



REST-API-Zugriff und Authentifizierung in Active IQ Unified Manager

Active IQ Unified Manager 9.9

NetApp
April 05, 2024

Inhalt

- REST-API-Zugriff und Authentifizierung in Active IQ Unified Manager 1
 - REST-Zugriff..... 1
 - Authentifizierung 3
 - In Active IQ Unified Manager verwendete HTTP-Statuscodes 3
 - Empfehlungen für die Verwendung der APIs für Active IQ Unified Manager 5
 - Protokolle für die Fehlerbehebung 5
 - Auftragsobjekte asynchrone Prozesse..... 6
 - Hallo API Server..... 7

REST-API-Zugriff und Authentifizierung in Active IQ Unified Manager

Auf die Active IQ Unified Manager REST API kann über jeden Webbrowser oder eine Programmierplattform zugegriffen werden, um HTTP-Anfragen zu stellen. Unified Manager unterstützt den grundlegenden HTTP-Authentifizierungsmechanismus. Bevor Sie die UNIFIED Manager REST API aufrufen, müssen Sie einen Benutzer authentifizieren.

REST-Zugriff

Sie können jeden Webbrowser oder jede Programmierplattform verwenden, die HTTP-Anfragen zum Zugriff auf die Unified Manager REST-API ausgeben. Nach der Anmeldung bei Unified Manager können Sie beispielsweise die URL in einen beliebigen Browser eingeben, um die Attribute aller Management-Stationen, wie beispielsweise der Name der Management Station, die Taste und die IP-Adresse, abzurufen.

- **Anfrage**

GET https://<IP address/hostname>:<port_number>/api/v2/datacenter/cluster/clusters

- **Antwort**

```
{
  "records": [
    {
      "key": "4c6bf721-2e3f-11e9-a3e2-00a0985badbb:type=cluster,uuid=4c6bf721-2e3f-11e9-a3e2-00a0985badbb",
      "name": "fas8040-206-21",
      "uuid": "4c6bf721-2e3f-11e9-a3e2-00a0985badbb",
      "contact": null,
      "location": null,
      "version": {
        "full": "NetApp Release Dayblazer__9.5.0: Thu Jan 17 10:28:33 UTC 2019",
        "generation": 9,
        "major": 5,
        "minor": 0
      },
      "isSanOptimized": false,
      "management_ip": "10.226.207.25",
      "nodes": [
        {
          "key": "4c6bf721-2e3f-11e9-a3e2-00a0985badbb:type=cluster_node,uuid=12cf06cc-2e3a-11e9-b9b4-00a0985badbb",
          "uuid": "12cf06cc-2e3a-11e9-b9b4-00a0985badbb",
```

```

    "name": "fas8040-206-21-01",
    "_links": {
      "self": {
        "href": "/api/datacenter/cluster/nodes/4c6bf721-2e3f-11e9-
a3e2-00a0985badbb:type=cluster_node,uuid=12cf06cc-2e3a-11e9-b9b4-
00a0985badbb"
      }
    },
    "location": null,
    "version": {
      "full": "NetApp Release Dayblazer__9.5.0: Thu Jan 17 10:28:33
UTC 2019",
      "generation": 9,
      "major": 5,
      "minor": 0
    },
    "model": "FAS8040",
    "uptime": 13924095,
    "serial_number": "701424000157"
  },
  {
    "key": "4c6bf721-2e3f-11e9-a3e2-
00a0985badbb:type=cluster_node,uuid=1ed606ed-2e3a-11e9-a270-00a0985bb9b7",
    "uuid": "1ed606ed-2e3a-11e9-a270-00a0985bb9b7",
    "name": "fas8040-206-21-02",
    "_links": {
      "self": {
        "href": "/api/datacenter/cluster/nodes/4c6bf721-2e3f-11e9-
a3e2-00a0985badbb:type=cluster_node,uuid=1ed606ed-2e3a-11e9-a270-
00a0985bb9b7"
      }
    },
    "location": null,
    "version": {
      "full": "NetApp Release Dayblazer__9.5.0: Thu Jan 17 10:28:33
UTC 2019",
      "generation": 9,
      "major": 5,
      "minor": 0
    },
    "model": "FAS8040",
    "uptime": 14012386,
    "serial_number": "701424000564"
  }
],
"_links": {

```

```

    "self": {
      "href": "/api/datacenter/cluster/clusters/4c6bf721-2e3f-11e9-a3e2-00a0985badbb?type=cluster,uuid=4c6bf721-2e3f-11e9-a3e2-00a0985badbb"
    }
  }
},

```

- IP address/hostname Ist die IP-Adresse oder der vollqualifizierte Domain-Name (FQDN) des API-Servers.
- Port 443

Der Standard-HTTPS-Port 443 ist. Sie können den HTTPS-Port bei Bedarf anpassen.

