Konzepte

ONTAP Select

NetApp
February 09, 2024

Inhalt

Konzepte

REST-Web-Services-Grundlage

Representational State Transfer (REST) ist ein Stil für die Erstellung von verteilten Web-Anwendungen. Bei der Anwendung auf das Design einer Web-Services-API werden eine Reihe von Technologien und Best Practices erstellt, um serverbasierte Ressourcen freizulegen und deren Status zu verwalten. Die Technologie nutzt Mainstream-Protokolle und -Standards, um eine flexible Grundlage für die Bereitstellung und das Management von ONTAP Select Clustern zu bieten.

Architektur und klassische Einschränkungen

REST wurde formell von Roy Fielding in seinem PhD artikuliert "Dissertation" An der UC Irvine im Jahr 2000. Es definiert einen Architekturstil durch eine Reihe von Einschränkungen, die gemeinsam Web-basierte Anwendungen und die zugrunde liegenden Protokolle verbessert haben. Durch diese Einschränkungen wird eine RESTful Web Services-Applikation basierend auf einer Client-/Serverarchitektur mit einem statusfreien Kommunikationsprotokoll hergestellt.

Ressourcen- und Zustandsdarstellung

Ressourcen sind die Grundkomponenten eines webbasierten Systems. Beim Erstellen einer ANWENDUNG FÜR REST-Webservices umfassen die frühen Designaufgaben Folgendes:

- Identifizierung von System- oder serverbasierten Ressourcen
  Jedes System nutzt und verwaltet Ressourcen. Eine Ressource kann eine Datei-, Geschäftstransaktion-, Prozess- oder Verwaltungseinheit sein. Eine der ersten Aufgaben bei der Entwicklung einer auf REST-Webservices basierenden Applikation ist die Identifizierung der Ressourcen.
- Definition von Ressourcenstatus und zugehörigen Statusoperationen
  Die Ressourcen befinden sich immer in einer endlichen Anzahl von Staaten. Die Zustände sowie die damit verbundenen Operationen, die zur Auswirkung der Statusänderungen verwendet werden, müssen klar definiert werden.

Nachrichten werden zwischen dem Client und dem Server ausgetauscht, um auf den Zustand der Ressourcen gemäß dem generischen CRUD-Modell (Create, Read, Update, Delete) zuzugreifen und diesen zu ändern.

URI-Endpunkte

Jede REST-Ressource muss definiert und über ein gut definiertes Adressierungssystem verfügbar gemacht werden. Die Endpunkte, in denen die Ressourcen gefunden und identifiziert werden, verwenden einen einheitlichen Resource Identifier (URI). Der URI bietet ein allgemeines Framework zum Erstellen eines eindeutigen Namens für jede Ressource im Netzwerk. Der Uniform Resource Locator (URL) ist ein URI-Typ, der mit Webservices zur Identifizierung und zum Zugriff von Ressourcen verwendet wird. Ressourcen werden in der Regel in einer hierarchischen Struktur ausgesetzt, die einem Dateiverzeichnis ähnelt.

HTTP-Meldungen

Hypertext Transfer Protocol (HTTP) ist das Protokoll, das vom Webservice-Client und -Server zum Austausch von Anforderungs- und Antwortmeldungen zu den Ressourcen verwendet wird. Im Rahmen der Entwicklung einer Webservices-Anwendung werden HTTP-Verben (wie GET und POST) den Ressourcen und

entsprechenden Statusverwaltungsaktionen zugeordnet.

HTTP ist statusfrei. Um eine Reihe verwandter Anforderungen und Antworten innerhalb einer Transaktion zuzuordnen, müssen zusätzliche Informationen in die HTTP-Header enthalten sein, die mit den Request/Response-Datenströmen übertragen werden.

JSON-Formatierung

Während Informationen auf verschiedene Weise zwischen Client und Server strukturiert und übertragen werden können, ist die beliebteste Option (und die bei der REST-API implementieren verwendete Option) JavaScript Object Notation (JSON). JSON ist ein Branchenstandard für die Darstellung einfacher Datenstrukturen im Klartext und wird zur Übertragung von Zustandsdaten zur Beschreibung der Ressourcen verwendet.

So erhalten Sie Zugriff auf die Bereitstellungs-API

Aufgrund der inhärenten Flexibilität VON REST-Webservices ist der Zugriff auf die ONTAP Select Deploy API auf verschiedene Weise möglich.

