# 概念
## ONTAP Select

NetApp
April 29, 2024

# 目录

# 概念

## REST Web 服务基础

表述性状态传输（ Representational State Transfer ， REST ）是一种用于创建分布式 Web 应用程序的模式。在设计 Web 服务 API 时，它会建立一组技术和最佳实践，用于公开基于服务器的资源并管理其状态。它使用主流协议和标准为部署和管理 ONTAP Select 集群提供了灵活的基础。

### 架构和传统限制

而其余内容则由 Roy Fielding 博士正式阐述 "Dissertation" 2000 年在 UC Irvine 大会上。它通过一组限制来定义架构模式，这些限制共同改进了基于 Web 的应用程序和底层协议。这些限制会根据使用无状态通信协议的客户端 / 服务器架构建立 RESTful Web 服务应用程序。

### 资源和状态表示

资源是基于 Web 的系统的基本组件。创建 REST Web 服务应用程序时，早期设计任务包括：

- 识别系统或基于服务器的资源
  每个系统都使用和维护资源。资源可以是文件，业务事务，流程或管理实体。在设计基于 REST Web 服务的应用程序时，首先要完成的任务之一是识别资源。
- 资源状态和关联状态操作的定义
  资源始终处于数量有限的状态之一。必须明确定义状态以及用于影响状态更改的关联操作。

客户端和服务器之间会交换消息，以便根据通用 CRUD （创建，读取，更新和删除）模式访问和更改资源的状态。

### URI 端点

必须使用定义明确的寻址方案定义和提供每个 REST 资源。资源所在的端点和标识的端点使用统一资源标识符（ Uniform Resource Identifier ， URI ）。URI 提供了一个通用框架，用于为网络中的每个资源创建唯一名称。统一资源定位器（ Uniform Resource Locator ， URL ）是一种用于 Web 服务的 URI 类型，用于标识和访问资源。资源通常以类似于文件目录的分层结构公开。

### HTTP消息

超文本传输协议（ HTTP ）是 Web 服务客户端和服务器用来交换有关资源的请求和响应消息的协议。在设计 Web 服务应用程序时， HTTP 动词（例如 GET 和 POST ）会映射到资源以及相应的状态管理操作。

HTTP 为无状态。因此，要在一个事务下关联一组相关请求和响应，追加信息 必须包含在请求 / 响应数据流附带的 HTTP 标头中。

### JSON 格式化

虽然信息可以通过多种方式在客户端和服务器之间进行结构化和传输，但最常用的选项（以及 Deploy REST API 中使用的选项）是 JavaScript 对象表示法（ JSON ）。JSON 是一种行业标准，用于以纯文本形式表示简单数据结构，并用于传输描述资源的状态信息。

# 如何访问 Deploy API

由于 REST Web 服务具有固有的灵活性，因此可以通过多种不同的方式访问 ONTAP Select Deploy API 。

### Deploy 实用程序原生 用户界面

访问 API 的主要方式是通过 ONTAP Select Deploy Web 用户界面。浏览器调用 API 并根据用户界面的设计重新格式化数据。您还可以通过 Deploy 实用程序命令行界面访问此 API 。

### ONTAP Select Deploy 联机文档页面

使用浏览器时， ONTAP Select Deploy 联机文档页面提供了一个备用访问点。除了提供直接执行单个 API 调用的方法之外，此页面还包括 API 的详细问题描述 ，包括每个调用的输入参数和其他选项。API 调用分为多个不同的功能区域或类别。

### 自定义程序

您可以使用多种不同的编程语言和工具访问 Deploy API 。常见选项包括 Python ， Java 和 CURL 。使用 API 的程序，脚本或工具充当 REST Web 服务客户端。通过使用编程语言，您可以更好地了解 API ，并有机会自动执行 ONTAP Select 部署。

# 部署 API 版本控制

ONTAP Select Deploy 附带的 REST API 分配有一个版本号。API 版本号与 Deploy 版本号无关。您应了解您的 Deploy 版本附带的 API 版本，以及此版本可能会对您使用此 API 产生何种影响。