Um HTTP-Anfragen aus einem Webbrowser zu veröffentlichen, ZU PATCHEN und ZU LÖSCHEN, müssen Sie Browser-Plugins verwenden. Sie können auch über Skripting-Plattformen wie Curl und Perl auf DIE REST-API zugreifen.

Authentifizierung

Unified Manager unterstützt das grundlegende HTTP-Authentifizierungsschema für APIs. Für einen sicheren Informationsfluss (Anfrage und Antwort) sind die REST-APIs nur über HTTPS zugänglich. Der API-Server stellt allen Clients ein selbstsigniertes SSL-Zertifikat zur Server-Überprüfung zur Verfügung. Dieses Zertifikat kann durch ein benutzerdefiniertes Zertifikat (oder ein CA-Zertifikat) ersetzt werden.

Sie müssen den Benutzerzugriff auf den API-Server konfigurieren, um die REST-APIs zu aufrufen. Die Benutzer können lokale Benutzer (Benutzerprofile, die in der lokalen Datenbank gespeichert sind) oder LDAP-Benutzer (wenn Sie den API-Server für die Authentifizierung über LDAP konfiguriert haben) sein. Sie können den Benutzerzugriff verwalten, indem Sie sich an der Benutzeroberfläche der Unified Manager Administration Console anmelden.

In Active IQ Unified Manager verwendete HTTP-Statuscodes

Bei Ausführung der APIs oder bei der Fehlerbehebung sollten Sie die verschiedenen HTTP-Statuscodes und -Fehlercodes kennen, die von Active IQ Unified Manager-APIs verwendet werden.

In der folgenden Tabelle sind die Fehlercodes für die Authentifizierung aufgeführt:

HTTP-Statuscode	Titel des Statuscodes	Beschreibung
200	OK	Wird bei der erfolgreichen Ausführung von synchronen API-Aufrufen zurückgegeben.
201	Erstellt	Erstellung neuer Ressourcen durch synchrone Anrufe, wie z. B. Konfiguration von Active Directory.

HTTP-Statuscode	Titel des Statuscodes	Beschreibung
202	Akzeptiert	Wird bei der erfolgreichen Ausführung von asynchronen Aufrufen für Bereitstellungsfunktionen zurückgegeben, z. B. Erstellen von LUNs und File Shares.
400	Ungültige Anforderung	Zeigt Fehler bei der Eingabevalidierung an. Der Benutzer muss die Eingaben korrigieren, z. B. gültige Schlüssel in einem Anforderungskörper.
401	Nicht autorisierte Anforderung	Sie sind nicht berechtigt, die Ressource/Unbefugte anzuzeigen.
403	Anfrage verweigert	Der Zugriff auf die Ressource, die Sie erreichen wollten, ist verboten.
404	Ressource nicht gefunden	Die Ressource, die Sie erreichen wollten, wurde nicht gefunden.
405	Methode Nicht Zulässig	Methode nicht zulässig.
429	Zu Viele Anfragen	Dieser Wert wird zurückgegeben, wenn der Benutzer zu viele Anfragen innerhalb eines bestimmten Zeitraums sendet.
500	Interner Serverfehler	Interner Serverfehler. Fehler beim Abrufen der Antwort vom Server. Dieser interne Serverfehler ist möglicherweise permanent oder nicht permanent. Beispiel: Wenn Sie ein ausführen GET Oder GET ALL Vorgang und Empfang dieses Fehlers. Es wird empfohlen, diesen Vorgang für mindestens fünf Wiederholungen zu wiederholen. Wenn es sich um einen permanenten Fehler handelt, ist der zurückgegebene Statuscode weiterhin 500. Wenn der Vorgang erfolgreich ist, wird der zurückgegebene Statuscode 200 zurückgegeben.