Implementieren Sie die native Benutzeroberfläche von Utility

Der primäre Weg, um auf die API zuzugreifen, ist über die ONTAP Select Deploy Web-Benutzeroberfläche. Der Browser ruft die API auf und formatiert die Daten entsprechend dem Design der Benutzeroberfläche neu. Sie haben auch Zugriff auf die API über die Befehlszeilenschnittstelle von Deploy Utility.

ONTAP Select Seite zur Online-Dokumentation bereitstellen

Die Seite ONTAP Select Online-Dokumentation bereitstellen stellt bei Verwendung eines Browsers einen alternativen Zugriffspunkt dar. Die Seite bietet nicht nur die Möglichkeit, einzelne API-Aufrufe direkt auszuführen, sondern enthält auch eine detaillierte Beschreibung der API, einschließlich Eingabeparameter und anderer Optionen für jeden Aufruf. Die API-Aufrufe sind in verschiedene funktionale Bereiche oder Kategorien gegliedert.

Benutzerdefiniertes Programm

Die Bereitstellungs-API kann über eine von mehreren verschiedenen Programmiersprachen und Tools auf die Bereitstellungsschnittstelle zugegriffen werden. Beliebte Optionen sind Python, Java und Curl. Ein Programm, Skript oder Tool, das die API verwendet, fungiert als REST-Web-Services-Client. Mithilfe einer Programmiersprache lernen Sie die API besser kennen und erhalten die Möglichkeit, die ONTAP Select Implementierungen zu automatisieren.

Implementieren der API-Versionierung

Der REST API, die in ONTAP Select Deploy enthalten ist, wird eine Versionsnummer zugewiesen. Die API-Versionsnummer ist unabhängig von der Versionsnummer für die Bereitstellung. Sie sollten die API-Version kennen, die in Ihrer Version von Deploy enthalten ist, und wissen, welche Auswirkungen dies auf Ihre Verwendung der API hat.

Die aktuelle Version des Deploy Administration Utility enthält Version 3 DER REST API. Frühere Versionen des Deploy Utility umfassen die folgenden API-Versionen:

Implementierung 2.8 und höher

ONTAP Select Deploy 2.8. Alle neueren Versionen enthalten Version 3 der REST API.

Implementierung 2.7.2 und früher

ONTAP Select Deploy 2.7.2; alle älteren Versionen enthalten Version 2 DER REST API.

Die Versionen 2 und 3 DER REST-API sind nicht kompatibel. Wenn Sie nach einem früheren Release, das Version 2 der API enthält, ein Upgrade auf die Bereitstellung von 2.8 oder höher durchführen, müssen Sie jeden vorhandenen Code aktualisieren, der direkt auf die API zugreift, sowie alle Skripte über die Befehlszeilenschnittstelle.

Grundlegende betriebliche Eigenschaften

IM RUHEZUSTAND werden einheitliche Technologien und Best Practices erstellt, jedoch können die Details jeder API je nach dem verfügbaren Design variieren. Vor der Verwendung der API sollten Sie auf die Details und betrieblichen Merkmale der ONTAP Select Deploy-API achten.

Hypervisor-Host oder ONTAP Select-Node

Ein *Hypervisor-Host* ist die zentrale Hardware-Plattform, die eine ONTAP Select Virtual Machine hostet. Wenn eine ONTAP Select Virtual Machine auf einem Hypervisor-Host bereitgestellt und aktiv ist, gilt die Virtual Machine als „*ONTAP Select Node*“. Ab Version 3 der Deploy REST API sind Host- und Node-Objekte voneinander getrennt und unterscheiden sich. Dadurch wird eine 1:n-Beziehung möglich, bei der ein oder mehrere ONTAP Select Nodes auf demselben Hypervisor-Host ausgeführt werden können.

Objektkennungen

Jeder Ressourceninstanz oder jedem Objekt wird eine eindeutige Kennung zugewiesen, wenn sie erstellt wird. Diese Kennungen sind global eindeutig in einer bestimmten Instanz von ONTAP Select Deploy. Nachdem ein API-Aufruf ausgegeben wurde, der eine neue Objektinstanz erstellt, wird der zugeordnete id-Wert an den Anrufer im zurückgegeben `location` Kopfzeile der HTTP-Antwort. Sie können die Kennung extrahieren und bei nachfolgenden Anrufen verwenden, wenn Sie sich auf die Ressourceninstanz beziehen.

