



Stellen Sie das BeeGFS-Dateisystem bereit

BeeGFS on NetApp with E-Series Storage

NetApp
August 23, 2024

Inhalt

- Stellen Sie das BeeGFS-Dateisystem bereit 1
 - Ansible – Playbook-Überblick 1
 - Implementieren Sie das BeeGFS HA-Cluster 2
 - Bereitstellen von BeeGFS-Clients 6
 - Überprüfen Sie die BeeGFS-Bereitstellung 11

Stellen Sie das BeeGFS-Dateisystem bereit

Ansible – Playbook-Überblick

BeeGFS HA-Cluster implementieren und managen mithilfe von Ansible

Überblick

In den vorherigen Abschnitten wurden die Schritte aufgeführt, mit denen ein Ansible-Inventar erstellt werden konnte, der ein BeeGFS HA-Cluster darstellt. In diesem Abschnitt wird die von NetApp entwickelte Ansible-Automatisierung für die Implementierung und das Management des Clusters vorgestellt.

Ansible – Wichtige Konzepte

Bevor Sie fortfahren, ist es hilfreich, sich mit ein paar Schlüsselkonzepten von Ansible vertraut zu machen:

- Aufgaben, die für einen Ansible-Bestand ausgeführt werden müssen, werden in einem sogenannten **Playbook** definiert.
 - Die meisten Aufgaben in Ansible sind **idempotent**, d. h., sie können mehrmals ausgeführt werden, um zu überprüfen, ob die gewünschte Konfiguration/der gewünschte Zustand noch angewendet wird, ohne dass Dinge zu stören oder unnötige Updates zu machen.
- Die kleinste Ausführungseinheit in Ansible ist ein **Modul**.
 - Typische Playbooks nutzen mehrere Module.
 - Beispiele: Laden Sie ein Paket herunter, aktualisieren Sie eine Konfigurationsdatei, starten/aktivieren Sie einen Dienst.
 - NetApp verteilt Module zur Automatisierung von NetApp E-Series Systemen.
- Komplexe Automatisierungsoptionen sind besser als Rollen integriert.
 - Im Wesentlichen ein Standardformat zur Verteilung eines wiederverwendbaren Playbooks.
 - NetApp verteilt Rollen für Linux-Hosts und BeeGFS-Filesysteme.

BeeGFS HA-Rolle für Ansible: Schlüsselkonzepte

Die gesamte Automatisierung, die für das Implementieren und Managen jeder Version von BeeGFS auf NetApp erforderlich ist, ist als Ansible-Rolle verpackt und im Rahmen der verteilt "[NetApp E-Series Ansible Collection für BeeGFS](#)":

- Diese Rolle kann als irgendwo zwischen einem **Installer** und einer modernen **Deployment/Management Engine** für BeeGFS gedacht werden.
 - Nutzt moderne Infrastruktur als Code-Praktiken und -Philosophien um das Management der Storage-Infrastruktur in jeder Größenordnung zu vereinfachen
 - Das "[Kubesbete](#)"-Projekt ermöglicht Benutzern die Implementierung und Wartung einer gesamten Kubernetes-Distribution für die Scale-out-Computing-Infrastruktur.
- Dabei handelt es sich um das **softwaredefinierte**-Format von NetApp zur Verpackung, Verteilung und Wartung von BeeGFS auf NetApp Lösungen.
 - Versuchen Sie, eine „Appliance-ähnliche“ Erfahrung zu schaffen, ohne eine gesamte Linux-Distribution oder ein großes Bild zu verteilen.

- Dazu gehören die von NetApp entwickelten Open Cluster Framework (OCF)-konformen Cluster-Ressourcen-Agents für benutzerdefinierte BeeGFS-Ziele, IP-Adressen und Monitoring für intelligente Pacemaker/BeeGFS-Integration.
- Diese Rolle spielt nicht einfach die Implementierung der „Automatisierung“ und soll den gesamten Lebenszyklus des Filesystems managen, darunter:
 - Konfigurationsänderungen und Updates pro Service oder Cluster-weite Konfiguration
 - Automatisierung von Cluster-Reparatur und Recovery nach Behebung von Hardware-Problemen
 - Vereinfachte Performance-Optimierung mit Standardeinstellungen, die auf umfangreichen Tests mit BeeGFS und NetApp Volumes basieren
 - Überprüfung und Korrektur von Konfigurationstendenzen.