当前版本的 Deploy 管理实用程序包括 REST API 版本 3 。Deploy 实用程序的以往版本包括以下 API 版本：

部署 **2.8** 及更高版本

ONTAP Select Deploy 2.8 及所有更高版本均包含 REST API 版本 3 。

部署 **2.7.2** 及更早版本

ONTAP Select Deploy 2.7.2 及所有早期版本均包含 REST API 版本 2 。

> REST API 版本 2 和 3 不兼容。如果从包含 API 版本 2 的早期版本升级到 Deploy 2.8 或更高版本，则必须更新直接访问此 API 的任何现有代码以及使用命令行界面的任何脚本。

# 基本操作特征

虽然 REST 建立了一组通用的技术和最佳实践，但每个 API 的详细信息可能因设计选择而异。在使用 ONTAP Select Deploy API 之前，您应了解 API 的详细信息和操作特征。

## 虚拟机管理程序主机与 ONTAP Select 节点

虚拟机管理程序 host_ 是托管 ONTAP Select 虚拟机的核心硬件平台。在虚拟机管理程序主机上部署 ONTAP Select 虚拟机并使其处于活动 ONTAP Select 状态时，该虚拟机将被视为 _vp 节点 _ 。在 Deploy REST API 版本 3 中，主机和节点对象是独立的。这样就可以建立一对多关系，其中一个或多个 ONTAP Select 节点可以在同一虚拟机管理程序主机上运行。

## 对象标识符

创建每个资源实例或对象时，系统会为其分配一个唯一标识符。这些标识符在 ONTAP Select Deploy 的特定实例中具有全局唯一性。发出创建新对象实例的API调用后、关联的ID值将返回到中的调用方 `location` HTTP响应的标题。在引用资源实例时，您可以提取此标识符并在后续调用中使用它。

> 对象标识符的内容和内部结构可以随时更改。仅当引用关联对象时，才应根据需要在适用的 API 调用上使用标识符。

## 请求标识符

每个成功的 API 请求都会分配一个唯一标识符。此标识符将在中返回 `request-id` 关联HTTP响应的标头。您可以使用请求标识符来统称单个特定 API 请求响应事务的活动。例如，您可以根据请求 ID 检索事务的所有事件消息。

## 同步和异步调用

服务器执行从客户端收到的 HTTP 请求的主要方式有两种：

- 同步
  服务器立即执行请求、并使用状态代码200、201或204进行响应。
- 异步
  服务器接受请求并使用状态代码202进行响应。这表示服务器已接受客户端请求并启动后台任务来完成此请求。最终成功或失败不会立即出现，必须通过其他 API 调用来确定。

## 确认已完成长时间运行的作业

通常、任何可能需要很长时间才能完成的操作都会使用异步处理
服务器上的后台任务。通过Deploy REST API、每个后台任务都由锚定
作业对象、用于跟踪任务并提供当前状态等信息。作业对象，
创建后台任务后、HTTP响应将返回其唯一标识符。

您可以直接查询作业对象以确定关联 API 调用的成功或失败。
有关追加信息 ，请参见 *asynchronous processing using the Job objection* 。

除了使用作业对象之外、还可以通过其他方法确定的成功或失败
请求，包括：

- 事件消息
  您可以使用随原始响应返回的请求ID检索与特定API调用关联的所有事件消息。事件消息通常包含成功或失败的指示，在调试错误情况时也很有用。
- 资源状态
  有几个资源保持状态或状态值、您可以查询这些状态或值来间接确定请求的成功或失败。

## 安全性

Deploy API 使用以下安全技术：

- 传输层安全性
  通过网络在Deploy服务器和客户端之间发送的所有流量都会通过TLS进行加密。不支持在未加密的通道上使用 HTTP 协议。支持 TLS 1.2 版。
- HTTP身份验证
  基本身份验证用于每个API事务。每个请求都会添加一个 HTTP 标头，其中包含 base64 字符串中的用户名和密码。