Empfehlungen für die Verwendung der APIs für Active IQ Unified Manager

Bei Verwendung der APIs in Active IQ Unified Manager sollten Sie bestimmte empfohlene Methoden befolgen.

- Alle Arten von Antwortinhalten müssen für eine gültige Ausführung das folgende Format aufweisen:

```
application/json
```

- Die API-Versionsnummer steht nicht zur Produktversionsnummer. Sie sollten die neueste Version der für Ihre Unified Manager Instanz verfügbaren API verwenden. Weitere Informationen zu den Unified Manager API-Versionen finden Sie im Abschnitt „reST API Versionierung in Active IQ Unified Manager“.
- Beim Aktualisieren der Array-Werte mithilfe einer Unified Manager API müssen Sie die gesamte Zeichenfolge von Werten aktualisieren. Sie können einem Array keine Werte anhängen. Sie können nur ein vorhandenes Array ersetzen.
- Filteroperatoren wie Pipe (,) und Wild Card () **können für alle Abfrageparameter verwendet werden, mit Ausnahme von doppelten Werten, z. B. IOPS und Performance in den Kennzahlen-APIs. Vermeiden Sie das Abfragen von Objekten, indem Sie eine Kombination aus Wildcard () und Rohr (,) des Filterbedieners verwenden.** Es kann eine falsche Anzahl von Objekten abrufen.
- Beachten Sie, dass die GET (Alle) die Anforderung für eine beliebige API gibt maximal 1000 Datensätze zurück. Auch wenn Sie die Abfrage ausführen, indem Sie die einstellen max_records Parameter auf einen Wert über 1000, nur 1000 Datensätze werden zurückgegeben.
- Für administrative Aufgaben wird empfohlen, die Unified Manager-Benutzeroberfläche zu verwenden.

Protokolle für die Fehlerbehebung

Mithilfe von Systemprotokollen können Sie die Ursachen eines Ausfalls und die Behebung von Problemen analysieren, die bei der Ausführung der APIs auftreten können.

Rufen Sie die Protokolle vom folgenden Speicherort ab, um Probleme im Zusammenhang mit den API-Aufrufen zu beheben.

Speicherort protokollieren	Nutzung
/var/log/ocie/access_log.log	<p>Enthält alle API-Anrufrdetails, z. B. den Benutzernamen des Benutzers, der die API aufruft, Startzeit, Ausführungszeit, Status und URL.</p> <p>In dieser Protokolldatei können Sie die häufig verwendeten APIs überprüfen oder einen Fehler in jedem GUI-Workflow beheben. Sie können die Analyse auch anhand der Ausführungszeit skalieren.</p>

Speicherort protokollieren	Nutzung
/var/log/ocum/ocumserver.log	Enthält alle API-Ausführungsprotokolle. Sie können diese Protokolldatei zur Fehlerbehebung und Fehlersuche bei API-Aufrufen verwenden.
/var/log/ocie/server.log	Enthält alle WildFly-Server-Bereitstellungen und Start/Stop-Service-bezogene Protokolle. Sie können diese Protokolldatei verwenden, um die Ursache von Problemen zu finden, die während des Starts, Stoppens oder der Bereitstellung des WildFly-Servers auftreten.
/var/log/ocie/au.log	Enthält Protokolle für die Erfassungseinheit. Sie können diese Protokolldatei verwenden, wenn Sie Objekte in ONTAP erstellt, geändert oder gelöscht haben, sie sich jedoch nicht für die Active IQ Unified Manager REST-APIs widerspiegeln.

Auftragsobjekte asynchrone Prozesse

Active IQ Unified Manager stellt den bereit `jobs API`, die Informationen über die während der Ausführung anderer APIs durchgeführten Jobs abrufen. Sie müssen wissen, wie die asynchrone Verarbeitung mit dem Job-Objekt funktioniert.

Einige der API-Aufrufe, insbesondere solche, die zum Hinzufügen oder Ändern von Ressourcen verwendet werden, können länger dauern als andere Anrufe. Unified Manager verarbeitet diese langfristigen 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.