NetApp bietet auch eine Ansible-Rolle für ["BeeGFS-Clients"](#), Die optional verwendet werden kann, um BeeGFS zu installieren und mounten Sie Dateisysteme auf Compute/GPU/Login-Nodes.

Implementieren Sie das BeeGFS HA-Cluster

Geben Sie an, welche Aufgaben ausgeführt werden sollen, um BeeGFS HA-Cluster mithilfe eines Playbooks zu implementieren.

Überblick

In diesem Abschnitt wird beschrieben, wie Sie das Standard-Playbook zur Bereitstellung/zum Managen von BeeGFS auf NetApp zusammenstellen können.

Schritte

Erstellen Sie das Ansible Playbook

Erstellen Sie die Datei `playbook.yml` Und füllen Sie es wie folgt aus:

1. Definieren Sie zunächst einen Satz von Aufgaben (allgemein als A bezeichnet) ["Spielen"](#)) Die nur auf Block-Nodes der NetApp E-Series ausgeführt werden sollte. Wir verwenden eine Pause-Aufgabe, um vor dem Ausführen der Installation eine Aufforderung zu geben (um versehentliche Playbook-Läufe zu vermeiden), und importieren dann die `nar_santricity_management` Rolle: Diese Rolle übernimmt die Anwendung aller in definierten allgemeinen Systemkonfiguration `group_vars/eseries_storage_systems.yml` Oder einzeln `host_vars/<BLOCK NODE>.yml` Dateien:

```

- hosts: eseries_storage_systems
gather_facts: false
collections:
  - netapp_eseries.santricity
tasks:
  - name: Verify before proceeding.
    pause:
      prompt: "Are you ready to proceed with running the BeeGFS HA
role? Depending on the size of the deployment and network performance
between the Ansible control node and BeeGFS file and block nodes this
can take awhile (10+ minutes) to complete."
  - name: Configure NetApp E-Series block nodes.
    import_role:
      name: nar_santricity_management

```

2. Definieren Sie die Wiedergabe, die für alle Datei- und Blockknoten ausgeführt wird:

```

- hosts: all
any_errors_fatal: true
gather_facts: false
collections:
  - netapp_eseries.beegfs

```

3. In diesem Sales Play können wir optional einen Satz von „Vorausgaben“ definieren, die vor der Bereitstellung des HA-Clusters ausgeführt werden sollten. Dies kann nützlich sein, um alle Voraussetzungen wie Python zu überprüfen/zu installieren. Zudem können Überprüfungen vor dem Flug durchgeführt werden, beispielsweise die Unterstützung der bereitgestellten Ansible-Tags:

```

pre_tasks:
  - name: Ensure a supported version of Python is available on all
file nodes.
    block:
      - name: Check if python is installed.
        failed_when: false
        changed_when: false
        raw: python --version
        register: python_version

      - name: Check if python3 is installed.
        raw: python3 --version
        failed_when: false
        changed_when: false
        register: python3_version
        when: 'python_version["rc"] != 0 or (python_version["stdout"]

```

```

| regex_replace("Python ", "")) is not version("3.0", ">=")'

- name: Install python3 if needed.
  raw: |
    id=$(grep "^ID=" /etc/*release* | cut -d= -f 2 | tr -d '"')
    case $id in
      ubuntu) sudo apt install python3 ;;
      rhel|centos) sudo yum -y install python3 ;;
      sles) sudo zypper install python3 ;;
    esac
  args:
    executable: /bin/bash
  register: python3_install
  when: python_version['rc'] != 0 and python3_version['rc'] != 0
  become: true

- name: Create a symbolic link to python from python3.
  raw: ln -s /usr/bin/python3 /usr/bin/python
  become: true
  when: python_version['rc'] != 0
when: inventory_hostname not in
groups[beegfs_ha_ansible_storage_group]

- name: Verify any provided tags are supported.
  fail:
    msg: "{{ item }}" tag is not a supported BeeGFS HA tag. Rerun
your playbook command with --list-tags to see all valid playbook tags."
    when: 'item not in ["all", "storage", "beegfs_ha",
"beegfs_ha_package", "beegfs_ha_configure",
"beegfs_ha_configure_resource", "beegfs_ha_performance_tuning",
"beegfs_ha_backup", "beegfs_ha_client"]'
    loop: "{{ ansible_run_tags }}"