# 请求和响应 API 事务

每次 Deploy API 调用都会作为 HTTP 请求执行给 Deploy 虚拟机，此请求会向客户端生成关联的响应。此请求 / 响应对被视为 API 事务。在使用 Deploy API 之前，您应熟悉可用于控制请求的输入变量以及响应输出的内容。

## 控制 API 请求的输入变量

您可以控制如何通过 HTTP 请求中设置的参数处理 API 调用。

### 请求标题

您必须在 HTTP 请求中包含多个标头，包括：

- 内容类型
  如果请求正文包括JSON、则必须将此标头设置为application/json。
- 接受
  如果响应正文将包括JSON、则必须将此标题设置为application/json。
- 授权
  必须使用base64字符串编码的用户名和密码设置基本身份验证。

### 请求正文

请求正文的内容因具体调用而异。HTTP 请求正文包含以下内容之一：

- 包含输入变量（例如新集群的名称）的 JSON 对象
- 空

### 筛选对象

发出使用 GET 的 API 调用时，您可以根据任何属性限制或筛选返回的对象。例如，您可以指定一个要匹配的精确值：

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

除了精确匹配之外，还有其他运算符可用于返回一组值范围内的对象。ONTAP Select 支持如下所示的筛选运算符。

| 运算符 | Description |
| --- | --- |
| = | 等于 |
| < | 小于 |
| > | 大于 |
| &lt； = | 小于或等于 |
| &gt； = | 大于或等于 |
| | 或 |
| ! | 不等于 |
| * | 贪婪的通配符 |

您还可以在查询中使用 null 关键字或其否定（！ null ）来根据是否设置了特定字段返回一组对象。

选择对象字段

默认情况下，使用 GET 发出 API 调用时，仅返回唯一标识一个或多个对象的属性。这组最小的字段可用作每个对象的密钥，并因对象类型而异。您可以通过以下方式使用 fields query 参数选择其他对象属性：

- 低成本字段
  指定 `fields=*` 检索本地服务器内存中维护的对象字段或只需少量处理即可访问的对象字段。
- 昂贵的字段
  指定 `fields=**` 检索所有对象字段、包括需要额外服务器处理才能访问的对象字段。
- 自定义字段选择
  使用 ... `fields=FIELDNAME` 以指定所需的确切字段。请求多个字段时，必须使用逗号分隔值，不能包含空格。

> 作为最佳实践，您应始终确定所需的特定字段。您只能在需要时检索一组廉价或昂贵的字段。成本低且昂贵的分类由 NetApp 根据内部性能分析确定。给定字段的分类可以随时更改。

对输出集中的对象进行排序

资源收集中的记录将按对象定义的默认顺序返回。您可以使用order_by查询参数以及字段名称和排序方向更改顺序、如下所示：

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

例如，您可以按降序对类型字段排序，然后按升序对 ID 排序：

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

如果包含多个参数，则必须使用逗号分隔各个字段。

分页

使用 GET 发出 API 调用以访问同一类型的对象集合时，默认情况下会返回所有匹配的对象。如果需要，您可以在请求中使用 max_records 查询参数限制返回的记录数。例如：

```
max_records=20
```