Der Inhalt und die interne Struktur der Objektkennungen können jederzeit geändert werden. Wenn Sie auf die zugeordneten Objekte verweisen, sollten Sie die Kennungen für die entsprechenden API-Aufrufe nur nach Bedarf verwenden.

Identifikatoren anfordern

Jeder erfolgreichen API-Anforderung wird eine eindeutige Kennung zugewiesen. Die Kennung wird im zurückgegeben `request-id` Kopfzeile der zugehörigen HTTP-Antwort. Sie können eine Anforderungskennung verwenden, um sich kollektiv auf die Aktivitäten einer einzelnen spezifischen API-Anforderungstransaktion zu beziehen. Sie können beispielsweise alle Ereignismeldungen einer Transaktion basierend auf der Anfrage-ID abrufen

Synchrone und asynchrone Anrufe

Es gibt zwei primäre Möglichkeiten, wie ein Server eine von einem Client empfangene HTTP-Anfrage durchführt:

- Synchron
  Der Server führt die Anforderung sofort aus und antwortet mit einem Statuscode von 200, 201 oder 204.
- Asynchron
  Der Server akzeptiert die Anfrage und antwortet mit dem Statuscode 202. Dies zeigt an, dass der Server die Clientanforderung angenommen hat und eine Hintergrundaufgabe gestartet hat, um die Anforderung abzuschließen. Der endgültige Erfolg oder Fehler ist nicht sofort verfügbar und muss durch zusätzliche API-Aufrufe ermittelt werden.

Bestätigen Sie den Abschluss eines lang laufenden Jobs

Im Allgemeinen wird jede Operation, die lange Zeit in Anspruch nehmen kann, asynchron mit einem verarbeitet Hintergrundaufgabe auf dem Server. Mit der Deploy REST API wird jede Hintergrundaufgabe durch ein verankert
Jobobjekt, das die Aufgabe verfolgt und Informationen bereitstellt, z. B. den aktuellen Status. Ein Job-Objekt, Einschließlich seiner eindeutigen Kennung wird in der HTTP-Antwort zurückgegeben, nachdem eine Hintergrundaufgabe erstellt wurde.

Sie können das Jobobjekt direkt abfragen, um den Erfolg oder den Fehler des zugeordneten API-Aufrufs zu ermitteln.
Weitere Informationen finden Sie unter „*Asynchronous Processing Using the Job Object*“.

Neben der Verwendung des Objekts Job gibt es weitere Möglichkeiten, wie Sie den Erfolg oder das Scheitern eines bestimmen können
Anforderung, einschließlich:

- Ereignismeldungen
  Sie können alle Ereignismeldungen, die einem bestimmten API-Aufruf zugeordnet sind, mithilfe der mit der ursprünglichen Antwort zurückgegebenen Anforderungs-id abrufen. Die Ereignismeldungen enthalten in der Regel Hinweise auf Erfolg oder Fehler und können auch nützlich sein, wenn ein Fehlerzustand behoben wird.
- Ressourcenstatus oder -Status
  Mehrere der Ressourcen behalten einen Status oder Statuswert bei, den Sie abfragen können, um indirekt den Erfolg oder das Fehlschlagen einer Anfrage zu bestimmen.

Sicherheit

Die Deploy-API nutzt die folgenden Sicherheitstechnologien:

- Sicherheit In Transportschicht
  Der gesamte Datenverkehr, der zwischen dem bereitzustellenden Server und dem Client über das Netzwerk gesendet wird, wird über TLS verschlüsselt. Die Verwendung des HTTP-Protokolls über einen unverschlüsselten Kanal wird nicht unterstützt. TLS-Version 1.2 wird unterstützt.
- HTTP-Authentifizierung
  Für jede API-Transaktion wird die Basisauthentifizierung verwendet. Jeder Anforderung wird ein HTTP-Header hinzugefügt, der den Benutzernamen und das Passwort in einem base64-String enthält.