```

4. Schließlich importiert dieses Spiel die BeeGFS HA-Rolle für die Version von BeeGFS, die Sie bereitstellen möchten:

```

tasks:
- name: Verify the BeeGFS HA cluster is properly deployed.
  import_role:
    name: beegfs_ha_7_4 # Alternatively specify: beegfs_ha_7_3.

```



Für jede unterstützte Major.Minor Version von BeeGFS wird eine BeeGFS HA-Rolle beibehalten. Auf diese Weise können Benutzer festlegen, wann ein Upgrade von Major-/Minor-Versionen durchgeführt werden soll. Derzeit (beegfs_7_3(`beegfs_7_2` werden BeeGFS 7.3.x) oder BeeGFS 7.2.x) unterstützt. Standardmäßig werden beide Rollen zum Zeitpunkt der Veröffentlichung die neueste BeeGFS-Patch-Version bereitstellen. Benutzer können dies jedoch überschreiben und gegebenenfalls den neuesten Patch bereitstellen. "[Upgrade-Leitfaden](#)" Weitere Informationen finden Sie auf dem neuesten Stand.

5. Optional: Wenn Sie zusätzliche Aufgaben definieren möchten, sollten Sie beachten, ob die Aufgaben an geleitet werden sollen `all` Hosts (einschließlich der E-Series Storage-Systeme) oder nur die Datei-Nodes Definieren Sie bei Bedarf ein neues Spiel speziell für Dateiknoten mit `- hosts: ha_cluster`.

Klicken Sie Auf "[Hier](#)" Beispiel für eine vollständige Playbook-Datei.

NetApp Ansible Sammlungen installieren

Die BeeGFS-Sammlung für Ansible, und alle Abhängigkeiten werden aufrechterhalten "[Ansible-Galaxie](#)". Führen Sie auf Ihrem Ansible-Steuerungsknoten den folgenden Befehl aus, um die neueste Version zu installieren:

```
ansible-galaxy collection install netapp_eseries.beegfs
```

Obwohl nicht in der Regel empfohlen, ist es auch möglich, eine bestimmte Version der Sammlung zu installieren:

```
ansible-galaxy collection install netapp_eseries.beegfs:  
==<MAJOR>.<MINOR>.<PATCH>
```

Führen Sie das Playbook aus

Aus dem Verzeichnis auf Ihrem Ansible-Steuerungsknoten, der den enthält `inventory.yml` Und `playbook.yml` Dateien, führen Sie das Playbook wie folgt aus:

```
ansible-playbook -i inventory.yml playbook.yml
```

Basierend auf der Cluster-Größe kann die ursprüngliche Implementierung 20+ Minuten dauern. Wenn die Implementierung aus irgendeinem Grund fehlschlägt, korrigieren Sie einfach Probleme (z. B. Fehlverkabelung, Knoten wurde nicht gestartet usw.) und starten Sie das Ansible Playbook neu.

"[Allgemeine Konfiguration der Datei-Nodes](#)" Wenn Sie angeben , wenn Sie die Standardoption wählen, damit Ansible die verbindungs-basierte Authentifizierung automatisch verwaltet, `connAuthFile` kann ein als gemeinsamer Schlüssel verwendet jetzt unter `<playbook_dir>/files/beegfs/<sysMgmtHost>_connAuthFile` (standardmäßig) gefunden werden. Alle Clients, die auf das Dateisystem zugreifen müssen, müssen diesen gemeinsam genutzten Schlüssel verwenden. Dies wird automatisch verarbeitet, wenn Clients über die konfiguriert werden "[BeeGFS-Client-Rolle](#)".

Bereitstellen von BeeGFS-Clients

Optional kann Ansible verwendet werden, um BeeGFS-Clients zu konfigurieren und das Dateisystem zu mounten.

