概念
ONTAP Select

NetApp
November 08, 2026

This PDF was generated from https://docs.netapp.com/zh-tw/ontap-select-9191/concept_api_rest.html on November 08, 2026. Always check docs.netapp.com for the latest.

目錄

概念 . . . 1
了解用於部署和管理 ONTAP Select 叢集的 REST Web 服務 . . . 1
架構與傳統限制 . . . 1
資源和狀態表示法 . . . 1
URI 端點 . . . 1
HTTP 訊息 . . . 1
JSON 格式化 . . . 1
了解如何存取 ONTAP Select Deploy API . . . 2
部署公用程式原生使用者介面 . . . 2
ONTAP Select Deploy 線上文件頁面 . . . 2
自訂程式 . . . 2
ONTAP Select Deploy API 基本操作特性 . . . 2
Hypervisor 主機與 ONTAP Select 節點 . . . 2
物件識別碼 . . . 2
請求識別碼 . . . 3
同步和非同步呼叫 . . . 3
確認長時間執行工作的完成 . . . 3
安全性 . . . 3
ONTAP Select 的請求和回應 API 交易 . . . 3
控制 API 要求的輸入變數 . . . 3
解讀 API 回應 . . . 5
使用 ONTAP Select 的 Job 物件進行非同步處理 . . . 6
使用 Job 物件描述的非同步請求 . . . 6
查詢與 API 請求相關聯的 Job 物件 . . . 6
發出非同步請求的一般程序 . . . 7

概念

了解用於部署和管理 ONTAP Select 叢集的 REST Web 服務

Representational State Transfer（REST）是一種用於建立分散式 Web 應用程式的風格。當應用於 Web 服務 API 的設計時、它建立了一套技術和最佳實務做法、用於公開伺服器型資源並管理其狀態。它使用主流傳輸協定和標準、為部署和管理 ONTAP Select 叢集提供靈活的基礎。

架構與傳統限制

REST 由 Roy Fielding 在 2000 年於 UC Irvine 攻讀博士學位 "論文"期間正式提出。它透過一系列約束定義了一種架構風格，這些約束共同改進了基於 Web 的應用程式及其底層協定。這些約束建立了一種基於客戶端/伺服器架構並使用無狀態通訊協定的 RESTful Web 服務應用程式。

資源和狀態表示法

資源是網路型系統的基本元件。建立 REST Web 服務應用程式時，早期設計工作包括：

- 系統或伺服器資源的識別 每個系統都會使用和維護資源。資源可以是檔案、業務交易、流程或管理實體。基於 REST Web 服務設計應用程式的首要任務之一就是識別資源。
- 資源狀態及其相關狀態操作的定義 資源始終處於有限數量的狀態之一。必須明確定義這些狀態以及用於影響狀態變化的相應操作。

用戶端和伺服器之間交換訊息,以根據通用的 CRUD (Create、Read、Update 和 Delete) 模型存取和變更資源的狀態。

URI 端點

每個 REST 資源都必須使用定義完善的定址方案進行定義和提供。資源所在和識別的端點使用統一資源識別碼（URI）。URI 提供了一個通用框架，用於為網路中的每個資源建立唯一名稱。統一資源定位器（URL）是一種與 Web 服務搭配使用的 URI 類型，用於識別和存取資源。資源通常以類似於檔案目錄的階層式結構公開。

HTTP 訊息

超文本傳輸協定（HTTP）是 Web 服務用戶端和伺服器用於交換資源請求和回應訊息的協定。在設計 Web 服務應用程式時，HTTP 動詞（例如 GET 和 POST）會被對應到資源和對應的狀態管理操作。

HTTP 是無狀態的。因此，要將一組相關的請求和回應關聯到一個交易中，必須在請求/回應資料流攜帶的 HTTP 標頭中包含額外資訊。

JSON 格式化

雖然資訊可以透過多種方式在客戶端和伺服器之間進行結構化和傳輸，但最常用的方式（也是 Deploy REST API 使用的方式）是 JavaScript 物件表示法（JSON）。JSON 是一種行業標準，用於以純文字形式表示簡單的資料結構，並用於傳輸描述資源的狀態資訊。

