



概念 ONTAP Select

NetApp
January 29, 2026

目錄

概念	1
用於部署和管理ONTAP Select叢集的 REST Web 服務基礎	1
建築與經典約束	1
資源和國家代表	1
URI 端點	1
HTTP 郵件	1
JSON 格式	1
如何存取ONTAP Select Deploy API	2
部署實用程式原生使用者介面	2
ONTAP Select Deploy 線上文件頁面	2
定製程式	2
ONTAP Select Deploy API 基本操作特性	2
虛擬機器管理程式主機與ONTAP Select節點	2
物件標識符	2
請求標識符	3
同步和異步調用	3
確認長期運行作業的完成	3
安全	3
ONTAP Select的請求和回應 API 事務	3
控制 API 請求的輸入變數	3
解釋 API 回應	5
使用ONTAP Select 的作業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 請求關聯的 Job 對象	6
發出非同步請求的一般過程	7

概念

用於部署和管理ONTAP Select叢集的 REST Web 服務基礎

表述性狀態傳輸 (REST) 是一種用於建立分散式 Web 應用程式的樣式。當應用於 Web 服務 API 的設計時，它可以建立一套用於公開基於伺服器的資源並管理其狀態的技術和最佳實踐。它使用主流協定和標準，為部署和管理ONTAP Select叢集提供靈活的基礎。

建築與經典約束

REST 理論由 Roy Fielding 在博士論文中正式闡述。["論文"](#) 2000年畢業於加州大學爾灣分校。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。這些約束基於客戶端/伺服器架構，使用無狀態通訊協議，建構了一個RESTful Web服務應用程式。

資源和國家代表

資源是基於 Web 系統的基本元件。在建立 REST Web 服務應用程式時，早期設計任務包括：

- 識別系統或基於伺服器的資源。每個系統都會使用和維護資源。資源可以是文件、業務事務、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關操作。

用戶端和伺服器之間交換訊息，根據通用 CRUD（建立、讀取、更新和刪除）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明確的尋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標識符 (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。統一資源定位符 (URL) 是一種用於 Web 服務識別和存取資源的 URI。資源通常以類似文件目錄的層級結構公開。

HTTP 郵件

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

HTTP 是無狀態的。因此，為了將一組相關的請求和回應關聯到一個事務下，必須在請求/回應資料流攜帶的 HTTP 標頭中包含附加資訊。

JSON 格式

雖然資訊可以透過多種方式在客戶端和伺服器之間建置和傳輸，但最受歡迎的選項（以及與部署 REST API 一起使用的選項）是 JavaScript 物件表示法 (JSON)。JSON是一種行業標準，用於以純文字表示簡單資料結構，並用於傳輸描述資源的狀態資訊。

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 隨附的 REST API 已指派一個版本號碼。API版本號碼與 Deploy 版本號無關。ONTAP Select 9.17.1 Deploy 管理公用程式包含 REST API 版本 3。

部署實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 實用程式命令列介面存取 API。

ONTAP Select Deploy 線上文件頁面

ONTAP Select Deploy 線上文件頁面提供了使用瀏覽器時的備用存取點。除了提供直接執行各個 API 呼叫的方法外，該頁面還包含 API 的詳細說明，包括每個呼叫的輸入參數和其他選項。API呼叫分為幾個不同的功能區域或類別。

定製程式

您可以使用多種不同的程式語言和工具存取 Deploy API。常用的選擇包括 Python、Java 和 cURL。使用該 API 的程式、腳本或工具可充當 REST Web 服務用戶端。使用程式語言可以幫助您更好地理解 API，並實現ONTAP Select部署的自動化。

ONTAP Select Deploy API 基本操作特性

雖然 REST 確立了一套通用的技術和最佳實踐，但每個 API 的細節可能會因設計選擇而異。在使用ONTAP Select Deploy API 之前，您應該了解該 API 的詳細資訊和操作特性。

虛擬機器管理程式主機與ONTAP Select節點

虛擬機器管理程式主機是託管ONTAP Select虛擬機器的核心硬體平台。當ONTAP Select虛擬機器部署在虛擬機器管理程式主機上並處於活動狀態時，該虛擬機器將被視為 ONTAP Select 節點。在 Deploy REST API 版本 3 中，主機物件和節點物件是獨立且不同的。這允許一對多關係，其中一個或多個ONTAP Select節點可以在同一虛擬機器管理程式主機上運行。

物件標識符

每個資源實例或物件在建立時都會指派一個唯一識別碼。這些標識符在ONTAP Select Deploy 的特定實例中是全域唯一的。在發出建立新物件實例的 API 呼叫後，關聯的 id 值將傳回給呼叫者。location HTTP回應頭。您可以提取標識符，並在後續呼叫中引用資源實例時使用它。