Überblick

Für den Zugriff auf BeeGFS-Dateisysteme muss der BeeGFS-Client auf jedem Node installiert und konfiguriert werden, der das Dateisystem bereitstellen muss. In diesem Abschnitt wird beschrieben, wie Sie diese Aufgaben mit dem verfügbaren ausführen "[Ansible-Rolle](#)".

Schritte

Erstellen Sie die Client-Bestandsdatei

1. Richten Sie bei Bedarf über den Ansible-Steuerungsknoten passwortlose SSH für jeden Host ein, den Sie als BeeGFS-Clients konfigurieren möchten:

```
ssh-copy-id <user>@<HOSTNAME_OR_IP>
```

2. Unter `host_vars/`` Erstellen Sie für jeden BeeGFS-Client eine Datei mit dem Namen `<HOSTNAME>.yaml` Füllen Sie den Platzhaltertext mit den korrekten Informationen für Ihre Umgebung aus:

```
# BeeGFS Client
ansible_host: <MANAGEMENT_IP>
```

3. Geben Sie optional eine der folgenden Optionen ein, wenn Sie die Rollen der NetApp E-Series Host Collection verwenden möchten, um InfiniBand- oder Ethernet-Schnittstellen für Clients zu konfigurieren, damit eine Verbindung mit BeeGFS File-Nodes hergestellt werden kann:
 - a. Wenn der Netzwerktyp ist "[InfiniBand \(mit IPoIB\)](#)":

```
eseries_ipoib_interfaces:
- name: <INTERFACE> # Example: ib0 or ilb
  address: <IP/SUBNET> # Example: 100.127.100.1/16
- name: <INTERFACE> # Additional interfaces as needed.
  address: <IP/SUBNET>
```

- b. Wenn der Netzwerktyp ist "[RDMA über Converged Ethernet \(RoCE\)](#)":


```
eseries_roce_interfaces:
- name: <INTERFACE> # Example: eth0.
  address: <IP/SUBNET> # Example: 100.127.100.1/16
- name: <INTERFACE> # Additional interfaces as needed.
  address: <IP/SUBNET>
```

c. Wenn der Netzwerktyp ist "Ethernet (nur TCP, kein RDMA)":

```
eseries_ip_interfaces:
- name: <INTERFACE> # Example: eth0.
  address: <IP/SUBNET> # Example: 100.127.100.1/16
- name: <INTERFACE> # Additional interfaces as needed.
  address: <IP/SUBNET>
```

4. Erstellen Sie eine neue Datei `client_inventory.yml`. Geben Sie den Benutzer an, den Ansible für die Verbindung mit den einzelnen Clients verwenden soll, und das Passwort, das Ansible zur Eskalation von Berechtigungen verwenden soll (dies erfordert `ansible_ssh_user` „Root“ oder „Sudo“-Berechtigungen besitzen):

```
# BeeGFS client inventory.
all:
  vars:
    ansible_ssh_user: <USER>
    ansible_become_password: <PASSWORD>
```



Speichern Sie Passwörter nicht im Klartext. Verwenden Sie stattdessen den Ansible-Vault (siehe "[Ansible-Dokumentation](#)" Für Inhalte mit Ansible Vault) oder verwenden Sie den `--ask-become-pass` Option beim Ausführen des Playbooks.

5. Im `client_inventory.yml` Datei, Listen Sie alle Hosts auf, die als BeeGFS-Clients unter dem konfiguriert werden sollen `beegfs_clients` Gruppe, und dann beachten Sie die Inline-Kommentare und Uncomment zusätzliche Konfiguration erforderlich, um das BeeGFS-Client-Kernel-Modul auf Ihrem System zu erstellen:

```

children:
  # Ansible group representing all BeeGFS clients:
  beegfs_clients:
    hosts:
      <CLIENT_HOSTNAME>:
        # Additional clients as needed.

    vars:
      # OPTION 1: If you're using the NVIDIA OFED drivers and they are
already installed:
      #eseries_ib_skip: True # Skip installing inbox drivers when
using the IPoIB role.
      #beegfs_client_ofed_enable: True
      #beegfs_client_ofed_include_path:
"/usr/src/ofa_kernel/default/include"

      # OPTION 2: If you're using inbox IB/RDMA drivers and they are
already installed:
      #eseries_ib_skip: True # Skip installing inbox drivers when
using the IPoIB role.

      # OPTION 3: If you want to use inbox IB/RDMA drivers and need
them installed/configured.
      #eseries_ib_skip: False # Default value.
      #beegfs_client_ofed_enable: False # Default value.