如果需要，您可以将此参数与其他查询参数结合使用，以缩小结果集的范围。例如、以下命令最多返回在指定时

间之后生成的10个系统事件：
`time⇒ 2019-04-04T15:41:29.140265Z&max_records=10`

您可以通过问题描述 发送多个请求来分页查看事件（或任何对象类型）。后续的每个 API 调用应根据最后一个结果集中的最新事件使用一个新的时间值。

## 解释 API 响应

每个 API 请求都会生成对客户端的响应。您可以检查响应以确定是否成功并根据需要检索其他数据。

### HTTP 状态代码

下面介绍了 Deploy REST API 使用的 HTTP 状态代码。

| 代码 | 含义 | Description |
|---|---|---|
| 200 | 确定 | 表示未创建新对象的调用成功。 |
| 201 | 已创建 | 已成功创建对象；位置响应标头包含对象的唯一标识符。 |
| 202. | 已接受 | 已启动长时间运行的后台作业来执行请求，但操作尚未完成。 |
| 400 | 请求错误 | 此请求输入无法识别或不适当。 |
| 403. | 已禁止 | 由于授权错误，访问被拒绝。 |
| 404 | 未找到 | 请求中引用的资源不存在。 |
| 405. | 不允许使用此方法 | 此资源不支持请求中的 HTTP 动词。 |
| 409. | 冲突 | 尝试创建对象失败，因为此对象已存在。 |
| 500 | 内部错误 | 服务器发生一般内部错误。 |
| 501. | 未实施 | 此 URI 已知，但无法执行此请求。 |

#### 响应标头

Deploy 服务器生成的 HTTP 响应包含多个标头，其中包括：

- 请求ID
  每个成功的API请求都会分配一个唯一的请求标识符。
- location
  创建对象时、位置标头包含新对象的完整URL、其中包括唯一对象标识符。

#### 响应正文

与 API 请求关联的响应内容因对象，处理类型以及请求的成功或失败而异。响应正文将在 JSON 中呈现。

- 单个对象
  可以根据请求返回一个对象并显示一组字段。例如，您可以使用 GET 使用唯一标识符检索集群的选定属性。
- 多个对象
  可以从一个资源收集返回多个对象。在所有情况下、都会使用一致的格式 `num_records` 指示包含对象实例

数组的记录和记录的数量。例如，您可以检索特定集群中定义的所有节点。

- 作业对象
  如果异步处理 API 调用，则会返回作业对象，用于将后台任务固定。例如，用于部署集群的 POST 请求会异步处理并返回作业对象。
- 错误对象
  如果发生错误，则始终返回 Error 对象。例如，在尝试创建名称已存在的集群时，您将收到错误消息。
- 空
  在某些情况下、不会返回任何数据、并且响应正文为空。例如，使用 delete 删除现有主机后，响应正文为空。

# 使用作业对象进行异步处理

某些 Deploy API 调用（尤其是创建或修改资源的调用）可能需要比其他调用更长的时间才能完成。ONTAP Select Deploy 会异步处理这些长时间运行的请求。

## 使用作业对象描述的异步请求

发出异步运行的 API 调用后， HTTP 响应代码 202 表示此请求已成功验证并被接受，但尚未完成。此请求将作为后台任务进行处理，在对客户端进行初始 HTTP 响应后，此任务将继续运行。响应包括作业对象锁定请求，包括其唯一标识符。

> 您应参阅 ONTAP Select Deploy 联机文档页面以确定哪些 API 调用异步运行。

## 查询与**API**请求关联的作业对象

HTTP 响应中返回的作业对象包含多个属性。您可以查询 state 属性以确定请求是否成功完成。作业对象可以处于以下状态之一：

- 已排队
- 正在运行
- success
- 失败

在轮询作业对象以检测任务的终端状态时，可以使用两种方法：成功或失败：

- 标准轮询请求
  当前作业状态将立即返回
- 长时间轮询请求
  只有在发生以下情况之一时、才会返回作业状态：
  - 状态的更改日期比轮询请求提供的日期时间值更晚
  - 超时值已过期（ 1 到 120 秒）

标准轮询和长轮询使用相同的 API 调用来查询作业对象。但是、长轮询请求包含两个查询参数：`poll_timeout` 和 `last_modified`。

您应始终使用长轮询来减少 Deploy 虚拟机上的工作负载。

## 用于发出异步请求的常规操作步骤

您可以使用以下高级操作步骤完成异步 API 调用。

1. 问题描述异步 API 调用。
2. 接收表示已成功接受请求的 HTTP 响应 202 。
3. 从响应正文中提取作业对象的标识符。
4. 在环路中，在每个周期中执行以下操作：
   a. 获取具有长时间轮询请求的作业的当前状态
   b. 如果作业处于非终端状态（已排队，正在运行），请重新执行环路。
5. 当作业达到终端状态（成功，失败）时停止。