Abfragen des mit einer API-Anforderung verknüpften Jobobjekts

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

- NORMAL
- WARNING
- PARTIAL_FAILURES
- ERROR

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

- Standard-Abfrage: Der aktuelle Job-Status wird sofort zurückgegeben.
- Lange Abfrage: Wenn der Job-Status zu `bewegt NORMAL, ERROR, Oder PARTIAL_FAILURES`.

Schritte in einer asynchronen Anforderung

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. Warten Sie in einer Schleife, bis das Objekt Job den Terminalstatus erreicht hat `NORMAL, ERROR, Oder PARTIAL_FAILURES`.
5. Überprüfen Sie den Terminalstatus des Jobs, und rufen Sie das Jobergebnis ab.

Hallo API Server

Der *Hello API-Server* ist ein Beispielprogramm, das zeigt, wie eine REST-API in Active IQ Unified Manager mit einem einfachen REST-Client aufgerufen wird. Das Beispielprogramm enthält grundlegende Details zum API-Server im JSON-Format (nur der Server unterstützt `application/json` Format).

Der verwendete URI ist: <https://<hostname>/api/datacenter/svm/svms>. Dieser Beispielcode nimmt die folgenden Eingabeparameter auf:

- Die IP-Adresse oder FQDN des API-Servers
- Optional: Portnummer (Standard: 443)
- Benutzername
- Passwort
- Antwortformat (`application/json`)

Um REST-APIs aufzurufen, können Sie auch andere Skripte wie Jersey und RESTEasy verwenden, um einen Java REST-Client für Active IQ Unified Manager zu schreiben. Beachten Sie die folgenden Überlegungen zum Beispielcode:

- Verwendet eine HTTPS-Verbindung zu Active IQ Unified Manager, um den angegebenen REST-URI aufzurufen
- Ignoriert das von Active IQ Unified Manager bereitgestellte Zertifikat
- Überspringt die Überprüfung des Host-Namens während des Handshakes
- Verwendet `javax.net.ssl.HttpsURLConnection` Für eine URI-Verbindung
- Verwendet eine Bibliothek eines Drittanbieters (`org.apache.commons.codec.binary.Base64`) Für die Erstellung der Base64 kodierten Zeichenfolge in der HTTP-Grundauthentifizierung verwendet

Um den Beispielcode kompilieren und ausführen zu können, müssen Sie Java Compiler 1.8 oder höher verwenden.

```
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.net.URL;
import java.security.SecureRandom;
import java.security.cert.X509Certificate;
import javax.net.ssl.HostnameVerifier;
import javax.net.ssl.HttpURLConnection;
import javax.net.ssl.SSLContext;
import javax.net.ssl.SSLSession;
import javax.net.ssl.TrustManager;
import javax.net.ssl.X509TrustManager;
import org.apache.commons.codec.binary.Base64;

public class HelloApiServer {

    private static String server;
    private static String user;
    private static String password;
    private static String response_format = "json";
    private static String server_url;
    private static String port = null;

    /*
     * * The main method which takes user inputs and performs the *
    necessary steps
     * to invoke the REST URI and show the response
    */ public static void main(String[] args) {
        if (args.length < 2 || args.length > 3) {
            printUsage();
            System.exit(1);
        }
        setUserArguments(args);
        String serverBaseUrl = "https://" + server;
        if (null != port) {
            serverBaseUrl = serverBaseUrl + ":" + port;
        }
        server_url = serverBaseUrl + "/api/datacenter/svm/svms";
        try {
            HttpURLConnection connection =
getAllTrustingHttpsURLConnection();
            if (connection == null) {
                System.err.println("FATAL: Failed to create HTTPS
connection to URL: " + server_url);
            }
        }
    }
}
```

```

        System.exit(1);
    }
    System.out.println("Invoking API: " + server_url);
    connection.setRequestMethod("GET");
    connection.setRequestProperty("Accept", "application/" +
response_format);
    String authString = getAuthorizationString();
    connection.setRequestProperty("Authorization", "Basic " +
authString);
    if (connection.getResponseCode() != 200) {
        System.err.println("API Invocation Failed : HTTP error
code : " + connection.getResponseCode() + " : "
+ connection.getResponseMessage());
        System.exit(1);
    }
    BufferedReader br = new BufferedReader(new
InputStreamReader((connection.getInputStream())));
    String response;
    System.out.println("Response:");
    while ((response = br.readLine()) != null) {
        System.out.println(response);
    }
    connection.disconnect();
} catch (Exception e) {
    e.printStackTrace();
}
}

/* Print the usage of this sample code */ private static void
printUsage() {
    System.out.println("\nUsage:\n\tHelloApiServer <hostname> <user>
<password>\n");
    System.out.println("\nExamples:\n\tHelloApiServer localhost admin
mypassword");
    System.out.println("\tHelloApiServer 10.22.12.34:8320 admin
password");
    System.out.println("\tHelloApiServer 10.22.12.34 admin password
");
    System.out.println("\tHelloApiServer 10.22.12.34:8212 admin
password \n");
    System.out.println("\nNote:\n\t(1) When port number is not
provided, 443 is chosen by default.");
}

/* * Set the server, port, username and password * based on user
inputs. */ private static void setUserArguments(