```



Wenn Sie die NVIDIA OFED-Treiber verwenden, stellen Sie sicher, dass `beegfs_Client_ofed_include_PATH` auf den korrekten „Header include path“ für Ihre Linux-Installation verweist. Weitere Informationen finden Sie in der BeeGFS-Dokumentation für ["RDMA-Unterstützung"](#).

6. Im `client_inventory.yml` Datei, Listen Sie die BeeGFS-Dateisysteme auf, die Sie unter einem zuvor definierten gemountet haben möchten vars:

```

    beegfs_client_mounts:
      - sysMgmtHost: <IP ADDRESS> # Primary IP of the BeeGFS
management service.
      mount_point: /mnt/beegfs # Path to mount BeeGFS on the
client.
    connInterfaces:
      - <INTERFACE> # Example: ibs4f1
      - <INTERFACE>
    beegfs_client_config:
      # Maximum number of simultaneous connections to the same
node.
      connMaxInternodeNum: 128 # BeeGFS Client Default: 12
      # Allocates the number of buffers for transferring IO.
      connRDMABufNum: 36 # BeeGFS Client Default: 70
      # Size of each allocated RDMA buffer
      connRDMABufSize: 65536 # BeeGFS Client Default: 8192
      # Required when using the BeeGFS client with the shared-
disk HA solution.
      # This does require BeeGFS targets be mounted in the
default "sync" mode.
      # See the documentation included with the BeeGFS client
role for full details.
      sysSessionChecksEnabled: false
      # Specify additional file system mounts for this or other file
systems.

```

7. Ab BeeGFS 7.2.7 und 7.3.1 "[Verbindungsauthentifizierung](#)" müssen konfiguriert oder explizit deaktiviert sein. Je nachdem, wie Sie die verbindungs-basierte Authentifizierung bei der Angabe konfigurieren "[Allgemeine Konfiguration der Datei-Nodes](#)", müssen Sie möglicherweise Ihre Clientkonfiguration anpassen:
- Standardmäßig konfiguriert die HA-Cluster-Implementierung die Verbindungsauthentifizierung automatisch und generiert einen `connauthfile` Die auf dem Ansible-Kontroll-Node bei platziert/gewartet werden `<INVENTORY>/files/beegfs/<sysMgmtHost>_connAuthFile`. Standardmäßig ist die BeeGFS-Client-Rolle so eingerichtet, dass sie diese Datei an die in definierten Clients liest/verteilt `client_inventory.yml`, Und es ist keine zusätzliche Aktion erforderlich.
 - Weitere Optionen finden Sie in der vollständigen Liste der Standardwerte, die im enthalten sind "[BeeGFS-Client-Rolle](#)".
 - Wenn Sie ein benutzerdefiniertes Geheimnis mit angeben `beegfs_ha_conn_auth_secret` Geben Sie ihn im an `client_inventory.yml` Außerdem:

```

beegfs_ha_conn_auth_secret: <SECRET>

```

- Wenn Sie die verbindungs-basierte Authentifizierung vollständig mit deaktivieren `beegfs_ha_conn_auth_enabled`, Geben Sie das im an `client_inventory.yml` Außerdem:

```
beegfs_ha_conn_auth_enabled: false
```

Eine vollständige Liste der unterstützten Parameter und weitere Details finden Sie im "[Vollständige BeeGFS-Client-Dokumentation](#)". Klicken Sie für ein vollständiges Beispiel eines Clientbestands auf "[Hier](#)".