API-Transaktion bei Anfrage und Reaktion

Jeder API-Aufruf zur Bereitstellung wird als HTTP-Anforderung an die virtuelle Maschine Bereitstellen ausgeführt, die eine entsprechende Antwort auf den Client erzeugt. Dieses Anforderungs-/Antwortpaar wird als API-Transaktion betrachtet. Bevor Sie die Deploy-API

verwenden, sollten Sie mit den zur Steuerung einer Anfrage verfügbaren Eingabevariablen und dem Inhalt der Antwortausgabe vertraut sein.

Eingabevariablen, die eine API-Anforderung steuern

Sie können steuern, wie ein API-Aufruf über in der HTTP-Anforderung festgelegte Parameter verarbeitet wird.

Anfragekopfzeilen

In der HTTP-Anfrage müssen mehrere Header enthalten sein, darunter:

- Content-Typ
  Wenn der Anforderungskörper JSON enthält, muss dieser Header auf Application/json gesetzt werden.
- Akzeptieren
  Wenn der Antworttext JSON enthält, muss dieser Header auf Application/json gesetzt werden.
- Autorisierung
  Die Basisauthentifizierung muss mit dem Benutzernamen und dem Passwort in einem base64-String eingerichtet werden.

Text anfordern

Der Inhalt der Anfraentext variiert je nach Anruf. Der HTTP-Request-Text besteht aus einem der folgenden Elemente:

- JSON-Objekt mit Eingabevariablen (z. B. der Name eines neuen Clusters)
- Leer

Objekte filtern

Wenn Sie einen API-Aufruf ausgeben, der GET verwendet, können Sie die zurückgegebenen Objekte anhand eines beliebigen Attributs einschränken oder filtern. Sie können beispielsweise einen genauen Wert angeben, der übereinstimmt:

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

Zusätzlich zu einer genauen Übereinstimmung stehen anderen Operatoren zur Verfügung, um einen Satz von Objekten über einen Wertebereich zurückzugeben. ONTAP Select unterstützt die unten aufgeführten Filteroperatoren.

Operator	Beschreibung
=	Gleich
<	Kleiner als
>	Größer als
≪=	Kleiner oder gleich
>=	Größer oder gleich
	Oder
!	Nicht gleich

Operator	Beschreibung
*	Gierige Wildcard

Sie können auch einen Satz von Objekten zurückgeben, basierend darauf, ob ein bestimmtes Feld gesetzt wird oder nicht, indem Sie das Null-Schlüsselwort oder dessen Negation (!null) als Teil der Abfrage verwenden.

Auswählen von Objektfeldern

Standardmäßig gibt die Ausgabe eines API-Aufrufs mithilfe VON GET nur die Attribute zurück, die das Objekt oder die Objekte eindeutig identifizieren. Dieser minimale Feldsatz dient als Schlüssel für jedes Objekt und variiert je nach Objekttyp. Sie können zusätzliche Objekteigenschaften mithilfe des Abfrageparameters Felder wie folgt auswählen:

- Günstige Felder
  Angeben `fields=*` Zum Abrufen der Objektfelder, die im lokalen Serverspeicher verwaltet werden oder für den Zugriff nur wenig verarbeitet werden müssen.
- Teure Felder
  Angeben `fields=**` Zum Abrufen aller Objektfelder, einschließlich solcher, die für den Zugriff zusätzliche Serververarbeitung erforderlich sind.
- Benutzerdefinierte Feldauswahl
  Nutzung `fields=FIELDNAME` Um das genaue Feld anzugeben, das Sie wünschen. Wenn Sie mehrere Felder anfordern, müssen die Werte durch Kommas ohne Leerzeichen getrennt werden.

Als Best Practice sollten Sie immer die gewünschten Felder identifizieren. Sie sollten nur die Reihe von kostengünstigen oder teuren Feldern abrufen, wenn Sie benötigen. Die kostengünstige und teure Klassifizierung wird durch NetApp auf der Grundlage interner Leistungsanalysen festgelegt. Die Klassifizierung für ein bestimmtes Feld kann sich jederzeit ändern.

Objekte im Ausgabesatz sortieren

Die Datensätze in einer Ressourcensammlung werden in der vom Objekt definierten Standardreihenfolge zurückgegeben. Sie können die Reihenfolge mit dem Abfrageparameter order_by mit dem Feldnamen und der Sortierrichtung wie folgt ändern:

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

Sie können beispielsweise das Typfeld in absteigender Reihenfolge, gefolgt von id in aufsteigender Reihenfolge sortieren:

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

Wenn Sie mehrere Parameter eingeben, müssen Sie die Felder mit einem Komma trennen.