物件標識符的內容和內部結構可能隨時變更。在引用關聯物件時，應僅在適用的 API 呼叫中根據需要使用這些標識符。

請求標識符

每個成功的 API 請求都會被指派一個唯一的識別碼。傳回的標識符 `request-id` 相關 HTTP 響應的頭部。您可以使用請求識別碼來統稱單一特定 API 請求-回應事務的活動。例如，您可以根據請求 ID 檢索交易的所有事件訊息。

同步和異步調用

伺服器執行從客戶端收到的 HTTP 請求主要有兩種方式：

- 同步伺服器立即執行請求並以狀態代碼 200、201 或 204 回應。
- 非同步：伺服器接受請求並以狀態碼 202 回應。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。最終的成功或失敗情況無法立即確定，必須透過其他 API 呼叫來確定。

確認長期運行作業的完成

通常，任何需要很長時間才能完成的操作都會使用伺服器上的後台任務非同步處理。使用 Deploy REST API，每個後台任務都由一個 Job 物件固定，該物件追蹤任務並提供資訊（例如當前狀態）。建立後台任務後，HTTP 回應中會傳回一個 Job 物件（包含其唯一識別碼）。

您可以直接查詢 Job 物件來確定相關 API 呼叫是否成功。有關更多信息，請參閱 [使用 Job 物件進行非同步處理](#)。

除了使用 Job 物件之外，還有其他方法可以確定請求的成功或失敗，包括：

- 事件訊息您可以使用原始回應傳回的請求 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。
- 資源狀態或狀況 有些資源維持一個狀態或狀況值，您可以查詢該狀態或狀況值來間接確定請求的成功或失敗。

安全

Deploy API 使用以下安全技術：

- 傳輸層安全性 (TLS)：Deploy 伺服器和用戶端之間透過網路傳送的所有流量均透過 TLS 加密。不支援在未加密的通道上使用 HTTP 協定。支援 TLS 1.2 版本。
- HTTP 驗證 每個 API 交易都使用基本驗證。每個請求都會新增一個 HTTP 標頭，其中包含以 base64 字串形式表示的使用者名稱和密碼。

ONTAP Select的請求和回應 API 事務

每個 Deploy API 呼叫都會以 HTTP 請求的形式傳送到 Deploy 虛擬機，並產生相關的回應傳回給客戶端。此請求/回應對被視為一次 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 請求的輸入變數

您可以透過 HTTP 請求中設定的參數來控制如何處理 API 呼叫。

請求標頭

您必須在 HTTP 請求中包含幾個標頭，其中包括：

- content-type 如果請求主體包含 JSON，則此標頭必須設定為 application/json。
- accept 如果回應主體將包含 JSON，則此標頭必須設定為 application/json。
- 授權必須使用以 base64 字串編碼的使用者名稱和密碼設定基本驗證。

請求正文

請求主體的內容根據具體呼叫而有所不同。HTTP 請求主體由以下之一組成：

- 帶有輸入變數的 JSON 物件（例如，新叢集的名稱）
- 空的

篩選對象

發出使用 GET 的 API 呼叫時，您可以根據任意屬性限制或過濾傳回的物件。例如，您可以指定要符合的精確值：

<field>=<query value>

除了完全匹配之外，還有其他運算子可用於傳回一組包含特定值範圍的物件。ONTAP Select 支援以下篩選運算子。

操作員	描述
=	等於
<	少於
>	大於
≤	小於或等於
≥	大於或等於
	或者
!	不等於
*	貪婪通配符

您也可以透過使用 null 關鍵字或其否定（! null）作為查詢的一部分，根據特定欄位是否設定來傳回一組物件。

選擇對象字段

預設情況下，使用 GET 發出 API 呼叫僅傳回唯一標識一個或多個物件的屬性。這組最小欄位將充當每個物件的鍵，並根據物件類型而有所不同。您可以使用 fields 查詢參數透過以下方式選擇其他物件屬性：

- 廉價字段指定 `fields=\*` 檢索保存在本機伺服器記憶體中或存取時幾乎不需要處理的物件欄位。
- 昂貴的字段 `fields=\*\*` 檢索所有物件字段，包括那些需要額外伺服器處理才能存取的字段。
- 自訂欄位選擇 `fields=FIELDNAME` 請指定您想要的確切欄位。請求多個欄位時，必須使用逗號分隔值，且

不能使用空格。



作為最佳實踐，您應該始終識別所需的特定欄位。您應僅在需要時檢索廉價或昂貴欄位集。廉價和昂貴的分類由NetApp根據內部效能分析決定。給定字段的分類可能隨時變更。

對輸出集中的物件進行排序

資源集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，透過指定欄位名稱和排序方向來變更排序順序，如下所示：