Erstellen Sie die BeeGFS Client Playbook-Datei

1. Erstellen Sie eine neue Datei `client_playbook.yml`

```
# BeeGFS client playbook.
- hosts: beegfs_clients
  any_errors_fatal: true
  gather_facts: true
  collections:
    - netapp_eseries.beegfs
    - netapp_eseries.host
  tasks:
```

2. Optional: Wenn Sie die Rollen der NetApp E-Series Host Collection verwenden möchten, um Schnittstellen für Clients zu konfigurieren, mit denen sich eine Verbindung zu BeeGFS-Dateisystemen herstellen lässt, importieren Sie die Rolle entsprechend dem Schnittstellentyp, den Sie konfigurieren:

- a. Wenn Sie InfiniBand (IPoIB) verwenden:

```
- name: Ensure IPoIB is configured
  import_role:
    name: ipoib
```

- b. Bei Verwendung von RDMA over Converged Ethernet (RoCE):

```
- name: Ensure IPoIB is configured
  import_role:
    name: roce
```

- c. Wenn Sie Ethernet verwenden (nur TCP, kein RDMA):

```
- name: Ensure IPoIB is configured
  import_role:
    name: ip
```

3. Schließlich importieren Sie die BeeGFS-Client-Rolle, um die Client-Software zu installieren und das Dateisystem-Mounts einzurichten:

```
# REQUIRED: Install the BeeGFS client and mount the BeeGFS file
system.
- name: Verify the BeeGFS clients are configured.
  import_role:
    name: beegfs_client
```

Klicken Sie zum vollständigen Beispiel für ein Client-Playbook auf "[Hier](#)".

Führen Sie das BeeGFS Client Playbook aus

Führen Sie den folgenden Befehl aus, um den Client zu installieren/zu erstellen und BeeGFS zu mounten:

```
ansible-playbook -i client_inventory.yml client_playbook.yml
```

Überprüfen Sie die BeeGFS-Bereitstellung

Überprüfen Sie die Bereitstellung des Dateisystems, bevor Sie das System in die Produktion bringen.

Überblick

Bevor Sie das BeeGFS-Dateisystem in Produktion setzen, führen Sie einige Überprüfungsprüfungen durch.

Schritte

1. Melden Sie sich bei einem Client an und führen Sie Folgendes aus, um sicherzustellen, dass alle erwarteten Knoten vorhanden/erreichbar sind, und es werden keine Inkonsistenzen oder andere Probleme gemeldet:

```
beegfs-fsck --checkfs
```

2. Fahren Sie den gesamten Cluster herunter und starten Sie ihn dann neu. Führen Sie von jedem beliebigen Dateiknoten Folgendes aus:

```
pcs cluster stop --all # Stop the cluster on all file nodes.
pcs cluster start --all # Start the cluster on all file nodes.
pcs status # Verify all nodes and services are started and no failures
are reported (the command may need to be reran a few times to allow time
for all services to start).
```

3. Platzieren Sie jeden Knoten in den Standby-Modus, und überprüfen Sie, ob BeeGFS-Dienste einen Failover auf sekundäre Knoten ausführen können. So melden Sie sich an einem beliebigen Dateiknoten an und führen Sie Folgendes aus:

```
pcs status # Verify the cluster is healthy at the start.
pcs node standby <FILE NODE HOSTNAME> # Place the node under test in
standby.
pcs status # Verify services are started on a secondary node and no
failures are reported.
pcs node unstandby <FILE NODE HOSTNAME> # Take the node under test out
of standby.
pcs status # Verify the file node is back online and no failures are
reported.
pcs resource relocate run # Move all services back to their preferred
nodes.
pcs status # Verify services have moved back to the preferred node.
```

4. Verwenden Sie Tools zum Leistungsvergleich wie IOR und MDTest, um zu überprüfen, ob die Performance des Dateisystems den Erwartungen entspricht. Beispiele für häufige Tests und Parameter, die mit BeeGFS verwendet werden "[Design-Verifizierung](#)", finden Sie im Abschnitt BeeGFS on NetApp Verified Architecture.

Zusätzliche Tests sollten auf Grundlage der Abnahmekriterien durchgeführt werden, die für einen bestimmten Standort/eine bestimmte Installation definiert sind.

Copyright-Informationen

Copyright © 2024 NetApp. Alle Rechte vorbehalten. Gedruckt in den USA. Dieses urheberrechtlich geschützte Dokument darf ohne die vorherige schriftliche Genehmigung des Urheberrechtlich geschützten Urhebers 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, BEISPIELHAFT 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.