了解如何存取 ONTAP Select Deploy API

由於 REST Web 服務固有的靈活性，可以透過多種不同的方式存取 ONTAP Select Deploy API。

ONTAP Select Deploy 中所包含的 REST API 被指派了一個版本號碼。此 API 版本號與 Deploy 版本號無關。ONTAP Select 9.19.1 Deploy 管理公用程式包含 REST API 的版本 3。

部署公用程式原生使用者介面

存取 API 的主要方式是透過 ONTAP Select Deploy Web 使用者介面。瀏覽器會呼叫 API 並根據使用者介面的設計重新格式化資料。您也可以透過 Deploy 公用程式命令列介面存取 API。

ONTAP Select Deploy 線上文件頁面

使用瀏覽器時，ONTAP Select Deploy 線上文件頁面提供了另一種存取途徑。除了可以直接執行各個 API 呼叫之外，頁面還包含 API 的詳細描述，包括每個呼叫的輸入參數和其他選項。API 呼叫以不同的功能領域或類別進行組織。

自訂程式

您可以使用多種程式語言和工具存取 Deploy API。常用的語言包括 Python、Java 和 cURL。使用該 API 的程式、腳本或工具可作為 REST Web 服務用戶端。使用程式語言可以幫助您更好地理解 API，並有機會實現 ONTAP Select 部署的自動化。

ONTAP Select Deploy API 基本操作特性

雖然 REST 建立了一套通用的技術和最佳實務做法，但每個 API 的具體細節會因設計選擇而異。在使用 ONTAP Select Deploy API 之前，您應該了解其詳細資訊和操作特性。

Hypervisor 主機與 ONTAP Select 節點

hypervisor host 是承載 ONTAP Select 虛擬機器的核心硬體平台。當 ONTAP Select 虛擬機器部署並在 hypervisor host 上作用中時，此虛擬機器被視為 *ONTAP Select* 節點。在 Deploy REST API 版本 3 中，主機和節點物件是分離且不同的。這允許一對多的關係，即一個或多個 ONTAP Select 節點可以在同一個 hypervisor host 上執行。

物件識別碼

每個資源執行個體或物件在建立時都會被指派一個唯一識別碼。這些識別碼在 ONTAP Select Deploy 的特定執行個體中是全域唯一的。發出建立新物件執行個體的 API 呼叫後,相關的 ID 值會在 HTTP 回應的 `location` 標頭中傳回給呼叫方。您可以擷取該識別碼,並在後續呼叫中參照該資源執行個體時使用它。

物件識別碼的內容和內部結構可能隨時變更。在引用關聯物件時，您應該僅在需要時在對應的 API 呼叫中使用這些識別碼。

請求識別碼

每個成功的 API 請求都會被指派一個唯一的識別碼。該識別碼會在相關聯的 HTTP 回應的 `request-id`標頭中傳回。您可以使用請求識別碼來統稱單一特定 API 請求-回應交易的活動。例如、您可以根據請求 ID 擷取交易的所有事件訊息。

同步和非同步呼叫

伺服器處理從客戶端收到的 HTTP 請求主要有兩種方式：

- 同步伺服器會立即執行要求，並以狀態碼 200、201 或 204 回應。
- 非同步處理：伺服器接受請求並傳回狀態碼 202。這表示伺服器已接受客戶端請求，並啟動後台任務以完成該請求。最終的成功或失敗結果不會立即顯示，必須透過額外的 API 呼叫來確定。

確認長時間執行工作的完成

通常，任何耗時較長的操作都會在伺服器端使用後台任務非同步處理。在 Deploy REST API 中，每個後台任務都由一個 Job 物件關聯，該物件追蹤任務並提供諸如當前狀態之類的資訊。後台任務建立後，HTTP 回應會傳回一個包含其唯一識別碼的 Job 物件。

您可以直接查詢 Job 物件，以判斷相關 API 呼叫是否成功。如需其他資訊，請參閱_使用 Job 物件進行非同步處理_。

除了使用 Job 物件之外，還有其他方法可以判斷要求是否成功或失敗，包括：

- 事件訊息您可以使用原始回應中傳回的請求 ID 來擷取與特定 API 呼叫關聯的所有事件訊息。事件訊息通常包含成功或失敗的指示，在偵錯錯誤情況時也很有用。
- 資源狀態或狀態 許多資源都維護一個狀態或狀態值、您可以查詢該值以間接確定請求是否成功或失敗。