```

```

        String[] args) {
            server = args[0];
            user = args[1];
            password = args[2];
            if (server.contains(":")) {
                String[] parts = server.split(":");
                server = parts[0];
                port = parts[1];
            }
        }

    /*
     * * Create a trust manager which accepts all certificates and * use
    this trust
     * manager to initialize the SSL Context. * Create a
    HttpURLConnection for this
     * SSL Context and skip * server hostname verification during SSL
    handshake. * *
     * Note: Trusting all certificates or skipping hostname verification *
    is not
     * required for API Services to work. These are done here to * keep
    this sample
     * REST Client code as simple as possible.
    */ private static HttpURLConnection
    getAllTrustingHttpsURLConnection() {
        HttpURLConnection conn =
        null;
        try {
            /* Creating a trust manager that does not
            validate certificate chains */
            TrustManager[]
            trustAllCertificatesManager = new
            TrustManager[]{new
            X509TrustManager(){
                public X509Certificate[] getAcceptedIssuers(){return null;}
                public void checkClientTrusted(X509Certificate[]
                certs, String authType){}
                public void checkServerTrusted(X509Certificate[]
                certs, String authType){}
            }};
            /* Initialize the
            SSLContext with the all-trusting trust manager */
            SSLContext sslContext = SSLContext.getInstance("TLS");
            sslContext.init(null, trustAllCertificatesManager, new
            SecureRandom());
            HttpURLConnection.setDefaultSSLSocketFactory(sslContext.getSocketFactory(
            ));
            URL url = new URL(server_url);
            conn =
            (HttpURLConnection) url.openConnection();
            /* Do not perform an
            actual hostname verification during SSL Handshake.
            Let all
            hostname pass through as verified.*/
            conn.setHostnameVerifier(new HostnameVerifier() {
                public
                boolean verify(String host, SSLSession
                session) {
                return true;
            }
            });
        } catch (Exception e)
    }

```

```

{
    e.printStackTrace();
}
return conn;
}

/*
 * * This forms the Base64 encoded string using the username and
password *
 * provided by the user. This is required for HTTP Basic
Authentication.
 */
private static String getAuthorizationString() {
    String userPassword = user + ":" + password;
    byte[] authEncodedBytes =
Base64.encodeBase64(userPassword.getBytes());
    String authString = new String(authEncodedBytes);
    return authString;
}
}

```

Copyright-Informationen

Copyright © 2024 NetApp. Alle Rechte vorbehalten. Gedruckt in den USA. Dieses urheberrechtlich geschützte Dokument darf ohne die vorherige schriftliche Genehmigung des Urheberrechtsinhabers in keiner Form und durch keine Mittel – weder grafische noch elektronische oder mechanische, einschließlich Fotokopieren, Aufnehmen oder Speichern in einem elektronischen Abrufsystem – auch nicht in Teilen, vervielfältigt werden.

