

NetApp 保留隨時變更本文所述之任何產品的權利，恕不另行通知。NetApp 不承擔因使用本文所述之產品而產生的責任或義務，除非明確經過 NetApp 書面同意。使用或購買此產品並不會在依據任何專利權、商標權或任何其他 NetApp 智慧財產權的情況下轉讓授權。

本手冊所述之產品受到一項（含）以上的美國專利、國外專利或申請中專利所保障。

有限權利說明：政府機關的使用、複製或公開揭露須受 DFARS 252.227-7013（2014 年 2 月）和 FAR 52.227-19（2007 年 12 月）中的「技術資料權利 - 非商業項目」條款 (b)(3) 小段所述之限制。

此處所含屬於商業產品和 / 或商業服務（如 FAR 2.101 所定義）的資料均為 NetApp, Inc. 所有。根據本協議提供的所有 NetApp 技術資料和電腦軟體皆屬於商業性質，並且完全由私人出資開發。 美國政府對於該資料具有非專屬、非轉讓、非轉授權、全球性、有限且不可撤銷的使用權限，僅限於美國政府為傳輸此資料所訂合約所允許之範圍，並基於履行該合約之目的方可使用。除非本文另有規定，否則未經 NetApp Inc. 事前書面許可，不得逕行使用、揭露、重製、修改、履行或展示該資料。美國政府授予國防部之許可權利，僅適用於 DFARS 條款 252.227-7015(b)（2014 年 2 月）所述權利。

商標資訊

NETAPP、NETAPP 標誌及 http://www.netapp.com/TM 所列之標章均為 NetApp, Inc. 的商標。文中所涉及的所有其他公司或產品名稱，均為其各自所有者的商標，不得侵犯。