安全性

Deploy API 使用以下安全技術：

- 傳輸層安全性協定 (TLS) Deploy 伺服器和用戶端之間透過網路傳輸的所有流量均透過 TLS 加密。不支援透過未加密通道使用 HTTP 協定。支援 TLS 1.2 版。
- HTTP 驗證基本驗證用於每個 API 交易。每個請求都會新增一個 HTTP 標頭，其中包含 base64 字串中的使用者名稱和密碼。

ONTAP Select 的請求和回應 API 交易

每次 Deploy API 呼叫都是向 Deploy 虛擬機器發送 HTTP 請求，虛擬機器隨後會產生相應的回應傳送給客戶端。這個請求/回應對被視為 API 事務。在使用 Deploy API 之前，您應該熟悉可用於控制請求的輸入變數以及回應輸出的內容。

控制 API 要求的輸入變數

您可以透過在 HTTP 請求中設定的參數來控制 API 呼叫的處理方式。

請求標頭

您必須在 HTTP 請求中包含多個標頭，包括：

- content-type 如果請求正文包含 JSON，則此標頭必須設定為 application/json。
- accept 如果回應正文將包含 JSON，則此標頭必須設定為 application/json。
- 授權基本驗證必須使用以 base64 字串編碼的使用者名稱和密碼進行設定。

請求本文

請求本文的內容會因特定呼叫而異。HTTP 請求本文包含下列其中一項：

- 包含輸入變數（例如新叢集的名稱）的 JSON 物件
- 空白

篩選物件

使用 GET 方法發出 API 呼叫時，您可以根據任何屬性限制或篩選傳回的物件。例如，您可以指定要匹配的確切值：

```
<field>=<query value>
```

除了精確匹配之外，還有其他運算子可用於傳回一組值範圍內的物件。ONTAP Select 支援以下所示的篩選運算子。

操作員	說明
=	等於
<	少於
>	大於
⇐	小於或等於
>=	大於或等於
|	或
!	不等於
*	貪婪萬用字元

您也可以透過在查詢中使用 null 關鍵字或其否定（!null）來根據特定欄位是否已設定傳回一組物件。

選取物件欄位

根據預設、使用 GET 發出 API 呼叫只會傳回唯一識別物件的屬性。這組最小欄位可做為每個物件的金鑰、並會因物件類型而異。您可以使用 fields 查詢參數、以下列方式選取其他物件內容：

- 低成本欄位 指定 `fields=*` 以擷取儲存在本機伺服器記憶體中或存取時處理量很少的物件欄位。
- 昂貴的欄位 指定 `fields=**` 以擷取所有物件欄位，包括需要額外伺服器處理才能存取的欄位。
- 自訂欄位選擇使用 `fields=FIELDNAME`指定所需的確切欄位。請求多個欄位時，值之間必須用逗號分隔，

且不能有空格。



最佳實務做法是，您應該始終明確所需欄位。僅在需要時才擷取低成本或高成本欄位集。低成本和高成本的分類是由 NetApp 根據內部效能分析確定。特定欄位的分類可能隨時變更。

對輸出集中的物件進行排序

資源集合中的記錄將按照物件定義的預設順序傳回。您可以使用 order_by 查詢參數，並指定欄位名稱和排序方向來變更順序，如下所示：

```
order_by=<field name> asc|desc
```

例如，您可以先按降序排列類型欄位，再按升序排列 id 欄位：

```
order_by=type desc, id asc
```

當包含多個參數時，必須用逗號分隔各個欄位。

分頁

使用 GET 發出 API 呼叫以存取相同類型的物件集合時，預設會傳回所有相符的物件。如有需要，您可以使用請求中的 max_records 查詢參數來限制傳回的記錄數。例如：

```
max_records=20
```

如有需要，您可以將此參數與其他查詢參數結合使用，以縮小結果集。例如，下列查詢將傳回指定時間之後產生的最多 10 個系統事件：

```
time⇒ 2019-04-04T15:41:29.140265Z&max_records=10
```