Paginierung

Wenn Sie einen API-Aufruf über GET ausgeben, um auf eine Sammlung von Objekten desselben Typs zuzugreifen, werden alle übereinstimmenden Objekte standardmäßig zurückgegeben. Bei Bedarf können Sie die Anzahl der zurückgegebenen Datensätze mithilfe des Abfrageparameters max_Records mit der Anforderung begrenzen. Beispiel:

```
max_records=20
```

Bei Bedarf können Sie diesen Parameter mit anderen Abfrageparametern kombinieren, um den Ergebnissatz

einzugrenzen. Beispiel: Im Folgenden werden bis zu 10 Systemereignisse angezeigt, die nach der angegebenen Zeit generiert wurden:
`time⇒ 2019-04-04T15:41:29.140265Z&max_records=10`

Sie können mehrere Anfragen zur Seite über die Ereignisse (oder jeden Objekttyp) ausgeben. Jeder nachfolgende API-Aufruf sollte einen neuen Zeitwert verwenden, der auf dem letzten Ereignis des letzten Ergebnisset basiert.

Eine API-Antwort interpretieren

Jede API-Anfrage generiert eine Antwort an den Client. Sie können die Antwort prüfen, um festzustellen Ob die Daten erfolgreich waren und zusätzliche Daten nach Bedarf abgerufen werden konnten.

HTTP-Statuscode

Im Folgenden werden die von der REST-API „Bereitstellen“ verwendeten HTTP-Statuscodes beschrieben.

Codieren	Bedeutung	Beschreibung
200	OK	Zeigt Erfolg für Anrufe an, die kein neues Objekt erstellen.
201	Erstellt	Ein Objekt wurde erfolgreich erstellt. Der Header für die Standortantwort enthält die eindeutige Kennung für das Objekt.
202	Akzeptiert	Ein schon seit langem laufender Hintergrundjob wurde gestartet, um die Anforderung auszuführen, der Vorgang wurde jedoch noch nicht abgeschlossen.
400	Schlechte Anfrage	Die Eingabe der Anfrage ist nicht erkannt oder nicht angemessen.
403	Verboten	Der Zugriff wird aufgrund eines Autorisierungsfehlers verweigert.
404	Nicht gefunden	Die Ressource, auf die in diesem Antrag verwiesen wird, ist nicht vorhanden.
405	Methode nicht zulässig	Das HTTP-Verb in der Anforderung wird für die Ressource nicht unterstützt.
409	Konflikt	Der Versuch, ein Objekt zu erstellen, ist fehlgeschlagen, weil das Objekt bereits vorhanden ist.
500	Interner Fehler	Ein allgemeiner interner Fehler ist auf dem Server aufgetreten.
501	Nicht implementiert	Der URI ist bekannt, kann die Anforderung jedoch nicht ausführen.

Antwortkopfzeilen

In der vom Deploy-Server erzeugten HTTP-Antwort sind mehrere Header enthalten, darunter:

- Anforderungs-id
  Jeder erfolgreichen API-Anforderung wird eine eindeutige Anforderungskennung zugewiesen.
- Standort
  Wenn ein Objekt erstellt wird, enthält die Kopfzeile des Speicherorts die vollständige URL zum neuen Objekt einschließlich der eindeutigen Objektkennung.

Antwortkörper

Der Inhalt der mit einer API-Anfrage verknüpften Antwort ist je nach Objekt, Verarbeitungstyp und Erfolg oder Misserfolg der Anforderung unterschiedlich. Der Antwortkörper wird in JSON gerendert.