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

例如，您可以先按類型欄位降序排序，再按 ID 欄位升序排序：

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

當包含多個參數時，必須用逗號分隔欄位。

分頁

使用 GET 方式發出 API 呼叫來存取相同類型的物件集合時，預設會傳回所有符合的物件。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
```

如果需要，您可以將此參數與其他查詢參數組合使用，以縮小結果集。例如，以下命令傳回指定時間後產生的最多 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）。每個後續 API 呼叫都應根據最後一個結果集中的最新事件使用新的時間值。

解釋 API 回應

每個 API 請求都會產生一個回應傳回給客戶端。您可以檢查該回應以確定請求是否成功，並根據需要檢索其他資料。

HTTP 狀態碼

下面描述了部署 REST API 使用的 HTTP 狀態碼。

程式碼	意義	描述
200	好的	表示沒有創建新物件的呼叫成功。
201	創建	物件已成功建立；位置回應標頭包含該物件的唯一識別碼。
202	公認	已啟動長時間運行的背景作業來執行請求，但操作尚未完成。
400	錯誤的請求	請求輸入無法識別或不合適。
403	禁止	由於授權錯誤，存取被拒絕。
404	未找到	請求中引用的資源不存在。
405	方法不允許	此資源不支援請求中的 HTTP 動詞。
409	衝突	嘗試建立物件失敗，因為該物件已存在。

程式碼	意義	描述
500	內部錯誤	伺服器發生一般內部錯誤。
501	未實施	URI 已知但無法執行請求。

響應標頭

Deploy 伺服器產生的 HTTP 回應中包含幾個標頭，包括：

- request-id 每個成功的 API 請求都會指派一個唯一的請求識別碼。
- 位置 當建立物件時，位置標頭包含新物件的完整 URL，其中包含唯一物件識別碼。

回應正文

與 API 請求相關的回應內容會根據物件、處理類型以及請求的成功或失敗而有所不同。回應主體以 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 操作，透過唯一識別碼檢索叢集的選定屬性。
- 多個物件 可以傳回資源集中的多個物件。所有情況下都使用一致的格式，即：`num\_records` 表示記錄數和包含物件實例陣列的記錄數。例如，您可以檢索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 對象，該物件用於錨定後台任務。例如，用於部署叢集的 POST 請求是非同步處理的，並傳回一個 Job 物件。
- 錯誤物件 如果發生錯誤，則總是會傳回一個 Error 物件。例如，當您嘗試建立名稱已存在的叢集時，就會收到錯誤。
- 空 某些情況下，沒有回傳任何數據，響應體為空，例如使用 DELETE 刪除一個已經存在的主機，響應體為空。

使用ONTAP Select 的作業物件進行非同步處理

某些 Deploy API 呼叫（尤其是建立或修改資源的呼叫）可能需要比其他呼叫更長的時間才能完成。ONTAP Select Deploy 會非同步處理這些長時間運行的請求。

使用 Job 物件描述的非同步請求

進行非同步運行的 API 呼叫後，HTTP 回應代碼 202 表示請求已成功驗證並接受，但尚未完成。該請求將作為後台任務處理，並在客戶端收到初始 HTTP 回應後繼續執行。回應中包含錨定該請求的 Job 物件及其唯一識別碼。



您應該參考ONTAP Select Deploy 線上文件頁面來決定哪些 API 呼叫是非同步操作的。

查詢與 API 請求關聯的 Job 對象

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已成功完成。Job物件可以處於以下狀態之一：

- 排隊

- 跑步
- 成功
- 失敗

輪詢 Job 物件來偵測任務的最終狀態（成功或失敗）時，可以使用兩種技術：

- 標準輪詢請求立即返回目前作業狀態
- 長輪詢請求僅當發生以下情況之一時才傳回作業狀態：
 - 狀態變更的時間比輪詢請求中提供的日期時間值更近
 - 超時值已過期（1 至 120 秒）

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



您應該始終使用長輪詢來減少 Deploy 虛擬機器上的工作負載。

發出非同步請求的一般過程

您可以使用下列進階程序來完成非同步 API 呼叫：

1. 發出異步 API 呼叫。
2. 收到 HTTP 回應 202，表示成功接受請求。
3. 從回應主體中提取 Job 物件的識別碼。
4. 在循環內，每次循環執行以下操作：
 - a. 使用長輪詢請求取得作業的目前狀態
 - b. 如果作業處於非終止狀態（排隊、執行），則再次執行循環。
5. 當作業達到最終狀態（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。