您可以發出多個請求來分頁瀏覽事件（或任何物件類型）。每次後續的 API 呼叫都應使用基於上一個結果集中最新事件的新時間值。

解讀 API 回應

每個 API 請求都會向用戶端傳回一個回應。您可以檢查該回應以確定請求是否成功，並根據需要擷取其他資料。

HTTP 狀態碼

以下說明 Deploy REST API 所使用的 HTTP 狀態碼。

程式碼	意義	說明
200	確定	表示不會建立新物件的呼叫成功。
201	已建立	已成功建立物件；位置回應標頭包含物件的唯一識別碼。
202	已接受	已啟動長時間執行的背景工作來執行要求，但作業尚未完成。
400	錯誤的請求	要求輸入無法辨識或不適當。
403	禁止	由於授權錯誤，存取被拒絕。
404	找不到	請求中引用的資源不存在。
405	不允許使用此方法	此資源不支援請求中的 HTTP 謂詞。
409	衝突	嘗試建立物件失敗，因為該物件已存在。

程式碼	意義	說明
500	內部錯誤	伺服器發生一般性內部錯誤。
501	未實作	URI 已知，但無法執行請求。

回應標頭

Deploy 伺服器產生的 HTTP 回應中包含多個標頭、包括：

- request-id 每個成功的 API 請求都會被指派一個唯一的請求識別碼。
- location 當建立物件時，location 標頭包含新物件的完整 URL，包括唯一的物件識別碼。

回應本文

與 API 請求相關聯的回應內容會根據物件、處理類型以及請求成功或失敗而有所不同。回應本文以 JSON 格式呈現。

- 單一物件：可以根據請求傳回包含一組欄位的單一物件。例如，您可以使用 GET 請求，透過唯一識別碼檢索叢集的特定屬性。
- 多個物件可以從資源集合中傳回多個物件。在所有情況下，都使用一致的格式，其中 `num_records`表示記錄數，而記錄包含物件實例的陣列。例如，您可以擷取特定叢集中定義的所有節點。
- Job 物件如果 API 呼叫是非同步處理的，則會傳回一個 Job 物件，該物件用於錨定背景工作。例如，用於部署叢集的 POST 要求是非同步處理的，並傳回一個 Job 物件。
- Error 物件如果發生錯誤，則會始終傳回 Error 物件。例如，嘗試建立名稱已存在的叢集時，將會收到錯誤。
- 空白在某些情況下，不會傳回任何資料，且回應本文為空白。例如，使用 DELETE 刪除現有主機後，回應本文為空白。

使用 ONTAP Select 的 Job 物件進行非同步處理

某些部署 API 調用，特別是建立或修改資源的調用，可能比其他調用耗時更長。ONTAP Select Deploy 會非同步處理這些耗時較長的請求。

使用 Job 物件描述的非同步請求

在進行非同步執行的 API 呼叫後，HTTP 回應代碼 202 表示請求已成功驗證並被接受，但尚未完成。此請求會作為背景工作進行處理，在向用戶端發出初始 HTTP 回應後仍會繼續執行。回應中包含錨定該請求的 Job 物件，以及其唯一識別碼。

您應該參考 ONTAP Select Deploy 線上文件頁面，以確定哪些 API 呼叫是非同步運作的。

查詢與 API 請求相關聯的 Job 物件

HTTP 回應中傳回的 Job 物件包含多個屬性。您可以查詢 state 屬性來決定請求是否成功完成。Job 物件可以處於以下狀態之一：

- 已排入佇列

- 執行中
- 成功
- 失敗

在輪詢 Job 物件以偵測任務的終止狀態（成功或失敗）時，可以使用兩種技術：

- 標準輪詢請求會立即傳回目前 Job 狀態。
- 長輪詢請求工作狀態僅在發生下列其中一種情況時傳回：
  - 狀態變更時間比輪詢請求中提供的日期時間值更近。
  - 逾時值已過期（1 至 120 秒）

標準輪詢和長輪詢使用相同的 API 呼叫來查詢 Job 物件。但是，長輪詢請求包含兩個查詢參數：`poll_timeout` 和 `last_modified`。



您應該一律使用長輪詢，以減少 Deploy 虛擬機器的工作負載。

發出非同步請求的一般程序

您可以使用下列高階程序來完成非同步 API 呼叫：

1. 發出非同步 API 呼叫。
2. 收到 HTTP 回應 202，表示請求已成功接受。
3. 從回應正文中提取 Job 物件的識別碼。
4. 在一個迴圈中，在每個週期內執行以下操作：
   a. 透過長輪詢請求取得工作的目前狀態
   b. 如果工作處於非終止狀態（已排入佇列、執行中），請再次執行迴圈。
5. 當工作達到終止狀態（成功、失敗）時停止。

版權資訊

Copyright © 2026 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。