- Einzelnes Objekt
  Je nach Anforderung kann ein einzelnes Objekt mit einer Reihe von Feldern zurückgegeben werden. Beispielsweise können Sie GET verwenden, um ausgewählte Eigenschaften eines Clusters mit der eindeutigen Kennung abzurufen.
- Mehrere Objekte
  Es können mehrere Objekte aus einer Ressourcensammlung zurückgegeben werden. In allen Fällen wird ein konsistentes Format verwendet, mit `num_records` Angabe der Anzahl der Datensätze und Datensätze, die ein Array der Objektinstanzen enthalten. Beispielsweise können Sie alle in einem bestimmten Cluster definierten Nodes abrufen.
- Jobobjekt
  Wenn ein API-Aufruf asynchron verarbeitet wird, wird ein Job-Objekt zurückgegeben, das den Hintergrund-Task ankers. Beispielsweise wird DIE POST-Anforderung, die zum Bereitstellen eines Clusters verwendet wird, asynchron bearbeitet und ein Job-Objekt zurückgegeben.
- Fehlerobjekt
  Wenn ein Fehler auftritt, wird immer ein Fehlerobjekt zurückgegeben. Beispielsweise erhalten Sie einen Fehler beim Versuch, ein Cluster mit einem bereits vorhandenen Namen zu erstellen.
- Leer
  In bestimmten Fällen werden keine Daten zurückgegeben und der Antworttext ist leer. Beispielsweise ist der Antwortkörper leer, nachdem Sie ZUM Löschen eines vorhandenen Hosts AUF „LÖSCHEN“ setzen.

Asynchrone Verarbeitung mit dem Job-Objekt

Einige der API-Aufrufe für die Bereitstellung, insbesondere solche, die eine Ressource erstellen oder ändern, können länger dauern als andere Anrufe. ONTAP Select implementieren verarbeitet diese langen Anforderungen asynchron.

Asynchrone Anforderungen, die mit Job Object beschrieben werden

Nach einem API-Aufruf, der asynchron ausgeführt wird, weist der HTTP-Antwortcode 202 darauf hin, dass die Anforderung erfolgreich validiert und akzeptiert, aber noch nicht abgeschlossen wurde. Die Anforderung wird als Hintergrundaufgabe verarbeitet, die nach der ersten HTTP-Antwort auf den Client weiter ausgeführt wird. Die Antwort umfasst das Job-Objekt, das die Anfrage einschließlich der eindeutigen Kennung anverankert.

Mithilfe der Seite ONTAP Select Deploy Online-Dokumentation können Sie ermitteln, welche API-Aufrufe asynchron funktionieren.

Abfrage des Job-Objekts, das einer API-Anforderung zugeordnet ist

Das in der HTTP-Antwort zurückgegebene Job-Objekt enthält mehrere Eigenschaften. Sie können die Statuseigenschaft abfragen, um festzustellen, ob die Anfrage erfolgreich abgeschlossen wurde. Ein Job-Objekt kann einen der folgenden Status haben:

- Warteschlange
- Wird Ausgeführt

- Erfolg
- Ausfall

Es gibt zwei Verfahren, die Sie beim Abfragen eines Jobobjekts verwenden können, um einen Terminalstatus für die Aufgabe zu erkennen: Erfolg oder Fehler:

- Standard-Polling-Anfrage
  Der aktuelle Jobstatus wird sofort zurückgegeben
- Lange Abfrageanfrage
  Der Jobstatus wird nur zurückgegeben, wenn einer der folgenden Fehler auftritt:
  - Status hat sich vor kurzem geändert als der Datumswert, der auf der Abfrage angegeben wurde
  - Timeout-Wert abgelaufen (1 bis 120 Sekunden)

Standardabfrage und langes Abfragen verwenden denselben API-Aufruf, um ein Auftragsobjekt abzufragen. Eine lange Abfrageanforderung umfasst jedoch zwei Abfrageparameter: `poll_timeout` Und `last_modified`.

Sie sollten immer Long Polling verwenden, um die Arbeitslast auf der virtuellen Maschine bereitstellen zu reduzieren.

Allgemeines Verfahren für die Ausgabe einer asynchronen Anfrage

Sie können den folgenden grundlegenden Vorgang verwenden, um einen asynchronen API-Aufruf abzuschließen:

1. Geben Sie den asynchronen API-Aufruf aus.
2. Sie erhalten eine HTTP-Antwort 202, die darauf hinweist, dass die Anfrage erfolgreich angenommen wurde.
3. Extrahieren Sie die Kennung für das Job-Objekt aus dem Antwortkörper.
4. Führen Sie in einer Schleife in jedem Zyklus die folgenden Schritte aus:
   a. Den aktuellen Status des Jobs mit einer langen Umfrage abrufen
   b. Wenn sich der Job in einem nicht-Terminal-Status befindet (Warteschlange, wird ausgeführt), führen Sie die Schleife erneut aus.
5. Beenden Sie, wenn der Job einen Terminalstatus erreicht (Erfolg, Fehler).