Software, die von urheberrechtlich geschütztem NetApp Material abgeleitet wird, unterliegt der folgenden Lizenz und dem folgenden Haftungsausschluss:

DIE VORLIEGENDE SOFTWARE WIRD IN DER VORLIEGENDEN FORM VON NETAPP ZUR VERFÜGUNG GESTELLT, D. H. OHNE JEGLICHE EXPLIZITE ODER IMPLIZITE GEWÄHRLEISTUNG, EINSCHLIESSLICH, JEDOCH NICHT BESCHRÄNKT AUF DIE STILLSCHWEIGENDE GEWÄHRLEISTUNG DER MARKTGÄNGIGKEIT UND EIGNUNG FÜR EINEN BESTIMMTEN ZWECK, DIE HIERMIT AUSGESCHLOSSEN WERDEN. NETAPP ÜBERNIMMT KEINERLEI HAFTUNG FÜR DIREKTE, INDIREKTE, ZUFÄLLIGE, BESONDERE, BEISPIELHAFTE SCHÄDEN ODER FOLGESCHÄDEN (EINSCHLIESSLICH, JEDOCH NICHT BESCHRÄNKT AUF DIE BESCHAFFUNG VON ERSATZWAREN ODER -DIENSTLEISTUNGEN, NUTZUNGS-, DATEN- ODER GEWINNVERLUSTE ODER UNTERBRECHUNG DES GESCHÄFTSBETRIEBS), UNABHÄNGIG DAVON, WIE SIE VERURSACHT WURDEN UND AUF WELCHER HAFTUNGSTHEORIE SIE BERUHEN, OB AUS VERTRAGLICH FESTGELEGTER HAFTUNG, VERSCHULDENSUNABHÄNGIGER HAFTUNG ODER DELIKTSHAFTUNG (EINSCHLIESSLICH FAHRLÄSSIGKEIT ODER AUF ANDEREM WEGE), DIE IN IRGEND EINER WEISE AUS DER NUTZUNG DIESER SOFTWARE RESULTIEREN, SELBST WENN AUF DIE MÖGLICHKEIT DERARTIGER SCHÄDEN HINGEWIESEN WURDE.

NetApp behält sich das Recht vor, die hierin beschriebenen Produkte jederzeit und ohne Vorankündigung zu ändern. NetApp übernimmt keine Verantwortung oder Haftung, die sich aus der Verwendung der hier beschriebenen Produkte ergibt, es sei denn, NetApp hat dem ausdrücklich in schriftlicher Form zugestimmt. Die Verwendung oder der Erwerb dieses Produkts stellt keine Lizenzierung im Rahmen eines Patentrechts, Markenrechts oder eines anderen Rechts an geistigem Eigentum von NetApp dar.

Das in diesem Dokument beschriebene Produkt kann durch ein oder mehrere US-amerikanische Patente, ausländische Patente oder anhängige Patentanmeldungen geschützt sein.

ERLÄUTERUNG ZU „RESTRICTED RIGHTS“: Nutzung, Vervielfältigung oder Offenlegung durch die US-Regierung unterliegt den Einschränkungen gemäß Unterabschnitt (b)(3) der Klausel „Rights in Technical Data – Noncommercial Items“ in DFARS 252.227-7013 (Februar 2014) und FAR 52.227-19 (Dezember 2007).

Die hierin enthaltenen Daten beziehen sich auf ein kommerzielles Produkt und/oder einen kommerziellen Service (wie in FAR 2.101 definiert) und sind Eigentum von NetApp, Inc. Alle technischen Daten und die Computersoftware von NetApp, die unter diesem Vertrag bereitgestellt werden, sind gewerblicher Natur und wurden ausschließlich unter Verwendung privater Mittel entwickelt. Die US-Regierung besitzt eine nicht ausschließliche, nicht übertragbare, nicht unterlizenzierbare, weltweite, limitierte unwiderrufliche Lizenz zur Nutzung der Daten nur in Verbindung mit und zur Unterstützung des Vertrags der US-Regierung, unter dem die Daten bereitgestellt wurden. Sofern in den vorliegenden Bedingungen nicht anders angegeben, dürfen die Daten ohne vorherige schriftliche Genehmigung von NetApp, Inc. nicht verwendet, offengelegt, vervielfältigt, geändert, aufgeführt oder angezeigt werden. Die Lizenzrechte der US-Regierung für das US-Verteidigungsministerium sind auf die in DFARS-Klausel 252.227-7015(b) (Februar 2014) genannten Rechte beschränkt.

Markeninformationen

NETAPP, das NETAPP Logo und die unter <http://www.netapp.com/TM> aufgeführten Marken sind Marken von NetApp, Inc. Andere Firmen und Produktnamen können Marken der jeweiligen Eigentümer sein.